Biomedical Data and Applications
The molecular biology community faces an inundation of data. This implies that most of the contents of databanks will be recently determined data, and features, such as quality, will be characteristic of the newest methods of measurement. New experimental techniques will increase the amount and diversity of data; for example, the Genomics, Proteomics and Expression Profiles projects. Compared with data from general application domains, modern biological data has many unique characteristics. Biological data are often characterized as having large volumes, complex structures, high dimensionality, evolving biological concepts, and insufficient data modelling practices. These characteristics require database researchers and developers to make many special considerations while developing biological databases and database systems. They also have made biological data management and knowledge discovery in databases challenging.

Management of scientific data is critical in supporting discoveries in the life sciences field. Over the past several years, bioinformatics has become an all-encompassing term for everything relating to both computer science and biology. The goal of this book is to cover biomedical data and applications identifying new issues and directions for future research in biomedical domain. The book will become a useful guide for researchers, practitioners, and graduate-level students interested in learning state-of-the-art development in biomedical data management, data-intensive bioinformatics systems, and other miscellaneous biological database applications. The content of this book is at an introductory and medium technical level.

There are 14 chapters presented in this book. Individual chapters have been written by selected and accomplished research teams active in the research of respective topics. Each chapters covers an important aspect of the fast growing topic of bioinformatics. Complication of these book chapters on the whole addresses this book's topic with varying degrees of balance between biomedical data models and their real-world applications. Chapter 1 discusses trends in biomedical data and applications. It focuses on the areas of biomedical data integration, access, and interoperability as these areas form the cornerstone of the field. In chapters 2 through 10, we introduce biomedical data management and general data analysis practices essential to post-genome biology. Chapters 11 through to 14 discuss methodologies of some of the biomedical applications.
Chapter 2 proposes to intuitively organize into Resourceomes the hierarchical vision of a scientific domain perceived by its scientists, connecting the related resources to the topics they concern. In chapter 3, a small catalog of 74 selected genomic databases are analyzed and classified by considering both their biological contents and their technical features (e.g., how they may be queried, the database schemas they are based on, the different data formats, etc.). Chapter 4 considers the challenges of information integration in proteomics from the prospective of researchers using information technology as an integral part of their discovery process. Chapter 5 presents a series of aspects in the field of multimedia medical databases. These databases are the result of structuring the alphanumeric and imagistic data gathered in large quantities in the patient investigation and diagnosis processes. It also presents the necessity of creating and managing the multimedia medical databases and the advantages of using these operations in order to increase medical act efficiency.

Chapter 6 surveys and reviews change management in bio-ontologies as well as some of the available tools and techniques in this area. It also surveys various potential changes in biomedical ontologies, with actual examples from some of the most popular ontologies in the biomedical domain. It investigates the potential of some of the advanced formalisms in this context by proposing formal method for analyzing and supporting ontology evolution and change management. In chapter 7 a framework for the application of data mining tools to constraint extraction in the biological domain is presented focusing on tuple constraint and functional dependency detection in representative biological databases by means of association rule mining. Chapter 8 proposes a method for classifying patterns in bioinformatics databases using Alpha-Beta Associative Memories. Chapter 9 described two pattern discovery analyses conducted on clinical data of patients in the follow-up of a liver transplantation. The two pattern discovery analyses used different techniques for different objectives. Chapter 10 presents a graph representation of metabolic pathways to describe all features of metabolic pathways and describes the application of graph-based relational learning for structure analysis on metabolic pathways in both supervised and unsupervised scenarios.

In Chapter 11, a new Personal Health Office Framework (PhoF) is described which is designed to enable patients effectively manage their own. Chapter 12 focuses on the implementation of agent-based systems within the health domain, more specifically, in the study of total well-being. Chapter 13 evaluates the ability of the minimal model of glucose disappearance to describe experimental data collected from 9 diabetic patients controlled subcutaneously by an insulin pump. Lastly, chapter 14 presents the ab initio protein structure prediction as a conformational search problem in low resolution model using genetic algorithm.

We hope this book will become a useful resource for bioinformatics graduate students, researchers, and practitioners interested in managing post-genome biology. We hope that bioinformatics students will use the book material as a guide to acquire basic concepts and theories of post-genome biomedical data management, bioinformatics practitioners will find valuable lessons for building future similar
biomedical systems, and researchers will find rewarding research data management questions to address in years to come.

April 2009
Perth, Australia

Amandeep S. Sidhu
Tharam S. Dillon
## Contents

Current Trends in Biomedical Data and Applications  
_Amandeep S. Sidhu, Matthew Bellgard, Tharam S. Dillon_  \[1\]

**Part I: Biomedical Data**

### Towards Bioinformatics Resourceomes  
_Nicola Cannata, Flavio Corradini, Emanuela Merelli, Francesca Piersigilli, Leonardo Vito_  \[13\]

### A Summary of Genomic Databases: Overview and Discussion  
_Erika De Francesco, Giuliana Di Santo, Luigi Palopoli, Simona E. Rombo_  \[37\]

### Protein Data Integration Problem  
_Amandeep S. Sidhu, Matthew Bellgard_  \[55\]

### Multimedia Medical Databases  
_Liana Stanescu, Dumitru Dan Burdescu, Marius Brezovan_  \[71\]

### Bio-medical Ontologies Maintenance and Change Management  
_Arash Shaban-Nejad, Volker Haarslev_  \[143\]

### Extraction of Constraints from Biological Data  
_Daniele Apiletti, Giulia Bruno, Elisa Ficarra, Elena Baralis_  \[169\]

### Classifying Patterns in Bioinformatics Databases by Using Alpha-Beta Associative Memories  
_Israel Román Godínez, Itzamá López-Yáñez, Cornelio Yáñez-Márquez_  \[187\]
Mining Clinical, Immunological, and Genetic Data of Solid Organ Transplantation
Michele Berlingerio, Francesco Bonchi, Michele Curcio, Fosca Giannotti, Franco Turini .................... 211

Substructure Analysis of Metabolic Pathways by Graph-Based Relational Learning
Chang hun You, Lawrence B. Holder, Diane J. Cook .................. 237

Part II: Biomedical Applications

Design of an Online Physician-Mediated Personal Health Record System
Jake Y. Chen, Shailaja Taduri, Frank Lloyd ......................... 265

Completing the Total Wellbeing Puzzle Using a Multi-agent System
Maja Hadzic, Meifania Chen, Rick Brouwer ......................... 281

The Minimal Model of Glucose Disappearance in Type I Diabetes
Margarita Fernandez, Minaya Villasana, Dan Streja ............... 295

Genetic Algorithm in Ab Initio Protein Structure Prediction Using Low Resolution Model: A Review
Md. Tamjidul Hoque, Madhu Chetty, Abdul Sattar ................. 317

Author Index .......................................................... 343
Current Trends in Biomedical Data and Applications

Amandeep S. Sidhu¹, Matthew Bellgard¹, and Tharam S. Dillon²

¹ WA Centre for Comparative Genomics, Murdoch University, Perth, Australia
{asidhu,mbellgard}@ccg.murdoch.edu.au
² Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia
Tharam.Dillon@cbs.curtin.edu.au

Abstract. Bioinformatics tools and systems perform a diverse range of functions including: data collection, data mining, data analysis, data management, and data integration. Computer-aided technology directly supporting medical applications is excluded from this definition and is referred to as medical informatics. This book is not an attempt at authoritatively describing the gamut of information contained in this field. Instead, it focuses on the areas of biomedical data integration, access, and interoperability as these areas form the cornerstone of the field. However, most of the approaches presented are generic integration systems that can be used in many similar contexts.

1 Background

Scientists argue that bioinformatics began in 1965. The first edition of the Atlas of Protein Sequence and Structure, compiled by Margaret Dayhoff, appeared in print form in 1965 (Dayhoff et al., 1965). The Atlas later became the basis for the PIR protein sequence database (Wu et al., 2003). However, this is stretching the point a little. The term bioinformatics was not around in 1965, and barring a few pioneers, bioinformatics was not an active research area at the time. As a discipline, bioinformatics is more recent. It arose from the recognition that efficient computational techniques were needed to study the huge amount of biological sequence information that was becoming available. If molecular biology arose at the same time as scientific computing, then we may also say that bioinformatics arose at the same time as the Internet. It is possible to imagine that biological sequence databases could exist without the Internet, but they would be much less useful.

Since the first efforts of Maxam (Maxam and Gilbert, 1977) and Sanger (Sanger et al., 1977), the DNA sequence databases have been doubling in size every 18 months or so. This trend continues unabated. This has forced the development of systems of software and mathematical techniques for managing and searching these collections. In the past decade, there has been an explosion in the amount of DNA sequence data available, due to the very rapid progress of genome sequencing projects. There are three principal comprehensive databases of nucleic acid sequences in the world today.
The EMBL (European Molecular Biology Laboratory) database is maintained at European Bioinformatics Institute in Cambridge, UK (Stoesser et al., 2003).

GenBank is maintained at the National Center for Biotechnology Information in Maryland, USA (Benson et al., 2000).

The DDBJ (DNA Databank of Japan) is maintained at National Institute of Genetics in Mishima, Japan (Miyazaki et al., 2003).

These three databases share information and hence contain similar but not necessarily identical sets of sequences. The objective of these databases is to ensure that DNA sequence information is stored in a way that is publicly and freely accessible, and that it can be retrieved and used by other researchers in the future.

Clearly, we have reached a point where computers are essential for the storage, retrieval, and analysis of biological sequence data. The sheer volume of data made it difficult to find sequences of interest in each release of sequence databases. The data were distributed as collection of flat files, each of which contained some textual information (the annotation) such as organism name and keywords as well as the DNA sequence. The main method of searching for a sequence of interest was to use a string-matching program. This forced the development of relational database management systems in the main database centres, but the databases continued to be delivered as flat files. One important system that is still in use for browsing and searching the databases, was ACNUC (Gouy et al., 1985), from Manolo Gouy and colleagues in Lyon, France. This was developed in the mid-'eighties and allowed fully relational searching and browsing of database annotation. SRS (Etzold and Argos, 1993) is a more recent development of this system.

Another important type of biological data that is exponentially increasing is protein structures. Protein Data Bank or PDB (Bernstein et al., 1977, Weissig and Bourne, 2002, Wesbrook et al., 2002) is a database of protein structures obtained from X-ray crystallography and NMR experiments. The traditional format used by PDB is PDB format. It consists of a collection of fixed format records that describe the atomic coordinates, chemical and biochemical features, experimental details of structure determination, and some structural features such as hydrogen bonds and secondary structure assignments. In recent years, dictionary-based representations emerged to give data a consistent interface (Bourne et al., 2004), making it easier to parse. The problem of management of biological macromolecular sequence data is as old as the data themselves. In 1998, a special issue of Nucleic Acids Research listed 64 different databanks covering diverse areas of biological research, and the nucleotide sequence data alone at over 1 billion bases. It is not only the flood of information and heterogeneity that make the issues of information representation, storage, structure, retrieval and interpretation critical. There also has been a change in the community of users. In the middle 1980s, fetching a biological entry on a mainframe computer was an adventurous step that only few dared. Now, at the end of the 1990s, thousands of researchers make use of biological databanks on a daily basis to answer queries, e.g. to find sequences similar to a newly sequenced gene, or to retrieve bibliographic references, or to
investigate fundamental problems of modern biology (Koonin and Galperin, 1997). New technologies, of which the World Wide Web (WWW) has been the most revolutionary in terms of impact on science, have made it possible to create a high density of links between databanks. Database systems today are facing the task of serving ever increasing amounts of data of ever growing complexity to a user community that is growing nearly as fast as the data, and is becoming increasingly demanding. In the next section, we review and summarize the recent development of biological databases.

2 Biomedical Data

The current scope of databases ranges from large-scale archiving projects to individual, private, specialized collections serving the needs of particular user communities. These include the following. General biological databanks: GenBank (Benson et al., 2000), EMBL (Stoesser et al., 2003), PIR (Barker et al., 1998), SWISS-PROT (Boeckmann et al., 2003), Online Mendelian Inheritance in Man or OMIM (McKusick, 1998), and Protein Data Bank (Bourne et al., 2004). Species-specific full genome databases of human and other organisms: *Saccharomyces cerevisiae* (Cherry et al., 1998), FlyBase (Gelbart et al., 1998), and a variety of small genomes (White and Kerlavage, 1996). Databases specializing in subject matter, such as the database of transcription factors and their binding sites (Wingender et al., 1996) and the restriction enzyme resource (Roberts and Macelis, 1998). Derived databases containing added descriptive material on top of the primary data, or providing novel structuring of these data. Annotation is provided by automatic and/or manual methods. The most popular are the protein motif database PROSITE; (Bairoch et al., 1997), structural classification of proteins SCOP; (Murzin et al., 1995), protein structure-sequence alignments HSSP; (Dodge et al., 1998), protein domains PFAM; (Sonnhammer et al., 1998) and conserved regions of proteins BLOCKS; (Henikoff et al., 1998). Special databases can grow and evolve very quickly as the result of the enormous data flow produced by automated sequencing and functional analysis. Whereas the large primary databases collect and collate information from literature and from the scientific community, specialized data collections integrate, via curatorial expertise, information from a multiplicity of primary sources, including sequence, structure, function, evolutionary relationships and bibliographic references. Rigid database classification has become obsolete, and users choose according to individual needs from the rich WWW-accessible data. More significant than growth in volume of the databases is the increasing complexity of information available. Sequences are linked to structure, chemical properties and functionality. In 40% of the sequences order doesn’t matter. 40% of sequences in which order matters determine the chemical reactivity and the remaining 20% of sequences determine the functionality.

Mechanization of data acquisition necessitates searching for ways to improve the annotation. It is possible and desirable to automate many annotation subjects, or to automate the process partially by providing biological experts with intelligently organized evidence. The first large piece of molecular data to be subjected to computer-based annotation was yeast chromosome III (Bork et al., 1992). The
A.S. Sidhu, M. Bellgard, and T.S. Dillon

A set of programs used in this effort was the progenitor of GeneQuiz genome analysis system (Casari et al., 1996, Scharf et al., 1994). Several other programs have been developed (Frishman and Mewes, 1997, Gaasterland and Sensen, 1996, Walker and Koonin, 1997). Many research centres where primary data is being generated developed customized annotation systems tailored to their own processes (Eckman et al., 1997, Westbrook and Fitzgerald, 2003). Typical features of such tools are the systematic application of selected bioinformatics methods to sequence sets of any size, integration of all available evidence in the form of well organized summaries for each data entry, application of hierarchical logical rules for producing functional and structural inferences with appropriate reliability estimates, and data storage, retrieval and visualization capabilities. Methods are available that provide additional functional insights into biological sequence data without similarity comparisons. For example, (des Jardins et al., 1997) described a way to delineate, with reasonable accuracy, enzyme EC numbers from easily computable protein sequence features through the application of machine intelligence approaches. (Andrade and Valencia, 1997) described a procedure for associating protein sequence data with bibliographic references stored in the MEDLINE database through frequency analysis of word occurrence. No matter what program system is used, there are problems inherent in automated annotation. Molecular biology databanks are reluctant to adopt automatic techniques, as these may erode annotation quality. One of the suggested solutions is to split a databank into two parts: the core section with carefully processed entries and a supplementary part for which the first-pass analysis is done automatically (Apweiler et al., 2004).

Databanks can, at two extremes, function as passive data repositories (archives), or as active references, issuing modifications of data and information content. Data in biological databanks contain facts, e.g. representations of biological macromolecules as strings or coordinates, and associated information, which might be fuzzy, incomplete or subject to individual interpretation or conflicting nomenclature. Data quality has several elements: correctness, completeness, timeliness of capture, applied both to the newly measured properties, e.g. a new gene sequence, and the annotation. Quality control should not be restricted to semantic checking of individual entries, but also include relationships to other parts of the database (George et al., 1987). The growth in rates of data generation has implications for data quality. On the one hand, most new data entries are related to previously described objects and can inherit some part of their annotation. However, many newly determined data entries have no associated experimentally confirmed facts, and their annotation is based on predictions. We observe that databanks are tending to be more invasive in their approach to processing incoming data. Whereas databanks previously tended to function as repositories or archives, and acted only passively in distributing the data to the community, they are now playing a more active role in interacting with the data. This interaction may involve checking procedures and/or addition of information in the form of annotations and links with other databanks. In particular, genome-related databases actively curate and update data on a regular basis (e.g. MIPS and Stanford’s SGD for the yeast
Current Trends in Biomedical Data and Applications

5

The types of data contained in macromolecular databanks differ in the extent to which errors can be detected and corrected. For sequences, their general absence from the archive of raw data makes it difficult to detect errors unless there are multiple determinations of the same sequence, or if detailed structural analysis of the protein corresponding to the sequence makes an error appear extremely likely (Bashford et al., 1987), and the correct version of sequence suspected of being erroneous is generally impossible to infer. For protein and nucleic acid structures, in contrast, knowledge that the molecules must obey the general rules of stereochemistry and specialized rules of protein architecture makes it possible to try to evaluate the quality of a structure from the coordinate set alone. Several approaches to evaluating structures from coordinates alone have been proposed (Hooft et al., 1997). It is clear that effective error detection and correction of macromolecular structural data require the deposition and availability of these raw data, and it is very difficult to envisage any argument against requiring their deposition and distribution. Databanks have always been under competing pressures to provide data quickly and completely, but also to aim for optimal data quality. For suspect data (that doesn’t meet data quality control) one possibility is to withhold until data have been corrected (assuming that this is possible) and the other is to release data with a suitable warning. We tend to prefer the latter regime, but it is for the community as a whole to decide.

Redundancy in primary molecular biology information arises as the consequence of parallel acquisition of the same or highly similar data from independent sources, inherent redundancy in the data itself, as well as small natural variation in the subjects or errors in the measurements that prevent the identification of essentially identical entries. Additional redundancy is often introduced by insufficiently coordinated annotation efforts. The situation is further complicated by the existence of two or more information resources in many subject areas, exemplified by two parallel protein sequence databanks: PIR-International (Barker et al., 1998) and SWISS-PROT (Bairoch and Apweiler, 1997); and two parallel sequence databanks: GenBank (Benson et al., 2000) and EMBL (Stoesser et al., 2003). Perhaps the most severe case of data redundancy is represented by the EST sequence collections (Adams et al., 1991). Multiple occurrences of nearly identical sequences in EST data collections are due to re-sampling of the same gene and is especially common for highly expressed genes. Several groups proposed methods to collapse clusters of related ESTs into distinct data entries representing individual genes, thereby reducing this element of redundancy by several orders of magnitude (Hide et al., 1997, Schuler et al., 1996).

3 Interoperability of Biological Data

Biological data must be described in context rather than in isolation (Karp, 1996). Hence, many databases provide multiple links to other resources, but efficient use of these links requires intelligent retrieval systems. Attempts have been made to create interdatabase links automatically, restricted to few selected data
resources, and with limited accuracy (Achard and Dessen, 1998). The user needs to be able to extract responses to a probe query from all possible sources through a transparent and easy-to-use interface. The need for interoperability gave rise to the idea of an autonomous database federation (Robbins, 1994) through partial sharing of underlying database schema permitting cross-database queries, using, for example, SQL-based commercial database implementations. For example, attempts have been made to create a unified federated resource for microbiological information (Wertheim, 1995) suitable for intelligent interrogation. The prototype of such a system is represented by the Ribosome Database Project (Maidak et al., 1996).

An alternative approach is the concept of a warehouse, or a centralized data resource that manages a variety of data collections translated into a common format (Ritter, 1994). Linking the community of databases through common semantics is impossible because of their extreme heterogeneity. The recently emerged ‘middleware’ approach affords a chance to uncouple data access from data management and to allow for remote retrieval beyond the simple scripts fetching data from external databases. The most prominent industrial standard for a client-server based middleware is Common Object Request Broker Architecture or CORBA (Ben-Natan, 1995) as defined by the Object Management Group OMG. CORBA is a distributed object architecture that allows objects to communicate across networks through defined interfaces using the syntax of the Interface Definition Language (IDL). The object management architecture of CORBA specifies an application-level communication infrastructure. Several CORBA-based applications have already appeared. (Achard and Barillot, 1997) suggest a set of interface definitions for molecular biology to access a simple but realistic data bank of Sequence Tag Sites. The European Commission supports a project to provide CORBA access to a set of public databases (EMBL, SWISS-PROT, PIR, TRANSFAC, and several others).

(Stein et al., 1998) described an alternative approach to database interconnection. Their software system Jade establishes a connection between the database servers and the application programs, and organizes data exchange through standardized relational tables and parameters. Information retrieved on the data server side is transformed into these tables with the help of a specialized application called Jade adapter. Jade currently supports the AceDB, although incorporation of other database systems is anticipated.

4 Summary

The existing biomedical databases on the web form a loose federation of autonomous, distributed and heterogeneous data repositories. In order to make all this data really useful, one needs tools that will access and retrieve exactly the information one needs. The online available information needs to be intelligently queried. There is a need within the biomedical community for intelligent and dynamic information retrieval systems for different knowledge domains.
References


In science fiction, human beings have been depicted able to colonize planets of far stars exploiting their chemical and mining resources. We can image bioinformatics as a very dynamic universe, continuously growing under our eyes. More and more scientists are approaching it and would like to easier explore it, discovering the resources that can be found in every space region (i.e. related to every bioinformatics topic). We propose to intuitively organize into Resourceomes the hierarchical vision of a scientific domain perceived by its scientists, connecting the related resources to the topics they concern. A Resourceome can be seen as a map of a scientific “universe”. A semantic browser for Resourceomes permits to intuitively navigate in the domain, eventually zooming it in and out. Once discovered a resource of interest it is possible to be “tele-ported” to it. More importantly, the maps are “machine understandable”, being built on ontologies and published on the Web with Semantic Web technologies.

1 Bioinformatics Resources

We introduce the reader to the universe of bioinformatics, concentrating our attention on bioinformatics resources, like e.g. databases, articles and programs.

1.1 A Universe in Expansion

Bioinformatics can be seen as a very dynamic and turbulent universe in continuous expansion. Its expansion speed is increasing and new regions are suddenly created. Bursts of new topics can be often observed as well as the deadening of other ones. Buzzwords quickly enter the scene. Metaphors are frequently used in teaching and research as well as in divulgation of science. One of the most famous metaphors in life science paints the genome as the “book of life”.

Bioinformatics originates from the encounter of the two universes of computer science and life sciences. The growing amount of knowledge acquired
in molecular biology, coupled with the technological evolutions (constant improvements and miniaturizations; the first impacts from the nanotechnologies) initiated to produce unimaginable quantity of data. Therefore the marriage with a science able to organize, manage, elaborate and represent such heaps of information was an unavoidable one. On the other hand, also the (much more recent than biology) science of informatics is strongly characterized by constant technological advancement and exponential growth. The empirical Moore’s law1 (the number of transistors on a chip doubles about every two years) can naturally be transferred from semiconductor industry to many other fields (e.g. growth of DNA sequence database, number of internet Web sites). The whole system evolves as a positive feedback circuit. From data, new information and knowledge are derived, in turn catalyzing new technologies which consequently yield higher amount of data. Scientists working in this area are becoming used to data deluges and overflows [7]. An impressing logarithmic graph at GenomeNet2 reports the growth of the main molecular databases. Concerning DNA sequences the new generation of sequencers has hundreds times higher production rates with a cost per Megabase of a tenth or a hundredth in comparison to the previous generation. So we can expect a further hike in the production of DNA sequences and in the number of sequenced organisms. In general it is also increasing the variety of the data. The anarchy in microarray experiments at the beginning of the new millennium required the definition of standards (i.e. MIAME [10]). Similarly, to bring order in the - now maybe too abused - field of systems biology, standards have been defined for data in metabolic modeling (MIRIAM [43]), proteomics experiments (MIAPE [63]), molecular interaction experiments (MIMIx [50]).

The pioneering computational analysis of molecular data and information started even before the 1970s [52]. The term “bioinformatics” was first used by Hwa Lim in the late 1980s [54]. But the community, after all these years is still asking “what is bioinformatics?” [27] and “who are the bioinformaticians?”. We suggest to identify three different levels of bioinformaticians with respect to their relations to resources.

At the basic level we can collocate the simple users of bioinformatics resources. In the pre-WWW age, resources were databases (composed of a few records) and programs, to be installed in local computers. The WWW contributed undoubtedly to the affirmation of bioinformatics. The fact that every data could be freely accessed on a database in the internet from everybody, from everywhere in the world and at every time, revolutioned science. HTML interfaces rendered simpler the execution - online - of algorithms on generally remote, huge and complex data. It was not necessary anymore to bother about programs and databases installation, license, versioning, operating systems, hardware requirements and system management skills. Nowadays, for every molecular biologists and every scientist involved in “omics” disciplines,

1 http://www.intel.com/technology/mooreslaw/
bioinformatics simply offers the common daily tools. The suffix “omics” became very famous up from the late 1990s to indicate the paradigm shift of research: from the study of a single molecule (e.g. a gene, a protein) to the study of the entire population of such molecules as a whole (e.g. the genome, the proteome). Probably with the announcement of the completion of the first draft of the human genome in the year 2000, bioinformatics reached the status of “big science”, becoming popular in the media and to the common people.

At the second level we can find the users which assemble resources. In the decade around the year 2000, complex repetitive bioinformatics analyses were automatized in form of programs written with script languages (e.g. Perl). Besides being very compact and permitting rapid programs development, such languages are mostly ideal also for performing sophisticated text analysis and pattern matching. Some of them are even endowed with dedicated bioinformatics libraries (e.g. Bioperl[3]). Now, most of the bioinformatics programs as well as database queries, can be invoked remotely as Web Services available somewhere in the internet. Web pages with HTML form have been developed for human beings, but being not standards and without any precise semantics and syntax, they could not be tackled automatically by programs. Also the affirmation of XML[4] as a standard for data interchange contributed firmly to acquire a complete machine interoperability. Remote programs sparse over the internet can talk - and understand - with each other using standard protocols. Scripts to execute “in-silico” bioinformatics experiments have been now substituted by workflows of Web Services, which can be composed in a very simple visual manner. Therefore, also users without any programming skill can easily assemble them.

At the higher level we pose the developers of resources. Either they could be the computational biologists, which develop the algorithms, subsequently implemented as stand alone programs or Web applications, accessible through HTML forms or Web Services. Or they could be the database developers, which expose their data or information, probably obtained also through complex computations and filtering. Here still emerges the foundational binomial of bioinformatics bioinformatics = database + computational biology. The latter could in some way remember the notable algorithms + data structure = programs [6].

To give an idea of the dynamicity of bioinformatics we do list the titles of the articles in the “trends guide to bioinformatics” edited by Mark Boguski in 1998 [9]: Text-based database searching, Fundamentals of database searching, Practical database searching, Computational genefinding, Multiple-alignment and -sequence searches, Protein classification and functional assignment, Phylogenetic analysis and comparative genomics, Databases of biological information, Functional genomics. We can make a comparison with the categories adopted from 2005 - only seven years later - by the journal Bioinformatics for

---

3 http://www.bioperl.org/
4 http://www.w3.org/XML/
classifying its articles: Genome analysis, Sequence analysis, Structural bioinformatics, Gene expression, Genetics and population analysis, Data and text mining, Databases and ontologies, Systems biology. Gene Expression, with its inflation of microarray experiments - but now already in a descending phase - could maybe have been glimpsed in 1998’s Functional genomics. Text mining, ontologies and systems biology arose instead rather “unexpectedly” and today constitute probably some of the most active research area in bioinformatics. A good introductions to biological literature mining can be found in [40]. The “manifesto” of computational systems biology was launched by Kitano at the dawn of the new millennium [41, 42]. Evolving research trends in bioinformatics have been analyzed in 2006 also by Wren and collaborators [53], through a statistical analysis of the occurrence of MeSH terms in the Bioinformatics Journal abstracts. Unfortunately MeSH terms concern biomedicine and do not actually provide a proper classification of bioinformatics tasks.

Now it has become extremely hard for a scientist to be a “plenary” bioinformatician, following the development of the whole area. The galaxies of the bioinformatics universe are too many, too big and too distant among each other. Being aware of all the resources available in the whole domain is a “mission impossible”. For this reason, adopting the same “ome” suffix used in indicating the wide amount of a molecular population, we termed “resourceome” the full set of bioinformatics resources [16]. In this chapter we describe how an intuitive semantic index of bioinformatics resources could be built for being understood (and reasoned) also by computers.

1.2 Searching for Bioinformatics Resources Today

What to Search for?

Students probably would like to have a general introduction to bioinformatics. They may be interested in text books and other scholar stuff: on-line courses - better when providing practical experiments -, lectures, presentations or e-learning objects. But also Web pages or portals summarizing the state of the art and simple reviews could be helpful when approaching specific arguments. Already formed scientists, active in other fields and wishing to know more about bioinformatics, could be interested in gaining an overall overview. For them the best would be to have a taxonomy of the domain, to easily individuate the characterizing areas, the historical subjects and the most promising arenas. Reviews, editorials and perspective articles may represent a good starting point for them. Scientists concerned in interdisciplinary collaborations or projects would be very interested in detailed insights in limited subjects. Very valuable would be for them to identify the reference papers, but also related persons, institutions, meetings, running projects and source of fundings. A molecular biologist would be gratified individuating a

http://www.nlm.nih.gov/mesh/
Towards Bioinformatics Resourceomes

particular database that could provide important data for him/her research. Similar enthusiasm could come from the discovery of an on-line program suitable to perform a key in-silico experiment on that data. A bioinformatician would be more interested in building such database or program. For both of them it could be important to find articles describing related resources. In research it is vital to understand as soon as possible if someone else already had the same promising idea that you have had. A usual recommendation is “not to reinvent the wheel” to avoid waste of time and money. Computer scientists or engineers may be definitely interested in articles describing the technical details of computational artifacts.

Where to Search for?

The most obvious starting point are search engines. But, how many minutes - maybe hours? - does a scientist spend every day making Google and bibliographic searches? Finding the “right” resource is a difficult and time-consuming activity. The WWW is not semantically organized and the nature of resources (which type of resource?) and their semantics (what are they for?) are not clearly identifiable. The problem is analogous with academic publications. The papers can be searched using keywords indexing but also in this case it is still missing a semantic annotation of publications and of the resources therein presented. Of course journals are the primary source of knowledge and information exchange in the scientific communities. Due to the “publish or perish” syndrome, unfortunately also the “bibliome” is growing out of any control. All the articles are available on-line, more often freely accessible thanks to an “Open Access” policy. In a few years, for bioinformatics only, the number of journals has raised from a handful to some tenths, including all the “omics” and post-”omics” subjects which have bioinformatics at their base. It is then clear why it is becoming essential the support of “intelligent” software agents equipped with text-mining and formal reasoning capabilities.

Databases are essential resources in bioinformatics. Since 1996 Nucleic Acid Research (NAR) reserves a special issue to articles presenting biomolecular databases. Figure provides a summary of the number of articles published in these database special issues. The trend is clearly increasing passing from the about 50 articles published in 1996 to the more than 174 published ten years afterward. In the table it is indicated also the number of databases, listed since 1999 in the associated “Molecular Biology Database Collection”. The online collection lists today almost thousand databases organized into the following categories: Nucleotide Sequence Databases, RNA sequence databases, Protein sequence databases, Structure Databases, Genomics Databases (non-vertebrate), Metabolic and Signaling Pathways, Human and other Vertebrate Genomes, Human Genes and Diseases, Microarray
Data and other Gene Expression Databases, Proteomics Resources, Other Molecular Biology Databases, Organelle databases, Plant databases, Immunological databases. Since 2003, a similar issue is provided also for the most relevant Web servers [31]. To have a full overview of this amount of well annotated resources it already constitutes a challenge for the brain of a human being. Nevertheless they represent only a little percentage of the quantity of computational resources constantly presented in the academic community.

When it is unknown what to search for, a better solution could be to navigate through manually curated directories. The quality of the listed resources is certified by annotators but building and maintaining directories represents a big effort. In bioinformatics there are many of such portals, mainly listing resources useful to some Special Interest Group or in very limited domains. Historical are the Pedro’s List (its status in 1995 has been “hibernated” and it is now still available in a mirror site [6]) and the Amos’s WWW links page (now updated as Expasy Life Sciences Directory [7]). A generic bioinformatics directory is available in Google Directory [8], assembled in the collaborative Open Directory Project [9] effort. The Bioinformatics Links Directory [30] today contains almost 1200 curated links to bioinformatics resources, organized into eleven main categories, including all the databases and Web servers yearly listed in the dedicated Nucleic Acids Research special issues. More recent is the Online Bioinformatics Resources Collection [18]. Another difficulty comes from the fact that resources once presented in the literature are not necessary anymore available on the Web [67]. In the age of WWW resources rapidly appear and disappear [22].

---

### Fig. 1. Number of articles yearly published in the Database special issues of NAR and number of databases listed in the Molecular Biology Database Collection

<table>
<thead>
<tr>
<th>Year</th>
<th>Articles</th>
<th>DB listed</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>174</td>
<td>968</td>
</tr>
<tr>
<td>2006</td>
<td>164</td>
<td>838</td>
</tr>
<tr>
<td>2005</td>
<td>137</td>
<td>719</td>
</tr>
<tr>
<td>2004</td>
<td>142</td>
<td>548</td>
</tr>
<tr>
<td>2003</td>
<td>95</td>
<td>386</td>
</tr>
<tr>
<td>2002</td>
<td>94</td>
<td>335</td>
</tr>
<tr>
<td>2001</td>
<td>73</td>
<td>281</td>
</tr>
<tr>
<td>2000</td>
<td>95</td>
<td>226</td>
</tr>
<tr>
<td>1999</td>
<td>86</td>
<td>201</td>
</tr>
</tbody>
</table>

---

2 The New Waves

Here we briefly summarize some recent innovations in Information Technology and their impact in bioinformatics.

2.1 Ontologies

In philosophy, ontology is, since the ancient Greeks, the study of being or existence. In applied computer science recently we witnessed a massive diffusion of ontologies. They are now used to model, by mean of a ontology formal language, the knowledge concerning a particular domain. The concepts of the domain can be precisely defined as well as the relationships between those concepts. The actual objects of the domain are represented as instances of the concepts and it is possible to perform formal reasoning on them, automatically inferring new knowledge.

Bioinformatics and biomedicine undoubtedly contributed to (and benefited of) this generalized diffusion of ontologies (or ontology-like structures [35]) in science. In order to provide some coordination to these efforts the Open Biomedical Ontologies (OBO [10]) consortium was established. The renewed OBO Foundry website lists today more than sixty ontologies, ranging from “Animal natural history” to “Zebrafish anatomy and development”. In particular in bioinformatics the role of ontologies has moved from a niche activity to a mainstream one [8]. According to Bodenreider and Stevens the Gene Ontology [20] has been phenomenally successful. Starting from about 3500 terms in 1998 now it covers around 20000 terms and its controlled vocabulary is adopted in more than 20 databases. Its success is due to the deep and continuous involvement of the molecular biology community in the development and maintenance of the ontology. Other important factors are its clear goals, limited scope and simple structure.

The TAMBIS Ontology [2, 61], represents a remarkable attempt to semantically organize bioinformatics concepts, including resources. The aim of the TAMBIS project was that of providing transparent access to disparate biological databases and analysis tools enabling users to utilize a wide range of resources with the minimum effort. Proton [17] is a ontology prototype for the classification of databases and algorithms for proteomics.

2.2 Web Services

Web Services, represent one of the most used middleware in distributed programming. Thanks to the WSDL [11] descriptors (which characterize the interface of the service) and the SOAP [12] messages (which make the communication possible), Web Services are a great way to integrate distributed
systems. WSDL and SOAP are W3C standards. Orchestration and choreography \cite{21}, with their languages (respectively WSBPEL\footnote{http://www.oasis-open.org/committees/wsbpel/} and WSCI\footnote{http://www.w3.org/TR/wsc110/}), represent another advantage for Web Services composition for the fulfillment of peculiar tasks. The automation, both in the discovery and in the access to the services still remains a problem to solve.

SOAP-based interfaces have been developed for a wide variety of bioinformatics applications. BioMOBY \cite{65} is an international, Open Source Web Service integration project. The MOBY-S system defines an ontology-based messaging standard through which a client (e.g. Taverna \cite{49}) is able to automatically discover and interact with task-appropriate biological data and analytical service providers, without requiring manual manipulation of data formats as data flows from one provider to the next. Also the BioMOBY registry is becoming semantic\footnote{http://semanticmoby.org/}.

2.3 Software Agents

Agent-oriented software development is a software engineering approach for designing complex, flexible applications in terms of agents \cite{64}. Agents can be considered as a distinct kind of software abstraction, in the same way that methods, functions and objects are software abstractions. More specifically, an agent is a high-level software abstraction that provides a convenient and powerful way to describe a complex software entity in terms of its behaviour within a contextual computational environment \cite{46}. An agent is a computer system capable of flexible, autonomous problem-solving actions; it is capable of operating as a stand-alone process, and performing actions without user intervention by maintaining a description of its own processing state and the state of environment in which it is situated \cite{39}.

The kinds of resources available in the bioinformatics domain, with numerous databases and analysis tools independently administered in geographically distinct locations, lend themselves almost ideally to the adoption of a multi-agent approach \cite{46}. Here, the environment is open, distributed and dynamic, with resources entering and leaving the system over time. There are likely to be large numbers of interactions between entities for various purposes, and the need for automation is substantial and pressing. In \cite{46} are reported some significant experiences in adopting agents in bioinformatics, computational and systems biology. Biological systems are complex and consist of a huge set of components interacting with each other and with an external (dynamic) environment. A conceptual framework for engineering an agent society that simulates the behavior of a biological system has been proposed in \cite{15}. The research in this area is today very active and numerous are the attempts of modeling cellular processes and systems with a multi-agents approach.
2.4 Semantic Web

Ontologies and intelligent agents constitute the foundations of the Semantic Web. The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. Whereas the current Web provides links between pages that are designed for human consumption, the Semantic Web augments this with pages designed to contain machine-readable descriptions of Web pages and other Web resources. In the foreseeable future the web of (semantic) links between documents, databases, and programs can provide a new level of interaction among scientific communities. The Semantic Web uses new Web languages based on RDF (the Resource Description Framework), which go beyond the presentation capabilities of HTML. OWL is a language for making ontological statements, developed as a follow-on from RDF and RDFS, as well as earlier ontology language projects including OIL, DAML and DAML+OIL. OWL is intended to be used over the World Wide Web, and all its elements (classes, properties and individuals) are defined as RDF resources, and identified by URI.

A new generation of (Semantic Web) browsers will raise, not only to render the semantic documents visually but also to aggregate(mashup) information referenced by documents and data objects. Based on the semantic browser Haystack, Biodash attempts to aggregate heterogeneous yet related facts and statements concerning drug development into an intuitive, visually descriptive and interactive display. The W3C consortium itself, in its mission to drive the WWW toward a Semantic Web, founded a Special Interest Group (SIG) in Semantic Web for Health Care and Life Science. Bringing together scientists, medical researchers, science writers and computer scientists coming from academia, government, no-profit organizations, and industry, this group intends to facilitate the development of future standards and tools.

A relevant example of Semantic Web for life science is FungalWeb, whose supporting ontology integrates various biological database schemas, web accessible textual resources, domain specific ontologies and experts' knowledge. Another Semantic Web ongoing project in life science (neuromedicine) is SWAN, attempting to develop a practical, semantically structured, web-compatible framework for scientific discussion using Semantic Web technologies. In particular, a public ontology about scientific reasoning and aimed to discovery life-cycle (including concepts like hypotheses, experiments, claims, models, documents) has been represented in RDF to support group sharing and collaboration, as well as personal and community knowledge-based construction. Related to neuroscience, it is also the

---

16 http://www.w3.org/RDF/
17 http://www.w3.org/2004/OWL/
18 http://www.faqs.org/rfcs/rfc2396.html
19 http://www.w3.org/2001/sw/hcls/
Neurocommons project\textsuperscript{20}, which intends to create an Open Source knowledgebase of RDF annotations and biomedical abstracts.

2.5 Grids

Cyberinfrastructures \cite{38, 12} provide today a distributed global environment, permitting multidisciplinary, geographically dispersed, data, knowledge and computation intensive “e-Science”. Among the others, the Grid technology \cite{28} supports virtual communities through sharing of computational and data resources. A very nice metaphor to describe the Grid concept is that of electric power grids \cite{60}. Users can “consume” computer resources like when plugging their electric devices at any wall outlet and getting electricity from collective power plants.

Incorporating concepts and technologies from the Web Services community, the concept of Grid is flowing into that of Services Grid. The convergence of Grid with Web Services is perfectly represented in the UK e-Science myGRID project \cite{62}. Taverna \cite{49} is the myGRID Workflow Management System for bioinformatics.

Another evolution of the Grid is taking place thanks to semantics, similarly to the expected transition from the Web to the Semantic Web \cite{22}. The Semantic Grid is therefore indicated as “the future infrastructure needed to support the full richness of the e-Science vision” \cite{23}. In bioinformatics we may cite the Semantic Grid \cite{23} browser SeaLife \cite{58}. It integrates GOHSE \cite{5} and GOPubMed \cite{25} to dynamically provide appropriate grid services for ontology concepts identified in a text (e.g. in a Web page or in paper abstract).

Also software agents are foreseen in the future of the Grid \cite{29}. Cognitive capabilities typical of agents (e.g. learning, planning, meta-reasoning and task-level coordination) would permit to turn Semantic Grids into Cognitive Grids \cite{33}. An agent-based multilayer architecture for bioinformatics Grids with a semantic index of resources is described in \cite{4}.

3 Intuitive Organization of Domains and Resources into Resourceomes

Scientists are not able to manage and keep track of such a fast evolving universe like the bioinformatics one. Here we present our proposal (Resourceome) for organizing a domain and its resources in an intuitive, machine-understandable way.

3.1 Lost in the Universe

We already underlined the efforts of scientists in searching resources such as databases, programs, papers, or books. To better illustrate our vision we

\textsuperscript{20} http://sciencecommons.org/projects/data
present an example showing how a reader currently approaches the papers of an issue of a typical bioinformatics journal (e.g. Journal Of Biomedical Informatics, volume 39, year 2006, issue 4). Figure 2 part a) presents the Web page hosting the index of the journal issue with a “flat” list of the articles. The only information that can help the reader to decide if a paper can be interesting for his/her research, are its title and authors. Not any classification is provided for the articles and in any case it is rather difficult to summarize the content in the title. Furthermore, the reader could have not any idea of how this knowledge domain is organized and where the papers could be located in an ideal map of the domain. So we can state that this list of papers weakly assists in their retrieval, because of the lack of a semantical approach to the classification of papers.

In Figure 2 part b), we tried to organize the same set of papers in a hierarchy that can intuitively describe the topics treated by the papers. In this way the user is able to recognize the resources (in this example only papers) of interest at first glance, without opening other documents for reading their abstracts in order to understand their content.

This vision is the beginning of a bigger one, that sees the topics the resources concern organized through an ontology. In this way it is possible to give a more understandable description of the domains the papers describe. Another feature of this intuitive vision is the possibility of extending the field of bioinformatics resources also to other kind of entities. In this manner also e.g. programs, databases, scientists, universities, books, and other resources can be clearly described and retrieved. Like the domains, also the different kinds of bioinformatics resources can be described through the hierarchical formalism of ontologies. Besides a perceptible presentation of the reality, this kind of classification is able to be machine-understandable and consequently “reasonable” from “in-silico” scientists [68].

3.2 What Is a Resourceome?

In order to help both scientists and software agents in their awareness of the domain, we raised the exigence of classifying the huge world of bioinformatics resources (resourceome) building a general machine-understandable index. This could emerge bottom-up from the creation of - probably overlapping - Resourceomes (please note here the capital “R”). Organizations, individual scientist and other Resourceomes creators could share, merge and integrate their Resourceomes aiming to a general one. But in any case a Resourceome represents the vision of scientific domain as it is perceived by its creator. We define a Resourceome as structurally composed of two ontologies. One ontology is for the description of the kinds of bioinformatics resources (left side of figure 3) and the other is for the description of the topics that the resources concern (right side of figure 3).

21 www.sciencedirect.com/science/journal/15320464
a) The index of a journal issue in its classical format

![Index of journal issue in classical format](image)

b) The index of a journal issue presented in a more intuitive format

![Intuitive index of journal issue](image)

Fig. 2. Two different visualizations of the same set of articles
- the resource ontology represents the kind of resources existing in the universe of bioinformatics (or in other scientific domains), such as papers, persons, programs, data. From one resource instance it is possible to reach other instances through the semantic relationships defined in the ontology;
- the domain ontology represents the semantic relationships between the concepts of the domain. For example in the domain could find place the different functions performed by a set of programs;
- the concerns relation allows to connect the two ontologies; in particular through this relation, a resource can be connected to the domain topics which it refers to.

Fig. 3. Visualization of a Resourceome: on the left the resource ontology and on the right the domain ontology with resources related to the concepts of the domain

As an example, the paper *Knowledge guided analysis of microarray data* [26], can be represented in a Resourceome by the instantiation of the concept Paper in the resource ontology. Reading the paper an agent - human or software - can then state that it is about “GO clustering”, so an instance of the concerns relation can bind its instance with (the instance of) the topic *Gene Ontology guided clustering* of the domain ontology. The latter could be a child of the concept *Biological knowledge guided clustering*, in turn child of another concept, and so on.

A Resource Ontology

It is not easy to build a universally accepted classification schema for bioinformatics resources, as the assignment of semantics to the concepts of actor and artifact is not a trivial task. A shared ontology is needed to properly annotate resources and for this reason here we give an example of how this task
can be done. We describe a resource ontology that formalizes the basic entities with which scientists interact every day. This essential resource ontology - by definition built on a *is-a* taxonomy - allows to classify bioinformatics resources. Besides expressing particular concepts identifying specific kinds of resources, we introduced also non-hierarchical semantic relationships, to improve automatic inference processes and to understand the dependences between different resources. The first step consists in identifying the main classes of bioinformatics resources that we want to represent. Some domain independent methodologies and directions to properly design high quality and consistent biomedical ontologies can be found in [3], [36] and [69].

In a resource ontology a generic *Resource* can be represented by two general kinds of concepts: *Actor* and *Artifact* (see Figure 4). Intuitively *Actor* represents resources where humans have a protagonist role, e.g. *Person*, with sub-concept *Author*, and *Institution*, which in turn includes *Research Group, Firm, Organization, University, Institute, Department, Division*. As *Artifact* we mean concepts like *Event, Educational Artifact, Project, Informatics Artifact, Methodology, Literature Resource*. In particular as *Event* we consider *Seminar, Meeting and Conference*. *Educational Artifact* resources include *Tutorial, Course and Learning Object* (for e-learning). *Literature Resource* considers *Book, Conference Proceeding, Article and Journal*. Besides literature resource, another important class is that of *Informatics Artifact*. This concept includes *Artifact for Data Representation, Artifact for Knowledge Representation* and *Computational Resource*. Under *Artifact for Data Representation* we have *File* and *Database* which undoubtedly have an historical place in bioinformatics. *Artifact for Knowledge Representation* includes the *Classification System*, such as *Ontology*, and *Semantic Web Prototype*. Another very important branch is *Computational Resource*, which includes *Algorithm* and *Program*. The latter can be specialized into *Stand Alone* and *Web Based* programs, which includes the *Web Service* class of programs.

We felt the necessity to consider also some auxiliary concepts. In particular they permit to take into account data features like format (an instance could be e.g. “FASTA format”) and type (e.g. “DNA sequence” or “integer”). To maintain the ontology as generic and reusable as possible, we hypothesized a domain-specific *Domain Data Type* class. Features like data format and type are becoming essential in bioinformatics data and processes integration. For example, in a bioinformatics workflow management system, it would be very helpful to have formally defined the input and output data type of programs and web services. This will allow to correctly compose and enact them or to suggest to the user the available tools for a specific data type.

In the resource ontology some attributes should be assigned to the root concept, in order to give more specific information on the instances belonging to its sub-concepts. Some examples can be:

- **name**: the name of the instance. This is a functional attribute, i.e. it is possible to have only one name for each instance;
Fig. 4. The is-a hierarchy of the resource ontology
- **description**: a string allowing to give a precise description of the instance. It is possible to have just one description for each instance;
- **author**: the creator of the instance in the Resourceome;
- **creation date**: the date of creation of the instance. Clearly it is possible to have only one creation date for each instance;
- **URI**: the URI of the instance. If it is an URL than the resource can be accessed directly on the Web.

Besides these general attributes, sometimes there is the need of defining other attributes for the description of a particular resource, e.g. **abstract** (which must be functional) for an **Article** resource, or **license** for an **Informatics Artifact**.

![Diagram of non-hierarchical relationships in the resource ontology](image)

**Fig. 5.** Some examples of non-hierarchical relationships in the resource ontology

We suggest also some useful non-hierarchical relationships for the resource ontology (see fig. 5): **Informatics Artifact uses** **Informatics Artifact**; **Actor collaborates with** **Actor** and **creates** **Artifact**; **Author writes** **Literature Resource**. One of the most important relations is **Literature Resource cites** **Literature Resource**. Usually, in the bioinformatics domain **Literature Resource describes** **Resource**. For most of the relations just described it is possible to define inverse ones, e.g. **Literature Resource is cited by** **Literature Resource** or **Literature Resource is written by** **Author**. The collaborates with relation is symmetric.

### 3.3 A Semantic Browser for Resourceomes

Building correct ontologies of the resources and of the domain they concern for an efficient and faithful representation of the reality is an important and
Towards Bioinformatics Resourceomes 29

hard task. Besides this, also the availability of tools for knowledge visualization will undoubtedly helps knowledge understanding and sharing. The huge amount of scientific knowledge needs not only to be machine understandable, but also to be visualized through intuitive graphical representations, like e.g. through an interactive browser. Therefore new tools are required for translating the coded knowledge in a form easily digestible for scientists. With this aim we implemented a prototype of a Semantic Web browser for Resourceomes (Figure 6). The browser, being coded as Java applet on a Web page, can be accessed from every common Web browser.

Fig. 6. The Semantic Web browser for Resourceomes

On the left panel of the browser it is possible to expand and collapse the resource ontology. Through the selection of one or more of its concepts it is possible to choose the types of resources to visualize. Pressing the button “SHOW DOMAIN VIEW”, on the upper part of the right panel will be visualized the part of the domain ontology containing the concepts to which the previously chosen resources are related. Besides the domain ontology portion, also resources instances are shown. They are connected through an arrow to the topic/s of the domain to which they refer to. Customizable icons are used to represent different kinds of resources, e.g. the pdf logo for the papers and a cylinder for databases. The resource instances of the types previously chosen can be visualized in the lower part of the left panel by pressing the button “SHOW INSTANCE TREE”. The resources appear as

http://resourceome.cs.unicam.it:8080/resourceome/resourceomeweb.html
leaves of the instance tree. Once selected a particular resource instance, on
the right panel it is possible to see (by pressing “SHOW INSTANCE VIEW”)
the chosen instance with all its relationships with other resources. The right
clicking a generic concept or instance, opens a contextual menu. It allows
to visualize the attributes of the selected item and to access the resource
itself through its URI (if available). The menu will soon allow also to edit
a Resourceome, for adding new concepts to the domain, new instances of
resources and new relations. In order to distinguish different instances with
the same icon, a tooltip with the resource name appears by passing with the
mouse pointer over a resource icon.

4 Comparison with Related Works

A Resourceome, that can be browsed by human beings through our Seman-
tic Web browser, is formalized into machine readable OWL ontologies that
are available on the Web. Every other compatible Semantic Web browser
would therefore be able to understand and represent it. In comparison to
simple Web textual visualization of ontologies, our semantic browser adds
the intuitive power of graphical representation of domain and resources with
their semantic relationships. Resourceomes can be foreseen also at the heart
of next generation cyberinfrastructures [14]. In the following we compare Re-
sourceomes with other solutions for organizing indexes of resources.

4.1 Resourceomes and Web Directories

A Web directory is a directory on the World Wide Web curated by domain
experts. The main role it plays is the linking to other Web sites and the
categorization of the links. Anyway, a Web directory is not a search engine,
and does not display list of Web pages based on keywords, but on categories
and subcategories. They can be more succesfull than search engines in find-
ing bioinformatics resources but are intended for human consumption only.
Therefore software agents cannot understand them.

4.2 Resourceomes and BioNavigation

BioNavigation [19] aims at providing scientists access to biological data
sources, that may be used to express queries. In BioNavigation a concep-
tual graph represents scientific concepts (e.g. proteins), and a physical graph
represents the related biological resources (i.e. database of proteins). Sci-
ettists express queries at conceptual level and the system returns the related
results. BioNavigation’s physical graph connecting data sources can be com-
pared with our resource ontology and its conceptual graph can be compared
with our domain ontology. The mapping between the two graph is in our
system formalized by the relation concerns.
A Resourceome has a three level view: two conceptual ones (concepts of resources and domain) and a physical one (individuals of resources). In a Resourceome, the arcs representing the relationships between different resources, allow for a better customization. The features of the resource are represented by the properties described at their physical level. The access to the resources can be made through specific relations with other resources. For example, a resource such as a Web Service like ClustalW\textsuperscript{23}, can be characterized by its WSDL descriptor property and by a relation like stub\_with to the WSDL2Java resource. The stub\_with relation specifies that the Web Service resource ClustalW, needs the tool resource WSDL2Java for being accessed. In this way, access to heterogeneous resources is guaranteed. Another important difference between BioNavigation and a Resourceome is that the first concerns only biological data sources, whereas a Resourceome is general purpose and embraces every kind of resource. Furthermore, the ontologies of a Resourceome are published on the Semantic Web.

### 4.3 Resourceomes and Web Services

In a Resourceome, Web Services are just a specific class of resources, whose function can be described in term of domain concepts. Further semantic descriptors which characterize the Semantic Web Service and systems for their interoperability represent a further advantage for the use of those resources.

ServicesSemanticMap \textsuperscript{15} is a project compatible with BioMoby. ServicesSemanticMap provides a system for the registration of Web Services and a visualization interface for ontology and service map exploration. ServiceSemanticMap and BioMoby represent Web Services registries like UDDI\textsuperscript{24}) enriched with semantics. Also in this case, Resourceome can play the same role. In any case, BioMoby and ServiceSemanticMap are dedicated only to resources of Web Services type. The SemBowser \textsuperscript{57} registry is aimed to help glycoproteomics researchers to find the proper Web Service among a number of available ones. It represents an ontology based implementation of the UDDI specification (non W3C), which classify a Web Service according to the task performed and the domain it is associated with.

### 4.4 Resourceomes and TAMBIS

The TAMBIS ontology (TaO) \textsuperscript{2} describes a wide range of bioinformatics tasks and resources, and has a central role within the TAMBIS system. An interesting difference between the TaO and some of the other ontologies reviewed here, is that the TaO does not contain any instances. The TaO only contains knowledge about bioinformatics and molecular biology concepts and their relationships. The instances they represent still reside in an external database. As concepts represent instances, a concept can act as a question.

\textsuperscript{23} http://www.ebi.ac.uk/Tools/webservices/services/clustalw
\textsuperscript{24} http://www.uddi.org
The concept Receptor Protein represents the instances of proteins with a receptor function and gathering these instances is answering that question.

Acknowledgments

This work is supported by the Italian Investment Funds for Basic Research (FIRB) project “Laboratory of Interdisciplinary Technologies in Bioinformatics” (LITBIO) and by the Loccioni Group - University of Camerino Laboratory (LULab). Many thanks to Massimo Fedeli, Sergio Gabrieli, Victor Karmanski, Cristina Lancioni and Luana Leoni for their help.

References

Towards Bioinformatics Resourceomes


A Summary of Genomic Databases: Overview and Discussion

Erika De Francesco¹, Giuliana Di Santo², Luigi Palopoli², and Simona E. Rombo²

¹ Exeura, Via Pedro Alvares Cabrai, Edificio Manhattan, 87036 C.da Lecco, Rende (CS), Italy 
erika.defrancesco@exeura.com
² Università della Calabria, Via Pietro Bucci 41c, 87036 Rende, Italy 
disanto@si.deis.unical.it, palopoli@deis.unical.it, simona.rombo@deis.unical.it

Summary. In the last few years both the amount of electronically stored biological data and the number of biological data repositories grew up significantly (today, more than eight hundred can be counted thereof). In spite of the enormous amount of available resources, a user may be disoriented when he/she searches for specific data. Thus, the accurate analysis of biological data and repositories turn out to be useful to obtain a systematic view of biological database structures, tools and contents and, eventually, to facilitate the access and recovery of such data. In this chapter, we propose an analysis of genomic databases, which are databases of fundamental importance for the research in bioinformatics. In particular, we provide a small catalog of 74 selected genomic databases, analyzed and classified by considering both their biological contents and their technical features (e.g., how they may be queried, the database schemas they are based on, the different data formats, etc.). We think that such a work may be an useful guide and reference for everyone needing to access and to retrieve information from genomic databases.

1 The Biological Databases Scenario

In such an evolving field like the biological one, having continuously new data to store and to analyze seems natural. Thus, the amount of data stored in biological databases is growing very quickly [12], and the number of relevant biological data sources can be nowadays estimated in about 968 units [34]. The availability of so many data repositories is indisputably an important resource but, on the other hand, opens new quests, as to effectively access and retrieve available data. Indeed, the distributed nature of biological knowledge, and the heterogeneity of biological data and sources, can often cause trouble to the user trying specific demands. And, in fact, not only such data sources are so numerous, but they use different kinds of representations, access methods and various features and formats are offered.
In this scenario, a documented analysis of on-line available material in terms not only of biological contents, but also considering the diverse types of exploited data formats and representations, the different ways in which such databases may be queried (e.g., by graphical interaction or by text forms) or the various database schemas they are based on, can be useful to ease users’ activities spent towards fulfilling their information needs. With this aim in mind, we started from the about 968 biological databases listed in [34], where a classification reflecting some specific biological views of data is provided, and we focused on a specific biological database category, that of genomic databases, analyzing them w.r.t. database technologies and tools (which has not been considered in previous works, such as [31, 34]).

As a result of this work, in this chapter we will discuss the 74 genomic databases we have analyzed, specifying the main features they have in common and the differences among them in terms of both biological and database technologies oriented points of views. In particular, we used five different and orthogonal criteria of analysis, that are:

- **typologies of recoverable data** (e.g., genomic segments or clone/contig regions);
- **database schema types** (e.g., Genolist schema or Chado schema);
- **query types** (e.g., simple queries or batch queries);
- **query methods** (e.g., graphical interaction based or query language based methods);
- **export formats** (e.g., flat files or XML formats).

Section 2 is devoted to the illustration of such an analysis, and a subsection for each particular criterium under which the selected genomic databases have been analyzed in detail is included. To facilitate the full understanding of the discussion, the mentioned classification criteria would be also graphically illustrated by some graphs. In this way, given one of the analyzed databases, it would be possible to easily know in a synthetic way all the information captured by the described criteria. An on-line version of such an analysis is available at [http://siloe.deis.unical.it/biodbSurvey/](http://siloe.deis.unical.it/biodbSurvey/), where a numbered list of all the databases we analyzed is provided (as also reported here in the Appendix) and the classifications corresponding to the five criteria illustrated above are graphically reported. In particular, for each criterium, each sub-entry is associated to a list of databases; clicking on one of the databases, a table will be shown where all the main features of that database are summarized (see Section 2.6 for more details).

The two following subsections summarize some basic biological notions and illustrate the main biological database classification commonly adopted in the literature.

### 1.1 Some (Very) Basic Biology

The genome encodes all the information of the *Deoxyribonucleic Acid* (*DNA*), the macromolecule containing the genetic heritage of each living being. DNA
is a sequence of symbols on an alphabet of four characters, that are, A, T, C and G, representing the four nucleotides Adenine, Thymine, Cytosine and Guanine. In genomic sequences, three kinds of subsequences can be distinguished: i) genic subsequences, coding for protein expression; ii) regulatory subsequences, placed upstream or downstream the gene of which they influence the expression; iii) subsequences apparently not related to any function. Each gene is associated to a functional result that, usually, is a protein. Protein synthesis can be summarized in two main steps: transcription and translation. During transcription, the DNA genic sequence is copied into a Messenger Ribonucleic Acid (RNA<sub>m</sub>), delivering needed information to the synthesis apparatus. During translation, the RNA<sub>m</sub> is translated into a protein, that is, a sequence of symbols on an alphabet of 20 characters, each denoting an amino acid and each corresponding to a nucleotide triplet. Therefore, even changing one single nucleotide in a genic subsequence may cause a change in the corresponding protein sequence. Proteins as, more in general, macromolecules, are characterized by different levels of information, related to the elementary units constituting them and to their spatial disposition. Such levels are known as structures. In particular, the biological functions of macromolecules also depend on their three-dimensional shapes, usually named tertiary structures. Moreover, behind the information encoded in the genomic sequence, additive knowledge about biological functions of genes and molecules can be attained by experimental techniques of computational biology. Such techniques are devoted, for example, to build detailed pathway maps, that are, directed graphs representing metabolic reactions of cellular signals [14].

1.2 Biological Database Classification

As pointed out in [31], all the biological databases may be classified, w.r.t. their biological contents, in the following categories (see also Figure[1]):

- **Macromolecular Databases:** contain information related to the three main macromolecules classes, that are, DNA, RNA and proteins.
  - **DNA Databases:** describe DNA sequences.
  - **RNA Databases:** describe RNA sequences.
  - **Protein Databases:** store information on proteins under three different points of view:
    - **Protein Sequence Databases:** describe amino acid sequences.
    - **Protein Structure Databases:** store information on protein structures, that are, protein shapes, charges and chemical features, characterizing their biological functions.
    - **Protein Motif and Domain Databases:** store sequences and structures of particular protein portions (motifs and domains) to which specific functional meanings can be associated, grouped by biological functions, cellular localizations, etc.
• **Genomic Databases**: contain data related to the genomic sequencing of different organisms, and gene annotations;
  - **Human Genome Databases**: include information on the human gene sequencing.
  - **Model Organism Databases (MOD)**: store data coming from the sequencing projects of *model* organisms (such as, e.g., MATDB); they are also intended to support the Human Genome Project (HGP).
  - **Other Organism Databases**: store information derived from sequencing projects not related to HGP.
  - **Organelle Databases**: store genomic data of cellular organelles, such as mitochondria, having their own genome, distinct from the nuclear genome.
  - **Virus Databases**: store virus genomes.
• **Experiment Databases**: contain results of lab experiments; these can be grouped in three main classes:
  - **Microarray Databases**: store data deriving from microarray experiments, useful to assess the variability of the gene expression.
  - **2D PAGE Databases**: contain data on proteins identified by *two-dimensional polyacrylamide gel electrophoresis* (2-D PAGE), an experimental technique used to physically separate and distinguish proteins.
  - **Other Experiment Databases**: store results of specific experimental techniques, such as **Quantitative PCR Primer Database** [23].
• **Immunological Databases**: store information associated to immunologic responses.
• **Gene Databases**: contain sequence data of DNA codifying subsequences, that are, the genes.
• **Transcript Databases**: store the set of transcribed genomic sequences of several organisms (that are, those genomic sequences characterizing organs’ and tissues’ cellular properties).
• **Pathway Databases**: contain information on metabolic reaction networks of cellular signals.
• **Polymorphism and mutation Databases**: report data on polymorphic variations and mutations.
• **Meta-Database**: store descriptions and links about (selected groups of) biological databases;
• **Literature Databases**: these are archives storing papers published on premier biological journals.
• **Other Databases**: store data related to specific application contexts, such as the **DrugBank** [10].

It is worth pointing out that the above classes in some cases overlap. As an example, **EcoCyc** [40] is to be looked at both as a genomic and as a pathway database. Note that this is intrinsic to the structure of the biological context. In fact, biological data sources contain data which can be classified under different perspectives, due to the various origins and goals of the associated repositories.
2 Genomic Databases Analysis

This section focuses on genomic databases, that represent one among the most important biological database categories. Differently from gene databases, containing only coding DNA sequences, genomic databases contain also non coding intergenic sequences.

In the following, we discuss a selected subset of genomic repositories (74 units) and illustrate in detail their differences and common features according
to five criteria of analysis, that are, the typologies of recoverable data, the database schema types, the query types, the query methods and, finally, the export formats.

For each classification criterium, the corresponding classification and the list of associated databases will be also graphically illustrated. In particular, in the graphs that follow, classes are represented by rectangles, whereas circles indicate sets of databases. Databases are indicated by numbers, according to the lists found in the Appendix and at http://siloe.deis.unical.it/biodbSurvey/.

2.1 Recoverable Data

Genomic databases contain a large set of data types. Some archives report only the sequence, the function and the organism corresponding to a given portion of genome, other ones contain also detailed information useful for biological or clinical analysis. As shown in Figure 2, data that are recoverable from genomic databases can been distinguished in six classes:

- **Genomic segments**: include all the nucleotide subsequences which are meaningful from a biological point of view, such as *genes*, *clone/clontig sequences*, *polymorphisms*, *control regions*, *motifs* and *structural features* of chromosomes. In detail:
  - **Genes**: are DNA subsequences originating functional product such as proteins and regulatory elements. All genomic databases contain information about genes.
  - **Clone/contig regions**: are sequences of in vitro copies of DNA regions, used to clone the DNA sequence of a given organism. Examples of databases storing information about clone/contig regions are ArkDB [1], SGD [32] and ZFIN [47].
  - **Polymorphisms**: are frequent changes in nucleotide sequences, usually corresponding to a phenotypic change (e.g., the variety of eye colors in the population). As an example, BeetleBase [3] contains information about polymorphisms.
  - **Control regions**: are DNA sequences regulating the gene expression. A few databases store information about control regions. Among them there are Ensembl [13], FlyBase [42], SGD [32], and WormBase [28].
  - **Motifs (or patterns)**: are specific DNA segments having important functions and repeatedly occurring in the genome of a given organism. Examples of databases containing information about motifs are AureoList [2], Leproma [20] and VEGA [38].
  - **Structural features**: can be further distinguished in *telomeres*, *centromeres* and *repeats*.
    - **Telomeres**: are the terminal regions of chromosomes. Génolevures [33], PlasmoDB [30] and SGD [32] store information about telomeres.
Fig. 2. Recoverable data typologies

- **Centromeres**: represent the conjunction between short arms and long arms of the chromosome. BGI-RiSe [51], PlasmoDB [30], RAD [24] and SGD [32] are databases containing information about centromeres.

- **Repeats**: are repetitions of short nucleotide segments, repeatedly occurring in some of the regions of the chromosome. Among the others, CADRE [7], PlasmoDB [30] and ToxoDB [37] store information about repeats.
• **Maps**: result from projects that produced the sequencing and mapping of the DNA of diverse organisms, such as the Human Genome Project [50]. In particular, *genetic maps* give information on the order in which genes occur in the genome, providing only an estimate of genes distances, whereas *physical maps* give more precise information on the physical distances among genes. GOBASE [45] and BeetleBase [3] are examples of databases storing maps.

• **Variations and mutations**: A nucleotide change is a mutation if it occurs with low frequency in the population (as opposed to polymorphisms). Mutations may cause alterations in the protein tertiary structures, inducing pathologic variations of the associated biological functions. These information are stored in databases such as, for example, FlyBase [42], and PlasmoDB [30].

• **Pathways**: describe interactions of sets of genes, or of proteins, or of metabolic reactions involved in the same biological function. Pathways are stored in Bovilist [5] and Leproma [20], for example.

• **Expression data**: are experimental data about the different levels of expression of genes. The levels of such an expression are related to the quantity of genic product of genes. Information about expression data are stored in, e.g., CandidaDB [6], PlasmoDB [30], SGD [32], and ToxoDB [37].

• **Bibliographic references**: are repositories of (sometimes, links to) relevant biological literature. Most genomic databases contain bibliographic references.

### 2.2 Database Schemas

Most genomic databases are relational, even if relevant examples exist based on the object-oriented or the object-relational model (see e.g., [28, 49]). We distinguished four different types of database schema designed “specifically” to manage biological data, from other unspecific database schemas, which are mostly generic relational schemas, whose structure is anyways independent from the biological nature of data.

Thus, w.r.t. database schemas, we grouped genomic databases in five classes, as also illustrated in Figure. More in detail, we referred to:

• The **Genomics Unified Schema (GUS)** [16]: this is a relational schema suitable for a large set of biological information, including genomic data, genic expression data, protein data. Databases based on this schema are CryptoDB [35], PlasmoDB [30] and TcruziDB [27].

• The **Genolist Schema** [43]: this is a relational database schema developed to manage bacteria genomes. In the Genolist Schema, genomic data are partitioned in subsequences, each associated to specific features. With it, partial and efficient data loading is possible (query and update operations), since the update of sequences and associated features only require accessing and analyzing the interested chromosomic fragments. Databases
A Summary of Genomic Databases: Overview and Discussion

Fig. 3. Database schemas

based on the genolist schema are, for example, AureoList [2], LegioList [19] and PyloriGene [22].

- The Chado Schema [8]: this is a relational schema having an extensible modular structure, and is used in some model organism databases such as BeetleBase [3] and FlyBase [12].

- The Pathway Tools Schema [39]: this is an object schema, used in Pathway/Genome databases. It is based on an ontology defining a large set of classes (there are about 1350 of them), attributes and relations to model biological data, such as metabolic pathways, enzymatic functions, genes, promoters and genic regulation mechanisms. Databases based on this schema are BioCyc [4], EcoCyc [11] and HumanCyc [18].

- Unspecific Schema refers to databases which do not correspond to any of the specific groups illustrated above; in particular, most of them are based on a relational schema, without any special adaptation to manage biological data.

2.3 Query Types

In Figure 4 query types supported by the analyzed databases are illustrated. By simple querying it is possible to recover data satisfying some standard search parameters such as, e.g., gene names, functional categories and others. Batch queries consist of bunches of simple queries that are simultaneously processed. The answers to such queries result as a combination of the answers obtained for the constituent simple queries. Analysis queries are more complex and, somehow, more typical of the biological domain. They consist in retrieving data based on similarities (similarity queries) and patterns (pattern search queries). The former ones take in input a DNA or a protein (sub)sequence, and return those sequences found in the database that are the most similar to the input sequence. The latter ones take in input a pattern $p$ and a DNA sequence $s$ and return those subsequences of $s$, which turn out to be most strongly related to the input pattern $p$. As an example, consider the pattern TATA, that is, a pattern denoting a regulative sequence upstream
the genic sequence of every organism: filling the pattern search query form with it on a database such as CandidaDB [6], and allowing at most one mismatch, 311 genic subsequences extracted from the Candida (a fungus) DNA and containing the TATA box in evidence are returned. Both sequence similarity and pattern search queries exploit heuristic search algorithms such as BLAST [29].

2.4 Query Methods

A further analysis criterium we considered is related to methods used to query available databases, as illustrated in Figure 5. We distinguished four main classes of query methods: text based methods, graphical interaction based methods, sequence based methods and query language based methods.

The most common methods are the text based ones, further grouped in two categories: free text and forms. In both cases, the query can be formulated specifying some keywords. With the free text methods, the user can specify
sets of words, also combining them by logical operators. Such keywords are to be searched for in specific and not a-priori determined fields and tables of the database. With forms, searching starts by specifying the values to look for that are associated to attributes of the database. Multi-attributes searches can be obtained by combining the sub-queries by logical operators. As an example, ArkDB [1] supports both free text and forms queries.

Graphical interaction based methods are also quite common in genomic database access systems. In fact, a large set of genomic information can be visualized by physical and genetic maps, whereby queries can be formulated by interacting with graphical objects representing the annotated genomic data. Graphical support tools have been indeed developed such as, e.g., the *Generic Genome Browser (GBrowser)* [18], which allows the user to navigate interactively through a genome sequence, to select specific regions and to recover the correspondent annotations and information. Databases such as CADRE [7] and HCVDB [17], for example, allow the user to query their repository by graphical interaction.

Sequence based queries rely on the computation of similarity among genomic sequences and patterns. In this case, the input is a nucleotides or amino-acid sequence and data mining algorithms, such as in [29], are exploited to process the query, so that alignments and similar information are retrieved and returned.

Query language based methods exploit DBMS-native query languages (e.g., SQL-based ones), allowing a “low level” interaction with the databases. At the moment, few databases (WormBase [28] and GrainGenes [15]) support them.

### 2.5 Result Formats

In genomic databases, several formats are adopted to represent query results. Web interfaces usually provide answers encoded in HTML, but other formats are often available as well. In the following discussion, we shall refer to the formats adopted for representing query results and not to those used for ftp-downloadable data.

*Flat files* are semi-structured text files where each information class is reported on one or more than one consecutive lines, identified by a code used to characterize the annotated attributes. The most common flat files formats are the *GenBank Flat File (GBFF)* [41] and the *European Molecular Biology Laboratory Data Library Format* [44]. These formats represent basically the same information contents, even if with some syntactic differences. Databases supporting the GenBank Flat File are, for example, CADRE [7] and Ensembl [13], whereas databases supporting the EMBL format are CADRE [7], Ensembl [13], and ToxoDB [37].

The *comma/tab* and the *attribute-value* separated files are similar to flat files, but featuring some mildly stronger form of structure. The former ones, supported for example by BioCyc [4], TAIR [46] and HumanCyc [18],
represent data in a tabular format, separating attributes and records by special characters (e.g., commas and tabulations). The latter ones (supported for example by BioCyc [4], EcoCyc [11] and HumanCyc [18]) organize data as pairs \langle attribute, value \rangle.

In other cases, special formats have been explicitly conceived for biological data. An example is the FASTA format [44], commonly used to represent sequence data (it is exploited in all genomic databases, except for ColiBase [9], PhotoList [21], and StreptoPneumoList [25]). Each entry of a FASTA document consists in three components: a comment line, that is optional and reports brief information about the sequence and the GenBank entry code; the sequence, represented as a string on the alphabet \{A, C, G, T\} of the nucleotide symbols, and a character denoting the end of the sequence.

More recently, in order to facilitate the spreading of information in heterogeneous contexts, the XML format has been sometimes supported. The main XML-based standards in the biological context are the System Biology Markup Language (SBML) [36] and the Biological Pathway Exchange (BioPAX) [26]. SBML is used to export and exchange query results such as pathways of cellular signals, gene regulation and so on. The main data types here are derived from XML-Schema. Similarly, BioPAX has been designed to integrate different pathways resources via a common data format. BioPAX specifically consists of two parts: the BioPAX ontology, that provides an abstract representation of concepts related to biological pathways, and
the BioPAX format, defining the syntax of data representation according to
definitions and relations represented in the ontology. BioCyc [4], EcoCyc [11]
and HumanCyc [18] support both SBML and BioPAX formats.
Figure 6 summarizes the described data formats.

2.6 The On-Line Synthetic Description

We remark that the main goal of our investigation has been that of screening
a significant subset of the available on-line biological resources, in order to
individuate some common directions of analysis, at least for a sub-category of
them (i.e., the genomic databases). Thus, as one of the main results of such
an analysis, we provide a synthetic description of each of the 74 analyzed
databases according to the considered criteria. All such synthetic descriptions
are freely available at http://siloe.deis.unical.it/biodbSurvey/.

![CryptoDB Table]

Fig. 7. An example of the database synthetic descriptions available at
http://siloe.deis.unical.it/biodbSurvey/
In Figure 7 an example of the synthetic description we adopted is illustrated. In particular, a form is exploited where both the acronym and the full name of the database under analysis are showed. We consider the CryptoDB [35] (Cryptosporidium Genome Resources) database, that is, the genome database of the AIDS-related apicomplexan-parasite, as reported in the brief description below the full name in the form. Also the web-link to the considered database is provided, along with its database features. In particular, CryptoDB is a relational DBMS based on the Genomic Unified Schema (illustrated above, in Section 2.2). All features corresponding to the other analysis criteria are reported. In detail, we highlight that CryptoDB contains information on genes, clone/contig sequences, polymorphism and motifs, and that it stores physical maps, pathways and bibliographic references as well. It supports simple queries and also analysis queries based on similarity search, and allows graphical interaction and sequence based query methods. Additional information are contained in the form about the search criteria supported by each database. In particular, CryptoDB can be queried by using both keywords and map positions as search parameters (map positions require the specification of the positions of chromosomic regions). The result formats supported by CryptoDB are HTML (as all the other genomic databases), FASTA and tab-separated files. It allows the user to download data stored in its repository, but not to submit new data, as specified in the last two field (“submit” and “download”) of the form.

3 Concluding Remarks

In this chapter we presented the main results of an analysis we have conducted on 74 genomic databases. Such an analysis has been done mainly focusing on a database technology point of view, by choosing five specific criteria under which all the considered databases have been investigated. Thus, the considered databases have been classified according to such criteria and their synthetic descriptions are given, with the aim of providing a preliminary instrument to access and manage more easily the enormous amount of data coming from the biological repositories.

References

7. Central aspergilus data repository, www.cadre.man.ac.uk
12. Embl nucl. seq. db statistics, http://www3.ebi.ac.uk/Services/DBStats
14. From sequence to function, an introduction to the kegg project, http://www.genome.jp/kegg/docs/intro.html
20. Leproma, http://genolist.pasteur.fr/Leproma
23. Quant. pcr primer db (qppd), http://web.ncifcrf.gov/rtp/gel/primerdb


Appendix

8: CADRE, http://www.cadre.man.ac.uk
9: CampyDB, http://campy.bham.ac.uk
12: ClostriDB, http://clostri.bham.ac.uk
13: coliBase, http://colibase.bham.ac.uk
16: CyanoBase, http://www.kazusa.or.jp/cyano
17: CyanoList, http://genolist.pasteur.fr/CyanoList
19: EcoCyc, http://ecocyc.org
23: Full-Malaria, http://fullmal.ims.u-tokyo.ac.jp
27: GIB, http://gib.genes.nig.ac.jp
36: Lepromatoida, http://genolist.pasteur.fr/Lepromatoida
39: ListiList, http://genolist.pasteur.fr/ListiList
41: MATDB, http://mips.gsf.de/proj/medmar/db
44. **MitoDat**, http://www-lecb.ncifcrf.gov/mitoDat/#searching
45. **MjCyc**, http://maine.ebi.ac.uk:1555/MJNRDB2/server.htm
    MageHome/index.php?webpage=microscope
47. **MolliGen**, http://cbi.labri.fr/outils/molligen
48. **MtDB**, http://www.medicago.org/MtDB
49. **MypuList**, http://genolist.pasteur.fr/MypuList
52. **OGRe**, http://drake.physics.mcmaster.ca/ogre
53. **Oryzabase**, http://www.shigen.nig.ac.jp/rice/oryzabase
54. **PEC**, http://shigen.lab.nig.ac.jp/ecoli/pec
55. **PhotoList**, http://genolist.pasteur.fr/PhotoList
56. **PlantGDB**, http://www.plantgdb.org
57. **PlasmoDB**, http://plasmodb.org
58. **PyloriGene**, http://genolist.pasteur.fr/PyloriGene
60. **RGD**, http://rgd.mcw.edu
61. **SagaList**, http://genolist.pasteur.fr/SagaList
62. **ScoCyc**, http://scocyc.jic.bbsrc.ac.uk:1555
63. **SGD**, http://www.yeastgenome.org
64. **StreptoPneumoList**, http://genolist.pasteur.fr/StreptoPneumoList
65. **SubtiList**, http://genolist.pasteur.fr/SubtiList
66. **TAIR**, http://www.arabidopsis.org
67. **TcruziDB**, http://tcruzidb.org
68. **TIGR-CMR**, http://www.tigr.org/CMR
69. **ToxoDB**, http://toxodb.org
70. **TubercuList**, http://genolist.pasteur.fr/TubercuList
71. **UCSC**, http://genome.ucsc.edu
72. **VEGA**, http://vega.sanger.ac.uk
73. **WormBase**, http://www.wormbase.org
74. **ZFIN**, http://zfin.org
Protein Data Integration Problem

Amandeep S. Sidhu and Matthew Bellgard

WA Centre for Comparative Genomics, Murdoch University, Perth, Australia
{asidhu,mbellgard}@ccg.murdoch.edu.au

Abstract. In this chapter, we consider the challenges of information integration in proteomics from the prospective of researchers using information technology as an integral part of their discovery process. Specifically, data integration, meta-data specification, data provenance and data quality, and ontology are discussed here. These are the fundamental problems that need to be solved by the bioinformatics community so that modern information technology can have a deeper impact on the progress of biological discovery.

1 Introduction

The advent of automated and high-throughput technologies in biological research and the progress in the genome projects has led to an ever-increasing rate of data acquisition and exponential growth of data volume. However, the most striking feature of data in life science is not its volume but its diversity and variability. The biological data sets are intrinsically complex and are organised in loose hierarchies that reflect our understanding of complex living systems, ranging from genes and proteins, to protein-protein interactions, biochemical pathways and regulatory networks, to cells and tissues, organisms and populations, and finally ecosystems on earth. This system spans many orders of magnitudes in time and space and poses challenges in informatics, modelling, and simulation that goes beyond any scientific endeavour. Reflecting the complexity of biological systems, the types of biological data are highly diverse. They range from plain text of laboratory records and literature publications, nucleic acid and protein sequences, three-dimensional atomic structure of molecules, and biomedical images with different levels of resolutions, to various experimental outputs from technology as diverse as microarray chips, light and electronic microscopy, Nuclear Magnetic Resonance (NMR), and mass spectrometry.

Different individuals and species vary tremendously, so naturally biological data does this also. For example, structure and function of organs vary across age and gender, in normal and diseased states, and across species. Essentially, all features of biology exhibit some degree of variability. Biological research is an expanding phase, and many fields of biology are still in the developing stages. Data from these systems is incomplete and often inconsistent. This presents a great challenge in modelling biological objects.
2 Need for Common Languages

Public databases distribute their contents as flat files, in some cases including indices for rapid data retrieval. In principle, all flat file formats are based on the organizational hierarchy of database, entry, and record. Entries are the fundamental entities of molecular databases, but in contrast to the situation in the living cell that they purport to describe, database entries store objects in the form of atomic, isolated, non-hierarchical structures. Different databases may describe different aspects of the same biological unit, e.g. the nucleic acid and amino acid sequences of a gene, and the relationship between them must be established by links that are not intrinsically part of the data archives themselves.

The development of individual databases has generated a large variety of formats in their implementations. There is consensus that a common language, or at least that mutual intelligibility, would be a good thing, but this goal has proved difficult to achieve. Attempts to unify data formats have included the application of a Backus–Naur based syntax (George et al., 1987), the development of an object-oriented database definition language (George et al., 1993) and the use of Abstract Syntax Notation 1 (Ohkawa et al., 1995, Ostell, 1990). None of these approaches has achieved the hoped for degree of acceptance. Underlying the questions of mechanisms of intercommunication between databases of different structure and format is the need for common semantic standards and controlled vocabulary in annotations (Pongor, 1998, Rawlings, 1998). This problem is especially acute in comparative genomics. From the technological point of view, intergenome comparisons are interdatabase comparisons, which means that the databases to be compared have to speak the same language: keywords, information fields, weight factors, object catalogues, etc.

Perhaps the technical problems of standardization discussed in the preceding paragraphs could be addressed more easily in the context of a more general logical structure. As noted by Hafner (Hafner and Fridman, 1996), general biological data resources are databases rather than knowledge bases: they describe miscellaneous objects according to the database schema, but no representation of general concepts and their relationships is given. (Schulze-Kremer, 1998) addressed this problem by developing ontologies for knowledge sharing in molecular biology. He proposed to create a repository of terms and concepts relevant to molecular biology, hierarchically organized by means of ‘is a subset of’ and ‘is member of’ operators.

Existing traditional approaches do not address the complex issues of biological data discussed in earlier sections. However, recent work on ontologies intends to provide solutions to these issues. The term ontology is originally a philosophical term referred to as “the object of existence”. The computer science community borrowed the term ontology to refer to a “specification of conceptualisation” for knowledge sharing in artificial intelligence (Gruber, 1993). Ontologies provide a conceptual framework for a structured representation of the meaning, through a common vocabulary, in a given domain — in this case, biological or medical— that can be used by either humans or automated software agents in the domain. This shared vocabulary usually includes concepts, relationships between concepts,
definitions of these concepts and relationships and also the possibility of defining ontology rules and axioms, in order to define a mechanism to control the objects that can be introduced in the ontology and to apply logical inference. Ontologies in biomedicine have emerged because of the need for a common language for effective communication across diverse sources of biological data and knowledge. In this section, we provide a survey of ontologies used in the biomedical domain. This section covers ontologies that focus on biological terminology.

2.1 The Gene Ontology

In 1998, efforts to develop the Gene Ontology (Lewis, 2004, Ashburner et al., 2001) began, leading ontological development in the genetic area. The Gene Ontology is a collaborative effort to create a controlled vocabulary of gene and protein roles in cells, addressing the need for consistent descriptions of gene products in different databases. The GO collaborators are developing three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. The GO Consortium was initially a collaboration among Mouse Genome Database (Blake et al., 1998), FlyBase (Ashburner, 1993), and Saccharomyces Genome database (Schuler et al., 1996a) efforts. GO is now a part of UMLS, and the GO Consortium is a member of the Open Biological Ontologies consortium to be discussed later in this section. One of the important uses of GO is the prediction of gene function based on patterns of annotation. For example, if annotations for two attributes tend to occur together in the database, then the gene holding one attribute is likely to hold for the other as well (King et al., 2003). In this way, functional predictions can be made by applying prior knowledge to infer the function of the new entity (either a gene or a protein).

GO consists of three distinct ontologies, each of which serves as an organizing principle for describing gene products. The intention is that each gene product should be annotated by classifying it three times, once within each ontology (Fraser and Marcotte, 2004). The three GO ontologies are:

1. **Molecular Function**: This ontology describes the biochemical activity of the gene product. For example, a gene product could be a transcription factor or DNA helicase. This classifies the gene product’s kind of molecule.
2. **Biological Process**: This ontology describes the biological goal to which a gene product contributes. For example, mitosis or purine metabolism. An ordered assembly of molecular functions accomplishes such a process. This describes what a molecule does or is involved in doing.
3. **Cellular Component**: This ontology describes the location in a cell in which the biological activity of the gene product is performed. Examples include the nucleus, telomere, or an origin recognition complex. This is where the gene product is located.

GO is the result of an effort to enumerate and model concepts used to describe genes and gene products. The central unit for description in GO is a **concept**. Concept
consists of a unique identifier and one or more strings (referred to as terms) that provide a controlled vocabulary for unambiguous and consistent naming. Concepts exist in a hierarchy of IsA and PartOf relations in a directed acyclic graph (DAG) that locates all concepts in the knowledge model with respect to their relationships with other concepts.

Eight years have now passed and GO has grown enormously. GO is now clearly defined and is a model for numerous other biological ontology projects that aim similarly to achieve structured, standardized vocabularies for describing biological systems. GO is a structured network consisting of defined terms and the relationships between them that describe attributes of gene products. There are many measures demonstrating its success. The characteristics of GO that we believe are most responsible for its success include: community involvement, clear goals, limited scope, simple, intuitive structure, continuous evolution, active curation, and early use. At present, there are close to 300 articles in PubMed referencing GO. Among large institutional databanks, Swiss-Prot now uses GO for annotating the peptide sequences it maintains. The number of organism groups participating in the GO consortium has grown every quarter-year from the initial three to roughly two dozen. Every conference has talks and posters either referencing or utilizing GO, and within the genome community it has become the accepted standard for functional annotation.

2.2 MGED Ontology

The MGED Ontology (MO) was developed by the Microarray Gene Expression Data (MGED) Society. MO provides terms for annotating all aspects of a microarray experiment from the design of the experiment and array layout, through to preparation of the biological sample and protocols used to hybridise the RNA and analyse the data (Whetzel et al., 2006). MO is a species-neutral ontology that focuses on commonalities among experiments rather than differences between them. MO is primarily an ontology used to annotate microarray experiments; however, it contains concepts that are universal to other types of functional genomics experiments. The major component of the ontology involves biological descriptors relating to samples or their processing; it is not an ontology of molecular, cellular, or organism biology, such as the Gene Ontology. MO version 1.2 contains 229 classes, 110 properties and 658 instances.

2.3 Open Issues in Biomedical Ontologies

Research into different biological systems uses different organisms chosen specifically because they are amenable to advancing these investigations. For example, the rat is a good model for the study of human heart disease, and the fly is a good model for the study of cellular differentiation. For each of these model systems, there is a database employing curators who collect and store the body of biological knowledge for that organism. Mining of Scientific Text and Literature is done to generate a list of keywords that are used as GO terms. However, querying heterogeneous, independent databases in order to draw these inferences...
is difficult: The different database projects may use different terms to refer to the same concept and the same terms to refer to different concepts. Furthermore, these terms are typically not formally linked with each other in any way. GO seeks to reveal these underlying biological functionalities by providing a structured controlled vocabulary that can be used to describe gene products, and is shared between biological databases. This facilitates querying for gene products that share biologically meaningful attributes, whether from separate databases or within the same database.

Association between ontology nodes and proteins, namely, protein annotation through gene ontology, is an integral application of GO. To efficiently annotate proteins, the GO Consortium developed a software platform, the GO Engine, which combines rigorous sequence homology comparison with text information analysis. During evolution, many new genes arose through mutation, duplication, and recombination of the ancestral genes. When one species evolved into another, the majority of orthologs retained very high levels of homology. The high sequence similarity between orthologs forms one of the foundations of the GO Engine.

Text information related to individual genes or proteins is immersed in the vast ocean of biomedical literature. Manual review of the literature to annotate proteins presents a daunting task. Several recent papers described the development of various methods for the automatic extraction of text information (Jenssen et al., 2001, Li et al., 2000). However, the direct applications of these approaches in GO annotation have been minimal. A simple correlation of text information with specific GO nodes in the training data predicts GO association for unannotated proteins. The GO Engine combines homology information, a unique protein-clustering procedure, and text information analysis to create the best possible annotations.

Recently, Protein Data Bank (PDB) also released versions of the PDB Exchange Dictionary and the PDB archival files in XML format collectively named PDBML (Westbrook et al., 2005). The representation of PDB data in XML builds from content of PDB Exchange Dictionary, both for assignment of data item names and defining data organization. PDB Exchange and XML Representations use the same logical data organization. A side effect of maintaining a logical correspondence with PDB Exchange representation is that PDBML lacks the hierarchical structure characteristic of XML data. A directed acyclic graph (DAG) based ontology induction tool (Mani et al., 2004) is also used to constructs a protein ontology including protein names found in MEDLINE abstracts and in UNIPROT. It is a typical example of text mining the literature and the data sources. It can’t be classified as protein ontology as it represents only the relationship between protein literatures and does not formalize knowledge about the protein synthesis process.

Recently, the Genomes to Life Initiative was introduced (Frazier et al., 2003a, Frazier et al., 2003b) close to completion of Human Genome Project (HGP) which finished in April 2003 (Collins et al., 2003). Lessons learnt from HGP will guide ongoing management and coordination of GTL. GTL states an objective: "To correlate information about multiprotein machines with data in major protein databases to better understand sequence, structure and function of protein machines.” This objective can be achieved to some extent only by creating a Generic Protein
Ontology for proteomics, and Specialized Domain Ontologies for each of the major protein families, based on the vocabulary of Generic Protein Ontology.

3 Protein Ontology Elements

We built the Protein Ontology (A. S. Sidhu, Dillon & Chang 2006; A. S Sidhu et al. 2005; A. S. Sidhu et al. 2004) to integrate protein data formats and provide a structured and unified vocabulary to represent protein synthesis concepts. Protein Ontology (PO) provides an integration of heterogeneous protein and biological data sources. PO converts the enormous amounts of data collected by geneticists and molecular biologists into information that scientists, physicians and other health care professionals and researchers can use to more easily understand the mapping of relationships inside protein molecules, interaction between two protein molecules and interactions between protein and other macromolecules at cellular level.

PO consists of concepts (or classes), which are data descriptors for proteomics data and the relationships among these concepts. PO has (1) a hierarchical classification of concepts represented as classes, from general to specific; (2) a list of properties related to each concept, for each class; (3) a set of relationships between classes to link concepts in ontology in more complicated ways then implied by the hierarchy, to promote reuse of concepts in the ontology; and (4) a set of algebraic operators for querying protein ontology instances. In this section, we will briefly discuss various concepts and relationships that make up the Protein Ontology.

3.1 Generic Concepts of Protein Ontology

There are seven concepts of PO, called Generic Concepts that are used to define complex PO Concepts: \{Residues, Chains, Atoms, Family, AtomicBind, Bind, and SiteGroup\}. These generic concepts are reused in defining complex PO concepts. We now briefly describe these generic concepts. Details and Properties of Residues in a Protein Sequence are defined by instances of Residues Concept. Instances of Chains of Residues are defined in Chains Concept. All the Three Dimensional Structure Data of Protein Atoms are represented as instances of Atoms Concept. Defining Chains, Residues and Atoms as individual concepts has the advantage that any special properties or changes affecting a particular chain, residue and atom can be easily added. Family Concept represents Protein Super Family and Family Details of Proteins. Data about binding atoms in Chemical Bonds like Hydrogen Bond, Residue Links, and Salt Bridges are entered into ontology as an instance of AtomicBind Concept. Similarly, the data about binding residues in Chemical Bonds like Disulphide Bonds and CIS Peptides is entered into ontology as an instance of Bind Concept. When defining the generic concepts of AtomicBind and Bind in PO we again reuse the generic concepts of Chain, Residue, and Atom. All data related to site groups of the active binding sites of Proteins are defined as instances of SiteGroup Concept. In PO, the notions classification, reasoning, and consistency are applied by defining new concepts from the
defined generic concepts. The concepts derived from generic concepts are placed precisely into a class hierarchy of the Protein Ontology to completely represent information defining a protein complex.

3.2 Derived Concepts of Protein Ontology

PO provides a description of protein data that can be used to describe proteins in any organism using derived concepts formed from the generic concepts.

Derived Concepts for Protein Entry Details

PO describes Protein Complex Entry and the Molecules contained in Protein Complex are described using Entry Concept and its sub-concepts of Description, Molecule and Reference. Molecule reuses the generic concepts of Chain to represent the linkage of molecules in the protein complex to the chain of residue sequences.

Derived Concepts for Protein Sequence and Structure Details

Protein Sequence and Structure data are described using Structure concept in PO with sub-concepts ATOMSequence and UnitCell. ATOMSequence represents protein sequence and structure and is made of generic concepts of Chain, Residue and Atom. Protein Crystallography Data is described using the UnitCell Concept.

Derived Concepts for Structural Folds and Domains in Proteins

Protein Structural Folds and Domains are defined in PO using the derived concept of StructuralDomains. Family and Super Family of the organism in which protein is present are represented in StructuralDomains by reference to the generic concept of Family. Structural Folds in protein are represented by sub-concepts of Helices, Sheets and Other Folds. Each definition of structural folds and domains also reuses the generic concepts of Chain and Residue for describing the Secondary Structure of Proteins. Helix, which is a sub-concept of Helices, identifies a helix. Helix has a sub-concept HelixStructure that gives the detailed composition of the helix. In this way, PO distinguishes concepts for identification and structure of secondary structures in a protein. Other secondary structures of proteins like sheets and turns (or loops) are represented in a similar way. Sheets have a sub-concept Sheet that identifies a sheet. Sheet has a sub-concept Strands that describes the detailed structure of a sheet. Similarly, turns in protein structure are represented in PO using OtherFolds Concept. Turn is a sub-concept of OtherFolds that identifies a turn and TurnStructure describes its structure. Turns in protein structure are categorized as OtherFolds in Protein Ontology as there are less frequent than Helices and Sheets in Protein Structure.

Derived Concepts for Functional Domains in Proteins

PO has the first Functional Domain Classification Model for proteins defined using the derived concept of FunctionalDomains. Like StructuralDomains, the
Family and Super Family of the organism in which protein is present, are represented in FunctionalDomains by reference to the generic concept of Family. FunctionalDomains describes the Cellular and Organism Source of Protein using SourceCell sub-concept, Biological Functionality of Protein using BiologicalFunction sub-concept, and describes Active Binding Sites in Protein using ActiveBindingSites sub-concept. Active Binding Sites are represented in PO as a collection of various Site Groups, defined using SiteGroup generic concept.

Derived Concepts for Chemical Bonds in Proteins

Various chemical bonds used to bind various substructures in a complex protein structure are defined using ChemicalBonds concept in PO. Chemical Bonds are defined by their respective sub-concepts are: DisulphideBond, CISPeptide, HydrogenBond, ResidueLink, and SaltBridge. They are defined using generic concepts of Bind and Atomic Bind. Chemical Bonds that have Binding Residues (DisulphideBond, CISPeptide) reuse the generic concept of Bind. Similarly the Chemical Bonds that have Binding Atoms (HydrogenBond, ResidueLink, and SaltBridge) reuse the generic concept of AtomicBind.

Derived Concepts for Constraints affecting the Protein Structural Conformation

Various constraints that affect the final protein structural conformation are defined using the Constraints concept of PO. The constraints described in PO at the moment are: Monogenetic and Polygenetic defects present in genes that are present in molecules making proteins described using GeneticDefects sub-concept, Hydrophobic properties of proteins described using Hydrophobicity sub-concept, and Modification in Residue Sequences due to changes in chemical environment and mutations are described using in ModifiedResidue sub-concept.

3.3 Relationships Protein Ontology

Semantics in protein data is normally not interpreted by annotating systems, since they are not aware of the specific structural, chemical and cellular interactions of protein complexes. A Protein Ontology Framework provides a specific set of rules to cover these application specific semantics. The rules use only the relationships whose semantics are predefined in PO to establish correspondence among terms. The set of relationships with predefined semantics is: \{SubClassOf, PartOf, AttributeOf, InstanceOf, and ValueOf\}. The PO conceptual modeling encourages the use of strictly typed relations with precisely defined semantics. Some of these relationships (like SubClassOf, InstanceOf) are somewhat similar to those in RDF Schema (W3C-RDFS schema 2004) but the set of relationships that have defined semantics in our conceptual PO model is too small to maintain the simplicity of the model. The following is a brief description of the set of pre-defined semantic relationships in our common PO conceptual model. SubClassOf relationship is used to indicate that one concept is a specialization of another concept. AttributeOf relationship indicates that a concept is an attribute of another concept. PartOf relationship indicates that a concept is a part of another concept. InstanceOf relationship indicates that an object is an instance of the concept. ValueOf
relationship is used to indicate the value of an attribute of an object. By them- selves, the relationships described above do not impose order among the children of the node. We defined a special relationship called Sequence(s) in PO to de- scribe and impose order in complex concepts defining Structure, Structural Folds and Domains and Chemical Bonds of Proteins.

Fig. 3. Class Hierarchy of Protein Ontology
3.4 Protein Ontology as a Structured Hierarchy

Protein Ontology consists of a hierarchical classification of concepts discussed above represented as classes, from general to specific. In PO, the notions classification, reasoning, and consistency are applied by defining new concepts from defined generic concepts. The concepts derived from generic concepts are placed precisely into the class hierarchy of Protein Ontology to completely represent information defining a protein complex, as depicted in the figure above. More details about Protein Ontology are available on the website (http://www.proteinontology.info/).

4 Summary

An ontology for Protein Domain must contain terms or concepts relevant to protein synthesis, describing Protein Sequence, Structure and Function and relationships between them. In this chapter, we address this issue by providing clear and unambiguous definitions of all major biological concepts of the protein synthesis process and the relationships between them through Protein Ontology or PO (Sidhu et al., 2007). PO provides a unified controlled vocabulary both for annotation data types and for annotation data. As complete protein information has to be obtained from diverse protein sources, it becomes necessary to compose ontology (corresponding to the diverse sources) that can serve the proteomics domain.

Instead of creating a new ontology right from scratch containing the terms that are new to the proteomics domain, it is preferable to create an ontology by reusing terms of existing protein data sources and composing them to form a protein ontology. The information in protein data sources change rapidly. Organizations introduce new terminology or expand the terminology they use in the data they publish. When such an update occurs, any ontology derived from these data sources also needs to change. An ideal solution to the synchronization problem between protein data sources and protein ontology when protein data sources are updated is to have a tool that automatically triggers a systematic update to the ontology when its sources change. Such a tool requires an algebraic foundation based upon which the updates can be systematically handled.

References


Pennisi, E.: Genome Data Shake Tree of Life. Science 280, 672–674 (1998)


Multimedia Medical Databases

Liana Stanescu, Dumitru Dan Burdescu, and Marius Brezovan

University of Craiova, Faculty of Automation, Computers and Electronics, Software Engineering Department
stanescu@software.ucv.ro, burdescu@software.ucv.ro, brezovan_marius@software.ucv.ro

1 General Overview

Nowadays there is an explosion of multimedia information. A huge quantity of static and video images has been stored on the Internet. A large number of images stored on different media were converted to digital format. For example, TV images and newspapers have been converted to digital form, making an easy task their processing, distribution and storage.

More than 2700 digital pictures are made in every second (in total 85 billion images yearly). For example, PhotoWorks includes tens of millions of images on its web site. The common images are completed by special purpose images, like medical images with an estimation of 2 billion per year.

The number of images will be increasing because of the tendency for digital (television, movies) and because everybody will have access to everything. The world production of digital information in 2007 is estimated to be more than 109 GB (250 MB for each man on the planet). It is estimated that in the next 10 years, each of us will manipulate terabytes of information (video, static images, music, photos and documents) every day.

In the medical domain, in the diagnosis, treatment and research processes, traditional alphanumerical information (patient personal data, diagnosis, results for the analysis and investigations) and a large quantity of images are accumulated. The tendency is to digitalize the images. In present there are used a large variety of digital image modalities: film scanners, computed tomography (CT), positron emission tomography (PET), single positron emission computed tomography (SPECT), ultrasounds, magnetic resonance imaging (MRI), digital subtraction angiography (DSA) and magnetic source imaging (MSI) [105].

It is considered that the cardiology and radiology are the medical domains that produce the highest number of images. Also the endoscopy images are produced in significant quantities. For example, the University Hospital of Geneva gathers more than 13,000 images daily, from more than 30 types of medical devices. Besides that, there are many other images stored on other media. But the largest volume of image data is produced in the hospitals from the United States where digital images represent 30% and the other 70% represent images acquired in
conventional X-rays and digital luminescent radiography modalities. The X-ray films can be digitized with different tools [105].

The storage of medical images in digital format makes an easy task their transferring from one device to another and enhances the achieving and manipulation process, in a useful and novel way.

Medical image management plays now an important role in designing and implementing medical information systems. It is needed to introduce new methods for representation, manipulation and search for multimedia information. The development of picture archiving and communications systems (PACS) created a unified structure for management of acquisition, storing, communication and display of image data, but they don’t provide enough flexibility in sharing and utilization of information associated to image [105]. As a result, the solution to all these problems is the development of database management systems that enhance the management and complex querying of medical multimedia information. Using these systems there is the possibility to integrate different medical informatics systems that are geographically dispersed, lowering the costs and enhancing the medical act. Managing and querying these large collections of medical images and alphanumerical information is a difficult task. The most efficient way for solving these problems is to have a multimedia database management system with the following characteristics [48, 61]:

- Support for multimedia data types;
- Possibility to manage a large number of multimedia objects;
- Hierarchical storage management;
- Conventional database capabilities;
- Possibility to retrieve multimedia information considering attributes or characteristics extracted from multimedia information and finding multimedia objects that have these types of characteristics in common; this technique is called content-based retrieval

Content-based visual retrieval can be implemented taking into consideration primitive visual characteristics (color, texture, shape), logical characteristics (object identity), or abstract characteristics (scene semantics). The easiest way for implementation is to use primitive characteristics as color, texture and shape [18, 23, 86].

1.1 A General Model for Information Retrieval

A general model for content-based retrieval is presented in figure 1.1 [60]. The information from the database is first pre-processed to extract specific characteristics and other semantic characteristics and then it is indexed based on these things. For information retrieval, the user’s query is processed and the main characteristics are extracted. These characteristics of the query are then compared with the characteristics of each record in the database. In the end, the user will see the information with the characteristics that are most closely to the query’s characteristics.
In the presented model, two of the most important features, are:

- Records indexing in the database
- Similarity measurement between query and the information in the database

The indexing is defined as a mechanism that reduces the searching space without losing important information. Like in the traditional databases management systems, indexing is necessary to increase the speed of the retrieval operation. It is also a very important operation in the multimedia databases where large amount of information need to be stored. The retrieval operation based on direct comparison between multimedia information records and the query, it is a slow process. As a result, it is important for multimedia information to have attributes that store the relevant information. These attributes must be compact and fast when they are used for retrieval. It is hard to find these attributes because multimedia information does not have a specific syntax or semantic.

The second important feature for information retrieval is the measure of similarity. The attributes that will be used for comparing must have properties like:

- Easy computation
- Correspondence to the human thinking

The second property is very important because the multimedia information retrieval is based on similarity and not on exact matching. The results of the retrieval are returned to the user in decreasing order of the similarity. All returned objects are considered to be relevant or similar to the query object. That is the reason why it is ideally considered that the similarity computing to be according to the human mind: the object considered to be similar by the human mind must be similar according to the similarity computing.

1.2 Content-Based Visual Query – Problem Definition

The objective of the content-based visual query is to search and retrieve in an efficient manner those images from the database that are most appropriate to the
image considered by the user as query [83]. The content-based visual query differs from the usual query by the fact it implies the similarity search.

There are two forms of content-based visual query [83, 86]:

- The k-nearest neighborhood query – that retrieves the most appropriate k images with the query image
- Range query – that retrieves all the images that respect a fixed limit of the similarity between the target and the query image.

Next it is given the difference between the two utilization of the content-based visual query, namely: content-based image query and content-based region query.

In the case of the content-based image query with the k-nearest neighborhood, it is imposed to solve the following problem [83, 86]:

**Being given a collection C of N images and a characteristics dissimilarity function v_r, it is required to find the k images \( \alpha_T \in C \) with the smallest dissimilarity, \( v_r(\alpha_Q, \alpha_T) \) taking into consideration the query image \( \alpha_Q \).**

In this case, a query always returns k images, which are usually sorted, in the ascending order of the dissimilarity, taking into consideration the query image.

In the case of the content-based image query limited to a certain domain, it is imposed to solve the problem [83, 86]:

**Being given a collection C of N images and a characteristics dissimilarity function v_r, it is required to find the images \( \alpha_T \in C \) such that \( v_r(\alpha_Q, \alpha_T) \leq \sigma_f \), where \( \alpha_Q \) is the query image, and \( \sigma_f \) is a limit for the characteristics similarity.**

In this case, the query returns a certain number of images, in function of the \( \sigma_f \) limit.

In a content-based region query, the images are compared on their regions. In the first step of the query, there are made content-based visual queries on the regions, and not on the images. In the final step of the query, there are determined the images corresponding to the regions and there is computed the total distance between the images by the weighting of the distances between regions. In the case of this type of query it is imposed to solve the following problem [83, 86]:

**Being given a collection C of N images and a features dissimilarity function v_f, is required to find the k images \( \alpha_T \in C \) that have at least R regions such that \( v_f(\alpha_Q^R, \alpha_T^R) \leq \sigma_f \), where \( \alpha_Q^R \) is the query image with R regions, and \( \sigma_f \) is a limit for the characteristics similarity.**

The content-based visual query may be improved by adding the spatial information to the query. So, the total measure of the dissimilarity takes into consideration both the features values (color, texture), and the spatial values of the regions.

In present, there are developed techniques for spatial indexing that allow the images retrieval based on their objects localization. These approaches compares images were there were defined a-priori regions and objects.

There are two types of spatial indexing, namely: relative and absolute.

In the case of spatial relative indexing, the images are compared based on their relative symbols locations. The following problem has to be solved [83, 86]:
Being given a collection $C$ of $N$ images, it is required to find all images $\alpha_r \in C$ that have at least $R$ symbols existing in the same spatial arrangement as the $R$ symbols of the query image.

In the case of the absolute spatial images query it have to be solved the following problem [83, 86]:

Being given a collection $C$ of $N$ images and a spatial distance function $v_f$, it is required to find the most appropriate $k$ images $\alpha_r \in C$ that have at least $R$ regions such that $u(\alpha_Q^R, \alpha_T^R) \leq \sigma$, where $\alpha_Q^R$ is the query image with $R$ regions, and $\sigma$ is a limit for the spatial distance.

The most powerful images retrieval system must allows queries where are specified both the visual features and spatial properties for the desired images. Such query offers to the user the possibility to control the selection of regions and attributes which are the most important in the similarity computation.

1.3 The Need for Content-Based Visual Query in Multimedia Medical Databases

The directions where content-based retrieval is needed in medical multimedia databases were fully specified. They are [67, 69]:

- Diagnostic aid – from the conversation with the doctors, the following situation appears frequently: the doctor visualizes a medical image, he/she cannot establish the diagnosis exactly, he is aware of the fact that he has seen something similar before but doesn’t have the means to search for it in the database; the problem can be solved establishing that image as query image and the content-based image query will provide the similar images from the database; it is very likely that among the retrieved images should be the searched image together with its diagnosis, observations, treatment; so the content-based image query can be directly used in the diagnosis process;

- Medical teaching – there is a series of applications for content-based visual query including other ways for access (text-based, hierarchical methods). Students can see the images in the database and the attached information in a simple and direct manner: they choose the query image and they see the similar images; this method stimulates learning by comparing similar cases and their particularities or comparing similar cases with different diagnosis;

- Medical research – using content-based visual query in this area brings up similar advantages for medical teaching. It can be used, for example, for finding certain types of images to be included in a study, for finding misclassified images, etc.;

- PACS and electronic records for patients – the integration of the content-based visual retrieval techniques and the simple text-based retrieval in PACS and the electronic patients’ record might bring important benefits in clinical applications. The information that is stored in the image can be used together with the alphanumerical information.
1.4 Content-Based Image Retrieval Systems

There is in present a series of database management systems capable to manage different types of media. Many of these systems permits indexing and searching for multimedia information taking into account only structured information, using traditional techniques. They work well only with short numeric and alphanumeric arrays. This includes traditional queries of the databases based on alphanumerical arrays. They cannot be used for multimedia information indexing and retrieval. That is why the researchers studied the possibility to create new systems that satisfy the high demands of the multimedia information.

An exception is Oracle Multimedia (Formerly Oracle inter-Media), a feature of Oracle Database that enables the efficient management and retrieval of image, audio, and video data. Oracle Multimedia has knowledge of the most popular multimedia formats and makes automate metadata extraction and basic image processing [13, 50, 73].

Most of the medical informatics systems use for multimedia medical databases management traditional database management servers (MySQL, MS SQL Server, Interbase, and Oracle). There have been implemented alternative methods for content-based visual retrieval, taking into consideration different characteristics like color, texture and shape.

For medical multimedia databases content-based query it is generally used the method called QBE (Query by example). It implies the selection of an image or a region as a query image (region). For improving the results, the simple text-based query and content-based visual query can be combined [67].

In general, the images are represented in the databases by automatically extracted visual features that are supposed to correspond to the visual image content or the way we perceive it. The features mainly used for image retrieval, are [30, 67]:

- Grey levels and color descriptors, in a local or global fashion
- Texture descriptors
- Shapes of segmented objects

Content-based retrieval has been investigated in a number of important projects. It can be mentioned the QBIC project from IBM [1, 26, 103] and Virage, a commercial project for content-based retrieval [103]. Most of the projects are academically projects: Photobook system from MIT [103], MARS (Multimedia Analysis and Retrieval System) developed to the University of Illinois [79, 103], Chabot system for image retrieval [101], WebSeek system, VisualSEEk and SaFe implemented to the University of Columbia [83, 84, 87, 103]. Using higher-level information, such as segmented parts of the image for queries, was introduced by the Blobworld system [9].

A system that is available free of charge is the GNU Image Finding Tool (GIFT). Some systems are available as demonstration versions on the web such as Viper, WIPE or Compass [67].
Generally, these systems have a similar architecture that includes: modules for medical characteristics extraction from images and their storage in the databases, modules for content-based retrieval taking into consideration the extracted characteristics and the user interface [67].

Although the number of the medical informatics systems that implement efficiently the content-based retrieval process is high, it is used in practice only a small number of them. An example is IRMA project that brings important contributions in two research fields [44, 98]:

- Automated classification of radiographs based on global features with respect to imaging modality, direction, body region examined and biological system under investigation.
- Identification of image features that are relevant for medical diagnosis. These features are derived from a-priori classified and registered images.

The system can retrieve images that are similar to a query image taking into consideration a selected set of features. The research was done on image data consists of radiographs, but will be extended on medical images from arbitrary modalities.

Another important CBIR system for the domain of HRCT (High-resolution Computed Tomography) images of the lung with emphysema-type diseases, is ASSERT [77, 82]. It was developed at Purdue University in collaboration with the Department of Radiology at Indiana University and the School of Medicine at the University of Wisconsin. Because the symptoms of these diseases can drastically alter the appearance of the texture of the lung, can vary widely across patients and based on the severity of the disease, ASSERT system characterizes the images using low-level features like texture features computed from the co-occurrence matrix of the image. The retrieval is performed hierarchically. At the first level the disease category of the query image is predicted. At the second level, the most similar images to the query image that belong to the predicted class are retrieved and displayed to the user.

Also, it must be mentioned the MedGIFT system, implemented to the University Hospital from Geneva [69]. It was developed to work together with CasImage, a radiological teaching file that has been used in daily routine for several years now. The system works with more than 60,000 images from more than 10,000 medical cases. The database is available on the Intranet of the hospital, with a smaller database being publicly available via Internet and MIRC.

The system contains modules for image feature extraction, feature indexing structures and a communication interface called MRML (Multimedia Retrieval Mark-up Language).

MedGIFT uses techniques from text retrieval such as [69]:

- Frequency-based feature weights
- Inverted file indexing structures
- Relevance feedback mechanisms
Four feature groups represent the image content [69]:

- Local and global texture features based on responses of Gabor filters;
- Color/grey scale characteristics on a global image scale and locally within image regions.

The interface allows for an easy integration into applications such as [69]:

- Teaching file
- Document management systems
- Tools for diagnostic aid.

At the Software Engineering Department, Faculty of Automation, Computers and Electronics (A.C.E.) Craiova an academic software tool with multi-threaded client/server architecture was developed. MIR (Medical Image Retrieval) includes the following functions [92]:

- The processing of the DICOM standard files provided by the medical tools in order to extract the alphanumeric data and images
- Storing the resulting data in a database
- Extracting the color feature from the image (the color histogram and the binary color set resulting from the HSV color space quantized to 166 colors)
- Extracting the color texture feature from the color images (using the co-occurrence matrices and the Gabor filters)
- Detecting color regions and storing the new data in the database
- The simple text-based query of the database
- The content-based visual query of the database on color and color texture features
- The content-based region query of the database

The medical personnel taking into consideration aspects like execution speed, retrieval quality and new necessary options test the software tool now. The MySql database contains 960 color images from the digestive area gathered with an endoscope.

Also, many experimental studies in the field of content-based visual retrieval on medical databases with color endoscopic images were effectuated in collaboration with University Hospital and Filantropia University Hospital from Craiova.

There is a large variety of applications and studies of content-based visual retrieval that takes into consideration the images from different medical departments. Most of them use databases with images produced in radiology departments, but there are many other departments where this type of algorithms have been implemented. Some of these categories of images are [67]:

- Dermatologic images
- Cytological specimens
• Pathology images have often been proposed for content-based access as the color and texture properties can relatively easy be identified; the pathologist can use a CBIR system instead of books when searching for reference cases
• Histopathology images
• Histology images
• Cardiology
• Within the radiology department, mammography are one of the most frequent application areas
• Ultrasound images of the breast and other ultrasound images
• High resolution computed tomography (HRCT) scans of the lung
• Endoscopic images from digestive area

It must be mentioned that all these applications and studies have been implemented using multimedia medical databases that are extremely varied in size and quality [67]. It starts from tens or hundreds, ending with thousands. The results of these studies are more solid as the dimension of the database is higher and if the images are acquired from the investigation and diagnosis process of the patients. One of the biggest databases used in studies use only simulated images. Although these simulated images are easy and cheap to obtain, their use for any qualitative assessments is more than questionable [67].

Databases that have been used in content-based visual retrieval study have only few tens or hundreds of medical images and it is considered to be too small for delivering any statistically significant measurements [67].

2 DICOM Standard

2.1 Introduction

Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It specifies a file format definition and a network communications protocol that uses TCP/IP to communicate between systems [19].

DICOM standard brings advantages like [105]:

• DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format
• It enables the integration of scanners, servers, workstations, printers, and network hardware from multiple manufacturers into a picture archiving and communication system.

DICOM has been widely adopted by hospitals, but also by doctors’ offices in smaller applications.

American College of Radiology (ACR) and National Electrical Manufacturers Association (NEMA) developed DICOM standard [19]. There were three steps in their development [105]. Their first standard, ACR/NEMA 300, was released in
1985, after the joining of the two organizations and formed a standard committee. In that period only the computed tomography and MRI devices could decode the produced images and the radiologists wanted to use the images for dose planning for radiation therapy. The initial goal in developing a standard for the transmission of digital images was to enable users to retrieve images and associated information from digital imaging equipment in a standard format that would be the same across multiple manufacturers. In 1988 the second version was released with improvements. The third version appeared in 1992, when new service classes were defined, network support added and the Conformance Statement was introduced.

Officially, the latest version of the standard is still 3.0, however, it has been constantly updated and extended since 1992. Instead of using the version number the standard is often version-numbered using the release year, like “the 2007 version of DICOM”.

2.2 The Organization of DICOM Files

A DICOM file has the following structure [19, 20, 21, 78, 55]:

- A preamble of 128 bytes
- Prefix (4 bytes) that stores the letters ‘D’, ‘I’, ‘C’, ‘M’ which represent the signature of the DICOM file
- Data Set, which stores a set of information such as: patient name, type of image, size of the image, etc.
- Pixels that compose the image(s) included into the DICOM file.

Figure 2.1 shows a small part of a DICOM file [22]. In figure 2.2, the structure of the DICOM file is sketched.

The Data Set is composed of a number of Data Elements. It represents an instance of a real world information object and the Data Elements contain the encoded values of attributes of that object [55].

A Data Element is composed of several fields [55]:

Data Element Tag – identifies the information in a unique way. This tag contains two parts: a Group Number (2 bytes) and an Element Number (2 bytes). For example, in (0010, 0020) tag the Group Number is 0010 and the Element Number is 0020. It is important the group with the number 0002 and the element with the number 0010 from this group which represent the Transfer Syntax Unique Identifier. The Transfer Syntax UID defines the byte order for raw data. The integer values can be stored using the Big Endian or the Little Endian ordering.

Little Endian byte ordering is defined as follows: in a binary number consisting of multiple bytes (e.g. a 32-bit unsigned integer value, the Group Number, the Element Number, etc.), the least significant byte is encoded first; with the remaining bytes encoded in increasing order of significance. In a character string consisting of multiple 8-bit single byte codes, the characters will be encoded in the order of occurrence in the string (left to right). Big Endian byte ordering differs
First 128 bytes: unused by DICOM format
Followed by the characters 'D','I','C','M'
This preamble is followed by extra information e.g.:

```
0002,0000,File Meta Elements Group Len: 132
0002,0001,File Meta Info Version: 256
0002,0010,Transfer Syntax UID: 1.2.840.10008.1.2.1.
0008,0000,Identifying Group Length: 152
0008,0050,Modality: MR
0008,0070,Manufacturer: MRlcor
0018,0000,Acquisition Group Length: 28
0018,0050,Slice Thickness: 2.00
0018,1020,Software Version: 4.6.64.37
0028,0000,Image Presentation Group Length: 148
0028,0002,Samples Per Pixel: 1
0028,0004,Photometric Interpretation: MONOCHROME2.
0028,0008,Number of Frames: 2
0028,0010,Rows: 109
0028,0011,Columns: 91
0028,0030,Pixel Spacing: 2.00\%2.00
0028,0100,Bits Allocated: 8
0028,0101,Bits Stored: 8
0028,0102,High Bit: 7
0028,0103,Pixel Representation: 0
0028,1052,Rescale Intercept: 0.00
0028,1053,Rescale Slope: 0.00092157
7FE0,0000,Pixel Data Group Length: 13850
7FE0,0010,Pixel Data: 13838
```

Fig. 2.1. DICOM file – an example

Fig. 2.2. DICOM file structure

from the little Endian byte ordering by the fact that the most significant byte is encoded first.

In the Data Set, Data Elements are arranged in an increasing order of the Tag Number and they appear only once.

**Value Representation (VR)** - describes the type of data and the size for the value contained in Data Element. It is an array of chars stored in 2 bytes. VR for a Data Element Tag is defined in Data Dictionary, and the array of chars is encrypted using the default array of chars defined in DICOM standard. Some of the available value representations are: PN (Person name), TM (Time), AS (Age String) and DA (Date). The VR may or may not be explicitly encoded in the data
set. When the Explicit VR function is used, four consecutive fields compose Data Element: Data Element Tag, VR, Value Length and Value.

An example of a Data Element with an Explicit VR, such as would be the case for data type OB, OW, SQ, or UN is shown in figure 2.3.

An example of a Data Element with an Explicit VR, such as would be the case for data types other than OB, OW, SQ, or UN is shown in figure 2.4.

When using the Implicit VR structure, the Data Element contains three consecutive fields: Data Element Tag, Value Length, and Value (figure 2.5). If the Value Field has an Explicit Length then the Value Length Field contains a value equal to the length (in bytes) of the Value Field. Otherwise, the Value Field has an Undefined Length and a Sequence Delimitation Item marks the end of the Value Field.

Value Length (VL) - either a 16 or 32-bit (dependent on VR and whether VR is explicit or implicit) unsigned integer containing the Explicit Length of the Value Field as the number of bytes (even) that makes up the Value. It does not include the length of the Data Element Tag, Value Representation, and Value Length Fields, or a 32-bit Length Field set to Undefined Length (FFFFFFFF).

Value Field - an even number of bytes containing the Value(s) of the Data Element. The data type of Value(s) stored in this field is specified by the Data Element's VR.

Value Multiplicity - specifies how many values with this VR can be placed in the Value Field.

<table>
<thead>
<tr>
<th>Group Number</th>
<th>Element Number</th>
<th>Value Representation</th>
<th>Reserved</th>
<th>Value Length</th>
<th>Value Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bytes</td>
<td>2 bytes</td>
<td>2 bytes</td>
<td>2 bytes</td>
<td>4 bytes</td>
<td>&quot;Value Length&quot; bytes</td>
</tr>
</tbody>
</table>

**Fig. 2.3.** Data Element with an Explicit VR

<table>
<thead>
<tr>
<th>Group Number</th>
<th>Element Number</th>
<th>Value Representation</th>
<th>Value Length</th>
<th>Value Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bytes</td>
<td>2 bytes</td>
<td>2 bytes</td>
<td>2 bytes</td>
<td>&quot;Value Length&quot; bytes</td>
</tr>
</tbody>
</table>

**Fig. 2.4.** Data Element with an Explicit VR

<table>
<thead>
<tr>
<th>Group Number</th>
<th>Element Number</th>
<th>Value Length</th>
<th>Value Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 bytes</td>
<td>2 bytes</td>
<td>4 bytes</td>
<td>&quot;Value Length&quot; bytes</td>
</tr>
</tbody>
</table>

**Fig. 2.5.** Data Element with an Implicit VR
2.3 Extracting Data from DICOM File

In order to extract data from the DICOM file, we are taking into account every tag from the DICOM dictionary. The tag will be searched in the file and in case of finding it the corresponding value will be extracted.

The steps of extracting information from DICOM files, designed at Software Engineering Department, Faculty of Automation, Computers and Electronics, Craiova, are:

2. Establishing the type of VR (ExplicitVR or ImplicitVR). This information is given by the UID (Unique Identifier), information stored in the value field corresponding to the Transfer Syntax Tag.
3. Establishing the Byte Ordering (BigEndian or LittleEndian). The information is also given by UID, stored in the value field of the same Transfer Syntax Tag. The DICOM standard contains all the values that UID can have.
4. Searching a tag in DICOM file according to the VR type and ByteOrdering
5. Value extraction of the corresponding found tag.

DICOM Standard contains over 20 types of binary data or ASCII. The type of information stored in the value field is given by VR. In accordance with this type will be extracted strings, integer or byte type information.

Next it is described the problem of image extracting from the standard DICOM files, taking into account the method of compression that was used: RLE, JPEG. A single DICOM object can contain only one image, but that image may have multiple “frames”, allowing storage of cine-loops or other multi-frame data.

The images from DICOM files can be classified using several criteria:

1. The number of images stored in a file: single frame or multi-frame.
2. Number of bits per pixel: 8 bits, 12 bits, 16 bits or 24 bits.
3. Compression: without compression (raw) or with compression (RLE JPEG, JPEG Lossless, JPEG 2000). LZW (zip) compression can be used for the whole data set (not just the pixel data) but is rarely implemented.

In the images without compression, the extraction of pictures is made pixel-by-pixel, taking into account the number of bits stored for each pixel and the photometric interpretation (for monochrome images a pixel is stored using maximum 2 bytes and for color images, a pixel is stored using 3 bytes). In the images that use compression it is necessary a decompression algorithm before saving.

The pseudo code for retrieving the frames is:

```
Set variable number to 0
Loop until all frames are retrieved
  Set file dimension as Rows*Columns*SamplePerPixel
  Read all file dimension pixels from file starting With (Header Length + number* file dimension)
```
If MONOCHROME image
    Save as image
    Store image using GIF or JPEG image format
    Return
End If
If PALETTE COLOR image
    Get the Palette tables (one for red values, one for green values and one for blue values)
    Get Pixels color from the Palette.
    Save as image
    Store image using GIF or JPEG image format
End If
If RGB image (24 bits)
    Get red, green and blue values
    Compute the color using the formula:
        \[
        \text{Color} = (255 << 24) | ((0xff\&r) << 16) | ((0xff\&g) << 8) | (0xff\&b)
        \]
    Save as image
    Store image using GIF or JPEG image format
End If
End Loop

2.4 DICOM Viewers

Taking into consideration that the computer does not recognize the DICOM files, there were created software tools that allow viewing the images included in the standard DICOM files and applying some transformations to them (zoom in, zoom out, increasing/decreasing brightness and contrast). The software tools are developed for Windows operating system, Linux operating system, or for both. There are implemented a high number of applications, many of them being free. Some of these instruments, are: EzDICOM, MRIcro, Julius, Sante, FP image, Rubo Medical Imaging, MillenTech, DICOM Works, Medal, IDICON and ACTIV 2000 [22]. Some of the functions they offer are:

- View Analyze format images (big or little endian)
- Create Analyze format headers (big or little endian)
- Create 3D regions of interest
- Overlap multiple regions of interest
- Applying different operations on images
- Export images in different formats
- Linked viewing of multiple images
- Software framework for medical applications
- Create a database of DICOM files located on a local or network drive.
- Choose from files organized by Patient / Series / Images.
- View DICOM tag information.
- Anonymize DICOM files
- Display images with zoom and real-time Window Width/Level capabilities. Copy the image and info into the Clipboard and paste them to other applications
- Full Cine Play Mode for 8 bit multi-frame image files. Save Cine runs as AVI file
- View/Process selected images with an external viewer

A DICOM Viewer was implemented, too, at Craiova, Faculty of Automation, Computers and Electronics, Software Engineering Department. The algorithms presented above for extracting the alphanumeric and imagistic data from DICOM files were developed and implemented using Java technologies in order to realize the DICOM Viewer. Also, we proposed a structure of the database different from the one existing in the DICOM standard. This structure has the advantage of permitting a flexible text-based query of the database. It is used the table which memorizes the entries from the data dictionary, specified by the DICOM standard. This table is also used to extract the information from the DICOM files.

![Fig. 2.6. The Entity-Relationship Model of the Database](image)

Entity-relationship model for the proposed database appears in figure 2.6. The database was implemented using MS SQL Server. The images that are extracted from DICOM files can be processed using algorithms for extracting color and texture characteristics and color regions detection. The results will be used for the content-based visual retrieval process.

Our DICOM Viewer, which is part of the content-based visual retrieval system on multimedia medical databases (MIR), has the following facilities:
1. A tree view of all the DICOM files from the database sorted by their modality (CT, MR, US, etc)
2. An image view panel
3. A tools panel which contains several image processing functions: (invert, blur, brighten and sharper functions, pseudo colors, edge detection)
4. Slider used to see a particular frame from multi frame images
5. Cine Mode for viewing a multi frame file as a motion picture
6. A DICOM Tag Browser organized in categories for displaying all the tags from each category found in the DICOM file together with all their information and value.

![DICOM Viewer](image)

**Fig. 2.7. DICOM Viewer**

The main frame, presented in figure 2.7 contains the following elements:

1. A DataBase Panel that has a tree view of all the DICOM files from the MSSQL database. The files are sorted by their modality (CT, MR, US and so on). When clicking a file path, the frame (or the first frame in case of multi-frames) of that file is shown in the ImageView Panel;
2. An ImageView Panel. When double-clicking the file path from the tree view, the first frame (in case of multi frames) or the frame from the DICOM file is shown in the ImageView Panel. The panel is also used to show the frame (or frames) of the file after image processing functions.
3. A Tools Panel, which contains several image processing, functions. It is composed of:

- An Image Processing panel which contains a Checkbox Group with image effects functions
- Some Rotate and Flip buttons used by physicians to determine the area of interest
- A Text Area and a Slider used for multi frame DICOM files. In the Text Area the number of the current frame is shown. The Slider dynamically changes with the DICOM file displayed. It can also be used to see a particular frame
- A Cine Mode option. By clicking this option a multi frame file can be viewed as a motion picture. It can be used by physicians when studying the human heart and not only
- OneFrame and AllFrames buttons used in case of multi frame DICOM files. By pressing AllFrames button all frames will be displayed. By pressing OneFrame, only the first frame of the file will be displayed
- Show Tag Info button that will open a new frame where all DICOM file tags are displayed by category

4. The DICOM Tag Browser frame contains a list where all the categories are displayed. By double clicking one of them, all tags from that category that were found in the DICOM file are displayed, together with all their information and value. It is possible for a category that no tag can be displayed, because no information was found in the file. The tags and all their information are taken from the database where they were stored after the file had been processed.

5. By selecting [File| Add a DICOM File] a DICOM file can be added to the database. The DICOM file package will be used to decode the DICOM file, save the images as jpegs or gifs and store the tags and the paths of the images in the database. When decoding is done, the file is added in the Database tree. It can be selected and the frames will be displayed in the ImageView panel.

6. If DICOM Dictionary has to be seen, select [Help| See Dicom Dictionary]. A frame will be displayed with all the tags from the dictionary sorted by category. A detailed description of this software tool can be found in [92, 93].

3 Content-Based Visual Query on Color Feature in Multimedia Medical Databases

3.1 Introduction

The color is the visual feature that is immediately perceived on an image. The color system used for representing image color information has a great importance in content-based image query, so this direction of research was intensely studied [18].
There is no color system that is universally used, because the notion of color can be modeled and interpreted in different ways. Each system has its own color models that represent the system parameters [30].

There were created several color systems, for different purposes: RGB (for displaying process), XYZ (for color standardization), rgb, xyz (for color normalization and representation), CieL*u*v*, CieL*a*b* (for perceptual uniformity), HSV (intuitive description) [29, 30].

There have been proposed systems for content-based retrieval that uses different technologies for color characteristics representation: color histograms, color moments, color edge orientation, color texture, color correlograms. These uses color systems like: RGB, HIS, L*a*b* or L*u*v* [29].

The color systems were studied taking into consideration different criteria imposed by content-based visual query [28, 30]:

- The independence of the imaging device
- Perceptual uniformity
- Linear transformation
- Intuitive for user
- Robust for imaging conditions.
  - Invariant to a change in viewing direction
  - Invariant to a change in object geometry
  - Invariant to a change in direction and intensity of the illumination
  - Invariant to a change in the spectral power distribution of the illumination

### 3.2 Color Fundamentals

Color is the visual perceptual property corresponding in humans to the categories called red, yellow, white, etc [15, 29]. Color derives from the spectrum of light (distribution of light energy versus wavelength) and interacts in the eye with the spectral sensitivities of the light receptors. Color categories and physical specifications of color are also associated with objects, materials, light sources, etc., based on their physical properties such as light absorption, reflection, or emission spectrum.

The visible spectrum is the portion of the electromagnetic spectrum that is visible to the human eye [29, 102]. The human eye is able to respond to wavelengths in air from 380 to 780 nm [29, 102].

Newton first used the word spectrum (Latin for “appearance” or “apparition”) in print in 1671 in describing his experiments in optics [29, 102]. Newton observed that, when a narrow beam of sunlight strikes the face of a glass prism at an angle, some is reflected and some of the beam passes into and through the glass, emerging as different colored bands. Newton hypothesized that light was made up of particles of different colors, and that the different colors of light moved at different speeds in transparent matter, with red light moving more quickly in glass than violet light. The result is that red light was refracted less
Multimedia Medical Databases

sharply than violet light as it passed through the prism, creating a spectrum of colors [102].

Newton divided the spectrum into seven colors: red, orange, yellow, green, blue, indigo, and violet (figure 3.1). These seven colors out of a belief derived from the ancient Greek sophists, that there was a connection between the colors, the musical notes, the known objects in the solar system, and the days of the week [102].

A system that describes color is called a color system or color space. Each color system has its own set of color models [29].

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components [29].

The three basic properties of the color radiation are: luminance, hue and saturation. The correspondent perceptual parameters are brightness, color and purity [29].

The luminance represents the visual perception attribute that makes an area to reflect more or less light [36]. The humans have a non-linear perception of brightness.

The hue represents visual perception attribute that makes an area to be similar to one of the following colors: red, yellow, green, blue, or a combination of them [36]. The colors found in nature are polychromes. They are a mixture of several radiations with different wavelength. Each natural color has a dominant wavelength that gives the visual perception of the hue. It can also include components with wavelengths beyond visible spectrum. The white or grey light appears when there are radiations from all wavelengths, in equal quantities.

The saturation is defined as the proportion of pure light with respect to white light needed to produce the color. Any color can be represented by a combination of white or grey light and a pure color, in a specific proportion. The ratio between energy magnitude of the spectral component and total energy represent the purity, or saturation. A pure color has 100% saturation, while the white or grey light has the saturation 0.

<table>
<thead>
<tr>
<th>The colors of the visible light spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
</tr>
<tr>
<td>red</td>
</tr>
<tr>
<td>orange</td>
</tr>
<tr>
<td>yellow</td>
</tr>
<tr>
<td>green</td>
</tr>
<tr>
<td>blue</td>
</tr>
<tr>
<td>violet</td>
</tr>
</tbody>
</table>

Fig. 3.1. The colors of the visible light spectrum
The color represents more than a property of an object, but that is in contradiction with the sense of the word we use in daily language. The association in our language between a color and an object (for example: this object is red) is wrong, because, with no doubt, the color exists only in our mind.

The color is the result of three factors: the nature of the physical world, physiological answer to light of the eye’s retina and the neural processing of the retina’s answer from the brain. As we can see, the number three has an almost magical association with the notion of color.

3.3 Color Perception

We use the eyes to receive most of the information from the outside world. Using the results provided by a researcher, 80% of our memories are obtained visually. The eye has the role to provide information – as color images – about depth, distance, and movement of the objects. Moving the eye up, down and sideways most of the nearby environment is seen.

Analyzing a photo camera we can understand better how the eye works. The portion in front of the eye works as an optical lens, similar with the glass lens of the camera. The optical lens is an object with one or two curbed faces, made from a transparent material. The light that enters in this object is refracted [96].

The dark side in the center of the eye, the eye pupil, regulates the quantity of the received light. When the light is weak, the dimension of the pupil is increased. When the light is bright, the dimension is reduced, allowing entering a small quantity of light. The same thing happens with the camera shutter behind the lens. The layer inside the eye corresponds to the camera film [96].

The eye is much more complex than the photo camera. Using the camera we can only put the images from outside world to a film. Instead, humans and animals can interpret the information received on the retina and act according with the received information. This is possible because the eye is connected with the brain using optical nerve. The optical nerve is connected to the posterior side of the eye, using a small pedicel [96].

The optical information intercepted by the retina is send to the brain using the optical nerve. The information is sent to the brain as electrical impulses. The brain receives the impulses and decodes them [96].

The human’s two eyes see the objects from different angles that make the information sent to the brain to be slightly different. Our brain “learns” how to ensemble these two images, so we don’t see double images. Combining the two images, the brain understands the spatial positioning and the distance of objects – that makes possible three-dimensional seeing.

The brain transforms the image from the upside down position, to normal position. The light is refracted in the crystalline and projected to the retina as an upside down image. The brain “reads” the image and rotates it in the normal position.

The retina is composed by approximately 130 million photo sensible cells – retinal cons and rods [96]. The rods are very sensible to light but they cannot
differentiate colors excepting the blue and green. The cons can differentiate the colors and increase the image clarity, but cannot work in low light. That is the reason why in twilit light we cannot see clear, the colors “disappear” and everything have bluish and greenish hues. In these conditions works only the rods. If the light is very bright, it works only the cons. There are three types of cons, each with its own spectral sensibility. They are red, green and blue depending to their sensibility to these colors. The cons are grouped in the posterior side of the retina, in the so-called yellow spot. Most of the rods are positioned outside the yellow spot, along with few rare cons [96].

In colors perception, the color is not only a wavelengths collection but it is also influenced by the adjacent colors. This process if called chromatic induction.

As we have presented above, the perception of color images is a complex process. To simplify this, several suppositions are being made. The first supposition is that the color is a process that involves a single pixel. The colors perception is not influenced by the surrounding colors.

It is supposed that some factors can be ignored: the distance to the color, the display quality and the ambient light. In general, these parameters are hard to be controlled in the content-based retrieval applications.

3.4 Color Systems

Taking into consideration that the content-based visual query process must use color systems with certain properties, some of them will be presented in detail. The studies made on color images showed that these color systems are the best candidates for usage in this process. It is also presented the RGB color system because the other color spaces result by linear or non-linear transformations of it.

3.4.1 RGB Color System

The first color space that is presented is the RGB color space (figure 3.2) [29, 36]. This color space is built on a cube with Cartesian coordinates. Each dimension of the cube represents a basic color. Each point inside the cube represents a certain hue. The coordinates of a point specify the contribution of each basic color to the final color. The values of the coordinates for each point must be between 0 and 255.

The pure red has the coordinates (255, 0, 0), the pure green has the coordinates (0, 255, 0) and the blue’s coordinates are (0, 0, 255). The yellow is (255, 255, 0) and the orange is between red and yellow and has the coordinates 255, 127, 0). The diagonal between black (0, 0, 0) and white (255, 255, 255) gives different grey hues.

This model has the possibility to represent $256^3$ meaning more than 16 million colors.

This color system has the following characteristics [29]: device dependent, not perceptual uniform (that is why adjacent colors in RGB space don’t imply color
similar), not intuitive, dependent on viewing direction, object geometry, direction, intensity or color of the illumination.

The distance between two points \((r_1, g_1, b_1)\) and \((r_2, g_2, b_2)\) from the RGB space is calculated in a simple manner:

\[
D_{1,2} = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}
\]  

(3.1)

3.4.2 Grey-Value System

In this system, the color feature or color model GREY or INTENSITY is calculated from the values \(R, G, B\) of the images provided by a color camera, with the next transformation [29]:

\[
\text{GREY} = 0.299 R + 0.587 G + 0.144 B
\]  

(3.2)

Characteristics [29]: device dependent, not perceptual uniform, intuitive, linear and dependent on viewing direction, object geometry, direction, intensity or color of the illumination.

3.4.3 rgb Color System

It has three normalized color features: \(r, g, b\) calculated with the next transformations [29]:

\[
R = R/(R+G+B) \\
G = G/(R+G+B) \\
B = B/(R+G+B)
\]  

(3.3)

The color models are normalized because the values of respectively \(R, G\) and \(B\) are divided by their sum. The values \(r, g, b\) depend only on the ratio of the values \(R, G, B\). As a result they have the property that they are not sensitive to surface orientation, illumination direction and intensity. Also, these color features can be represented in the chromaticity diagram.

Its characteristics are [29]: device dependent, not perceptual uniform, not intuitive, nonlinear, dependent on highlights and changes in the illumination color.
3.4.4 XYZ Color System

It was one of the first mathematically defined color spaces. Created by CIE in 1931, the CIE XYZ color space is special, because it is based on direct measurements of the human eye, and serves as the basis from which many other color spaces are defined [14, 29].

This color system was derived from a series of experiments done in the late 1920s by W. David Wright and John Guild. Their experimental results were combined into the specification of the CIE RGB color space, from which the CIE XYZ color space was derived.

This system is based on the additive mixture of the values X, Y, Z, the imaginary primaries introduced by CIE. The values of X, Y, Z are calculated with the next transformation [14, 29]:

\[
X = 0.607R + 0.174G + 0.200B \\
Y = 0.299R + 0.587G + 0.114B \\
Z = 0.000R + 0.066G + 1.116B
\]

(3.4)

These color features cannot be seen by a human eye, or produced, because are too saturated. The luminance is represented only by the Y value. This system is device dependent, because the RGB color system is device dependent.

It has the next characteristics [29]: device independent, not perceptual uniform, not intuitive, linear transformation, dependent on viewing direction, object geometry, direction, intensity or color of the illumination and highlights.

3.4.5 CIE L^*u^*v^* and CIE L^*a^*b^* Color Systems

The CIE 1976 (L^*, u^*, v^*) color space, also known as the CIELUV color space, was adopted by the International Commission on Illumination (CIE) in 1976, as a simple method to compute transformation of the 1931 CIE XYZ color space, but which attempted perceptual uniformity [29, 52].

The color model L^* is based on the scaling of luminance and it is determined only by Y value.

The conversion from XYZ color space to the L^*u^*v^* color space is [29]:

\[
L^* = \begin{cases} 
116 \left( \frac{Y}{Y_n} \right)^{1/3} & \text{if } Y/Y_n > 0.008856 \\
903.3 \left( \frac{Y}{Y_n} \right) & \text{if } Y/Y_n \leq 0.008856 
\end{cases} \\
\]

(3.5)

\[
u^* = 13 \left( \frac{u' - u_n'}{u_n'} \right) \\
v^* = 13 \left( \frac{v' - v_n'}{v_n'} \right)
\]

Where:

\[
u' = \frac{4X}{X + 15Y + 3Z} \\
u_n' = \frac{4X_n}{X_n + 15Y_n + 3Z_n} \\
v' = \frac{9Y}{X + 15Y + 3Z} \\
v_n' = \frac{9Y_n}{X_n + 15Y_n + 3Z_n}
\]

(3.6)

Where \(X_n, Y_n\) and \(Z_n\) are values of the nominally white object-color stimulus.
The distance between two stimulus is calculated using [86]:

$$\Delta E^*_{uv} = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{1/2}$$ (3.7)

*CIE L*a*b* (CIELAB)* is the most complete color model used conventionally to describe all the colors visible to the human eye. It was developed for this specific purpose by C.I.E. The three basic features represent the lightness of the color ($L^*$, $L^* = 0$ yields black and $L^* = 100$ indicates white), its position between magenta and green ($a^*$, negative values indicate green while positive values indicate magenta) and its position between yellow and blue ($b^*$, negative values indicate blue and positive values indicate yellow). The $L^*a^*b^*$ color model has been created to serve as a device independent model to be used as a reference [52].

The values for $a^*$ and $b^*$ are resulted from [52]:

$$a^* = 500 \left[ (X/X_n)^{1/2} - (Y/Y_n)^{1/2} \right]$$
$$b^* = 200 \left[ (Y/Y_n)^{1/2} - (Z/Z_n)^{1/2} \right]$$ (3.8)

The color distance between two color stimulus is calculated using [82]:

$$\Delta E^*_{ab} = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{1/2}$$ (3.9)

These two color systems have the following characteristics [29]: device independent, perceptual uniform, not intuitive, nonlinear transformation and dependent on viewing direction, object geometry, direction, intensity or color of the illumination and highlights.

### 3.4.6 HSV Color System

Alvey Ray Smith created the HSV color space in 1978. The HSV name is from the initials Hue, Saturation and Value. The HSV space describes the colors in more natural terms, close to the artists’ concepts. The HSV color space is cylindrical, as presented in figure 3.3 [84].

The Hue or color is provided by the angle around the axis. The hue with the 0 value is red. The green is for an angle of 120° and the blue is for an angle of 240°.

---

![Fig. 3.3. $T_c$ transformation from RGB to HSV](image_url)
The Saturation describes the intensity of the color. If the saturation is 0 (center of the cylinder), the color has no intensity (there is a grey color). The maximum value of the saturation (to the border of the cylinder) gives a maximum intensity of the color.

The long axis represents the Value, the brightness of the color. The 0 value for this component means the lack of light, resulting black and the maximum value gives the maximum brightness of the color.

The transformation from RGB to HSV is non-linear but it is easy reversible. The HSV color space is natural and approximately uniform from the perceptual point of view. That is why it can be defined a quantization \( Q_c \) of the HSV color space that produce a compact and complete collection of colors.

The transformation from RGB to HSV is done using the following equations:

Let:

\[ r' = \frac{v - r}{v - \min(r, b, g)} \]

\[ g' = \frac{v - g}{v - \min(r, b, g)} \]

\[ b' = \frac{v - b}{v - \min(r, b, g)} \]

\[ \beta h = \begin{cases} 5 + b' & \text{if } r = \max(r, g, b) \text{ and } g = \min(r, b, g) \\ 1 - g' & \text{if } r = \max(r, g, b) \text{ and } g \neq \min(r, b, g) \\ 1 + r' & \text{if } g = \max(r, g, b) \text{ and } b = \min(r, b, g) \\ 3 - b' & \text{if } g = \max(r, g, b) \text{ and } b \neq \min(r, b, g) \\ 3 + g' & \text{if } b = \max(r, g, b) \text{ and } r = \min(r, b, g) \\ 5 - r' & \text{otherwise} \end{cases} \]

The characteristics of this color space are [29]: device dependent, not perceptual uniform, intuitive, nonlinear transformation.

\( H \) is dependent on the color of the illumination, \( S \) is dependent on highlights and changes in the color of the illumination and \( V \) is dependent on viewing direction, object geometry, direction, intensity and color of the illumination.

### 3.4.7 \textit{ll1l2l3 } Color System

Gevers and Smeulders proposed a new color system \textit{ll1l2l3} uniquely determining the direction of the triangular color plane in RGB-space. The transformation is [28, 80]:
Its characteristics are [28, 80]: nonlinear transformation, $H$ is independent on viewing direction, object geometry, direction and intensity of the illumination and highlights.

### 3.5 The Representation of the Color Features

#### 3.5.1 Color Histograms

Indexing images by global color distribution has been achieved by using color histograms. For example, an image that contains different levels of grey can be transformed in a histogram that defines the number of pixels of each color (level of grey). For the color images, the color histograms can have a dimension higher than 2, because the colors are represented as arrays with three components [18, 86].

For a specific image there is only one histogram, but different images might have identical histograms. Swain and Ballard confirmed that histograms are invariant to translations, rotations and they are slightly variant when the viewing angle or the scale is changed, but when the image is transformed in the histogram, spatial information is lost. Color histograms can be considered a good solution for comparing images in content-based visual query.

The next definition can be given [86]:

*The distribution of colors in an image, region or object is represented by a histogram.*

Having an image $I[x,y]$ with three color channels $I = (I_R, I_G, I_B)$, the histogram is given by:

$$h_c[m] = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \begin{cases} 1, & \text{if } Q_c(T_c I[x, y]) = m \\ 0, & \text{otherwise} \end{cases}$$  \hfill (3.13)

Where $X$, $Y$ represent the width and the height of the image.

In the content-based retrieval process, the histograms are normalized. This process is necessary to make the histograms and image matching invariant to image size.

#### 3.5.2 Binary Color Sets

The images or regions color content can be represented more compact with the binary color sets. J.R. Smith and Shih-Fu Chang introduced this solution at the
The binary color sets can be obtained by selection from M colors. The process is represented in the following figure [84, 86]:

Let $B^M$ be an M dimensional binary space, such that each index value $m$ corresponds to one axis in $B^M$.

The following definition can be given [86]:

*The color set $s_c$ is a binary vector in $B^M$. A color set corresponds to a selection of colors from the quantized color space.*

A binary set is equivalent with a limited histogram. For example, having the threshold $\tau_m$ for a color $m$, the color set is obtained using the following equation:

$$S_c[m] = 1, \text{ if } h_c[m] \geq \tau_m, \text{ otherwise } S_c[m] = 0 \quad (3.14)$$

The color set specifies only the colors that are above the specified threshold.

The color sets give a good representation of colors, images and regions. If a color is not representative in a region and it is under the threshold $\tau_m$ it is ignored. In this manner, with the color set, the color content is represented using only the most preeminent colors from image or region.

### 3.5.3 Color Quantization

Because the color spaces are multi-dimensional, a partitioning of the color space is described by a space vector quantization [84, 85]. Generally speaking, a vector quantization $Q_c$ of dimension k and size M is a mapping of a vector from a k-dimensional space, in a finite set $C$ which has M outputs. $Q_c$ can be defined as a function $Q_c: \mathbb{R}^k \rightarrow C$, where $C = (y_0, y_1, \ldots, y_{M-1})$ and $y_m \in \mathbb{R}^k$ for each $m \in 0, 1, \ldots, M-1$ [84, 86].

The set $C$ is called the collection of codes and has the size M. For the case of vector quantization of a color space, $k = 3$ and for each input in the codes collection, the $y_m$ corresponds to a color point.

There is a coverage of $\mathbb{R}^3$ with M partitions that is associated with the vector quantization, where each partition $R_m$ has all the points $w_c$ assigned to the same code $y_m$. 

---

**Fig. 3.4.** Transformation between 3-D color histogram and binary color set
\[ R_m = w_c \in \mathbb{R}^k : Q_c(w_c) = y_m \]  \hspace{1cm} (3.15)

From the partitions definition results that the partitions completely covers \( \mathbb{R}^k \) and they are not intersected:

\[ \bigcup_m R_m = \mathbb{R}^k \quad \text{and} \quad R_m \cap R_n = \emptyset \quad \forall \ m \neq n \]  \hspace{1cm} (3.16)

In conclusion, the partitions represent a complete partitioning of \( \mathbb{R}^k \).

In these circumstances it can be defined the color with the index value \( m \), as a color unit that corresponds to a partition \( R_m \) and which is generated by the color set of points \( \{v_c\}_m \) that are assigned to the same code \( y_m \) for the vector transformation and quantization \[84, 86\].

\[ y_m = Q_c(T_c v_c) \]  \hspace{1cm} (3.17)

Fig. 3.5. The transformation \((T_c)\) and quantization \((Q_c)\) produce the color histogram \( h_c \) and the binary color set \( s_c \).

### 3.6 Computing Color Features Similarity in Content-Based Visual Retrieval

In this section we present a number of methods used for computing the color similarity of the images, using color histograms.

As specified above, the histograms indicate the color distribution in an image or regions. Because the histograms are discrete distributions, they can be represented as characteristic vectors in an M-dimensional space, where M represents the number of distinct colors in histogram. This space is defined as the histogram space, \( H^M \).

The histogram space \( H^M \) is considered to be a metric space and the histograms \( h \) are points in this space, if the following condition is true \[18\]: for each couple of histograms \( h_i, h_j \) it can be found a corresponding number \( v(h_i, h_j) \), called the distance between the points \( h_i \) and \( h_j \), that satisfies the followings:

- \( v(h_i, h_j) = 0 \) (identity)
- \( v(h_i, h_j) \geq 0 \) (non-negativity)
- \( v(h_i, h_j) = v(h_j, h_i) \geq 0 \) (if \( i \neq j \)) (commutability / symmetry)
- \( v(h_i, h_q) \leq v(h_i, h_j) + v(h_j, h_q) \) (triangle non-equality)
3.6.1 Histogram Intersection

Swain and Ballard were the ones that have investigated the use of histogram intersection for color image retrieval. Their objective was to find known objects inside the images, using color histograms. When the size of the object $q$ is smaller than the size of the image $t$ and the histograms are not normalized, then: $|h_q| \leq |h_t|$. The histogram intersection $h_q$ and $h_t$ is given by [84, 85, 86]:

$$d_{q,t} = 1 - \frac{\sum_{m=0}^{M-1} \min(h_q[m], h_t[m])}{\min(|h_q|, |h_t|)}$$

(3.18)

Where:

$$|h| = \sum_{m=0}^{M-1} h[m]$$

(3.19)

The complexity of the method is $O(m \times n)$ where $m$ represents the number of colors resulted from the quantization process and $n$ represents the number of images in the database.

3.6.2 Histogram Euclidian Distance

Having two histograms $h_q$ and $h_t$, then the Euclidian distance is given by [84, 85, 86]:

$$d_{q,t} = \sum_{m=0}^{M-1} (|h_q[m]| - |h_t[m]|)^2$$

(3.20)

The complexity of the method is $O(m \times n)$ where $m$ represents the number of colors resulted from the quantization process and $n$ represents the number of images in the database.

3.6.3 Quadratic Distance between Histograms

The quadratic distance uses the cross-correlation between histogram elements based on the perceptual similarity of the colors. The quadratic distance between the histograms $h_q$ and $h_t$ is given by [84, 85, 86]:

$$d_{q,t} = \sum_{m_0=0}^{M-1} \sum_{m_1=0}^{M-1} (h_q[m_0] - h_t[m_0])a_{m_0m_1}(h_q[m_1] - h_t[m_1])$$

(3.21)

Where $A = [a_{ij}]$, and $a_{ij}$ represent the similarity between elements having the indexes $i$ and $j$. The quadratic metric is a true metric distance when $a_{ij} = a_{ji}$ (symmetrical) and $a_{ii} = 1$. 
For a usual implementation, the calculation of quadratic distance is much more complex than calculation of distances based on Minkowski form, because it is calculated the similarity between all the elements.

The complexity of the method is \(O(m^2 \times n)\) where \(m\) represents the number of colors resulted from the quantization process and \(n\) represents the number of images in the database.

### 3.6.4 Evaluation of the Retrieval Efficiency

The scope of the indexing operations and similarity comparing is to obtain a good efficiency for the retrieval operation. The performance of the information retrieval operation is measured normally with three parameters [86]: speed, recall and precision. These three parameters are determined by the indexing schema and the method used for similarity comparing.

The meaning of the speed parameter is obvious. The performance is higher as the speed is higher.

The recall and precision parameters are used together to measure the efficiency of the retrieval process [36, 84].

The recall parameter measures the ability of the system to find relevant information in the database. Recall signifies the proportion of relevant images in the database that are retrieved in response to a query [41, 84]. To test the system performance, an expert has to determine the number of relevant articles in the database, for each query that is tested. The performance is higher as the value of this parameter is higher.

The precision parameter measures the accuracy of the retrieval operation. Precision is the proportion of the retrieved images that are relevant to the query [41, 84]. If it is taken into consideration only this parameter, the performance of the retrieval operation is higher as the value of this parameter is higher.

Let consider \(A\), the collection of relevant articles and \(B\) the collection of retrieved articles. The \(a, b, c\) and \(d\) are defined next (figure 3.6):

- \(a\) = relevant retrieved articles
- \(b\) = not relevant retrieved articles
- \(c\) = relevant articles that are not retrieved
- \(d\) = not relevant articles that are not retrieved

![Fig. 3.6. A – relevant articles collection; B – retrieved articles collection](image-url)
Then:

\[
\text{Recall} = \frac{a}{a + c} \\
\text{Precision} = \frac{a}{a + b}
\]  

(3.22)

In practice, there are considered both parameters: recall and precision. In this case when the recall is increasing, the precision is decreasing. This is happening because when the system tries to find all relevant articles for a query, it also finds non-relevant articles. The result is a decrease of the precision.

A system with a high value for recall, but with a low value for precision, will return a long list of retrieved articles, but a lot of them are irrelevant. On the other hand, a system with a high value for precision and a small value for the recall parameter indicates that there are many relevant records that haven’t been retrieved.

In conclusion, a good retrieval system must have a balance between those two parameters. A modality to do that is to determine the values of precision and recall (values between 0 and 1) and to build a drawing precision/recall for each system in part (as in figure 3.7). The system that has the drawing to a bigger distance from origin has a better performance.

![Recall/Precision Drawing](image)

**Fig. 3.7.** The Recall/Precision drawing. A system with the drawing to a higher distance from origin has a better performance. The system with the drawing B is better than the system with drawing A.

### 3.7 Content-Based Visual Retrieval on Color Features – Experiments and Results

For the content-based image retrieval systems that use in practice medical multimedia databases have been considered especially grey level features in local or global fashion and less color features.

For example, in the IRMA project (Image Retrieval in Medical Applications) the image data consists of radiographs, while in the later phases will deal with
medical images from arbitrary modalities. As a result, the characteristics that they focused on were texture and shape [44].

The project I-Browse developed by the City University Honk Kong has as application domain the Histology of GI Tract. It works with coarse and fine histological features characterized by color and textures. The color features are represented by color histograms [42].

Another active framework is ASSERT, developed by University Purdue. It is specialized to work with gray-level lung images. In this case, each pathology bearing region (PBR) – region of interest is characterized by a set of shape, texture, and other gray-level attributes. Here it is also calculated a histogram of the local gray levels [16, 82].

MedGIFT is the system developed by the University Hospitals of Geneva. It uses global color histogram based on the HSV (Hue, Saturation, Value) quantized into 18 hues, 3 saturations, 3 values and 4 grey levels. Also uses local color blocks. Each image is recursively partitioned into 4 blocks of equal size, and each block is represented by its mode color. During the experiments, they increased the number of grey levels used for the color block features and color histogram features to 8, 16 and 32 [71, 70].

There are presented next some experiments made at University of Craiova, Faculty of Automation, Computers and Electronics – Software Engineering Department. The color images from digestive area were acquisitioned using the endoscope. The reasons that suggested these experimental studies are:

• Most of the content-based visual retrieval systems from medical domain take into consideration only certain types of images, especially grey-level images. In this condition, we considered important to consider the color images also, as they are produced in a high quantity
• There were considered images with diagnostics from digestive area, that was not so intensely studied
• The conclusions that were presented above corresponds to color spaces experiments using only nature images and not medical images (that have a higher complexity)
• The researchers have not established yet the best color space for content-based visual retrieval on medical multimedia databases.

**Experiment 1**

The study realized on color images extracted from the DICOM files takes in consideration three solutions, like:

• The transformation of the RGB color space to HSV and the quantization at 166 colors -M1
• The use of the RGB color space quantized at 64 colors – M2
• The transformation of the RGB color space to the CIE-LUV and the quantization at 512 colors – M3
We have chosen HSV and CIE-LUV color systems because they have good characteristics for content-based visual query, as it was previously shown. It is taken into consideration also the RGB color space, because of its large use, even if it is not accomplished the above properties. There are considered different levels of quantization to determine the way they affect the retrieval quality.

We have computed the similarity between the query and the target image with three methods: the Euclidian distance (D1), the histogram intersection (D2) and the histogram quadratic distance (D3).

The experiments were performed in the following conditions:

- It was created the test database with medical images extracted from DICOM files.
- Each image from the database was processed before the execution of any query.
- For each experimental query, an image was chosen like query image and there were established by a human factor, the relevant images for query.
- Each of the images relevant for the considered query was utilized, one by one, for querying the database containing images. The final values of the precision and the recall represent an average of the values resulted in the case of each image taken one by one as query image.
- For comparing the obtained results, for each experimental query we draw the graphic of the precision vs. recall for each of the three distances in the case of each quantization method (figure 3.9). Also we present under a tabular form the values that represent the number of relevant images, existing in the first 5, respectively 10 retrieved images, and also number of images that must be retrieved for finding among them the first 5, respectively 10 relevant images (table 3.1).

### Table 3.1. Query 1: Stomach and Duodenum Ulcers. Comparison of three distances in the case of three methods of transformation and quantization.

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>5(9)</td>
<td>5(9)</td>
<td>5(9)</td>
</tr>
<tr>
<td></td>
<td>8(13)</td>
<td>6(11)</td>
<td>7(12)</td>
</tr>
<tr>
<td>M2</td>
<td>5(9)</td>
<td>5(9)</td>
<td>5(9)</td>
</tr>
<tr>
<td></td>
<td>7(18)</td>
<td>6(11)</td>
<td>5(11)</td>
</tr>
<tr>
<td>M3</td>
<td>4(7)</td>
<td>5(9)</td>
<td>4(7)</td>
</tr>
<tr>
<td></td>
<td>8(23)</td>
<td>6(13)</td>
<td>7(17)</td>
</tr>
</tbody>
</table>

We have performed four types of queries on color medical images representing the followings diagnostics: stomach and duodenum ulcer, ulcerate cancer, hernias and esophagus varicose. The values from table 3.1 represent an average of the resulted values in the case of each image taken, one by one, as query image, for the first diagnosis.
In figure 3.8 there are presented some of images retrieved in the case of quantization at 166 colors and using the histograms intersection. The first image is the query image. The “r” symbol indicates that the image was established as relevant, and “nr” like irrelevant.

![Fig. 3.8. Stomach and duodenum ulcers. The retrieved images using the histograms intersection in the case of the quantization at 166 colors for the query image from the first position.](image)

In our experiments, the best results were obtained, constantly, computing the histogram intersection or the histogram quadratic distance in all three cases of quantization. Also, the histogram Euclidian distance gave satisfying results. The color space and the number of quantized colors do not influence very much the queries results. As a result, the HSV color space quantized at 166 colors can be considered the best solution (because the execution speed is higher).

The effectuated studies shown that none of the distances mentioned above produces much better results than the others. In each case, relevant images for the query were retrieved with one of the distances and the others did not retrieve them. It could be observed that in most cases, all the results retrieved by computing the three distances may be useful for not loosing relevant images for the query; consequently, they complement one another. In conclusion, it can be executed a parallel display of the images provided by the all three distances that have been studied.
Fig. 3.9. Query 1: Stomach and Duodenum Ulcers. The graphic of the retrieving efficiency in the case of the transformation of the RGB space to HSV and quantization at 166 colors.

Experiment 2

This experiment makes a comparative study of the content-based query results effectuated on medical images database where the color information is represented by HSV and l1l2l3 color systems.

The originality of the study is given by the fact that the experiments are made on medical images from digestive area produced by an endoscope. The ill area is seen from different directions and in different illumination intensity. This study, unlike the others made on CorelDraw images, uses images produced in real condition, in patient diagnosis.

For the color space l1l2l3 the solution of quantization at 64 colors is chosen, keeping 4 values for each component of the system.

The fact that a color system is quantized at 166 colors and the other at 64 colors does not influence the quality of the content-based image query process, the research studies showing clearly this aspect.

In case of both color systems, to compute the distance between the color histograms of the query image and the target image, we have used the histogram intersection.

The experiments were performed in the following conditions:

- A database with 520 color images from the field of the digestive area was created.
- The images are from patients with the following diagnosis: polyps, ulcer, esophagitis, ulcerous tumors and colitis.
- For each image there are several images with affected area captured from 3 or 4 viewing directions. For each image in the database there is another identical image, but having the illumination intensity changed.

A image processing software tool executes the following steps:

- The transformation of image from RGB color space to HSV color space and the quantization at 166 colors
• The transformation of image from RGB color space to l1l2l3 color space and the quantization at 64 colors
• Computation of the two color histograms with 166, respectively 64 values, that represent the characteristics vectors and storing them in the database
In order to make the query the procedure is:
• A query image is chosen
• The dissimilarity between the query image and every target image from the database is computed, for each two specified criteria (color histograms with 166 colors and the color histogram with 64 colors);
• The images are displayed on 2 columns corresponding to the 2 methods in ascending order of the computed distance

For each query, the relevant images have been established. Each of the relevant images has become in turn a query image, and the final results for a query are an average of these individual results.

The experimental results are summarized in table 3.2. Method 1 represents the query using the HSV color space quantized at 166 colors and Method 2 represents the query on color using the l1l2l3 color space quantized at 64 colors.

<table>
<thead>
<tr>
<th>Query</th>
<th>Method 1</th>
<th>Method 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polyps</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Colitis</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Ulcer</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Ulcerous Tumor</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Esophagitis</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

The values in the table represent the number of relevant images in the first 5 retrieved images, for each query and each method.

It must be mentioned that the queries were made for each of the 5 diagnostics in part. The notion of relevant image was strictly defined. The images from the same patient captured at different illumination intensity and from different points of view were considered relevant for a query, and not the ones with the same diagnosis. The quality of the content-based image query process was strictly analyzed.

In figure 3.10 there is an example of content-based image query considering the two specified methods for images categorized as colitis.

The first column contains 5 images retrieved by Method 1 and the second contains the images retrieved using Method 2. In the first case there are 5 relevant images and in the second case, 4 relevant images.

The precision vs. recall graphic for this example of content-based visual query appears in figure 3.11.
Fig. 3.10. The retrieved images using the two specified methods

Fig. 3.11. The precision vs. recall graphic
Other experimental results can be found in [89, 94]. Several conclusions can be formulated after the experimental results were analyzed:

- To find images representing the same ill area, that were captured by an endoscope from several viewing directions, the solution that uses HSV color system quantized to 166 colors gives the best results.
- For images representing the same ill area, captured to different illumination intensities, the solution that uses l1l2l3l color system quantized to 64 colors, gives the best results in querying process.
- Globally, the solution that uses HSV color space gives most satisfying results, because the database includes both types of images.

In general, for medical images, the first case, with images representing ill area captured from different angles is the most frequent case. So, that is why the use of HSV color space, quantized to 166 colors, is recommended. The situation in the database that was studied was the same, namely, the number of images captured from different angles was higher than the number of images where only the illumination intensity was different.

4 Content-Based Visual Query on Texture Feature in Multimedia Medical Databases

4.1 Overview

There is no precise definition for the notion of texture because the natural textures present contradicting properties (regularity versus randomness, uniformity versus distortion) that are very hard to describe in a unified manner. Generally speaking, the word texture refers to surface characteristics and appearance of an object given by the size, shape, density, arrangement, proportion of its elementary parts, etc [108]. A texture is usually described as smooth or rough, soft or hard, coarse of fine, matt or glossy, etc.

Texture analysis deals with feature extraction and image coding. Feature extraction tries to identify and select a set of distinguishing and sufficient features to characterize a texture. Image coding brings out a compact texture description from selected features. By representing a complex texture with a small number of parameters automated texture processing is possible [108].

Many texture analysis methods have been proposed in the last years. The available methods might be categorized into geometrical, statistical, model-based and signal processing methods [100]. An observation can be made: many methods apparently stride over more than one above category [108]. For instance, a Markov-Gibbs Random Field (MGRF) model derives a joint probability distribution on statistical image features for texture description, so it can be included in both model-based and statistical categories.
In [108] are specified two main approaches for texture description:

- **Descriptive approach** - derives a quantitative description of a texture in terms of a manageable set of feature measures
- **Generic approach** - creates a geometric or a probability model for texture description

Further, the descriptive approach can be divided into **statistical** and **spectral methods**, because of the techniques used in feature extraction [108].

**Statistical methods** use as feature descriptors the image signal statistics from the spatial domain. The most commonly applied statistics are: 1D histograms, moments, grey-level co-occurrence matrices (GLCM), etc [108]. Usually lower-order image statistics, particularly first- and second-order statistics, are used in texture analysis. First-order statistics (the mean, standard deviation and higher-order moments of the histogram) work with properties of individual pixels. Second-order statistics also account for the spatial inter-dependency or co-occurrence of two pixels at specific relative positions. Grey level co-occurrence matrices [39], grey level differences [104], autocorrelation function, and local binary pattern operator [72] are the most commonly applied second-order statistics for texture description. Higher than second-order statistical features have also been investigated in [27, 99], but the computational complexity increases exponentially with the order of statistics. The Haralick features [39] derived from the GLCM, are one of the most popular feature set.

**Spectral methods** work in the frequency domain where features are related to statistics of filter responses [108] and there are several advantages. It was proved that a tuned bandpass filter bank resembles the structure of the neural receptive fields in the human visual system [17, 51]. This is the main motivation of spectral methods to extend feature extraction into the spatial frequency domain.

In constructing filter bank, two-dimensional Gabor filters have been frequently used [45]. Recently, Leung and Malik identify textons (the cluster centres) as feature descriptors from filter responses of a stack of training images [58, 59] and Konishi and Yuille proposed a Bayesian classifier based on the joint probability distribution of filter responses [49].

Also, the generic approaches can be divided into **syntactic** and **probability models** [108].

**Syntactic models** analyze the geometric structure of textures with the help of spatial analytical techniques [108]. The most important methods that belong to this type of models are fractal analysis and structural approach. A variety of fractal models have been proposed for modeling textures in natural scenes [75] and in medical imaging [12].

**Probability models** generalize the feature based descriptive approach by deriving a probability model from the joint distribution of selected image features [108]. Markov-Gibbs random fields are the most successful probability models for texture analysis [31, 40, 107].
Together with color, texture is a powerful characteristic of medical images too, where a disease can be indicated by changes in the color and texture of a tissue. Many methods have been studied to extract texture feature from medical images. The number of publications that presents these methods from the medical domain is impressive. Reading only a small part of them and some articles that presents an overview for them, it can be said that the most used techniques for texture detection in the medical images, are:

- Wavelets
- Gabor Filters
- Co-occurrence matrices
- Based on Fourier transform
- Markov-Gibbs Random Field
- Tamura method

In [95] the authors present a 2-stage method based on wavelet transforms for detecting and segmenting calcifications in mammograms that may be an early sign of disease. Individual grains are difficult to detect and segment due to size and shape variability and because the background mammogram texture is typically inhomogeneous.

The same wavelet transform was used and presented in [25] to carry out the supervised segmentation of echographic images corresponding to injured Achilles tendon of athletes. Texture features are calculated on the expansion wavelet coefficients of the images. The Mahalanobis distance between texture samples of the injured tissue and pattern texture is computed and used as a discriminating function.

Cardiac image properties are analyzed and evaluated with the help of Gabor filters [34]. The paper shows that in the case of cardiac imaging, these techniques can be used for indexing, retrieval by similarity queries, and to some extent, extracting clinically relevant information from the images.

In [63] there is presented a method for texture extraction from medical images that is different from the others presented above. It is based on vector quantization, a technique used in images compression.

An experimental study on a database with different human body tissues is presented in [24]. The study takes into consideration the following texture description: Gradient, entropy, homogeneity, variance, 3rd moment, inverse variance and energy, based on the co-occurrence matrices method. The Gradient method is a new method proposed by authors.

In [4] can be found a study that is a preliminary preparation for the application of some methods to medical images. It is presented a statistical approach of the texture description. Specifically, it introduces the use of first- and second-order statistics on texture color spaces.

An evaluation of texture analysis based on co-occurrence matrices (CMs) with respect to clinical data derived from laryngeal images with and without organic disease in order to examine the feasibility of objective computer-based evaluation
of laryngeal disease can be found in [43]. Haralick introduced co-occurrence matrices for grey-scale textures. They are defined as a histogram, in which the probability of the simultaneous occurrence of two grey-scale values according to a predefined neighborhood is stored. Recently, these CMs have been adapted for color imaging.

In [37, 98] a new method introduced by Tamura et al. was studied for detecting texture features. They proposed a set of texture features to capture global texture properties of an image, namely: coarseness, contrast, and directionality. This information is stored in a three-dimensional histogram, which is quantized to 384 bins.

IRMA is one of the most solid and advanced CBIR systems used in the medical domain. Texture descriptors are obtained from spatial gray-level difference statistics, circular Moran autocorrelation function, entropy and coarseness [56].

In [66] the authors describe that in the system medGIFT that derived from Viper and GNU Image Finding and that was mainly developed for medical domain at the University Hospitals of Geneva, the local texture features are detected by partitioning the image and applying Gabor Filters in various scales and directions. Gabor responses are quantized into 10 strengths. The global texture features are represented as a simple histogram of the responses of the local Gabor Filters in various directions and scales.

In [65] the authors consider that for the images gathered in radiology, textures can be described by wavelet filter responses that measure the changes of the grey levels in various directions and scales throughout the image, or features derived from co-occurrence matrices that count the frequency of neighboring grey levels in various directions and distances. This allows describing the scale of a texture, the principal directions and whether the changes are very quick or rather gradual. Texture descriptors make mainly sense when they are extracted from a region that is homogenous in texture.

A generalized statistical texture analysis technique for characterizing and recognizing typical, diagnostically most important, vascular patterns relating to cervical lesions from colposcopic images is made in [46].

A deep presentation of the concepts of texture analysis in general and specifically in medical imaging can be found in [38]. Magnetic resonance imaging is the particular focus and the range of established and possible clinical applications in that modality is dealt with in detail.

A very interesting comparative study on some new methods used for describing texture feature in medical images is presented in [57]. These methods introduced by Tamura, Castelli and Ngo are considered as most suitable to distinguish medical images. Also, in [11] the authors clarify the principles of texture analysis and give examples of its applications and reviewing studies of the technique.

There are many techniques used for texture extraction, but there isn’t a certain method that can be considered the most appropriate, this depending on the application and the type of images taken into account: breast imaging, mammograms, liver, lung
or cardiac images. There are made efforts for finding a texture detection method for medical images that produce good results, regardless of the type of images [67].

Although most images coming from nature and other fields are color, the majority of research has been done on grayscale textures, for several reasons: high costs for color cameras, high computational costs for color image processing, large complexity even for grayscale textures. However, over the past few years, research has been done in color textures recognition, proving that taking into account the color information improves the color texture classification [74].

In [67], the authors concluded that there are only few comparative studies for the methods used in texture detections and cannot be determined which of them produces the best results from the quality and complexity point of view.

That is why, we proposed ourselves to make a comparative study of two techniques for texture detection that are mostly used: Gabor filters and co-occurrence matrices. An element of originality of this study is that it takes into consideration color texture detection on images from databases with medical images from the digestive acquired using an endoscope, from patients with different diagnosis.

4.2 Gabor Filters

Starting from the representation of the HSV color space, the color in complex domain can be represented. The affix of any point from the HSV cone base can be computed as [74, 106]: $z_M = S \left(\cos H + i \sin H\right)$. Therefore, the saturation is interpreted as the magnitude and the hue as the phase of the complex value $b$; the value channel is not included. The advantages of this representation of complex color are: the simplicity due to the fact that the color is now a scalar and not a vector and the combination between channels is done before filtering. In conclusion, the color can be represented in complex domain [74]:

$$b(x, y) = S(x, y) \cdot e^{iH(x, y)} \quad (4.1)$$

The computation of the Gabor characteristics for the image represented in the HS-complex space is similar to the one for the monochromatic Gabor characteristics, because the combination of color channels is done before filtering [74]:

$$C_{f, \varphi} = \left(\sum_{x, y} (\text{FFT}^{-1}\{P(u, v) \cdot M_{f, \varphi}(u, v)\})^2\right) \quad (4.2)$$

The Gabor characteristics vector is created using the value computed $C_{f, \varphi}$ for 3 scales and 4 orientations [74]:

$$f = (C_{0,0}, C_{0,1}, \ldots, C_{2,3}) \quad (4.3)$$

The similarity between the texture characteristics of the query image $Q$ and the target image $T$ is defined by the metric [74]:
\[ D^2(Q,T) = \sum_J \sum_{\varphi} d_{\varphi}(Q,T), \text{where } d_{\varphi} = (f^0 - f^T)^2 \]  

The algorithm in pseudo-code for detecting color texture with Gabor method is:

**Procedure Gabor_filter**
```
int scale = 3; int orientation = 4; double Ul = 0.1;
double Uh = 0.4; int flag = 0; int side = 60;
```

// the main function
```
*function* GaborFilteredImg(feature[][], Matrix
img, Matrix img_i, int side, double Ul, double Uh, int
scale, int orientation, int flag)
```

begin
```
int border = side; int height = img.getNRows();
int width = img.getNCols();
int xs = height+2.0*border;
int ys = width+2.0*border;
Matrix IMG (xs,ys); Matrix IMG_imag (xs,ys);
Matrix F_real (xs,ys);
Matrix F_imag (xs,ys);
Matrix G_real (xs,ys); Matrix G_imag (xs,ys);
Matrix Gr (2*side+1, 2*side+1);
Matrix Gi (2*side+1, 2*side+1);
Matrix Tmp_1 (xs,ys); Matrix Tmp_2 (xs,ys);
Matrix F_1 (xs,ys); Matrix F_2 (xs,ys);
```

//The Fourier transform of the image matrix
```
*Function Matrix FFT2D(F_real, F_imag, IMG,
IMG_imag);
```

for s = 0; scale; do
```
for n = 0; orientation do
```
```
*Function Gabor(Gr, Gi, s+1, n+1, Ul, Uh, scale,
orientation, flag);
```
```
//The Fourier transform of the Gabor filter
*Function Matrix FFT2D(G_real, G_imag, F_1, F_2);
```
```
for i=0;xs do
```
```
for j=0;ys do
```
```
//The product of the two Fourier transforms
Tmp_1.set(i,j, G_real[i,j] * F_real[i,j]);
Tmp_2.set(i,j, G_imag[i,j] * F_imag[i,j]);
IMG.set(i,j, Tmp_1[i,j] - Tmp_2[i,j]);
Tmp_1.set(i,j, G_real[i,j] * F_imag[i,j]);
Tmp_2.set(i,j, G_imag[i,j] * F_real[i,j]);
IMG_imag.set(i,j, Tmp_1[i,j] + Tmp_2[i,j]);
```
```
End;
```
```
End;
```
// The Inverse Fast Fourier transform
*Function Matrix IFFT2D(Tmp_1, Tmp_2, IMG, IMG_imag);
// The sum of the square items
for k=2*border; width+2*border-1 do
    for m=2*border; height+2*border-1 do
        feature[s][n]+=Tmp_1[k,m]*Tmp_1[k,m] + Tmp_2[k,m]*Tmp_2[k,m];
    end;
end;
end;
end;
*output feature;
end;

Some of the functions called in the main function:

**Function Gabor (Matrix Gr, Matrix Gi, int s, int n, double Ul, double Uh, int scale, int orientation, int flag)
begin
    double base = Uh/Ul;
    double a = pow(base, 1.0/(scale-1));
    double u0 = Uh/ pow(a, (scale-s));
    double Uvar=(a-1.0)*u0/((a+1.0)* sqrt(2.0* log(2.0)));
    double z = -2.0* log(2.0)* (Uvar*Uvar);
    double Vvar=tan(PI/(2*orientation))* u0+z)/sqrt(2.0* log(2.0)-z*z/(Uvar*Uvar));
    double Xvar = 1.0/(2.0* PI*Uvar);
    double Yvar = 1.0/(2.0* PI*Vvar);
    double t1 = cos(PI/orientation*(n-1.0));
    double t2 = sinPI/orientation*(n-1.0));
    int side = (Gr.getNRows() -1)/2;
    for x = 0; 2*side+1 do
        for y = 0; 2*side+1 do
            X=(x-side)*t1+(y-side)*t2;
            Y=-(x-side)*t2+(y-side)*t1;
            G = 1.0/(2.0*PI*Xvar*Yvar)*
            pow(a, (scale-s))*exp(0.5*((X*X)/
                (Xvar*Xvar)+(Y*Y)/(Yvar*Yvar)));
            Gr[x,y]= G*cos(2.0*PI*u0*X));
            Gi[x,y]= G*sin(2.0*PI*u0*X));
            m += Gr[x,y];
            m /= pow( (2.0*side+1), 2.0);
        End;
for \( x_1 = 0; 2 \times \text{side}+1 \) do
for \( y_1 = 0; 2 \times \text{side}+1 \) do
\( \text{Gr}[x_1,y_1] = \text{Gr}[x_1,y_1]-m; \)
End;
End;
End;

Proposition 1: The Gabor_filter procedure has a temporal complexity \( O(n^2) \) where \( n \) is the maximum dimension of the image \( (n= \max\{\text{side}, \text{xs}, \text{ys}\}) \).

Proof: The operations from the beginning of the procedure are elementary and don’t have the complexity more than \( O(1) \). The products of the matrices have complexity \( O(\text{dim}_1 \times \text{dim}_2) \), where ‘dim’ is the matrix dimensions (no more than ‘n’, as it has been stated at the beginning of the procedure). The function calls aren’t recursive, and the functions have a complexity no more than \( O(\text{side}^2) \). So, the Gabor function has complexity \( O(\text{side}^3) \), due to the FOR loops which contain only elementary operations of \( O(1) \) complexity. The functions \text{MatrixFFT2D}, \text{MatrixFFT2D}, \text{MatrixIFFT2D} have complexity no more than \( O(n^2) \), because they contain at most 2 nested FOR loops each having the maximum dimension ‘n’ and operations of \( O(1) \) complexity. The result is that the whole procedure has the time complexity \( O(n^3) \).

4.3 Co-occurrence Matrices

For comparison, the method based on co-occurrence matrices is implemented.

In the case of color images, one matrix was computed for each of the three channels (R, G, B). For an image \( f(x, y) \), the co-occurrence matrix \( h_{df}(i, j) \) is defined so that each entry \((i, j)\) is equal to the number of times for that \( f(x_1,y_1)=i \) and \( f(x_2,y_2)=j \), where \( (x_2,y_2) = (x_1,y_1) + (d \cos \phi, d \sin \phi) \) [18].

This leads to three quadratic matrices with a dimension equal to the number of the color levels presented in an image (256 in our case), for each distance \( d \) and orientation \( f \).

The classification of texture is based on the characteristics extracted from the co-occurrence matrix: energy, entropy, maximum probability, contrast, inverse difference moment and correlation [18].

Energy:

\[
\sum_{a,b:a \neq b} P_{\Phi,d}^2(a,b) \frac{P_{\Phi,d}(a,b)}{|a-b|^\lambda}
\]

Entropy:

\[
\sum_{a,b} P_{\Phi,d}^2(a,b) \log_2 P_{\Phi,d}(a,b)
\]
Maximum probability:
\[
\max_{a,b} P_{\Phi,d}(a,b)
\]  
(4.7)

Contrast:
\[
\sum_{a,b} (a - b)^k P_{\Phi,d}^k(a,b)
\]  
(4.8)

Inverse difference moment:
\[
\sum_{a,b,a\neq b} \frac{P_{\Phi,d}(a,b)}{|a-b|^k}
\]  
(4.9)

Correlation:
\[
\sum_{a,b} \frac{(a,b)P_{\Phi,d}(a,b) - \mu_x \mu_y}{\sigma_x \sigma_y}
\]  
(4.10)

The three vectors of texture characteristics extracted from the three occurrence matrices are created using the 6 characteristics computed for \(d=1\) and \(\phi=0\). The texture similarity between the query image \(Q\) and target image \(T\) is computed by the Euclidian metric.

The algorithm in pseudo-code for generating the co-occurrence matrix is:

```plaintext
**function computecoMatrix (double map[][], int xshift, int yshift, int height, int width)
begin
    int total = 0, gray1, gray2;
    Matrix coMatrix(256,256);
    for i = 0; height; do
        for j = 0; width do
            if (not((j + xshift >= width) || (j + xshift < 0) || (i + yshift >= height) || (i + yshift < 0))) then
                gray1 = map[i][j];
                gray2 = map[i+yshift][j+xshift];
                coMatrix.set(gray1, gray2, coMatrix[gray1][gray2] + 1);
                total ++;
            end;
        end;
    end;
end;
```

The algorithm that generates the 6 characteristics (entropy, maximum probability, contrast, inverse difference moment and correlation) is:
**function analysecoMatrix ()**

begin

double sum=0; double miu_x=0, miu_y=0,
    tau_x=0, tau_y=0, sum_a1=0, sum_b1 =0;
double ss1=0;
double maxProb,inverseDiff, entropy, energy,
    contrast, correlation;
String vectorsString;
MaxProb =0; InverseDiff =0; Energy=0; Contrast=0;
for  i = 0; i < w  do
    for  j = 0;  h  do
        if (coMatrix.elementAt(i, j) > MaxProb) then
            maxProb = coMatrix.elementAt(i, j);
        end;
        inverseDiff += coMatrix.elementAt(i,j)/
            (1+Math.abs(i - j));
        Energy += coMatrix.elementAt(i, j) *
            coMatrix.elementAt(i, j);
        contrast += (i - j) * (i - j) *
            coMatrix.elementAt(i, j);
        if (coMatrix.elementAt(i, j) != 0) then
            sum += coMatrix.elementAt(i, j)
                *log(coMatrix.elementAt(i, j));
        end;
        entropy=-sum;
        sum_b1 += coMatrix[i, j];
        miu_x += i * sum_b1;
        sum_a1+= coMatrix[i, j];
        miu_y += j * sum_a1;
        tau_x += (i- miu_x)*(i - miu_x) * coMatrix[i,j];
        tau_y += (j - miu_y)*(j - miu_y)*coMatrix[i, j];
    end;
end;
tau_x = Math.sqrt(tau_x);
tau_y = Math.sqrt(tau_y);
for  i = 0; i < w  do
    for  j = 0;  h  do
        sum += (double) Math.abs((i * j *
            coMatrix.elementAt(i,j)-miu_x*miu_y))/
            (tau_x* tau_y);
    end;
end;
correlation = sum;
vectorsString = maxProb + ";" + inverseDiff + ";" + entropy + ";");" + energy + ";");" + contrast + ";");" + correlation + ";");
* output vectorsString;
end;

Proposition 2: The functions computeCoMatrix and analyseCoMatrix have the temporal complexity O(m^2), where 'm' is the maximum dimension of the image (m = max{ xshift, yshift, height, width}).

Proof: The function computeCoMatrix contains only 2 FOR nested loops, each having the dimension ‘m’, and operations of O(1) complexity. The function analyseCoMatrix contains a sequence of 2 FOR nested loops, each having the dimension ‘m’, and operations of O(1) complexity. The result is that the functions have a temporal complexity of O(m^2).

4.4 Experiments and Results

The experiments were taken into consideration the content-based visual retrieval on color feature represented by 166 values in HSV color system (the previous experiments showed clearly that it performs better than other methods) and color texture feature represented by the two methods. The experiments were performed in the following conditions:

It was created a database with 960 color images from the digestive area. A software tool that permits the processing of each image was developed. The software tool executes the following functions:

1. The images transformation from the RGB color space to the HSV color space and the quantization to 166 colors;
2. The co-occurrence matrices are computed for each component R, G, B and three vectors containing 6 sizes (energy, entropy, maximum probability, contrast, inverse difference moment, correlation) are generated; the matrices are computed for d=1 and \( \phi = 0 \); in this case the characteristics vector has 18 values;
3. The Gabor characteristics vector containing the values computed for 3 scales and 4 orientations is generated; in this case the characteristics vector has 12 values;
4. The characteristics vectors generated at points 1, 2 and 3 are stored in the database

In order to execute the query, the next procedure is followed:

- A query image is chosen;
- It is computed the dissimilarity between the query image and every target image from the database, for each of the three specified criteria (color histograms with 166 colors, the vector generated on the basis of the co-occurrence matrices and the vector for Gabor method);
- The images are displayed on 3 columns corresponding to the 3 methods in ascending order of the computed distance
For each query the relevant images have been established. Each of the relevant images has become in its turn a query image, and the final results for a query are an average of these individual results.

The experimental results are summarized in table 1. Met 1 represents the query on color feature, Met 2 represents the query on color texture feature using co-occurrence matrices and Met 3 represents the query on color texture feature using Gabor method.

The values in the table 1 represent the number of relevant images in the first 5 retrieved images for each query and each of the three methods.

In figure 4.1 there is an example of content-based image query considering the three specified methods. The first column contains the 5 images retrieved on color feature.
feature; the second contains the retrieved images on color texture using co-occurrence matrices, and the third the retrieved images using the Gabor method. In the first case there were 4 relevant images, in the second case 3, and in the third 2 relevant images.

As the values in the table 1 and other experiments have shown, the best results for medical color images from the field of digestive apparatus have constantly been obtained on color feature. The color textures obtained by the co-occurrence matrices and Gabor method have had poorer results. Between the two methods for color texture detection, the method using co-occurrence matrices, applied on RGB color space has led to better results.

An important observation has to be done, which leads to the improvement of the quality of the content-based query on this type of images. For each query, at least in half of the cases, the color texture method based on co-occurrence matrices has given at least one relevant image for the query, image that could not be found using the color feature. Consequently, it is proposed that the retrieval system should use two methods: one based on color feature and the other based on color texture detected with co-occurrence matrices. It is also proposed that the display of the results should be done in parallel, so that the number of relevant images can be increased from 3 to 4 in the first 5 retrieved images. For the example in figure 1, in the case of a union of the images retrieved using the first and the second method, the next relevant distinct images will result: 307, 303, 304, 328 and 342. Both feature detection methods have the same complexity $O(\text{width} \times \text{height})$, where width and height are the image dimensions. The two computed distances, the histogram intersection and the Euclidian distance are equally complex $O(m \times n)$ where $m$ is the number of values in the characteristics vector, and $n$ is the number of images in the database.

Also, a parallel computation of the two distances can be proposed in order to make the execution time for a query shorter.

5 Automatic Segmentation and Content-Based Region Query in Multimedia Medical Databases

5.1 Overview

Segmentation of medical images is the task of partitioning the data into contiguous regions representing individual anatomical objects. This task plays a vital role in

<table>
<thead>
<tr>
<th>Query</th>
<th>Met 1</th>
<th>Met 2</th>
<th>Met 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polyps</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Colitis</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Ulcer</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Ulcerous Tumor</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Esophagitis</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
many biomedical imaging applications: the quantification of tissue volumes, diagnosis, localization of pathology, study of anatomical structure, treatment planning, partial volume correction of functional imaging data, and computer-integrated surgery [76, 81].

Segmentation is a difficult task because in most cases it is very hard to separate the object from the image background. Also, the image acquisition process brings noise in the medical data. Moreover, inhomogeneities in the data might lead to undesired boundaries. The medical experts can overcome these problems and identify objects in the data due to their knowledge about typical shape and image data characteristics. But, manual segmentation is a very time-consuming process for the already increasing amount of medical images. As a result, reliable automatic methods for image segmentation are necessary [76, 81].

As in content-based visual retrieval on color or texture features, it cannot be said that there is a segmentation method for medical images that produces good results for all types of images. There have been studied several segmentation methods that are influenced by factors like: application domain, imaging modality or others [76, 67].

Image segmentation is defined as the partitioning of an image into non overlapping, constituent regions that are homogeneous, taking into consideration some characteristic such as intensity or texture [76].

If the domain of the image is given by $I$, then the segmentation problem is to determine the sets $S_k \subset I$ whose union is the entire image. Thus, the sets that make up segmentation must satisfy

$$I = \bigcup_{k=1}^{K} S_k$$

where $S_k \cap S_j = \emptyset$ for $k \neq j$ and each $S_k$ is connected [76, 86].

In an ideal mode, a segmentation method finds those sets that correspond to distinct anatomical structures or regions of interest in the image. When the constraint from the above definition is removed, then the operation of determining the sets is called pixel classification and the sets are called classes [76]. Pixel classification can be very important in medical application, especially when disconnected regions belonging to the same tissue class need to be identified. The determination of the total number of classes in pixel classification can be also a difficult problem.

Another process bounded by segmentation is called labeling, that means assigning a meaningful designation to each region or class [76]. It can be performed separately from segmentation. Labeling process maps the numerical index $k$ of set $S_k$, to an anatomical designation. In medical imaging, the labels are often visually obvious and can be determined upon inspection by a medical expert. Computer automated labeling is desirable when labels are not obvious, or in automated processing systems. This situation occurs in digital mammography where the image is segmented into distinct regions and the regions are subsequently labeled as being healthy tissue or with tumor.
The segmentation methods can operate in a 2-D image domain or a 3-D image domain [32, 33, 53, 54, 76]. This property is called dimensionality. Methods based only on image intensities are independent of the image domain. Certain methods such as deformable models, Markov random fields, and region growing, incorporate spatial information may therefore operate differently depending on the dimensionality of the image.

The segmentation methods were grouped in the following categories: thresholding, region growing, classifiers, clustering, Markov random field models, artificial neural networks and deformable models. Of course, there are other important methods that do not belong to any of these categories [76, 81].

In thresholding approaches an intensity value called the threshold must be established. This value will separate the image intensities in two classes: all pixels with intensity greater than the threshold are grouped into one class and all the other pixels into another class. As a result, a binary partitioning of the image intensities is created. If more than one threshold is determined, the process is called multi-thresholding [76].

Thresholding is often used as the first step in a sequence of image processing operations. It has some limitations:

- In its simplest form only two classes are generated
- It cannot be applied to multi-channel images
- Typically, does not take into account the spatial characteristics of the image.

Region growing is a technique for extracting a region from an image that contains pixels connected by some predefined criteria, based on intensity information and/or edges in the image. In its simplest form, region growing requires a seed point that is manually selected by an operator, and extracts all pixels connected to the initial seed having the same intensity value [47, 76].

Like thresholding, region growing is not often used alone. It can be used particularly for emphasizing small and simple structures such as tumors and lesions.

Limitations:

- It requires manual interaction to obtain the seed point
- Can also be sensitive to noise, causing extracted regions to have holes or even become disconnected

Split and merge algorithms are related to region growing but do not require a seed point.

Classifier methods represent pattern recognition techniques that try to partition a feature space extracted from the image using data with known labels. A feature space is the range space of any function of the image, with the most common feature space being the image intensities themselves. Classifiers are known as supervised methods because they need training data that are manually segmented by medical experts and then used as references for automatically segmenting new data [76].
A simple classifier is the nearest-neighbor classifier, where each pixel or voxel is classified in the same class as the training data with the closest intensity. The k-nearest-neighbor (kNN) classifier is a generalization of this approach, where the pixel is classified according to the majority vote of the k-closest training data [76]. Classifier methods have some advantages: are relatively computationally efficient and can be applied to multi-channel images.

The disadvantages are:

- They generally do not perform any spatial modeling
- The requirement of manual interaction for obtaining training data; training sets can be acquired for each image that requires segmenting, but this can be time consuming and laborious.
- Using of the same training set for a large number of scans can lead to results which do not take into account anatomical and physiological variability between different subjects.

Clustering algorithms work as classifier methods but they don’t use training data. As a result they are called unsupervised methods. Because there isn’t any training data, clustering methods iterate between segmenting the image and characterizing the properties of the each class. It can be said that clustering methods train themselves using the available data [76]. The next three commonly used clustering algorithms must be mentioned:

- K-means or ISODATA algorithm (clusters data by iteratively computing a mean intensity for each class and segmenting the image by classifying each pixel in the class with the closest mean)
- The fuzzy c-means algorithm (it is a generalization of the K-means algorithm; it is based on fuzzy set theory)
- The expectation-maximization (EM) algorithm (works with the same clustering principles and is based on the assumption that the data follows a Gaussian mixture model) [3, 10]

These algorithms have the following disadvantages: they require an initial segmentation and do not directly incorporate spatial modeling and can therefore be sensitive to noise and intensity in homogeneities.

Markov random field (MRF) is a statistical model that can be used within segmentation methods. For example, MRFs are often incorporated into clustering segmentation algorithms such as the K-means algorithm under a Bayesian prior model. MRFs model spatial interactions between neighboring or nearby pixels. In medical imaging, they are typically used to take into account the fact that most pixels belong to the same class as their neighboring pixels. In physical terms, this implies that any anatomical structure that consists of only one pixel has a very low probability of occurring under a MRF assumption [76].

The disadvantages are:

- Proper selection of the parameters controlling the strength of spatial interactions is necessary
- Usually require computationally intensive algorithms
Artificial neural networks (ANNs) are massively parallel networks of processing elements or nodes that simulate biological learning. Each node in an ANN is capable of performing elementary computations. Learning is possible through the adaptation of weights assigned to the connections between nodes [76]. ANNs are used in many ways for image segmentation. The most widely applied use in medical imaging is:

- As a classifier, where the weights are determined using training data, and the ANN is then used to segment new data
- In an unsupervised fashion as a clustering method
- For deformable models.

Deformable models are physically motivated, model-based techniques for outlining region boundaries using closed parametric curves or surfaces that deform under the influence of internal and external forces. To outline an object boundary in an image, a closed curve or surface must be placed first near the desired boundary that comes into an iterative relaxation process [2, 32, 33, 76].

They offer some advantages:

- The ability to directly generate closed parametric curves or surfaces from images
- The incorporation of a smoothness constraint that provides robustness to noise and spurious edges

A disadvantage is that they require manual interaction to place an initial model and choose right parameters.

Other methods for medical image segmentation must be mentioned [76]:

- Atlas-guided approaches use an atlas as a reference frame for segmenting new images. Conceptually, atlas-guided approaches are similar to classifiers except they are implemented in the spatial domain of the image rather than in a feature space.
- Model-fitting is a segmentation method that typically fits a simple geometric shape such as an ellipse or parabola to the locations of extracted image features in an image.
- The watershed algorithm uses concepts from mathematical morphology to partition images into homogeneous regions.

Detailed explanations on the methods presented above, examples and a large bibliography on medical image segmentation problem can be found in [76].

In image retrieval, several systems attempt to perform an automatic segmentation of the images in the collection for feature extraction [8, 10, 62, 97]. To have an effective segmentation of images using varied image databases the segmentation process has to be done based on the color and texture properties of the image regions [35, 67].

A very well known system is Blobworld. It is a system for image retrieval based on finding coherent image regions which roughly correspond to objects.
Each image is automatically segmented into regions named "blobs" with associated color and texture descriptors. Querying is based on the attributes of one or two regions of interest, rather than a description of the entire image. The blob descriptions were indexing using a tree in order to make a faster retrieval. Experiments showed encouraging results for both querying and indexing [3, 8, 10].

In [64] the Schema Reference System is presented. This is a content-based image retrieval system that employs multiple segmentation algorithms and indexing and retrieval subsystems. These algorithms are the following:

- Pseudo Flat Zone Loop algorithm (PFZL), contributed by Munich University of Technology - Institute for Integrated Systems.
- Modified Recursive Shortest Spanning Tree algorithm (MRSST), contributed by Dublin City University
- K-Means-with-Connectivity-Constraint algorithm (KMCC), contributed by the Informatics and Telematics Institute/Centre for Research and Technology - Hellas.
- Expectation Maximization algorithm (EM) in a 6D colour/texture space, contributed by Queen Mary University of London.

The automatic segmentation techniques were applied on various imaging modalities: brain imaging, chest radiography, computed tomography, digital mammography or ultrasound imaging.

A lot of studies were made on MR brain images in order to extract the brain volume, to outline structures such as the cerebral cortex or the hippocampus or to segment the brain tissue into gray matter, white matter and cerebrospinal fluid with the help of classifier approaches, clustering approaches, neuronal network and Markov random fields. For the segmentation of the cerebral cortex or other structures (the ventricles, the corpus callosum, the hippocampus) the deformable models were especially used. The atlas-guided methods are capable of fully segmentation of the brain structures [76].

Automatic segmentation techniques in computed tomography were applied to bone scans (thresholding, region growing, Markov random fields or deformable models), to thoracic scans (deformable models, region growing combined with watershed algorithms, region growing combined with fuzzy logic) or liver images (deformable models) [53, 76, 68].

The automatic segmentation applied on digital mammography tries to distinguish the tumors, the microcalcification clusters or other pathologies. In reference materials two approaches can be found:

- Image initial segmentation and labeling the candidate regions as normal or suspicious
- Image processing in order to detect the presence of pathology and then image segmentation to determine its precise location

The most used techniques are: thresholding, region growing and Markov random fields, because pathological regions have often different texture characteristics [76].
In ultrasound imaging the deformable models were successfully applied for segmentation of the echocardiograms, to detect the boundary of the fetus and the fetus head, or to outline the cysts in breast images. Though, the segmentation algorithms are limited in ultrasound images because of the high level of speckling present in this type of medical images [76].

5.2 The Color Set Back-Projection Algorithm

For detecting color regions, at Software Engineering Department Craiova, it was chosen the color set back-projection algorithm, introduced initially by Swain and Ballard and then developed in the research projects at Columbia University, in the content-based visual retrieval domain [83, 86, 87]. This technique provides the automatic extraction of regions and the representation of their color content. The extraction system for color regions has four steps [83, 86, 87]:

1. The image transformation, quantization and filtering (the transformation from the RGB color space to HSV color space and the quantization of the HSV color space at 166 colors)
2. Back-projection of binary color sets
3. The labeling of regions
4. The extraction of the region features

The algorithm reduces the insignificant color information and makes evident the significant color regions, followed by the generation, in automatic way, of the regions of a single color, of the two colors, of three colors.

To conclude with, the second step of the color set back-projection algorithm is the following [83, 86, 87]:

1. Detection of single color regions
   - Having the image histogram, \( H[m] \), all the values \( m' = m \) for which \( H[m] \geq p_0 \) are detected.
   - For each \( m' \) the color set \( c \) having the property \( c[k] = 1 \) for \( k = m \) and \( c[k] = 0 \) in other cases is found. On the image \( R[m,n] \) the back-projection algorithm for each color set \( c \) is applied and the color regions are found. For each region \( n \) the local histogram \( L_n[m] \) is stored.
   - The residue histogram \( H_r[m] = H[m] - \sum_n L_n[m] \) is computed

2. Detection of two colors regions
   - The values \( l' = 1 \) and \( m' = m, l \neq m \), \( H[1] \geq p_0 \), \( H[m] \geq p_0 \) and \( H_r[1] \geq p_1 \), \( H_r[m] \geq p_1 \) are found.
   - For each set \( l' \), \( m' \) the color set \( c \) having the property \( c[k] = 1 \) for \( k = l' \) or \( k = m' \) and \( c[k] = 0 \) in other cases is found. On the image \( R[m,n] \) the
back-projection algorithm for each set \( c \) is applied and the color regions are found. For each region the local histogram \( L_n[m] \) is recorded.

- The residue histogram \( H_r[m] = H[m] - \sum_n L_n[m] \) is updated.

### 3. Detection of the three colors regions...

Next, two implementations of the color set back-projection algorithm will be presented. These original implementations were designed at Software Engineering Department, Faculty of Automation, Computers and Electronics, Craiova.

In the first implementation of the color set back-projection algorithm (Method 1), the image is transformed in HSV color system and quantized to 166 colors. At the end of this process, both the global histogram of the image and the color set are available. To the matrix that stores only the quantized colors from 0 to 165 it is applied a 5x5 median filter, which has the role of to eliminate the isolated points. Having the HSV quantized matrix it is possible to begin the process of regions extraction presented above. It may be observed that this process it is in fact a depth-first traversal, described in pseudo-cod in the following way:

```plaintext
procedure FindRegions (Image I, colorset C)
    InitStack(S)
    Visited = ∅
    for *each node P in the I do
        if *color of P is in C then
            PUSH(P)
            Visited ← Visited ∪ {P}
    while not Empty(S) do
        CrtPoint <- POP()
        Visited ← Visited ∪ {CrtPoint}
        For *each unvisited neighbor S of CrtPoint do
            if *color of S is in C then
                Visited ← Visited ∪ {S}
                PUSH(S)
        //end if
    //end while
    *Output detected region
//end if
//end for
```

**Proposition 1**

The total running time of a call of the procedure `FindRegions` (Image I, colorset C) is \( O(m^2*n^2) \), where “\( m \)” is the width and “\( n \)” is the height of the image.

**Proof**

Recall that the number of pixels of the image is \( m*n \), where “\( m \)” is the width and “\( n \)” is the height of the image. As it is observed next the first loop FOR of the algorithm is executed at most once for each pixel P in the image. Hence, the total
time spent in this loop is $O(n^2m)$. The WHILE loop processes the stack $S$ for each pixel that has the same color of its neighbor. The inner loop FOR processes the pixels of unvisited neighbor. So, the total time spent in these loops is $O(m^2n)$, because all pixels of the image are processed at most once. From previous statements it is inferred that the total running time of this procedure is $O(m^2n)$.

In the second original implementation of the algorithm (Method2), the image pixels are arranged into hexagons. The edge of a hexagon has a certain number of pixels (3, 4, 5). There are taken into consideration only the pixels that correspond to the vertices of the hexagons with an established edge. The image is viewed as a graph not as a pixel matrix. The vertices represent the pixels and the edges represent neighborhoods between pixels.

For each binary set is executed:

1. The graph is inspected until it is found the first vertex having the color from the color set
2. Starting from this vertex, there are found all the adjacent vertices having the same color
3. The process will continue in the same manner for each neighbor, until there are not found vertices having the same color
4. It is verified if the detected region satisfies the imposed thresholds; in affirmative case, the region is labeled and introduced in the database

This process of regions extraction from a graph is in fact a breadth-first traversal, described in pseudo-cod in the following way:

```plaintext
procedure construct_graph (Image I, Graph g, Edge edge):
    for * i->0;width/edge
        for * j->0;height/edge
            if (i mod 3==0)
                *if(j mod 2==0)
                    g[i][j]=I[edge*i][edge*j+edge-1]
                *if(j mod 2==1)
                    g[i][j]=I[edge*i][edge*j+edge+2]
            if (i mod 3==1)
                *if(j mod 2==0)
                    g[i][j]=I[edge*i-1][edge*j+edge]
                *if(j mod 2==1)
                    g[i][j]=I[edge*i-1][edge*j+edge*2]
            if (i mod 3==2)
                *if(j mod 2==0)
                    g[i][j]=I[edge*i-2][edge*j+edge-1]
                *if(j mod 2==1)
                    g[i][j]=I[edge*i-2][edge*j+edge+2]
        //end for *j->0
    *output the graph g
//end for * i->0
```
procedure FindRegions (Graph G, colorset C) :
    InitQueue(Q)
    Visited = ∅
    for *each node P in the G do
        if *color of P is in C then
            PUSH(P)
            Visited ← Visited ∪ {P}
        end if
    end for
    while not Empty(q) do
        CrtPoint <- POP()
        Visited ← Visited ∪ {CrtPoint}
        for *each unvisited neighbor Q of CrtPoint do
            if *color of Q is in C then
                Visited ← Visited ∪ {Q}
                PUSH(Q)
            end if
        end for
    end while
    *output-detected region
end procedure

Proposition 2
The total running time of a call of the procedure \textit{FindRegions} (Graph G, colorset C) is $O(n^2)$, where “n” is the number of nodes of graph attached to the image.

Proof
Observe that the first FOR loop of the algorithm is executed at most once for each node of the graph. Hence, the total time spent in this loop is $O(n)$. The WHILE loop processes the queue Q for each node which has the same color of its neighbor. The inner loop FOR processes the nodes of unvisited neighbor. So, the total time spent in these loops is $O(n)$, because are processed all nodes of the graph at most once. From previous statements is inferred that the total running time of this procedure is $O(n^2)$.

In figure 5.1 there is an image representing gastric cancer diagnosis. The sick part of the image that indicates the diagnosis is the prominent oval zone that presents bleedings. The color regions detected by the color set back-projection algorithm are presented in figure 5.2. From the medical point of view the regions 5 and 6 are important.

![Fig. 5.1. An image representing gastric cancer diagnosis](image-url)
For each region, the color, the minimum bounding rectangle and the number of pixels that roughly indicates the dimension of the sick region, were stored in the database. The region localization is given by the minimal bounding rectangle (MBR). The region area is represented by the number of color pixels, and can be smaller than the minimum-bounding rectangle. Other experiments can be found in [90].

5.3 Content-Based Region Query – Experiments and Results

The regions detected by the image automatic segmentation techniques can be utilized in content-based region query of the multimedia medical databases [86].

In a content-based region query, the images are compared based on their regions. In the first step of the query, content-based visual queries are effectuated on the regions, and not on the images. Then, in the final step of the query, there are determined the images corresponding to the regions and there is computed the total distance between the images by the weighting of the distances between regions. So, the user selects one or several query regions in order to find in the database images containing similar regions from the color, size or spatial extent point of view.

The content–based visual query may be improved by adding spatial information to the query. So, the total measure of the dissimilarity takes into consideration both the values of the features (color, texture), and the spatial values of the regions. There are two types of spatial indexing, namely: relative and absolute [83, 86, 87, 91].

The most powerful images retrieval system is the one that allows queries in which are specified both the visual features and spatial properties for the desired images. Such query offers to the user the possibility to control the selection of
regions and attributes which are the most important in the determination of the similarity.

At Craiova we studied the content-based region query process on a database with medical images from digestive area captured by an endoscope. The color regions were obtained using the color set back-projection algorithm. The reasons for making this study are:

- There are not so many studies made on medical color images from the digestive area, although the number of these images, acquired in the diagnosis process, is high.
- Extraction of color regions from database containing nature images provided good results. That is why we tried this algorithm on color medical images.
- In content-based region query on medical images collections, the specialist chooses one or several detected regions for querying the database. The purpose is the retrieval of images that are similar by color, texture or both; this can be useful for clarifying some uncertain diagnosis or seeing the evolution and the treatment for images with the same diagnosis; another utilization can be in medical teaching – can be useful for students to see the important color regions from medical images, or images that contain the similar color regions.

Taking into account that the color information of each region is stored as a color binary set, the color similarity between two regions may be computed using the quadratic distance between binary sets \( s_q \) and \( s_t \) that is given by the following equation [86]:

\[
d_{q,t}^f = \sum_{m_0=0}^{M-1} \sum_{m_1=0}^{M-1} (s_q[m_0] - s_t[m_0]) \cdot (s_q[m_1] - s_t[m_1])
\]

Other two important distances are taken into consideration:

1. The distance in area between two regions \( q \) and \( t \) [86]:

\[
d_{q,t}^a = \left| \text{area}_q - \text{area}_t \right|
\]

2. The distance in MBR width (w) and height (h) between two regions \( q \) and \( t \) [86]:

\[
d_{q,t}^s = \sqrt{(w_q - w_t)^2 + (h_q - h_t)^2}
\]

The single region distance is given by the weighted sum of the color feature \( d_{q,t}^f \), area \( d_{q,t}^a \) and spatial extent \( d_{q,t}^s \) distances [86].

\[
D_{tot} = \alpha_a \cdot d_{q,t}^a + \alpha_s \cdot d_{q,t}^s + \alpha_f \cdot d_{q,t}^f
\]
The user may also assign a relative weight \( \alpha \) to each attribute. For example, the user may weigh the size parameter more heavily than feature value and location in the query. For multiple regions query, the overall image query strategy consists of joining the queries on the individual regions in the query image. The join identifies the candidate target images.

For testing the efficiency of the new method and for comparing the two methods of implementation of the color set back-projection algorithm, there have been made some experiments over the medical images collection. For Method2 the hexagon edge can be equal to 3, respective 4. For each query, the images from the databases were inspected and relevance was assigned to them (1- relevant, 0 – irrelevant) and the retrieval effectiveness using recall and precision was recorded. Below, there are presented the results of one of the experiments.

Experiment:
In figure 5.3 there is the image for which there were detected the color regions using Method2 with edge=3. Region6 was chosen as query region, which appears as marked and emphasizes the sick area. The obtained results are presented in figure 5.4. It can be observed that the first five retrieved images are all relevant for this query. The graphic of the retrieving efficiency in the case of the two presented algorithms (Method1 and Method2 with edge =3) is shown in figure 5.5. The graphics corresponding to the two methods are superposed, which means that the results of this one region query are identical.

![Fig. 5.3.](image)

The very good results obtained in the effectuated experiments indicate the fact that each of the two implementations methods (Method1 and Method2) of the color set back-projection algorithm can be used in the processing of the content-based visual query. The experiments show that the results obtained with the Method2 and edge=3 are closer in quality with those obtained with Method1. The advantage of the second method (Method 2 with edge equal to 3) is given by the fact that for detecting the color regions it is not necessary the pixel-by-pixel image traversal, but only the pixels arranged in the vertices of a hexagon with
Fig. 5.4. The retrieved images using Method2 and edge equal to 3, for the Region6 as query region.

Fig. 5.5. Experiment 1. The graphic of the retrieving efficiency for Method1 and Method2 with the hexagon edge equal to 3.

edge equal to 3 pixels. If the processing time of the Method 1 is $O(m^2n^2)$ (m is the width and n is the height of image), the processing time for the Method 2 presented here is $O(n^2)$ (n is the number of nodes of graph attached to an image).

Other experiments and results can be found in [5, 6, 7].
6 Conclusions

The chapter presents a series of aspects in the field of multimedia medical databases. These databases are the result of structuring the alphanumeric and imagistic data gathered in large quantities in the patient investigation and diagnosis processes.

We have also presented the necessity of creating and managing the multimedia medical databases and the advantages of using these operations in order to increase medical act efficiency.

A big part of medical data is stored in DICOM files that can’t be seen directly on a computer. As a result, we have given attention to this problem by presenting our original algorithms for extracting the alphanumeric and imagistic data from this type of files in order to be integrated in the multimedia medical databases.

Also, the problem of DICOM viewers has been presented. These tools have the ability to display and apply certain operations on the information from DICOM files, being very useful in the medical activity.

Taking into consideration that a database is created for querying and obtaining with accuracy and speed the information requested by the user, the chapter treats in great detail the problem of content-based visual query. This type of query is applied on multimedia databases and can be combined with text-based simple query.

As a rule, the content-based visual query takes into consideration the primitive characteristics automatically extracted from images: color, texture or shape.

We have presented in detail the notion of color and algorithms for automated extraction of the gray-level or color information from medical images and also aspects of color texture characteristics extracting, algorithms and experiments on color images from digestive area.

Another important aspect in medical applications is the automated segmentation of images. For segmenting the color images from digestive area, we have implemented the color set back-projection algorithm. The chapter presents in detail our original implementations of this method and the results of the content-based region query on color, size and spatial extent on a database with color medical images.

The presented experiments have been realized with the help of MIR (Medical Image Retrieval) – a software tool for creating, managing and content-based visual querying of the multimedia medical databases with color images from the digestive area.

These experiments were realized along with medical experts from the two biggest university hospitals from Craiova and with financial support from Romanian Academy and National University Research Council.

In the content-based visual query of the medical databases and automated segmentation of medical images areas, there are wide researches, a big variety of techniques applied on various imaging modalities, taking into consideration many systems of the human body and pathologies.
This variety is caused by the fact that there isn’t a method with universal validity for all types of medical images and the research continues in this direction. We also mention image mining and automate diagnosis as intense research directions with a great impact in medical activity.

References

19. DICOM Homepage (2005), http://medical.nema.org/
20. DICOM Research (2005), http://www.alumni.ca/~chia4a0/dicomresearch.htm
52. Lab color space (2007), http://en.wikipedia.org/wiki/Lab_color_space


55. LEAD Technologies (2005), http://www.leadtools.com/SDK/Medical/DICOM/ltdc1.htm


77. Purdue University. Content-based Image Retrieval from Large Medical Image Databases, http://cobweb.ecn.purdue.edu/RVL/Projects/CBIR/index.htm


Bio-medical Ontologies Maintenance and Change Management

Arash Shaban-Nejad and Volker Haarslev

Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec H3G 1M8, Canada
arash_sh@cs.concordia.ca, haarslev@cs.concordia.ca

Abstract. Things change. Words change, meanings and context change. To manage a large volume of evolving bio-medical data of various types, one needs to employ several techniques from areas such as knowledge representation, semantic web and databases. Many of these techniques require a formal description of a part of the real world. Ontologies can provide a set of shared and precisely defined terms in various degrees of formality to describe a particular domain of interest. When the knowledge changes, then the related definitions will be altered. Changes to ontologies may occur for many reasons. The issues arising from ontological change can affect the validity of information in applications that are tightly bound to concepts in a particular ontological context. Many knowledge-based systems are now reaching a stage where they need a change management strategy to update their ontological knowledge. This area is becoming increasingly important in science as high throughput techniques frequently necessitate updates to existing scientific ‘truths’. In this chapter, we survey and review state of the art change management in bio-ontologies as well as some of the available tools and techniques in this area. We also survey various potential changes in biomedical ontologies, with actual examples from some of the most popular ontologies in the biomedical domain. In addition we investigate the potential of some of the advanced formalisms in this context by proposing our formal method for analyzing and supporting ontology evolution and change management.

1 Introduction

Employing clinical terminologies has a long history in medicine and life sciences (Bodenreider and Stevens 2006). However, it is a new trend to use ontology as it is defined by Gruber (1993), the “specification of conceptualization” that aims to provide an underlying discipline of sharing knowledge and modeling biomedical applications by defining concepts, properties and axioms. Ontologies are extensively being employed in biomedical systems to share common terminologies,
provide annotation, and organize and extract knowledge from a domain of interest. Ontologies are constantly evolving to fix errors, reclassify the taxonomy, and add/remove concepts, attributes, relations and instances. Due to the importance of ontology as the conceptual backbone of modern decision support systems in life sciences, the maintenance phase is crucial in an ontology life cycle to preserving the validity and consistency of ontological knowledge. Ontology maintenance traditionally includes two main activities: ontology integration and change management. However, there is not always a clear line to distinguish between these activities. In many cases, the ontology change management process requires one to perform some data or semantic integration, and it is also very unlikely to keep all the integrated ontologies unchanged. The maintenance cost for evolving ontologies and knowledge bases is relatively expensive, and may range up to 90% (Jones 1998) of the total cost, depending on the size, complexity and domain of an ontology.

Ontologies evolve all the time, and each change in the ontological structure or nomenclature can have a crucial impact on the inferred knowledge. Especially in a heterogeneous environment, like the Web, with vast numbers of interdependencies, even simple changes to ontological elements can trigger a domino effect, and it is very hard—sometimes impossible—to guess all the effects of a simple change. Different versions of an ontology behave differently in response to posed queries. If one works with a system based on frequently changing ontologies, how can s/he even ask queries and be sure of the logical and scientific correctness of the answer? The issues arising from ontology evolution can affect the validity of information in applications that are tightly bound to concepts in a particular ontological context.

In the last decade, several studies (Cimino and Clayton 1994, Oliver et al. 1999, Klein and Noy 2003, Stojanovic 2004, Noy et al. 2006, Flouris 2006) on ontology evolution have been reported. Despite worldwide efforts, the topic of ontology evolution is still a source of much debate, as it brings together various issues that are central to philosophy, logic, artificial intelligence, cognitive science, neural nets, linguistics and physics, including identity, persistence and time.

This chapter is structured as follows. First, Section 2 describes a brief philosophical foundation of change. Section 3 reviews state of the art change management in some selected biomedical ontologies. In Section 4, we look at different types of common alterations with some actual examples from popular bio-ontologies. We survey some of the tools and approaches other researchers take to handle ontology evolution in Section 5. Section 6 describes our proposed framework for autonomous ontology change management in an agent-based environment grounded in category theory by using the FungalWeb ontology (Baker et al. 2006) as the application scenario. This manuscript is partially based on papers (Shaban-Nejad and Haarslev 2007(a), 2007(b), 2008) and other conferences, with many additions, including a variety of new materials and examples.
2 Philosophical Foundations

Designing a framework for ontology evolution by using available methods in the area of knowledge representation (KR) is the main strategic plan in the Semantic Web community. However, since the problem of change management is not completely computational, it seems necessary to incorporate complementary techniques from other disciplines such as philosophy, mathematics, biology, neural networks, semiotics, linguistics, psychology (to study the behavioral affects), etc. (Fig. 1) for the ontology evolution process. The topic of change—and particularly changing ontology (as the study of “being”)—brings together various issues that are central to philosophy, including identity, persistence and time (Wasserman 2006).

Discussion about change is as old as philosophy itself. Heraclitus (535–475 BCE), for example, argued that “All is flux,” and everything is changing all the time, so that it is impossible to step into the same river twice. Parmenides (b. 510 BCE) and Zeno of Elea (490–430 BCE) were not in agreement with Heraclitus’s statement; they believed in the constancy and stability of the world. Parmenides had stated that “reality is one, and this one, which only is, is unchanging” (Magee 1999). Zeno of Elea also believed all changes and motions are in fact illusions of the senses (Hardie and Gaye 2004), and to show the paradoxical nature of change and motion, he summarized his philosophy into several paradoxes, including The
Dichotomy, Achilles and the Tortoise and The Arrow (Kemerling 2006). Plato (427–347 BCE) in his allegory of the Cave tried to overcome this issue by separating the world into the visible world, which is uncertain and changes frequently, and the intelligible or real world, which is stable, arose from reason and includes the timeless unchanging “Forms”. Husserl (born 1859) tried to define the concept of changes by considering the notion of time, saying, “Things are always intended toward something, and are always ‘about’ something,” which shifts the notion of ontology from studying “being” towards studying “becoming”.

It has been commonly acknowledged that a change happens in relation to time. However, Aristotle (384–322 BCE) in his book Physics IV argued that since change, unlike time, occurs at different rates, it is distinct from time (Hardie and Gaye 2004). The nature of change may appear contradictory and a source of inconsistency, as “it requires both sameness and difference” in parts and attributes (Wasserman 2006) and deals with contrary facts about the identity of things. See Leibniz’s Law at (Wasserman 2006), Theseus’s paradox at (Cohen 2004) and The Heap (Sorites) paradox at (Zalta 2005) for more information on change, persistence and identity.

Due to the paradoxical nature of change, change in a thing causes various problems, including the problem of the consistency of change. Some have said that the only way to make sense of change is through inconsistency (Varzi 2005). Many philosophers believe that studying and reasoning out change only make sense when things extend through “time”. It means the temporal parts of a changing “concept” can have different properties at different times (Varzi 2005). In order to talk about the identity of objects, ontologists need to distinguish between Continuants/Occurrents and Dependent/Independent and Universals/Particulars (Smith et al. 2003). According to Smith et al. (2003), Continuants (objects) are things that continue to exist through time and their identities remain unchanged. Occurrents (processes) are time dependent entities, whose identities unfold at different points in time. The existence of a “Dependent” depends on the existence of other things (i.e., a bodily injury is dependent upon the injured organ), in contrast to an “Independent”, whose existence does not necessarily depend on other things (i.e., atoms, molecules). Also, “Universals” can be considered classes or groups of things (i.e., “student”) while “Particulars” are “instances” of those classes (i.e., a specific student).

In Section 6, we will consider “time” as a primary factor in our approach to analyzing changes in temporal biomedical ontologies.

3 Biomedical Ontologies and the Editorial Procedure – State of the Art

There are currently a growing number of ontologies and controlled vocabularies in various areas of life sciences. In this section, we review the state of the art of change management in some available bio-ontologies. It is not a surprise that many of them do not sufficiently meet the requirements to be considered a formal
ontology (Guarino 1995). Most ontologies in the biomedical domain are recognized to be acutely defective from both terminological and ontological perspectives (Kumar and Smith 2003, Smith et al. 2003, Kumar et al. 2004, Grenon et al. 2004, Ceusters et al. 2004, Smith and Rosse 2004, Ceusters and Smith 2006, Smith and Ceusters 2006, Smith 2006). A list of open-source ontologies used in life sciences can be found on the Open Biological Ontologies (OBO) website (http://obo.sourceforge.net/). Many of the available ontologies are still under active development, revision and improvement, and are subject to frequent changes. The following ontologies and controlled vocabularies have been selected for a study of their change management mechanism based on several criteria, such as availability, popularity, and complexity of and accessibility to the source and documentation. The Gene Ontology (GO) (Ashburner et al. 2000) is a community standard and the Unified Medical Language System (UMLS) (Humphreys et al. 1998) is quite popular, with its rich collection of biomedical terminologies. Clinical Terms Version 2 (Cimino 1996, Bailey and Read 1999) deals with actual patient care records. We also look at HL7, FMA and Terminologia Anatomica (TA) (Whitmore 1999) to see different examples of potential changes.

3.1 The Gene Ontology (GO)

The Gene Ontology (GO) is a collaborative project (Ashburner et al. 2000) that intends to provide a controlled vocabulary to describe gene and gene product attributes in existing organisms based on their associated biological processes, cellular components and molecular functions. GO has been modeled and implemented based on three distinct ontologies, represented as directed acyclic graphs (DAGs) or networks consisting of a number of terms, represented by nodes within the graph, connected by relationships that are represented by edges (Lord et al. 2003). The current GO term count as of Dec 18, 2008 at 14:00 (PST) (http://www.geneontology.org/GO.downloads.shtml) is 26475 terms with 1362 obsolete terms. The GO consortium makes cross-links between the ontologies and the genes and gene products in the collaborating databases (Sklyar 2001). The Gene Ontology is currently available in Flat File, FASTA, MySQL, RDF-XML, OBO-XML and OWL formats. Members of the consortium contribute to updates and revisions of the GO. Changes in GO occur on a daily basis and a new version of GO is published monthly, a snapshot of the current status of the database (Klein 2004). As GO becomes larger and complexity arises, it also becomes more difficult to control and maintain. To ensure consistency of the modified ontology, all changes are coordinated by a few biologists in the GO editorial office staff, who have write access to the Concurrent Versions System (CVS) (Cederqvist 1993) repository in which GO files are maintained. The users can make requests for modifications through an online system that tracks the suggestions and manages the change requests. All tracking information about requests and changes are archived and several curator interest groups have been established with associated actively archived mailing lists (Harris 2005). The GO editorial staff notifies others of the changes via monthly reports (http://www.geneontology.org/MonthlyReports) to
the users (by email), or at the GO site. Different sources of suggested changes in GO, as described by Harris (2005), are advances in biology that alter the knowledge of gene and protein roles in cells, joining new groups that require new terms and relations, fixing errors, completing unfinished parts of the ontology, updating legacy terms and improving the formal representation of the ontology by identifying missing or misplaced relationships and terms. One of the problems in Gene Ontology maintenance is related to the versioning tool. CVS repositories, which currently handle versioning in GO, work based on syntactic differences between ontologies. For instance, CVS is not able to differentiate class versions, being able only to differentiate text/file differences (Volkel et al. 2005). The research on conceptualization change over time (Volkel et al. 2005) is still promising.

### 3.2 UMLS Semantic Network

The Unified Medical Language System (UMLS) (McCray and Nelson 1995) is a composite of about 100 source vocabularies that contain 870,853 concepts and 2.27 million terms (UMLS documentation, 2008). It was created by the National Library of Medicine (NLM) to facilitate the development of computer systems that behave as if they "understand" the meaning of the biomedicine/health language. To that end, the NLM produces and distributes the UMLS knowledge sources (databases) and associated software tools (programs) to system developers for use in informatics research and in building or enhancing electronic information systems that create, process, retrieve, integrate, and aggregate biomedical/health data and information. The UMLS Knowledge Sources are multi-purpose, and can utilize a variety of data and information, such as patient records, scientific literature, guidelines and public health data (UMLS documentation 2008). Due to the popularity and multi-purpose nature of the UMLS, it seems to be a perfect candidate to study change management. The UMLS Semantic Network covers different levels of granularities, which have a key effect on interpreting the meaning that has been assigned to the Metathesaurus concepts (Fact sheet of UMLS 2006). Changes in the UMLS are usually recommended by the UMLS contractors and others who have experimented with the previous versions of the ontology. UMLS terms that share the same conceptual meaning are linked by a concept unique identifier (CUI) (Campbell et al. 1995). Two files called DELETED.CUI, which lists deleted concepts, and MERGED.CUI, which lists all pairs of CUIs that were merged, are associated with each new release of the UMLS (Olson et al. 1996). These files help users to determine whether a CUI that is no longer present in the new version was removed due to a deletion of the concept, or due to a merger of the concept with another concept (Oliver et al. 1999).

### 3.3 Clinical Terms Version 3 (The Read Codes)

The Clinical Terms Version 3 (CTV3) (http://www.nhsia.nhs.uk/terms/pages/) (NHS Information Authority 2000.a) or Read Codes are a set of coded terms arranged in a hierarchical structure for use in clinical practice, with such applica-
Bio-medical Ontologies Maintenance and Change Management

The CTV3 classifies chemicals by their name, i.e., alphabetically. The first version of Read Codes (CTV1) was initially developed to provide a terminology for describing relevant clinical summaries and administrative data for general practice. It is known as the 4-Byte Set since each code is four characters long. In the next version (CTV2), the codes were subsequently adapted for use in hospitals, and were extended to allow more detail. To hold more detailed information, a supplementary alphanumeric character was included in the Read Codes (5-Byte Sets) (NHS Information Authority 2000.b). CTV2 uses the code to specify a class and its unique place within the taxonomy, which has a limited, fixed number of levels. The CTV3, with its flexible structure unlike the previous versions, allows more changes in terminology (Jurisica et al., 2004). The Read Codes have been changed in each version (based on strict protocol under central control of NHS) by adding terms and codes to fix the errors and reflect the newly discovered knowledge (mostly to enrich the descriptions). Further alterations include changes to qualifiers and atoms (semantic definitions), the hierarchical structure and the mapping files (NHS Information Authority 2000.a). CTV1 and CTV2 changed relatively little between releases, due to their rigid file structure that was limited to five levels of offspring, and about 60 siblings. The CTV3 “Description Change File” (DCF) (NHS Information Authority 2000.a) shares the entire change management procedure between “terminology providers” and “terminology users” (i.e., clinicians). The DCF starts by recommending a new code for any terminology discovered to be incorrectly classified and suggesting that the user replace it. The process continues by labeling the obsolete concepts as “extinct”. An example from (NHS Information Authority 2000.a) describes the deletion of the relation between the terms ‘Cardiac rupture’ and ‘Myocardial infarct’, which turned out to have the same code in CTV2, and the addition of a new code to ‘Cardiac rupture’ in CTV3.

We also consider some other popular controlled vocabularies in life science including:

**Health Level 7 (HL7):** HL7 (http://www.hl7.org/) is an ontology that aims to provide a UML-based standard for the exchange, management, and integration of data to support clinical patient care and the management, delivery, and evaluation of healthcare services.

**The Foundational Model of Anatomy (FMA):** FMA (http://sig.biostr.washington.edu/projects/fm/AboutFM.html) represents a coherent body of explicit declarative knowledge about human anatomy and claims to be the most complete ontology of canonical human anatomy in a high granularity from the macromolecular to the macroscopic levels (Rosse and Mejino 2003).

**Terminologia Anatomica (TA):** Terminologia Anatomica (Whitmore 1999) is a standard controlled vocabulary on human anatomical terminology, developed by the Federative Committee on Anatomical Terminology (FCAT).
4 Different Types of Changes in Biomedical Ontologies

Based on our research of the literature, observations of different releases of ontologies, surveys, and interviews with several domain experts and ontology engineers, we distinguished 74 different types of changes that frequently occur in the life cycles of existing bio-ontologies. These changes are usually classified under ten general terms: addition, deletion, retirement (obsolescence), merging, splitting, replacement (editing or renaming), movement, importing, integration, or changes to file structure.

- **Addition**: This is one of the basic operations in ontology evolution, and aims to improve ontological structure by adding one or more components to the available structure. The most common additions in the observed bio-ontologies are of the following elements: namespace, identifier code, concept, attribute, abbreviation, super/subclass, attribute value, synonym, constraint (cardinality, type, min/max, inverse roles, and default value), associative relationships, annotation description and instance. As an example from Gene Ontology, in order to update the biological structure of GO for the annotation of genes, the GO maintenance team needs to improve the content of the ontology regularly. For instance, as described in (GO Newsletter, May 2006), the curators at MGI, who were reviewing the existing terms for the comprehensive annotation of “mammalian genes involved in the regulation of blood pressure”, realized that the existing GO terms were not sufficient to annotate these genes. They then proposed 43 new terms (http://www.informatics.jax.org/searches/GO.cgi?id=GO:0008217), which were accepted after refinement in the GO discussion forum, and new annotations for the mouse genes involved in blood pressure regulation came into existence. Another example from GO (GO Newsletter, Feb 2007) is the addition of a new evidence code (http://www.geneontology.org/GO.evidence.shtml) used for the annotation of gene products, "Inferred from Genomic Context" (IGC).

- **Deletion**: Deletion in ontology evolution process, refers to the erasure of a selected element of an ontology when it does not reflect the ontological “truth” anymore. The most common deletions in the bio-ontologies are of the following elements: namespace, identifier code, concept, synonym, abbreviation, annotation (description), constraint (cardinality, type and min/max, inverse roles, default value), attribute value, super/subclass, associative relationships, and instance. For example, the GO terms must characterize biological entities. As stated in (GO Newsletter, May 2006), the terms classified as “Unknown” violated this principle, so the decision was made to delete the following terms, biological process unknown; GO:0000004, molecular function unknown; GO:0005554 and cellular component unknown; GO:0008372, from the ontology. The new annotations signify that a given gene product should have a molecular function, biological process, or cellular component, but that no information was available as of the date of annotation.
• **Retirement (Obsolescence):** Retirement or, as it is referred to in (Cimino 1996), obsolescence means diminishing the role of an older ontological element when a newer, more functional element or meaning supersedes it. In this situation, the older version can be kept somewhere for future use, but its usage will be discouraged. The retirement process usually involves concepts, attributes, identifier codes, instances and associative relationships. For example in Release 2.0 of HL7, the certain components, ClinicalDocument.copyTime, MaintainedEntity, CodedEntry, inkHtml.name, table.border, table.cellspacing and table.cellpadding, are retained for backwards compatibility with HL7 Clinical Document Architecture (CDA), Release 1.0, and have been retired. Further use of these components is discouraged (Dolin 2004).

• **Merging:** This is defined as the process of creating a consistent and coherent ontological element that includes information from two or more basic elements. The process of merging can be seen in this format: merging two or more concepts/associative relationships/identifier codes/attributes into one of the same or into a new concept/relationship/code/attribute (Oliver et al. 1999). For instance, in HL7, the purpose of the header is to enable clinical document management, compilation and exchange across institutions (Dolin 2004). In HL7’s Clinical Document Architecture (CDA), Release 2.0, two concepts in the header (service_actor and service_target) have been merged (Dolin 2004).

• **Splitting:** An ontological element may be split into two or more new elements. This means that a concept, associative relationship, identifier code, or attribute can be split into two or more of the same. For example, in Terminologia Anatomica (TA) (Whitmore 1999), terms that share an identifier code are considered synonyms. However, this is not valid for sexually dimorphic anatomical parts, such as ‘Ovarian artery’ and ‘Testicular artery’, as they share the same TA code (A12.2.12.086). The two arteries are not synonymous but distinct, and are involved in different relationships. Therefore, they need to be treated as two separated concepts, meaning the code A12.2.12.086 can be split into A12.2.12.086-1 for ‘Ovarian artery’ and A12.2.12.086-2 for ‘Testicular artery’ (Whitmore 1999).

• **Replacement (Edit, Renaming):** This process is for editing available labels and values. Editing mainly occurs to change namespace, concept name/definition/role, and attribute value/name/definition. A typical scenario (Dolin, 2004) from HL7, Release 2.0, is a simple replacement of

\[
\text{ClinicalDocument.id } "1.2.345.6789.123"
\]

With

\[
\text{ClinicalDocument.id } "1.2.345.6789.266"
\]

Another example from Terminologia Anatomica (TA): TA renames the ‘Angiologia’ chapter of Nomina as ‘Cardiovascular system’. This new name resolves the inconsistency in Nomina Anatomica (the older anatomy
classification), which assigned the heart to the ‘system of vessels’ (Angiologia), rather than to viscera (Splanchnologia) (Rosse 2000).

- **Movement (Transition):** This is defined as the transition of one or more ontological elements across the ontological hierarchy. This transition can happen to identifier codes, concepts, attributes, super/subclass, associative relationships, and instances. For example, GO terms representing transporter activity in the Molecular Function sub-ontology are gradually being overtaken to better represent current scientific knowledge. A new high-level term called “transmembrane transporter activity” (GO:0022857) was introduced. In addition, the related child terms and subclasses have been moved and organized under GO terms that describe the activity of the transporters, such as channel, active transporter, symporter, antiporter and uniporter activities (GO Newsletter, Aug 2007).

- **Importing:** This refers to the process of bringing an existing ontology (a tree) or parts of an existing ontology (a sub-tree) into another ontological structure. As an example from Gene Ontology, in 2001, the GO developers imported the first pass annotation from SWISS-PROT (http://ca.expasy.org/sprot/), trEMBL (http://www.ebi.ac.uk/trembl/), and Ensembl (http://www.ensembl.org/index.html) (GO Meeting collected notes, 2001). Also, 7,316 GO annotations were imported from Proteome and literature associations (GO Meeting collected notes, 2001).

- **Integration:** In data integration, scattered process data is extracted from different sources with different data formats, and then “normalized into a consistent syntactic representation and semantic frame of reference” (Buttler et al. 2002). Semantic integration is more complex than data integration. For example (Martin et al. 2001), in the Foundational Model of Anatomy (FMA), to have a comprehensive knowledgebase in neuroinformatics, the FMA developers have integrated several terminologies of NeuroNames and Terminologia Anatomica into the FMA. They have enhanced the FMA to support unique information on neuronal structures (Martin et al. 2001).

- **Changes to release file (File structure):** By the advancement of technology for storing and retrieving data files and the emergence of new standards, the format of file structures can be changed. As an example, in the first version of Read Codes, four-character alphanumeric codes determined the position of a term in a hierarchy; this version is known as the 4-Byte Set (Bentley 1996). The restrictions imposed by only four levels of hierarchy led to the development of a 5-Byte Set, which expanded the set to support secondary and tertiary care. As stated in (Robinson et al. 1997) “this set was released in two structurally different versions, and has increased content and provided a more structured mechanism for representing synonyms”. Ver. 1.0 has shorter terms and keys than Version 2.0. Ver. 3.0 with its rich structure supports the character structures of both Versions 1.0 and 2.0 (Robinson et al. 1997).
A comprehensible survey on different types of ontological changes and relations between change management practices with others overlapping research disciplines can be found in (Flouris et al. 2008).

5 Tools and Methods to Support Ontology Change Management

There are a few tools (Haase and Sure 2004, Stojanovic 2004) to manage changes in ontologies. Some of the available tools are simply ontology editors, such as Protege (Noy et al. 2000), OntoEdit (Sure et al. 2003) and TopBraid Composer (http://www.topquadrant.com/topbraid/composer/index.html). Despite their differences, they all assist users in implementing, updating and managing elementary changes in ontologies. According to (Stojanovic 2004, Stojanovic and Motik 2002), the most critical requirements for ontology editors in order to be more robust in a changing environment are related to functionality, customizability, transparency, reversibility, auditing, refinement and usability. Other available tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protege (Noy et al. 2000)</td>
<td>A popular ontology design environment with support for RDF and OWL ontologies. It provides some editing facilities such as: adding/deleting/renameing ontological elements, undo/redo of changes and version archiving (Liang et al. 2005). Protege also includes plug-ins such as PROMPT for managing multiple ontologies. It can compare versions of the same ontology, merge two ontologies into one and extract part of an ontology (Noy and Musen 2003). PromptDiff (Noy and Musen 2004) also can determine the changes between two versions.</td>
</tr>
<tr>
<td>TopBraid Composer (TopBraid Composer Guide 2007)</td>
<td>A commercial ontology editor that supports editing RDF Schemas and OWL Ontologies, as well as executing rules and queries in the SPARQL Protocol and RDF Query Language (SPARQL) (Beckett 2006) and the Semantic Web Rule Language (SWRL) within a multi-user environment. It manages multiple versions of ontologies by using the following set of rules. Any changes to the statements are written into the source ontology. If the change is “overtyping” an entry, it will be saved in the original ontology as an update. In case of the “deletion” of an entry and then the “addition” of a new one, the deletion would be done in the original file and the new triple would be saved in the existing file. Also, by changing any class, the composer scans to see if there are any other ontologies that import this class. It keeps a log of the changes that is accessible from the Change History view. Unsaved changes can be undone. To prevent accidental changes, a file can be defined as “read only”.</td>
</tr>
<tr>
<td>Table 1. (continued)</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Concurrent Version System (CVS)</strong> (Cederqvist 1993) (<a href="http://www.nongnu.org/cvs">www.nongnu.org/cvs</a>)</td>
<td>Supports basic version control functionality and maintains a history of the changes. CVS can reveal syntactical and textual differences between two files. It mostly works on the syntactic level. Since ontology versioning and change management need operations on the conceptual level rather than the syntactic level, CVS might not seem an appropriate tool for ontology change management (Völkel and Groza 2006). However, CVS can provide basic support for managing structural changes in RDF and OWL files.</td>
</tr>
<tr>
<td><strong>CONCORDIA</strong> (Oliver and Shahar 2000)</td>
<td>A model for managing divergence in concept-based terminologies, developed to facilitate the study of synchronization in health care terminologies. CONCORDIA uses the models of Medical Subject Headings (MeSH) (Nelson et al. 2001), ICD-9-CM (Cimino 1996), and ICD-10. It enables one to manage 27 different kinds of changes, such as adding, deleting, retiring, or merging concepts, terms or attributes (Oliver and Shahar 2000). CONCORDIA does not provide any services to log motivations for the changes (Ceusters and Smith 2006).</td>
</tr>
<tr>
<td><strong>KAON</strong> (Maedche and Staab 2003), <a href="http://kaon.semantic-web.org/">http://kaon.semantic-web.org/</a> (Gabel et al. 2004).</td>
<td>An integrated open-source ontology management system targeted at semantics driven business applications, KAON components can be divided into 3 layers: (i) The applications/services layer realizes user interface applications and provides interfaces to non-human agents; (ii) The API, which is the major part of KAON, checks the validity of change sequences, and also requests user approval for performing a change, justifies the necessity of a particular change, executes the modifications, reverses the effect of some undesirable changes and keeps a history of changes; (iii) The data and remote services layer provides data storage facilities. See (Gabel et al. 2004) for more information.</td>
</tr>
<tr>
<td><strong>OntoView</strong> (Klein et al. 2002)</td>
<td>A web-based system that assists users in handling ontology evolution. The system helps to keep different versions of web-based ontologies interoperable by maintaining the transformations between ontologies and the relations between concepts in different versions. OntoView was inspired by and can be considered a Web interface for CVS. OntoView compares ontologies at a conceptual level, analyzes effects of changes (e.g., by checking consistency and highlighting the places in the ontology where conceptually changed concepts or properties are used) (Klein et al. 2002)) and utilizes changes.</td>
</tr>
<tr>
<td><strong>OntoManager</strong> (Stojanovic et al. 2003)</td>
<td>Has been designed to assist ontology managers in managing ontologies according to the users’ requirements. The technique used to evaluate users’ needs depends on the information source by tracking user interactions with the application in a log file. The OntoManager consists of three modules: (i) The data integration module, which aggregates, transforms, and correlates the usage data; (ii) The visualization module that presents the integrated data in a comprehensible visual form; and (iii) The analysis module, as the major part of the change management, provides guidance for adapting and consistently improving the ontology with respect to the users’ requirements. This module keeps track of the changes and has the ability to undo any action taken upon the ontology.</td>
</tr>
</tbody>
</table>
include but are not limited to Concurrent Version System (CVS) (Cederqvist 1993), CONCORDIA (Oliver and Shahar 2000), KAON (Maedche and Staab 2003, Gabel et al. 2004) management tool, OntoView (Klein et al. 2002), OntoManager (Stojanovic et al. 2003) and TextToOnto (Maedche and Volz 2001). Table 1 represents some of the popular ontology editors and management tools with their descriptions.

As can be seen from the current state-of-the-art change management in existing ontologies in life sciences, the current biomedical ontologies do not follow any standard, consistent, formal change models with clear semantics. Most of the available tools are just simple ontology editors with a few extended features. Some parts of ontology evolution, such as the change representation and conceptualization change, are not satisfactorily managed by existing tools and they are left to be handled by the users. The major issues in available ontology management tools can be summarized as: (i) Too much reliance on human decisions due to lack of fully automatic ontology change management tools and too much dependency of the existing systems on the human factor (Haase and Sure 2004), which both give rise to several issues relating to complexity, accuracy, security and reproducibility (Flouris 2006); (ii) Representation and tracking of complex changes using available technologies are limited; (iii) Lack of formal evaluation methods, which makes the comparison and evaluation of different algorithms extremely difficult (Flouris 2006); (iv) Little or no support for conceptualization change management; and (v) Change models that have been designed based on time/space independent ontologies.

6 The RLR Framework for Ontology Change Management

Here we propose our approach, based on the RLR framework (Shaban-Nejad and Haarslev 2008), for recruiting intelligent agents to capture the pattern of changes, track a change, predict the rate and direction of changes and validate the results. The RLR framework will be used to Represent, Legitimate and Reproduce the changes and their effects (Fig. 2).

In this structure, all changes are to be captured, represented and then validated logically and approved publicly and by experts. To support reproducibility of the outcomes and automate the ontology evolution process, agents can be used to learn the patterns of changes and their consequences in the Reproduction phase.
The Representation phase updates the representations (either formal or diagrammatical) of the new knowledge in a consistent way. For formal representation, we use Description Logic (DL) (Baader et al. 2003), and for diagrammatical representation, we employ a method based on discrete state model and category theory (Shaban-Nejad and Haarslev 2007(a)). The Legitimation stage determines the consistency and validity of a change. This phase assesses the impact of a potential change before the change is actually made. Software agents can search, notify, perceive and collect information about different actions from multiple sources in a frequently changing environment. Intelligent agents perform rationally in dynamic environments and provide prompt responses. In our framework, we have employed various types of agents: the change capture, learner, reasoning and negotiation agents.

The change capture agents perceive and capture different changes and accumulate the related information within the change logs. The learner agent can use these records of changes, which occur over and over in a change management procedure, to design a change pattern. This means that after several changes—most likely in different releases—the direction and rate of change can be predicted with reasonable accuracy. The learner agent and the change patterns are essential in the Reproduction phase. A reasoning agent controls the logical validity of the ontological structure and reveals inconsistencies, hidden dependencies, redundancies and misclassifications. RACER (Haarslev and Möller 2001) has been used in RLR as a Description Logic reasoning agent. The negotiation agent acts as a mediator to assist the ontology engineer and other autonomous agents in negotiations concerning the realization of particular change. The negotiation outcome will be confirmed, deleted, or modified by human experts in accordance to the purpose of the system.

We have used category theory for mathematical notation, which is independent of a specific choice of ontology language and any particular implementation, as
the main formalism of the RLR framework. Furthermore, we will demonstrate the applicability of category theory for representing changes in the ontological structure.

### 6.1 Representation of Evolving Ontologies Using Category Theory

We noticed that the main issues in change management strategies are simply problems about the representation of change. Category theory is an intuitive domain of mathematics, introduced in 1945 (Eilenberg and MacLane 1945). Category theory is grounded in logic and algebra, and allows an ontology engineer to define different states of conceptual models to represent the reality. Categories can be used for specifying different objects, capturing their interactions and the relations between them, determining patterns of interacting objects and decomposing a composite object into its basic elements (Ehresmann and Vanbremeersch 2006). Categorical representations consist of diagrams with arrows (i.e., \( f: X \rightarrow Y \)). A typical category includes classes of objects and morphisms (arrows), and for each morphism \( f \), there is one object as the domain of \( f \) (i.e., \( X \)) and one object as the codomain (i.e., \( Y \)). In addition, for each object, \( X \), we can define an identity morphism (\( \text{id}_X \)) which has domain and codomain \( X \). Also, for each pair of morphisms \( f: X \rightarrow Y \) and \( g: Y \rightarrow Z \), (in other words, \( \text{cod}(f) = \text{dom}(g) \)) a composite morphism \( (g \circ f: X \rightarrow Z) \) exists. The representation of a category can be formalized using the notation of a diagram. Ontologies in their simplest form can be seen as the categorization of things in the real world. It is quite common to use diagrammatical notations to analyze ontological structures. For example, the UMLS Semantic Network (SN) has been modeled as a graph, whose nodes denote concepts (McCray 2003). Category theory, with its analytical characteristics, can act as the prospective knowledge representation language, particularly for modeling dynamic abstract models, where ontologies can be defined as an interconnected hierarchy of categories (Healy and Caudell 2006) with directed ontological relationships as “morphisms” (Krötzsch 2005). As a knowledge representation language, category theory improves the incremental analysis of changes in ontological structures. Some of the primitive constructors of category theory, which we use in our framework for ontology change management, are as follows: primary categorical objects and morphisms, functors, pushouts, pullbacks, natural transformations and isomorphisms.

**Functors** are a special type of mapping between categories, defined as morphisms in the category of all small categories (classes are defined as categories) (Awodey 2006). As defined in (Van Oosten 2002), assume \( A, B \) are two categories, so a functor \( F: A \rightarrow B \) is joint functions: \( F_0: A_0 \rightarrow B_0 \) and \( F_1: A_1 \rightarrow B_1 \), such that:

\[
\forall f: \alpha \rightarrow \beta \in A_1 \Rightarrow F_1(f): F_0(\alpha) \rightarrow F_0(\beta) \in B_1 \\
\forall \alpha \in A_0 \Rightarrow F_1(\text{id}_\alpha) = \text{id}_{F_0(\alpha)} \\
\forall g \circ f \in A_1 \Rightarrow F_1(g \circ f) = F_1(g) \circ F_1(f) \in B_1
\]

A categorical model can represent transitions between different states of an ontological structure. To represent and track this kind of transition, we represent the conceptualization of things indexed by time (i.e., from the FungalWeb Ontology...
(Baker et al. 2006), “enzyme has_KM_value at t” is rendered as “enzyme-at-t has_KM_value”). Then we represent a set of time-indexed categories using functors to capture different states of ontological structure at different points in time. The category O at time t \( (O_t) \) models the state of the ontologies and all related interactions at this time. A functor can represent the transition from \( O_t \) to \( O_{t'} \) (Fig. 3), where the time changes from t to t’. Also, each sub-ontology A can be modeled by the series of its successive states, \( A_t \), from its creation to its destruction (Ehresmann and Vanbremeersch 2006).

**Fig. 3. Using Functor for studying model transition between different states**

Also, categorical arrows (morphisms) can be used to describe different conditional changes and to measure coupling as the extent of the connections between elements of a system (Shaban-Nejad and Haarslev 2007(a)). Coupling identifies the complication of a changing system. Measuring coupling is useful for predicting and controlling the scope of changes to an ontological application. Often a change in one class can cause some changes to the dependent classes. When the coupling is high, it indicates the existence of a large number of dependencies in an ontological structure, which must be checked to analyze and control the chain of changes. Coupling for ontological elements can be described by the number of connections and links between them. Therefore, we focus on arrows in category theory to study these connections. For analyzing a conditional change, we followed the formal model described in (Whitmire 1997) by identifying three types of arrows in our category: precondition, post-condition and message-send arrows for an existing category (Whitmire 1997). The type of message is determined by the types of changes caused by a method.

6.2 Application Scenario

For the application scenario, we have applied our method for managing changes to the FungalWeb Ontology (Baker et al. 2006). The FungalWeb Ontology is a formal ontology in the domain of fungal genomics, which provides a semantic web infrastructure for sharing knowledge using four distinct sub-ontologies: enzyme classification based on their reaction mechanism, fungal species, enzyme substrates and industrial applications of enzymes. The ontology was developed in OWL-DL by integrating numerous online textual resources, interviews with
domain experts, biological database schemas (e.g., NCBI (Wheeler et al., 2000), EC, NEWT (Phan et al. 2003), SwissProt (Bairoch 2000), Brenda (Schomburg et al. 2004)) and reusing some existing bio-ontologies, such as GO and TAMBIOS (Baker et al. 1998). Fungi are widely used in industrial, medical, food and biotechnological applications. They are also related to many human, animal and plant diseases. It is estimated that there are about 1.5 million fungal species (Heywood 1995) on the earth, but only about 10% of those are known and only a few of them have an identified usage. The fungal taxonomy is frequently changing. Most of the alterations are changes in names and taxonomic structure and relationships.

- **Changes in Names:** Fungal names reflect information about the organisms. Therefore, as our knowledge increases, names need to be changed when they no longer express the correct information (Crous 2005). These changes may cause misunderstanding and miscommunication, and affect the soundness of different queries. Crous et al. (2003) describe an example of eyespot disease in cereals and the issues related to naming its associated fungi. Most fungi names are currently based on their visible characteristics and appearances. To manage this process of constant evolution, one solution is employing ontologies, where names are only meaningful once linked to descriptive information that was extracted from trustworthy data sources. The incorporation of DNA data is also crucial to ensure stability in names and uniquely distinguish the species. At this time, only about 1.1% of the estimated 1.5 million species are represented by DNA sequence data, meaning very few have been preserved from change (Hawksworth 2004).

- **Changes in Taxonomic Structure:** By advancing our knowledge in life sciences, fundamental changes in taxonomical structure and relationships can be foreseen. For instance, by studying some molecular, morphological and ecological characteristics, *Glomeromycota* was discovered in 2001 (Schüßler et al. 2001) as a new fungal phylum. Another example is the sedge parasite *Kriegeria eriophori*, which has never been satisfactorily classified. Recently, ribosomal RNA gene sequences and nucleus-associated ultrastructural characters were analyzed separately and combined to define the new subclass, *Microbotryomycetidae* (Swann et al. 1999). Fig. 4 represents how the place of the concept “pH optimum” has been changed within the FungalWeb taxonomy (ver. 2.0) by adding the new concept “Functional Property”.

![Fig. 4. A simple change in taxonomical structures of two consecutive versions of the FungalWeb Ontology (FWOnt)](image-url)
Fig. 5 demonstrates a portion of the FungalWeb application in categorical representation.

Fig. 5. A category model of portion of the FungalWeb application

Most common operations during ontology evolution, such as adding or deleting a class/property/relationship, combining two classes into one, or adding a generalization/association relationship, can be represented using category theory (Shaban-Nejad and Haarslev 2007(b)). Based on our application, we designed our class diagrams following Whitmire (1997) to track different states of our ontological structure (Fig. 6). The $O_p$ arrows in this figure represent the operations, which cause an ontological state to change to another state. For instance, in Fig. 11, the operation $O_{p_1}$ causes the transition of an object from state $St_1$ to state $St_2$. The $O_{p_r}$ (reverse operation) returns the system to its initial state ($St_1$). For more details see (Shaban-Nejad and Haarslev 2007, 2008)

Fig. 6. A class diagram to track different operations and states in an evolving structure

Summary

Biology is known as a field with continuous evolution. As the knowledge about biology rapidly grows and new methods become available, one can anticipate a
fundamental change in the way we capture the knowledge of the real world. One of the important activities in knowledge representation and bioinformatics is properly responding to changes and coping with the ontological evolution. Research on ontology change management is an ongoing effort that is still in its early stages. Many tools and techniques are available to assist ontology engineers to implement changes. However, they still have long road ahead to be considered for practical usage due to following issues:

- Lack of formal change models with clear semantics
- Inconsistencies among change models and log models
- Too much reliance on human decisions
- Reproducibility of the results cannot be guaranteed
- Little or no support for the representation of complex changes
- Lack of formal evaluation methods
- Little support for handling changes in conceptualization

The proposed RLR framework can be used to Represent, Legitimate and Reproduce the changes and their effects. Using this framework can help capture, track, represent and manage the changes in a formal and consistent way, which enables the system to create reproducible results. In this framework, all changes should be represented in either formal or diagrammatical representations. The change then should be legitimated and validated logically, by a reasoning agent, and it should be approved publicly and by experts. In order to reproduce the results of changes and automate the change management process, agents can be used to learn the patterns of changes, using learner agents, while the negotiation agent acts as a mediator that allows the ontology engineer and other autonomous agents to negotiate the proper implementation of a specific change. We are currently using the RLR framework to manage the evolving structure of the FungalWeb Ontology. Due to importance of the Representation phase, we concentrated more on this issue in this manuscript. For diagrammatical representation, we proposed using category theory, which has significant potential as a supplementary tool for capturing and representing the full semantics of ontology-driven applications. Category theory can provide a formal basis for analyzing complex evolving biomedical ontologies. Our future research will be focused on improving the reasoning process and generalizing the usage of category theory along with other formalisms to improve ontological conceptualization change management. Due to the multidisciplinary nature of research on ontology change management, any advances in this field would be highly dependent on advances in the research of various related topics, such as ontology integration, translation, merging or alignment, and computer-human interactions.

References


http://www.dajobe.org/2005/04-sparql/
SPARQLreference-1.8.pdf (accessed January 10, 2009)

http://www.phcsg.org/main/pastconf/camb96/readcode.htm


http://faculty.washington.edu/smcohen/320/theseus.html

Fact Sheet of Unified Medical Language System (UMLS) Semantic Network (2006),
Gabel, T., Sure, Y., Voelker, J.: KAON – ontology management infrastructure. SEKT informal deliverable 3.1.1.a, Institute AIFB, University of Karlsruhe,
Go Browser representing blood pressure terminologies,
GO Meeting collected notes, Hosted by Judy Blake and the Jackson Lab in Bar Harbor, ME. compiled by L. Reiser (July 14-15, 2001),
http://www.geneontology.org/minutes/collected_minutes.txt
GO Newsletter, No. 1 (May 2006),
http://www.geneontology.org/newsletter/archive/200605.shtml#bp
GO Newsletter, No. 4 (February 2007),
GO Newsletter, No. 5 (May 2007)
GO Newsletter, No. 6 (August 2007)


Smith, B.: Realism, Concepts and Categories or: how realism can be pragmatically useful for information. Talk in OntoQuery at Copenhagen Business School, May 18-22 (2003)


**Abbreviations**

CDA: Clinical Document Architecture

CUI: Concept Unique Identifier

CVS: Concurrent Versions System

DAG: Directed Acyclic Graphs

DL: Description Logic

FMA: The Foundational Model of Anatomy
FWOnt: FungalWeb Ontology
GO: Gene Ontology
HL7: Health Level 7
KR: Knowledge Representation
MeSH: Medical Subject Headings
NLM: The National Library of Medicine
OBO: Open Biological Ontologies
OWL: The Web Ontology Language
RACER: Renamed Abox and Concept Expression Reasoner
RLR: Representation, Legitimation and Reproduction
SPARQL: SPARQL Protocol and RDF Query Language
SWRL: The Semantic Web Rule Language
TA: Terminologia Anatomica
UMLS: The Unified Medical Language System
Extraction of Constraints from Biological Data

Daniele Apiletti, Giulia Bruno, Elisa Ficarra, and Elena Baralis

Department of Control and Computer Engineering (DAUIN),
Politecnico di Torino
C.so Duca degli Abruzzi 24, 10129 Torino, Italy

1 Introduction

Data constraints are used in structured and unstructured databases to capture real-world semantics observed in the modeled application domain. In our context, a constraint can be defined as a set of predicates \( P_1 \land P_2 \land \ldots \land P_k \). Each predicate is in the form \( C_1 \theta C_2 \), where \( C_1 \) is an attribute, \( \theta \) is a comparison operator and \( C_2 \) is either an attribute or a constant [15]. Constraints are assertions on permissible or consistent database states, and specify certain properties of data that need to be satisfied by valid instances of the database.

Constraints show dependencies among data and add semantics to a database schema. Thus, they are useful for studying various problems such as database design, query optimization and dimensional reduction. Constraints are usually introduced at design time to describe a priori knowledge. Consequently, the valid instances of the database are those satisfying simultaneously all constraints. However, collected data can hide interesting and previously unknown information because of unstated constraints. For example, this happens when data is the result of an integration process of several sources or when it represents dynamic aspects. Furthermore, the design process is not always complete and the constraint definition may be omitted from the design. The analysis of heterogeneous data with the aim of detecting implicit information is an important and useful task, which may become complex due to the size of datasets.

Among the numerous constraint types, we focus on table constraints, which refer to a single relation of the database. Examples of such constraints are domain constraints and tuple constraints. A domain constraint restricts allowed values of a single attribute, i.e. it describes its domain. A tuple constraint limits the allowed values for several (related) attributes of the same tuple. Constraints are properties of the database schema, thus they can not be directly inferred from data. We will denote this type of constraints as *schema constraints*. However, if the constraints are not a priori known, they can be hypothesized by analyzing database instances. We can not directly infer schema constraints from data, but we can infer *instance constraints*, i.e., constraints which represent the current relationships holding among data. Instance constraints represent a snapshot on the current database state.
of possible schema constraints. However, since they are inferred from data, they highlight possible schema properties, but do not represent actual schema properties. An instance constraint may become a schema property when it is validated by an application domain expert.

Since our aim is to infer unknown constraints from data, we will focus on instances constraints. We will denote instance constraints with the term constraint in the remainder of this chapter. For example, a constraint may be inferred by analyzing the values of the Age attribute. If it always has positive values, we can infer the (instance) domain constraint \( \text{Age} > 0 \). Instead, by analyzing a mark database, we may detect a tuple constraint between the values of the attributes Mark and Honours. If the Honours attribute is true, Mark should take the highest value. Instead, if Mark does not take the highest value, Honours is false. If all the values of two attributes are linked by tuple constraints, then there is a functional dependency between the attributes. A functional dependency states that if in a relation two rows agree on the value of a set of attributes \( X \), then they also agree on the value of a set of attributes \( Y \). The dependency is written as \( X \rightarrow Y \). For example, in a relation such as \( \text{Buyers} (\text{Name, Address, City, Nation, Age, Product}) \), there is a functional dependency \( \text{City} \rightarrow \text{Nation} \), because for each row the value of the attribute City identifies the value of the attribute Nation (i.e., a city always belongs to the same nation).

Unfortunately, real datasets are affected by errors. Errors can be divided into two categories: syntactic and semantic. Among syntactic errors there are incompleteness (due to the lack of attribute values), inaccuracy (due to the presence of errors and outliers), lexical errors, domain format errors and irregularities. Among semantic errors there are discrepancies, due to a conflict between some attribute values (i.e. age and date of birth), ambiguities, due to the presence of synonyms, homonyms or abbreviations, redundancies, due to the presence of duplicate information, inconsistencies, due to an integrity constraint violation (i.e. the attribute age must be a value grater than 0) or a functional constraint violation (i.e. if the attribute married is false, the attribute wife must be null), invalidities, due to the presence of tuples that do not display anomalies of the classes defined above but still do not represent valid entities [12].

Public genomic and proteomic databases may be affected by the aforementioned errors, mainly because they grew some years ago, under the pressing need of storing the large amount of genetic information available at the time, without having neither a standard method to collect it, nor a standard format to represent it. Due to these problems it is difficult to clearly describe the actual relationships among data. Recently, a significant effort has been devoted to the integration of distributed heterogeneous databases, where researchers continuously store their new experimental results. However, the existence of erroneous or poor data may harmfully affect any further elaboration or application.

Errors can occur in different phases of data production: experiment (unnoticed experimental setup failure or systematic errors), analysis (misinterpretation of information), transformation (from one representation into another), propagation
(erroneous data used for the generation of new data), and staleness (unnoticed changes in data could produce the falsification of other data which depend on it). These problems lead to semantic errors and the resulting information does not represent the real-world facts correctly. Data dependencies inherent to data production process and data usage make genome data predestined for transmitting errors [21].

Most existing works, such as [11], [23] and [20], focus on inaccuracy, lexical errors, redundancy problems and enforcement of integrity constraints, but ignore the functional constraint violations. Moreover, due to the large amount of data in existing databases, a tool for automatically detecting relationships among data can be useful for biological specialists to improve the domain knowledge. The objective is to define an algorithm that automatically infers rules from data: rules can be maintained by means of an incremental approach even if data are updated.

We propose a method to discover tuple constraints and functional dependencies among data by means of association rule mining. Constraints and dependencies show semantic relationships among attributes in a database schema. Their knowledge can be exploited to improve data quality and integration in database design, and to perform query optimization and dimensional reduction. Association rules are a well-known data mining tool. They have been applied to biological data cleaning for detecting outliers and duplicates [17], and to Gene Ontology for finding relationships among terms of the three ontology levels (cellular components, molecular functions and biological processes) [18], [6], but not for finding constraints, dependencies or anomalies.

By means of association rules we detect correlation relationships among attribute values. Then, by analyzing the support and confidence of each rule, (probabilistic) tuple constraints and functional dependencies may be detected. They may both show the presence of erroneous data and highlight novel semantic information. We present experiments on biological databases and we validate our method by verifying its correctness and completeness. Finally, we show how, by means of a further analysis of the obtained results, domain knowledge and data quality may be improved.

2 Application of Constraint Extraction

Recently, new ways to mine patterns and constraints in biological databases have been proposed. In [16] the authors focus on constrained pattern mining on the “transposed” database, thus facing a smaller search space when the number of attributes (genes) is orders of magnitude larger than the number of objects (experiments). They present a theoretical framework for database and constraint transposition, discuss the properties of constraint transposition and look into classical constraints. Our approach does not require the data to be transposed and aims at discovering constraints among attributes instead of constraints associated with patterns.
Along with the ongoing trend to use the XML format in biological databases, some kinds of constraints for XML data have been explored by recent research papers [10][19], such as key, inclusion, inverse and path constraints. Instead functional dependencies other than key dependencies have been little investigated. In [14] the authors address this issue with a subgraph-based approach which captures new kinds of functional dependencies useful in designing XML documents. Furthermore, they analyze various properties such as expressiveness, semantic behaviour and axiomatizations.

Constraints in hierarchically structured data have been discussed in [13]. In particular, they focus on functional and key dependencies and investigate how constraints can be used to check the consistency of data being exchanged among different sources. This leads to the constraint propagation problem, which is addressed in the form of propagating XML keys to relational views by providing two algorithms. The importance of translating constraints in data exchanges is also discussed in [9], whose work proposes a suitable language for this task. The proposed language is able to express a wide variety of database constraints and transformations and its development was motivated by experiences on biomedical databases in informatic support to genome research centers.

In [8] the authors introduce the notion of pseudo-constraints, which are predicates having significantly few violations. This is a concept similar to the quasi functional dependencies, but they define pseudo-constraints on the Entity-Relationship model, whereas we use association rules to define quasi functional dependencies. Furthermore, they focus on cyclic pseudo-constraints and propose an algorithm for extracting this kind of cyclic pattern. On the contrary, our notion of dependency is an implication between sets of elements and is not bound to the structure of the data source used to mine the pattern.

In [7] quasi functional dependencies have been exploited to detect anomalies in the data. On the contrary, in this work the focus is on the extraction of constraints and the anomaly detection is only a further possible application of our method.

3 Background

In this section we provide an overview of the main concepts behind the successful discovery of constraints. We start with a survey of the relational model, which is widely adopted in the database design and whose strongly structured format eases the definition of constraints. Then we focus on a particular kind of constraints: integrity constraints and their subtypes. Eventually we conclude with an introduction to association rules, a well-established data mining technique for inferring unknown correlations from data.

3.1 Relational Model

There are several ways to describe a database at a logic level, e.g., hierarchical, relational, or object oriented. Nowadays the relational model is the most widely
adopted representation for databases. Every concept is represented by a relation, i.e. a table. In the biological domain well known examples of relational databases are cuticleDB\(^1\), RCSB Protein Data Bank\(^2\), Identitag\(^3\), AbMiner\(^4\), PfamRDB\(^5\), and Reactome\(^6\).

Since older databases often lack a relational structure, many tools to parse and load their data into a relational schema have been developed. Indeed, the relational model is more convenient than other data representations. For example the Bio-Warehouse\(^7\) toolkit enables to translate the flat file representation of some databases, such as SwissProt\(^8\), NCBI\(^9\), KEGG\(^10\) and GO\(^11\), into a relational schema.

A relation consists of many tuples (rows), each of which represents an instance of an entity (i.e., a concept), and many columns, which represent the attributes (i.e., properties) of the entity. For example, in a protein database, the protein is an entity and its structure, function and sequence are its attributes. Every row related to a specific protein with all its attribute values is an instance of the protein entity. The table that contains all the proteins with their attributes is a relation.

The relational model is characterized by a structured, fixed format, since data values have to be homogeneous and have to meet several constraints. The schema constraints may be known or unknown in advance. In both cases our analysis is helpful for detecting and investigating instance constraints, which may suggest novel information and detect errors.

### 3.2 Constraints

Constraints are assertions on permissible or consistent database states and specify data properties that should be satisfied by valid instances of the database. They are usually defined in the form of expressions that provide a boolean value, indicating whether or not the constraint holds.

Let us now consider, as an example, a relational database containing information about students of a university. In particular, we consider only a relation Student (StudentID, Name, Age, City, Country, Mail). Each student is described by some attributes, among which there are a unique identifier (StudentID), his/her name (Name), city (City), country (Country) and mail address (Mail).

---

1 http://biophysics.biol.uoa.gr/cuticle
2 http://pdbbeta.rcsb.org
3 http://pbil.univ-lyon1.fr/software/identitag
4 http://discover.nci.nih.gov/abminer
5 http://sanger.ac.uk/pub/databases/Pfam
6 http://www.reactome.org
7 http://brg.ai.sri.com/biowarehouse
8 www.ebi.ac.uk/swissprot
9 www.ncbi.nih.gov
10 www.genome.jp/kegg
11 www.geneontology.org
Constraints may be classified as follows [4].

- **Intra-relational constraints** if they are validated over the tuples of a single relation. Particular cases of such constraints are (in decreasing order of specificity) the following.
  - *Domain constraints* (also known as *value constraints* or *attribute constraints*) apply to the values of a single attribute. With respect to the relation described above, the following domain constraints hold:
    - \((\text{Age} \geq 18) \, \text{AND} \, (\text{Age} \leq 25)\)
    - Mail must contain the ‘@’ symbol.
  - *Tuple constraints* apply to combinations of different attribute values within a single tuple. They must hold separately on each tuple. For example:
    - \(\text{NOT( (City=\text{Rome}) \, \text{AND} \, (\text{Country} \neq \text{Italy}) )}\), which means that if the City is Rome, then the Country must be Italy (i.e., it is not possible that if the city is Rome, the country is not Italy).
  - *Functional dependencies* apply to combination of attributes and must hold for all the tuples of the relation. *Key constraints* are the most important functional dependency. For example:
    - \(\text{StudentID} \rightarrow \text{Name}\), which means that given a value of the StudentID attribute, the value of the Name attribute is univocally determined. If StudentID is also the key of the relation, this means that all the StudentID values must be different, i.e., unique within the considered relation.

Functional dependencies can be seen as a generalization of tuple constraints. If there is a tuple constraint between every value of two attributes, then there is also a functional dependency between the two attributes.

- **Inter-relational constraints** are integrity constraints which are validated over the tuples of different database relations. The most well known are referential integrity constraint, which describe the relationship among attributes in different relations. For example, if a mark relation stores marks for all students, the StudentID attribute may be exploited to establish a relationship between the student and mark relations.

In this work, we focus on intra-relational constraints, in particular on tuple constraints and functional dependencies.

### 3.3 Association Rules

Association rule discovery is an important data mining technique, which is commonly used for local pattern detection in unsupervised learning systems. It shows attribute values that occur together in a given dataset. These combinations of values are useful for finding correlations among sets of items in transactional or relational databases. Association rules describe the co-occurrence of data items (i.e., couples in the form (attribute, value)) in a large amount of collected data [2].
Rules are usually represented in the form: \( body \rightarrow head \ [support, confidence] \). \( Body \) and \( head \) are two arbitrary sets of data items, such that \( body \cap head = \emptyset \). \( Support \) \((s)\) and \( confidence \) \((c)\) are used to measure the quality of an association rule. They are computed as shown in equations (1) and (2).

\[
s = \frac{n_{bh}}{n} \tag{1}
\]

\[
c = \frac{n_{bh}}{n_b} \tag{2}
\]

In Eq. (1), \( n_{bh} \) is the number of data instances that contain both body and head, and \( n \) the cardinality of the relation (i.e., the number of data instances in the relation). In Eq. (2), \( n_b \) is the total number of data instances containing the body [12]. In Eq. (1), \( n_{bh} \) is also called absolute support, while the support \( s \) is a relative value with respect to the total number of tuples.

For example, the rule \(((City=Paris)\ AND\ (Age=30)) \rightarrow (Preferred\ product=Car)\ [0.5\%,\ 60\%]\) means that thirty years old people living in Paris whose preferred product is a car are the 0.5\% of the buyers stored in the database. It means also that the preferred product for 60\% of thirty years old people living in Paris (stored in the buyers database) is a car.

4 Constraint Extraction in Biological Data

In this section we describe how constraints can be extracted from a database using a combination of the previously introduced techniques. Fig. 1 synthesizes the phases of the proposed approach, exploited in [3] for a different application. The method is based on association rule extraction, based on the Apriori algorithm [1]. Any other association rule mining algorithm may be substituted as a building block. Association rules are extracted form the data. Next, tuple constraints and functional dependencies are identified by analyzing the extracted rules.

If the database constraints and dependencies are already known, they can be used as a criterion to evaluate the accuracy of the method. In this case, when a table row does not satisfy a tuple constraint or a functional dependency, its data is not correct.

Constraint knowledge may be exploited to improve data quality and integration in database design, and to perform query optimization and dimensional reduction. We show an application of our method for identifying anomalies with respect to the detected constraints which can be used to improve domain knowledge.

4.1 Quasi Tuple Constraints and Quasi Functional Dependencies

The concepts of support and confidence of a detected rule are used to determine its frequency and its strength. An association rule with confidence equal to 1 means that there is a tuple constraint between the attribute values in the head and the ones
If the confidence of a rule is close to 1, according to a specific threshold, we define the constraint as a *quasi tuple constraint*. It means that in most cases the specific value of the body attribute determines the corresponding head value, but there are few cases in which this relationship is not verified. This is a good hint for possible errors or interesting anomalies in the data.

If all the association rules that involve the same attributes have confidence equal to 1 and the sum of all association rule supports is 1, a functional dependency between the corresponding elements is found. The dependency degree between attributes can be expressed by considering confidence and support in a single expression [5], as reported in Eq. (3).

\[
P = \sum_{i \in A} s_i \cdot c_i
\]  

Eq. (3)

In Eq. (3), \( A \) is the set of all association rules representing tuple constraints on the same attributes, and \( s_i \) and \( c_i \) are the support and the confidence of every such rule. When \( p \) is equal to 1, there is a functional dependency between the considered attributes. Note that if the confidence value is not equal to 1, the \( p \) index will never be equal to 1. When the dependency degree is close to 1, according to a specific threshold discussed later, we define the relationship as a *quasi functional dependency* [5].

For example, we can consider the relation *Animal* (*AnimalID*, *Species*, *Class*, *Reproduction*), which represent biological information about animals in a zoo or in a research center. Each animal is described by some attributes, among which there are a unique identifier (*AnimalID*), its species (*Species*), and class (*Class*) of its tassonomic classification, and the way of reproduction (*Reproduction*), which can be *vivipary* if the animal gives birth to live young or *ovipary* if it lays eggs.
The primary key is \textit{AnimalID}. Thus, we assume that the following functional dependencies hold at design time:

- \textit{AnimalID} $\rightarrow$ \textit{Species},
- \textit{AnimalID} $\rightarrow$ \textit{Class},
- \textit{AnimalID} $\rightarrow$ \textit{Reproduction}.

Furthermore, every rule that involves an attribute value of \textit{AnimalID} is a tuple constraint.

According to the nature and the content of the considered relation, we also expect to find the functional dependency \textit{Class} $\rightarrow$ \textit{Reproduction}, because given a class, we can always know the way of reproduction. Hence, we expect exclusively to find association rules with confidence 100\% between the attributes \textit{Class} and \textit{Reproduction}.

Since in biological databases data dependencies may be incomplete, due to the complexity of the stored data, it is possible that none of the discovered rules have a confidence or dependency degree equal to 1. To address this issue, we extract also association rules with confidence lower than 100\%. Considering the subset of the relation \textit{Animal} shown in Table 1, we can find two classes (\textit{Mammalia} and \textit{Aves}) which correspond to both ways of reproduction. Table 2 reports examples of associations rules found in the database (support and confidence are the values extracted by the mining algorithm applied on the entire dataset whose portion is reported in Table 1).

\begin{table}[h]
\centering
\caption{A portion of the \textit{Animal} table}
\begin{tabular}{llll}
AnimalID & Species & Class & Reproduction \\
1 & \textit{Felis catus} & \textit{Mammalia} & vivipary \\
2 & \textit{Ornithorhynchus} & \textit{Mammalia} & ovipary \\
3 & \textit{Mus musculus castaneus} & \textit{Mammalia} & vivipary \\
& ... & ... & ... \\
100 & \textit{Passer Domesticus} & \textit{Aves} & vivipary \\
101 & \textit{Eurypygga Helias} & \textit{Aves} & ovipary \\
& ... & ... & ... \\
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Examples of association rules found in the \textit{Animal} table}
\begin{tabular}{llll}
Body & Head & Sup & Conf \\
Class = \textit{Mammalia} & Reproduction = vivipary & 45.1\% & 75.2\% \\
Class = \textit{Mammalia} & Reproduction = ovipary & 14.9\% & 24.8\% \\
Class = \textit{Aves} & Reproduction = ovipary & 28.7\% & 99.7\% \\
Class = \textit{Aves} & Reproduction = vivipary & 0.1\% & 0.3\% \\
\end{tabular}
\end{table}
Rules shown in Table 2 allow us to infer two quasi tuple constraints, which have a high confidence (e.g., greater than 70%):

1. (Class = Mammalia) → (Reproduction = vivipary) [with confidence of 75.2%]
2. (Class = Aves) → (Reproduction = ovipary) [with confidence of 99.7%]

The first rule means that the majority of mammalians (75.2%) stored in the database are vivipary, the second one that almost all the birds (99.7%) are ovipary.

To compute the dependency degree between Class and Reproduction we consider all the extracted rules that involve the two attributes (not only those that are reported in Table 2) and we obtain a value of 0.95. Supposing to set a threshold of 0.05, this value is high enough to state that there is a quasi functional dependency between the two attributes, as reported in Table 3.

<table>
<thead>
<tr>
<th>Quasi-functional dependency</th>
<th>Dependency degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class → Reproduction</td>
<td>0.95</td>
</tr>
</tbody>
</table>

4.2 Violations of Constraints and Dependencies

Constraint extraction has the double aim of discovering hidden relationships among data, which may improve domain knowledge and data quality, and investigating anomalies with respect to these relationships. If a quasi tuple constraint or a quasi functional dependency has been detected, it means that there are few cases in which the relationship is not valid. Such cases are anomalies with respect to the frequent ones. These anomalies can be errors or exceptions in the data. Analyzing such anomalies can be useful in different application domains to perform data cleaning or improve the context knowledge.

To better investigate the nature of the detected anomalies, we can analyze the confidence of the rules that involve the attribute values of the quasi tuple constraint or the quasi functional dependency. If this value is very low (compared to the confidence value of the other rules), we can strongly suggest that this is an error, otherwise it is more likely to be a correct exception.

With respect to the Animal relation introduced in the previous section, we investigate the rules that involve Class and Reproduction with a low confidence (i.e., lower than 30%). We find the following rules:

1. (Class = Mammalia) → (Reproduction = ovipary) [with confidence of 24.8%]
2. (Class = Aves) → (Reproduction = vivipary) [with confidence of 0.3%]

Both of them represent interesting cases. The first one is a correct, albeit infrequent, relationship, since there are some mammalians which lay eggs (such as the Ornithorhynchus). The second one is an error, since there is no bird that is not ovipary. There is a mistake in the 100th row because the Passer Domesticus has an ovipary reproduction.
Errors and anomalies may be distinguished by analyzing the confidence value of the rules. The confidence of the second rule (0.3%) is at least one order of magnitude smaller than the average confidence values of the other rules. Hence, we can conclude that this is probably an error.

5 Experiments on Biological Data

It is about thirty years that biological data are generated from a variety of biomedical devices and stored at an increasing rate in public repositories. Thus, data constraints can be unknown and it is difficult to recognize the structure and the relationships among data.

In this section we present experiments showing the application of the constraint extraction to biological databases. Two kinds of experiments are reported. The first aims at validating the constraint extraction method by applying the proposed framework to a database whose constraints and dependencies are already known in advance. The second extracts and analyzes quasi constraints to discover semantic anomalies and inconsistencies due to possible violations.

5.1 Biological Databases

We selected two databases whose structure and dependencies among data were known in order to verify the accuracy of our method. We considered the SCOP (Structural Classification Of Proteins, version 1.71, http://scop.berkeley.edu) and CATH (Class, Architecture, Topology and Homologous superfamily, version 3.0.0, http://www.cathdb.info) databases, which are characterized by a hierarchical structure, similarly to many biological data sources. However, the method is independent of the hierarchical structure, since the Apriori algorithm and our analysis can be applied to different data models ranging from the relational model to XML.

The SCOP database classifies about 30.000 proteins in a hierarchical tree of seven levels. From the root to the leaves they are: class, fold, superfamily, family, protein, species, as shown in Fig. 2. This tree structure is particularly suitable for verifying the tuple constraints and functional dependencies that can be extracted by our tool. In this database tuple constraints are represented by tree edges, while functional dependencies are represented by the tree hierarchy.

For example we expect to find the following association rules:

- \((\text{Superfamily}=\text{alpha helical ferredoxin}) \rightarrow (\text{Fold}=\text{globin-like})\) with confidence of 100%
- \((\text{Superfamily}=\text{globin-like}) \rightarrow (\text{Fold}=\text{globin-like})\) with confidence of 100%
- \(\sum_{\text{Superfamily} \rightarrow \text{Fold}} s_i = 1\)

In this way we can deduce that \((\text{NOT(Superfamily}=\text{alpha helical ferredoxin}) \text{ OR } (\text{Fold}=\text{globin-like}))\) and \((\text{NOT(Superfamily}=\text{globin-like}) \text{ OR } (\text{Fold}=\text{globin-like}))\) are tuple constraints and \(\text{Superfamily} \rightarrow \text{Fold}\) is a functional dependency.
The CATH database is a hierarchical classification of protein domain structures in the Protein Data Bank (http://www.rcsb.org/pdb/). Protein structures are classified using a combination of automated and manual procedures. There are four major levels in this hierarchy: Class, Architecture, Topology and Homologous superfamily, as shown in Fig. 2. Domains within each Homologous superfamily level are subclustered into sequence families using multi-linkage clustering, identifying five family levels (named S, O, L, I, D). Thus, the complete classification hierarchy consists of nine levels (CATHSOLID).

5.2 Extracting Association Rules

The first step in finding association rules is to look for attribute values that appear in the same tuple. Every attribute-value couple is an item. A group of items is
called *itemset*. We are interested in finding *frequent* itemsets, which are itemsets with support higher than a specific threshold.

A well-known technique for frequent itemset extraction is the Apriori algorithm [1]. It relies upon a fundamental property of frequent itemsets, called the *apriori property*: every subset of a frequent itemset must also be a frequent itemset. The algorithm proceeds iteratively, firstly identifying frequent itemsets containing a single item. In subsequent iterations, frequent itemsets with \( n \) items identified in the previous iteration are combined together to obtain itemset with \( n+1 \) items. A single scan of the database after each iteration suffices to determine which generated candidates are frequent itemsets.

Once the largest frequent itemsets are identified, each of them can be subdivided into smaller itemsets to find association rules. For every largest frequent itemset \( s \), all non-empty subsets \( a \) are computed. For every such subset, a rule \( a \rightarrow (s-a) \) is generated and its confidence is computed. If the rule confidence exceeds a specified minimum threshold, the rule is included in the result set.

### 5.3 Quasi Tuple Constraints

For tuple constraints, the minimum confidence must be equal to 1. The minimum support value allows us to concentrate on the most frequent constraints. If the support is set to the inverse of the total number of records, then all the constraints are considered (this support corresponds to rules contained in a single data entry). We defined the *tolerance* as the complement of the confidence value (e.g., if confidence is 0.95, then tolerance is 0.05).

By setting a tolerance value of 0.0, our method detected all and only the tuple constraints contained in the SCOP and in the CATH databases. Furthermore, we performed experiments with tolerance values ranging from 0.001 to 0.100 (corresponding to confidence values from 0.999 to 0.900), while the support value has been set to the inverse of the total number of records. Results show a consistent number of quasi tuple constraints whose presence increases as the tolerance value increases (see Fig. 3).

By investigating tuples that violate such constraints, anomalies can be identified. Such analysis allows experts interested in the domain to focus their study on a small set of data in order to highlight biological exceptions or inconsistencies in the data. As the results in Fig. 3 show, the lower the tolerance is, the stronger the tuple constraint between the two values is and the fewer rules are extracted. Tuning the tolerance, a domain expert is allowed to concentrate on the desired subset of most probable anomalies present in the data. As an example, on the SCOP database a tolerance value of 0.05 extracts less than 200 rules to be further investigated by domain experts, among the overall 30000 protein entries.

To distinguish between biological exceptions or inconsistencies, a further analysis of the discovered anomalies can be performed by means of three approaches:
1. querying different related databases (e.g., GO, PDB, Swiss Prot, CATH, SCOP),
2. searching relevant information in literature,
3. comparing the obtained results with the schema constraints of the examined database. If the database constraints are known, errors can be distinguished from exceptions automatically. In particular, if the attribute values of a tuple do not satisfy the constraints, we can conclude that it is an error, otherwise it is a biological exception.

For example, in the SCOP database, we can consider the following quasi tuple constraint.

\[(\text{fold}= \text{Ntn hydrolase-like}) \rightarrow (\text{superfamily}= \text{hypothetical protein MTH1020}) [c=0.99771] \]

The extracted anomaly rule with respect to this constraint is:

\[(\text{fold}= \text{Ntn hydrolase-like}) \rightarrow (\text{superfamily}= \text{N-terminal nucleophile aminohydrolases}) [c=0.00229] \]

Given such anomaly, the three above-described approaches can be applied to distinguish between an error and a biological exception.

1. In the CATH database, the same protein is classified in the same \textit{alpha+beta} class as in SCOP and it has a 4-layer sandwich architecture (and Glutamine topology) that consists of the 4 layers \textit{alpha/beta/beta/alpha} that are the same for the N-terminal nucleophile aminohydrolases fold found in the SCOP classification.
2. In literature evidences can be found that the crystal structure of MTH1020 protein reveals an Ntn-hydrolase fold [22].

3. The results can be compared to the SCOP constraints (which are known) and confirm the correctness of this relationship (i.e., it is not an error).

Finally, applying the same procedure also to the CATH database, we discovered the anomalies reported in Fig. 3. Among these, we noticed the one for the hypothetical protein MTH1020. Thus, the method applied independently on two different databases reveals the same anomaly. This result confirms the consistency of the proposed method for anomaly discovery.

5.4 Quasi Functional Dependencies

For the purpose of identifying functional dependencies, the dependency degree between couples of attributes must be equal to 1 (see Eq. (3)). We verified that the proposed method is sound and complete by means of two experiments, which show that it correctly identifies all and only the functional dependencies contained in the database.

In the first experiment we executed the algorithm on the SCOP database with the lowest support value. Table 4 shows the detected functional dependencies. The first two columns contain the attributes which are functionally dependent (their names are explained in the following example). The third column represents the dependency degree obtained by applying Eq. (3) on the original data. Only the couples of attributes with dependency degree of 1 are shown. Since the SCOP database constraints are known, we exploited its structural knowledge to prove the correctness of our method. Our method detects all and only the functional dependencies in the SCOP database. Hence, it is sound (i.e., it only detects correct functional dependencies) and complete (i.e., all known functional dependencies are detected).

In the second experiment we performed a simulation of fault injection tests. In this way, we demonstrate that also in presence of errors, our method is sound and complete, because it still detects all and only the non-faulted dependencies. We changed random values at different levels of the hierarchy, by substituting the actual value with one randomly taken from another branch of the tree. This kind of misclassification is rather subtle to identify, since the value is acceptable (i.e., it is valid and it is not misspelled), but it assigns a particular protein to the wrong class for one or more levels of the hierarchy. The fourth column of Table 4 represents the dependency degree after the fault injection simulation, where a random fault has been injected for some levels of the classification hierarchy. All the affected attributes, represented in bold in the first two columns, report dependencies whose value falls below 1. Thus, a quasi functional dependency analysis can be performed to detect all the injected faults.
Table 4. Functional dependencies among attributes in the SCOP database

<table>
<thead>
<tr>
<th>attributes</th>
<th>original</th>
<th>faulted</th>
</tr>
</thead>
<tbody>
<tr>
<td>cf</td>
<td>cl</td>
<td>1.000000</td>
</tr>
<tr>
<td>dm</td>
<td>cf</td>
<td>1.000000</td>
</tr>
<tr>
<td>dm</td>
<td>cl</td>
<td>1.000000</td>
</tr>
<tr>
<td>dm</td>
<td>fa</td>
<td>1.000000</td>
</tr>
<tr>
<td>dm</td>
<td>sf</td>
<td>1.000000</td>
</tr>
<tr>
<td>fa</td>
<td>cf</td>
<td>1.000000</td>
</tr>
<tr>
<td>fa</td>
<td>cl</td>
<td>1.000000</td>
</tr>
<tr>
<td>fa</td>
<td>sf</td>
<td>1.000000</td>
</tr>
<tr>
<td>px</td>
<td>cf</td>
<td>1.000000</td>
</tr>
<tr>
<td>px</td>
<td>cl</td>
<td>1.000000</td>
</tr>
<tr>
<td>px</td>
<td>dm</td>
<td>1.000000</td>
</tr>
<tr>
<td>px</td>
<td>fa</td>
<td>1.000000</td>
</tr>
<tr>
<td>px</td>
<td>sf</td>
<td>1.000000</td>
</tr>
<tr>
<td>px</td>
<td>sp</td>
<td>1.000000</td>
</tr>
<tr>
<td>sf</td>
<td>cf</td>
<td>1.000000</td>
</tr>
<tr>
<td>sf</td>
<td>cl</td>
<td>1.000000</td>
</tr>
<tr>
<td>sp</td>
<td>cf</td>
<td>1.000000</td>
</tr>
<tr>
<td>sp</td>
<td>cl</td>
<td>1.000000</td>
</tr>
<tr>
<td>sp</td>
<td>dm</td>
<td>1.000000</td>
</tr>
<tr>
<td>sp</td>
<td>fa</td>
<td>1.000000</td>
</tr>
<tr>
<td>sp</td>
<td>sf</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

For example, in the SCOP database, the record of the protein chain\(px=100068\) consists of the following attributes:

- class\(cl=46456\) (alpha proteins),
- fold\(cf=46457\) (globin-like),
- super family\(sf=46458\) (globin-like again),
- family\(fa=46459\) (truncated hemoglobin),
- protein domain\(dm=46460\) (hemoglobin),
- species\(sp=46461\) (ciliate).

One of the injected faults is the misclassification of the record by assigning the value\(fa=46463\) (globins) instead of\(fa=46459\) (truncated hemoglobin) to the family attribute. The faulty value\(fa=46463\) (globins) actually exists in the database and is the correct family attribute of many other proteins.

In this example, the rule protein domain\(dm=46640\) → family\(fa=46459\) occurs in 26 records with confidence=0.963. We classified records that do not respect such rule as anomalies. Thus, the records with protein domain\(dm=46640\) and
family \( fa \neq 46459 \) are selected as either candidate inconsistency or information for further investigation by biological experts.

6 Conclusions

In this chapter we present a framework for the application of data mining tools to constraint extraction in the biological domain. We focus on tuple constraint and functional dependency detection in representative biological databases by means of association rule mining. By analyzing association rules we can deduce not only constraints and dependencies, which provide structural knowledge on a dataset and may be useful to perform query optimization or dimensional reduction, but also anomalies in data, which could be errors or interesting exceptions to be highlighted to domain experts.

We have applied our analysis to the SCOP and CATH databases. We plan to extend our approach to different database models, such as XML or collections of relational tables, and to integrate automatic distributed inquiry about the detected anomalies on other databases, in order to help domain experts to distinguish biological anomalies from errors. Further developments of this work include the application of our method to heterogeneous data sources, to derive schema information that may be exploited during data integration.

Acknowledgements

We would like to thank Paolo Garza for his help in association rule extraction and for many stimulating discussions.

References


Summary. One of the most important genomic tasks is the identification of promoters and splice-junction zone, which are essential on deciding whether there is a gene or not in a genome sequence. This problem could be seen as a classification problem, therefore the use of computational algorithms for both, pattern recognition and classification are a natural option to face it. In this chapter we develop a pattern classifier algorithm that works notably with bioinformatics databases. The associative memories model on which the classifier is based is the Alpha-Beta model. In order to achieve a good classification performance it was necessary to develop a new heteroassociative memories algorithm that let us recall the complete fundamental set. The heteroassociative memories property of recalling all the fundamental patterns is not so common; actually, no previous model of heteroassociative memory can guarantee this property. Thus, creating such a model is an important contribution. In addition, an heteroassociative Alpha-Beta multimemory is created, as a fundamental base for the proposed classifier.

1 Introduction

In the later decades, very important scientific advances in the field of molecular biology have been achieved. Thanks to the enormous amounts of information derived from these advances, there has arisen a need to process such information in a faster way and just as effectively, or more, than by an expert. This gives birth to a new branch of science, known as Bioinformatics: a multidisciplinary field which combines, among others, two important fields of science, molecular biology and computer sciences [1].

Among the first and foremost problems boarded by Bioinformatics are: the development of databases, protein sequence alignment, DNA string sequencing, protein structure prediction, protein structure classification, promoter identification, splice-junction zone localization, and phylogenetic relationships determining [2, 3].

The solutions to some of these problems are based on the search and localization of patterns in certain sequences, in order to classify them. As an example, we can mention promoter identification and splice-junction zone
localization, which are the two tasks in which the algorithms proposed in this work are tested.

The topic of associative memories has been an active field of scientific research for some decades, attracting the attention in some research areas for the great power they offer, despite the simplicity of their algorithms. The most important characteristic and, at the same time, fundamental purpose of an associative memory, is to correctly recall whole output patterns from input patterns, with the possibility of having the latter altered, either by an additive, subtractive, or mixed alteration [4]-[6].

Associative memories, and specifically alpha-beta associative memories, are a powerful computational tool in pattern recognition due to the simplicity of their algorithms, their strong mathematical foundation, and the high efficacy shown by them in pattern recalling and classification.

The rest of this chapter is organized as follows: in section 2 some background notions are presented, regarding bioinformatics and associative memories. Section 3 is dedicated to the proposed solution to the problems boarded here, while in section 4 the experimental results are shown. Finally, in section 5 conclusions and future work are described.

## 2 Background Notions

In this section, some basic notions and concepts on bioinformatics and associative memories are introduced, in order to better understand the problems presented in this chapter and how we propose to solve them.

### 2.1 Bioinformatics

In later decades, the development of genomic technology has caused the generation of a great amount of data, which in turn gives rise to a greater and greater demand of computational resources for its storing, organization, recovering, and, above all, visualization and analysis [7].

According to [8], the most recent goal of bioinformatics is to allow the discovery of new biological elements which help in the creation of a global perspective from which uniform biological principles can be derived.

#### 2.1.1 Basic Concepts

Deoxyribonucleic acid (DNA) and proteins are biological macromolecules made up of long chains of chemical components. On one hand, DNA is made up of nucleotides, of which there are four: adenine (A), cytosine (C), guanine (G), and thymine (T), denoted by their initials. Also, DNA plays a fundamental role in different biochemical processes of living organisms, such as protein synthesis and hereditary information transmission from parents to children [9].
Inside DNA chains there are some nucleotide strings known as genes, which are sequences of nucleotide, also called exons. These exons are of particular interest, since they have a particular correspondence with a certain protein: they code for this particular protein. However, the exons are located in DNA chains in a disorderly fashion, making it very difficult to take a portion of the chain and say that it codes for a certain protein, as is. Between exons there are sequences which do not code for a protein, called introns; the sequences that do not code for proteins and are between genes are known as intergenic regions [7], [10], [11].

On the other hand, proteins are polypeptides formed inside cells as sequences of 20 different aminoacids [12], which are denoted by 20 different letters. Each of these 20 aminoacids is coded by one or more codons [9]. The chemical properties differentiating the 20 aminoacids make them group together to conform proteins with certain tridimensional structures, defining the specific functions of a cell [11].

2.1.2 Main Problems in Bioinformatics
The different problems addressed by Bioinformatics can be classified into three categories: genomic tasks, proteomic tasks, and gene expression tasks. The first refers to the study of various genomes when taken as entities with similar contents, while the second refers to the study of all the proteins which arise from a genome [13]. The last one is related to studying the relationships between a nucleotide string and the proteins generated by that string. In this chapter we are interested on the genomic task, particularly on both promoter and splice-junction identification. Promoters are regions located immediately before each gene, indicating that what follows is a gene; they also regulate the beginning of transcription [14]. A splice-junction zone is where an intron becomes an exon an viceversa, this is important in order to identified which segments of the sequence code for a protein [14].

2.1.3 Computational Tools Applied to Bioinformatics
One of the most used computational methods in Bioinformatics are artificial neural networks. These are a set of models which emulate biological neural networks. Some of the tasks in which artificial neural networks are most employed are classification and function approximation problems. However, it is noteworthy to mention that, even though neural networks and associative memories are equivalent models, as shown by Hopfield [15], associative memories have not been much used in Bioinformatics. Results obtained by feed-forward back-propagation neural networks have been compared to those shown by BLAST, one of the most widely used tools in Bioinformatics. In these comparisons, the neural network trained with the back-propagation algorithm presented a better processing speed, but not as effective results [16]-[20].
On the other hand, the work of Lukashin et al. [17] is one of the earliest researches done in promoters localization in DNA sequences by using artificial neural networks. The process therein mentioned uses a small subset (around 10%), taken randomly from the whole set of sequences of promoters, in order to train the network. The learning capacity is tested by presenting the whole set of sequences, obtaining an effectiveness of 94% to 99%.

With respect to gene localization, quantitative analysis of similarity between tRNA gene sequences have been done, mainly for the search of evolutive relationships. New sequences have been introduced to this neural network and these are known as tRNA genes [18].

Another example related to promoters localization is a multi-layer feedforward neural network whose architecture is trained in order to predict whether a nucleotide sequence is a bacterial promoter sequence or not [19].

### 2.2 Alpha-Beta Associative Memories

Basic concepts about associative memories were established three decades ago in [4], [5], [21], nonetheless here we use the concepts, results and notation introduced in the Yáñez-Márquez’s Ph.D. Thesis [22]. An associative memory $M$ is a system that relates input patterns and outputs patterns, during the operation, as follows: $x \rightarrow M \rightarrow y$ with $x$ and $y$ the input and output pattern vectors, respectively. Each input vector forms an association with a corresponding output vector. For $k$ integer and positive, the corresponding association will be denoted as: $(x^k, y^k)$.

Associative memory $M$ is represented by a matrix whose $ij$-th component is $m_{ij}$. Memory $M$ is generated from an a priori finite set of known associations, known as the fundamental set of associations. If $\mu$ is an index, the fundamental set is represented as:

$$\{ (x^\mu, y^\mu) \mid \mu = 1, 2, \ldots, p \}$$

with $p$ the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns.

If it holds that $x^\mu = y^\mu, \forall \mu \in \{ 1, 2, \ldots, p \}$, $M$ is autoassociative, otherwise it is heteroassociative; in this latter case it is possible to establish that $\exists \mu \in \{1, 2, \ldots, p\}$ for which $x^\mu \neq y^\mu$. A distorted version of a pattern $x^k$ to be recalled will be denoted as $\tilde{x}^k$. If when feeding a distorted version of $x^\varpi$ with $\varpi = \{1, 2, \ldots, p\}$ to an associative memory $M$, it happens that the output corresponds exactly to the associated pattern $y^\varpi$, we say that recall is correct.

Among the variety of associative memory models described in the scientific literature, there are two models that, because of their relevance, it is important to emphasize: morphological associative memories which were introduced by Ritter et al. [6], and alpha-beta associative memories, which were introduced in [22]-[24]. Because of their excellent characteristics, which allow them to be superior in many aspects to other models for associative
memories [6], morphological associative memories served as starter point for the creation and development of the alpha-beta associative memories.

2.2.1 The $\alpha$ and $\beta$ Operators
The heart of the mathematical tools used in the alpha-beta model, are two binary operators designed specifically for these memories. These operators are defined in [22] as follows: First, we define the sets $A = \{0, 1\}$ and $B = \{0, 1, 2\}$, then the operators $\alpha$ and $\beta$ are defined in tabular form:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$\alpha(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$\beta(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The sets $A$ and $B$, the $\alpha$ and $\beta$ operators, along with the usual $\wedge$ (minimum) and $\vee$ (maximum) operators, form the algebraic system $(A, B, \alpha, \beta, \wedge, \vee)$ which is the mathematical basis for the alpha-beta associative memories.

The $ij$-th entry of the matrix $y \boxtimes x^t$ is: $[y \boxtimes x^t]_{ij} = \alpha(y_i, x_j)$. If we consider the fundamental set of patterns: $\{(x^\mu, y^\mu) | \mu = 1, 2, \ldots, p\}$ then, the $ij$-th entry of the matrix $y^\mu \boxtimes (x^\mu)^t$ is: $[y^\mu \boxtimes (x^\mu)^t]_{ij} = \alpha(y^\mu_i, x^\mu_j)$

2.2.2 The Learning and Recalling Phases
Given that there are two kinds of alpha-beta associative memories, $\vee$ and $\wedge$, if we consider that each of these kinds is able to operate in two different modes, heteroassociative and autoassociative, we have four different available choices.

Due to expediency, we will only discuss here the alpha-beta autoassociative memories of kind $\vee$. Then, the input and output patterns have the same dimension $n$, and the memory is a square matrix $V = [v_{ij}]_{n \times n}$. Also, the fundamental set takes the form $\{(x^\mu, x^\mu) | \mu = 1, 2, \ldots, p\}$.

LEARNING PHASE
For each $\mu = 1, 2, \ldots, p$ , and from each couple $(x^\mu, x^\mu)$ build the matrix $[x^\mu \boxtimes (x^\mu)^t]_{n \times n}$. Then apply the binary $\vee$ operator to the matrices obtained to get $V$ as follows: $V = \bigvee_{\mu=1}^{p} [x^\mu \boxtimes (x^\mu)^t]$. 
The $ij$-th entry is given as: $v_{ij} = \bigvee_{\mu=1}^{p} \alpha \left( x^\mu_i, x^\mu_j \right)$. It is obvious that $v_{ij} \in B, \forall i \in \{1, 2, \ldots, n\}, \forall j \in \{1, 2, \ldots, n\}.$

**RECALLING PHASE**

A pattern $x^\omega$, that could be or not a fundamental pattern, with $\omega \in \{1, 2, \ldots, p\}$ is presented to the alpha-beta autoassociative memory of kind $\vee$ and the following operation is done: $V \Delta_{\beta} x^\omega$. The result is a column vector of dimension $n$, with the $i$-th component given as:

$$(V \Delta_{\beta} x^\omega)_i = \bigwedge_{j=1}^{n} \beta \left( v_{ij}, x^\omega_j \right)$$

*Remark 1.* Note that alpha-beta autoassociative memories of kind $\land$ are built by duality from alpha-beta autoassociative memories of kind $\vee$. In order to do so, the following changes are made:

- Use operator $\lor$ instead of operator $\land$.
- Use operator $\land$ instead of operator $\lor$.
- Use operator $\nabla_{\beta}$ instead of operator $\Delta_{\beta}$; operator $\nabla_{\beta}$ is defined as:

$$(\Lambda \nabla_{\beta} x^\omega)_i = \bigvee_{j=1}^{n} \beta(\lambda_{ij}, x^\omega_j)$$

The greatest limitation of alpha-beta associative memories is perhaps that they only work with binary data, being unable to manipulate integer or real numbers. However, there have been numerous efforts to extend the original model $[25]$-$[28]$. With these advances, alpha-beta memories can be applied to binary and integer pattern recognition, as well as gray-level and color image processing, and all the different fields these imply.

### 3 Proposed Solution

This section is dedicated to the main contribution presented in this chapter, the Alpha-Beta MultiMemories Classifier, used to broach the subjects of promoter identification and splice-junction zone localization. First, an extension to the heteroassociative kind of the original alpha-beta associative memory model is introduced, followed by a new model of associative memory based in the former extension: the alpha-beta heteroassociative multimemories. Finally, the concepts presented in the model of multimememories are applied to build the proposed classifier.

#### 3.1 New Alpha-Beta Heteroassociative Memory

The alpha-beta associative memories model guarantee, through theorems 4.28 and 4.32 from $[22]$, the complete recall of the fundamental set, but only for
the autoassociative kind and not for the heteroassociative one. Therefore, in this subsection we propose a new heteroassociative algorithm, based on the original model, to ensure this correct recall.

In order to guarantee the complete recall of the fundamental set it is necessary to redefine the learning and recalling phase of the original model, building new ones. The lemmas and theorems that prove the complete recall in this new alpha-beta heteroassociative memories are discussed in [29],[30].

**Definition 1.** Let $V$ be an alpha-beta heteroassociative memory type $\text{Max}$ and $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ its fundamental set with $x^\mu \in A^n$ and $y^\mu \in A^p$, $A = \{0, 1\}$, $B = \{0, 1, 2\}$, $n \in Z^+$. The sum of the components of the $i$-th row of $V$ with value equal to one is given as:

$$s_i = \sum_{j=1}^{n} T_j$$

where $T \in B^n$ and its components are defined as: $T_i = \begin{cases} 1 & \leftrightarrow \nu_{ij} = 1 \\ 0 & \leftrightarrow \nu_{ij} \neq 1 \end{cases}$

$\forall j \in \{1, 2, ..., n\}$ and the $s_i$ components conform the max sum vector with $s \in Z^p$.

**Definition 2.** Let $x^\omega \in A^n$ with $\omega, n \in Z^+$, $A = \{0, 1\}$; each component of the negated vector of $x^\omega$, denoted by $\tilde{x}^\omega$, is given as:

$$\tilde{x}^\omega_i = \begin{cases} 1 & x^\omega_i = 0 \\ 0 & x^\omega_i = 1 \end{cases}$$

**Definition 3.** Let $\Lambda$ be an alpha-beta heteroassociative memory type $\text{Min}$ and $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ its fundamental set with $x^\mu \in A^n$ and $y^\mu \in A^p$, $A = \{0, 1\}$, $B = \{0, 1, 2\}$, $n \in Z^+$. The sum of the components with value equal to zero of the $i$-th row of $\Lambda$ is given as:

$$r_i = \sum_{j=1}^{n} T_j$$

where $T \in B^n$ and its components are defined as: $T_i = \begin{cases} 1 & \leftrightarrow \lambda_{ij} = 0 \\ 0 & \leftrightarrow \lambda_{ij} \neq 0 \end{cases}$

$\forall j \in \{1, 2, ..., n\}$ and the $r_i$ components conform the min sum vector with $r \in Z^p$.

**Definition 4.** Let $y^\omega \in A^n$ be a binary vector with $\omega, n \in Z^+$, $A = \{0, 1\}$. Its $k$-th component is assigned as follows: $y^\omega_k = 1$, and $y^\omega_j = 0$ for $j = 1, 2, \ldots, k-1, k+1, \ldots, m$ where $k \in \{1, 2, \ldots, m\}$ and $k = \mu$. This is known as one-hot vector.

**Definition 5.** Let $y^\omega \in A^n$ be a binary vector with $\omega, n \in Z^+$, $A = \{0, 1\}$. Its $k$-th component is assigned as follow: $y^\omega_k = 0$, and $y^\omega_j = 1$ for $j = 1, 2, \ldots, k-1, k+1, \ldots, m$ where $k \in \{1, 2, \ldots, m\}$ and $k = \mu$. This is known as zero-hot vector.
3.1.1 New Alpha-Beta Heteroassociative Memory Type Max

LEARNING PHASE
Let $A = \{0, 1\}, n, p \in \mathbb{Z}^+, \mu \in \{1, 2, \ldots, p\}, i \in \{1, 2, \ldots, p\}$ and $j \in \{1, 2, \ldots, n\}$ and let $x \in A^n$ and $y \in A^p$ be an input and output vectors, respectively. The corresponding fundamental set is denoted by $\{(x^\mu, y^\mu) | \mu = 1, 2, \ldots, p\}$.

The fundamental set must be built according to the following rules. First, all the $y$ vectors are built with the one-hot codification. Second, to each $y^\mu$ vector corresponds one and only one $x^\mu$ vector, this is, there is only one couple $(x^\mu, y^\mu)$ in the fundamental set.

**STEP 1:**
For each $\mu \in \{1, 2, \ldots, p\}$, from the couple $(x^\mu, y^\mu)$ build the matrix:

$$[y^\mu \otimes (x^\mu)^t]_{m \times n}$$

**STEP 2:**
Apply the binary $\lor$ operator to the matrices obtained in step 1 to get the new alpha-beta heteroassociative memory type max $V$ as follow:

$$V = \bigvee_{\mu=1}^{p} [y^\mu \otimes (x^\mu)^t]$$

with the $ij$–th component given by:

$$v_{ij} = \bigvee_{\mu=1}^{p} \alpha(y^\mu_i, x^\mu_j)$$

RECALLING PHASE

**STEP 1:**
A pattern $x^\omega$ is presented to $V$, the $\Delta_\beta$ operation is done and the resulting vector is assigned to a vector called $z^\omega$: $z^\omega = V \Delta_\beta x^\omega$

The $i$–th component of the resulting column vector is:

$$z^\omega_i = \bigwedge_{j=1}^{n} \beta(v_{ij}, x^\omega_j)$$

**STEP 2:**
It is necessary to build the *max sum vector* $s$ according to the definition therefore the corresponding $y^\omega$ is given as:

$$y^\omega_i = \begin{cases} 1 & \text{if } s_i = \bigvee_{k \in \theta} s_k \text{ AND } z^\omega_i = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\theta = \{i | z^\omega_i = 1\}$.

**Example 1.** Let $x^1, x^2, x^3, x^4$ be the input patterns

$$x^1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, x^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, x^3 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, x^4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
and the corresponding output vectors

\[
y^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad y^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad y^3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad y^4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
\]

the output vectors are built with the one-hot codification, and to each output pattern corresponds one and only one input pattern, therefore the fundamental set is expressed as follow:

\[
\{(x^1, y^1), (x^2, y^2), (x^3, y^3), (x^4, y^4)\}
\]

Once we have the fundamental set, the learning phase of the new heteroassociative memories type Max is applied:

**STEP 1:**

\[
\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \bigtriangledown (10110) = \begin{pmatrix} 12112 \\ 01001 \\ 01001 \\ 01001 \end{pmatrix}
\]

\[
\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \bigtriangledown (01000) = \begin{pmatrix} 10111 \\ 21222 \\ 10111 \\ 10111 \end{pmatrix}
\]

\[
\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \bigtriangledown (10010) = \begin{pmatrix} 01101 \\ 01101 \\ 12212 \\ 01101 \end{pmatrix}
\]

\[
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \bigtriangledown (11011) = \begin{pmatrix} 00100 \\ 00100 \\ 00100 \\ 11211 \end{pmatrix}
\]

**STEP 2:**

The binary operator max $\bigvee$ is applied to the matrix obtained before to build the matrix $V$

\[
V = \begin{pmatrix} 12112 \\ 21222 \\ 12212 \\ 11211 \end{pmatrix}
\]

Once we have the $V$ matrix, to recall $x^\varpi$ with $\varpi \in \{1, 2, 3, ..., p\}$, particularly $x^4$, $x^4$ is presented to $V$: 
\[ V\Delta_\beta x^4 = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 2 & 1 \end{pmatrix} \Delta_\beta \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \]

the resulting vector is not an output pattern from the fundamental set, in other words, it is an ambiguous pattern. According to step 2 of the recall phase of the new heteroassociative memory type \textit{Max}, the resulting vector is known as \( z^4 \), then the \textit{max sum vector} \( s \) must be built:

\[ z^4 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, s = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 4 \end{pmatrix} \]

after that, according to the expression of the recall phase the corresponding output pattern is:

\[ y^4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \]

since the minimum value of \( s_j \) where \( z^4_j = 1 \forall j \in \{1, 2, 3, 4\} \) is 4.

\textit{Remark 2.} Note that alpha-beta heteroassociative memories of kind \( \wedge \) are built by duality from alpha-beta heteroassociative memories of kind \( \lor \). In order to do so, the following changes are made:

- Use operator \( \lor \) instead of operator \( \land \).
- Use operator \( \land \) instead of operator \( \lor \).
- Use operator \( \nabla_\beta \) instead of operator \( \Delta_\beta \); operator \( \nabla_\beta \) is defined as:

\[ (\Lambda \nabla_\beta x^\omega)_i = \bigvee_{j=1}^n \beta(\lambda_{ij}, x^\omega_j) \]

- Use zero-hot vectors instead of one-hot vectors when building the fundamental set.
- Use min sum vectors instead of max sum vectors to obtain the output vector according to:

\[ y^\omega_i = \begin{cases} 0 & \text{if } r_i = \bigwedge_{k \in \theta} r_k AND z^\omega_i = 0 \\ 1 & \text{otherwise} \end{cases} \]

where \( \theta = \{i | z^\omega_i = 0\} \).
3.2 Alpha-Beta Heteroassociative Multimemories

Former associative memory models, particularly the alpha-beta model, used to keep a one-to-one correspondence between the fundamental set and the associative memory. That is, for each fundamental set there is one associative memory. To build many associative memories, which behave as one, from one fundamental set, could be helpful in both the treatment of mixed alterations and the development of new applications.

As presented in [29], it is possible to divide $\mathbf{x}^\omega$ in $q$ equal partitions if $\frac{n}{q} \in \mathbb{Z}^+$. 

**Definition 6.** Let $\mathbf{x}^\mu \in A^n$ be a column vector, with $\mu, n \in \mathbb{Z}^+$, $A = \{0, 1\}$, and $q$ be the number of partitions in which $\mathbf{x}^\mu$ will be divided. The vector partition operator, denoted by $\rho$, defined as the set of $\frac{n}{q}$-dimensional column vectors and it is denoted by:

$$\rho(\mathbf{x}^\mu, q) = \{\mathbf{x}^{\mu_1}, \mathbf{x}^{\mu_2}, ..., \mathbf{x}^{\mu_q}\}$$

such that $\mathbf{x}^{\mu_l} \in A^{\frac{n}{q}}$ for $\forall l \in \{1, 2, ..., q\}$, $\forall i \in \left\{1, 2, ..., \frac{n}{q}\right\}$ and $\forall j \in \{1, 2, ..., n\}$ is expressed as:

$$\mathbf{x}^{\mu_l}_{i} = \mathbf{x}^\mu_{j} \quad \text{such as} 
\quad j = \left( (l - 1) \ast \frac{n}{q} \right) + i$$

3.2.1 Alpha-Beta Heteroassociative Multimemories of Type Max

**LEARNING PHASE**

Let $A = \{0, 1\}, n, p \in \mathbb{Z}^+, \mu \in \{1, 2, ..., p\}, i \in \{1, 2, ..., p\}, j \in \{1, 2, ..., n\}$ and let $\mathbf{x} \in A^n$ and $\mathbf{y} \in A^p$ be input and output vectors, respectively. The corresponding fundamental set is denoted by $\{(\mathbf{x}^{\mu}, \mathbf{y}^{\mu}) \mid \mu = 1, 2, ..., p\}$. The fundamental set must be built according to the following rules: First, the $\mathbf{y}$ vectors are built with the *one-hot* codification. Second, to each $\mathbf{y}^{\mu}$ vector correspond one and only one $\mathbf{x}^{\mu}$ vector, this is, there is only one couple $(\mathbf{x}^{\mu}, \mathbf{y}^{\mu})$ in the fundamental set.

For every $\mu \in \{1, 2, ..., p\}$, from the couple $(\mathbf{x}^{\mu}, \mathbf{y}^{\mu})$ the vector partition operator is applied to each $\mathbf{x}^{\mu}$, then the new fundamental set is expressed by:

$$\{(\mathbf{x}^{\mu_1}, \mathbf{y}^{\mu}), (\mathbf{x}^{\mu_2}, \mathbf{y}^{\mu}), ..., (\mathbf{x}^{\mu_q}, \mathbf{y}^{\mu}) \mid \mu = 1, 2, ..., p, \ l = 1, 2, ..., q\}$$

Now, from the couple $(\mathbf{x}^{\mu_l}, \mathbf{y}^{\mu})$ the $q$ matrices are built as follow: $[\mathbf{y}^{\mu} \boxtimes (\mathbf{x}^{\mu_l})^t]_{m \times (\frac{n}{q})}$. Then apply the binary $\lor$ operator to the corresponding matrices obtained to get $\mathbf{V}^l$ as follow: $\mathbf{V}^l = \bigvee_{\mu=1}^{p} \left[\mathbf{y}^\mu \boxtimes (\mathbf{x}^{\mu_l})^t\right]_{m \times (\frac{n}{q})}$. The $ij$–th component of the $l$ matrix is given by:

$$v^l_{ij} = \bigvee_{\mu=1}^{p} \alpha(y^\mu_i, x^{\mu_l}_j)$$
RECALLING PHASE

STEP 1:
A pattern $x^\omega$, that could be or not from the fundamental set, is presented to each $V^l$, $l = 1, 2, \ldots, q$. First, we have to apply the vector partition operator to $x^\omega$:

$$\rho(x^\omega, q) = \{x^\omega_1, x^\omega_2, \ldots, x^\omega_q\}$$

then for each $V^l$ matrix and $x^\omega_l$ partition with $l = \{1, 2, \ldots, q\}$, the $\Delta_\beta$ operation is done and the resulting vector is assigned to a vector called $z^\omega_l$:

$$z^\omega_l = V^l \Delta_\beta x^\omega_l.$$ 

The $i$-th component of the resulting vector is given as:

$$z^\omega_l_i = \bigwedge_{j=1}^n \beta(V^l_{ij}, x^\omega_l_j).$$

STEP 2:
It is necessary to build the max sum vector $s$ according to the definition in order to obtain the corresponding $z_{\text{int}}^\omega_l$ given as:

$$z_{\text{int}}^\omega_l_i = \begin{cases} 
1 & \text{if } s_i = \bigvee_{k \in \theta} s_k \land z_{\text{int}}^\omega_l_i = 1 \\
0 & \text{otherwise}
\end{cases}$$

where $\theta = \{i|z_{\text{int}}^\omega_l_i = 1\}$.

STEP 3:
An intermediate vector $I$ is created. This vector will contain the sum of the $i$-th components of the $z_{\text{int}}^\omega_l$ vectors:

$$I^\omega_i = \sum_{l=0}^q z_{\text{int}}^\omega_l_i$$

then the corresponding $y^\omega$ vector is obtained by:

$$y^\omega_i = \begin{cases} 
1 & I^\omega_i = \bigvee_{k=1}^p I^\omega_k \\
0 & \text{otherwise}
\end{cases}$$

(2)

3.2.2 Alpha-Beta Heteroassociative Multimemories of Type Min

LEARNING PHASE

Let $A = \{0, 1\}, n, p \in \mathbb{Z}^+, \mu \in \{1, 2, \ldots, p\}, i \in \{1, 2, \ldots, p\}$ and $j \in \{1, 2, \ldots, n\}$ and let be $x \in \mathbb{A}^n$ and $y \in \mathbb{A}^p$ input and output vectors, respectively. The corresponding fundamental set is denoted by $\{(x^\mu, y^\mu) | \mu = 1, 2, \ldots, p\}$.

The fundamental set must be built according to the following rules. First, the $y$ vectors are built with the zero-hot codification. Second, to each $y^\mu$ vector correspond one and only one $x^\mu$ vector, this is, there is only one couple $(x^\mu, y^\mu)$ in the fundamental set.
For every $\mu \in \{1, 2, \ldots, p\}$, from the couple $(x^\mu, y^\mu)$ the vector partition operator is applied to each $x^\mu$, then the new fundamental set is expressed by:

$$\{ (x^{\mu_1}, y^\mu), (x^{\mu_2}, y^\mu), \ldots, (x^{\mu_q}, y^\mu) \mid \mu = 1, 2, \ldots, p, l = 1, 2, \ldots q \}$$

Now, from the couple $(x^{\mu l}, y^\mu)$ the $q$ matrices are built as follows: $[y^\mu \otimes (x^{\mu l})^t]_{mx(nq)}$. Then apply the binary $\wedge$ operator to the corresponding matrices obtained to get $\Lambda^l$ as follows: $\Lambda^l = \bigwedge_{\mu=1}^{p} [y^\mu \otimes (x^{\mu l})^t]_{mx(nq)}$. The $ij$–th component of each $l$ matrices is given by:

$$\lambda_{ij}^l = \bigwedge_{\mu=1}^{p} \alpha(y^\mu_i, x^\mu_j)$$

**RECALLING PHASE**

**STEP 1:**

A pattern $x^\omega$, that could be or not from the fundamental set, is presented to $\Lambda^l$. First, we have to apply the vector partition operator to $x^\omega$:

$$\rho(x^\omega, q) = \{ x^{\omega 1}, x^{\omega 2}, \ldots, x^{\omega q} \}$$

For every $\Lambda^l$ matrix and $x^{\omega l}$ partition with $l = \{1, 2, \ldots, q\}$, the $\nabla_\beta$ operation is done and the resulting vector is assigned to a vector called $z^{\omega l}$: $z^{\omega l} = \Lambda^l \nabla_\beta x^{\omega l}$ the $i$–th component of the resulting vector is given as:

$$z_{i}^{\omega l} = \bigvee_{j=1}^{n} \beta(\lambda_{ij}^l, x^{\omega l}_j)$$

**STEP 2:**

It is necessary to build the *min sum vector* $r$ according to the definition therefore the corresponding $z^{\omega l}_{int}$ is given as:

$$z^{\omega l}_{int} = \begin{cases} 1 \text{ if } r_i = \bigwedge_{k \in \theta} r_k \text{ AND } z^{\omega l}_{int} = 0 \\ 0 \text{ otherwise} \end{cases}$$

where $\theta = \{i | z^{\omega l}_{int} = 0\}$.

Once we have obtained each $z^{\omega l}_{int}$ vector from step 2, in order to obtain one resultant vector the step 3 is applied.

**STEP 3:**

An intermediate vector $I$ is created. This vector will contain the sum of the $i$–th components of the $z^{\omega l}_{int}$ vectors:

$$I_i^\omega = \sum_{l=1}^{q} z^{\omega l}_{int}$$

then the corresponding $y^\omega$ vector is obtained by:
\[
y_i^\omega = \begin{cases} 
1 & I_i^\omega = \bigwedge_{k=0}^p I_k^\omega \\
0 & \text{otherwise}
\end{cases}
\]

Due to a matter of space we only show an example of the alpha-beta heteroassociative multimemory type max, the type min could be induced from this.

**Example 2.** Let \(x^1, x^2, x^3, x^4\) be the input patterns

\[
x^1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad x^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad x^3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad x^4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

The output vectors are built with the One-Hot codification, and for each output pattern corresponds *one and just one* input pattern, therefore the fundamental set is expressed as follows:

\[
\{(x^1, y^1), (x^2, y^2), (x^3, y^3), (x^4, y^4)\}
\]

Each input pattern from the fundamental set must be divided into as many partitions as we choose, in this case in 3. Therefore, it is necessary to verify whether it is possible to divide the input pattern through the definition \([6]\)

\[
\frac{n}{q} = \frac{6}{3} = 2 \in \mathbb{Z}^+
\]

Since \(2 \in \mathbb{Z}^+\), it is possible to divide each input pattern in 3 partitions of equals dimension. After that, the *vector partition operator* is applied, obtaining the new input patterns similar to this:

\[
\rho(x^1, 3) = \{(10), (11), (00)\}
\]
\[
\rho(x^2, 3) = \{(01), (00), (01)\}
\]
\[
\rho(x^3, 3) = \{(10), (01), (01)\}
\]
\[
\rho(x^4, 3) = \{(11), (01), (11)\}
\]

then the new fundamental set is:

\[
\{(x^{11}, y^1), (x^{21}, y^2), (x^{31}, y^3), (x^{41}, y^4), (x^{12}, y^1), (x^{22}, y^2), (x^{32}, y^3), (x^{42}, y^4), (x^{13}, y^1), (x^{23}, y^2), (x^{33}, y^3), (x^{43}, y^4)\}
\]

finally, from the new fundamental set, the *q* matrices \(V^l\) are built:
\[
V^1 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{pmatrix}, \quad V^2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 2 & 1 \\ 2 & 1 \end{pmatrix}, \quad V^3 = \begin{pmatrix} 2 & 2 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \end{pmatrix}
\]

Once we have the \(V^l\) matrices, to recall \(x^\omega\) with \(\omega \in \{1, 2, 3, ..., p\}\), particularly \(x^4\), the vector partition operator is applied to \(x^4\) with \(q = 3\):

\[
\rho(x^4, 3) = \{(1 1), (0 1), (1 1)\}
\]

then each one of the new vectors, \(x^{41}, x^{42}, x^{43}\), are presented to its corresponding \(V^l\) matrix according to the new alpha-beta heteroassociative memory type Max algorithm. Therefore three new vectors, \(z^{41}, z^{42}, z^{43}\), are obtained.

\[
z^{41} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad z^{42} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \quad z^{43} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
\]

after that, from the three new resulting vectors the intermediate vector \(I^4\) is built

\[
I^4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 3 \end{pmatrix}
\]

finally, the expression [2] is applied in order to obtain the corresponding \(y^4\) vector:

\[
y^4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
\]

3.3 Proposed Classifier

In this subsection we present the pattern classifier algorithm based on the alpha-beta heteroassociative multimemories that was mainly developed for bioinformatics applications. This new algorithm is known as Alpha-Beta MultiMemories Classifier (ABMMC).

3.3.1 ABMMC Algorithm

**LEARNING PHASE**

1. The fundamental set is built taking \(k\) instances from each class; that is, the fundamental set is assumed to be balance with respect to each class. Thus, the first \(k\) instances belong to the first class, the next \(k\) instances belong to the second class, and so on.
2. The $\mathbf{W}^I$ matrices are built according to the learning phase of the alpha-beta heteroassociative multimemories type $Max$. The output patterns of these memories are denoted as $y$.

3. The $\Lambda^I$ matrices are built according to the learning phase of the alpha-beta heteroassociative multimemories type $Min$. The output patterns of these memories are denoted as $yz$.

RECALLING PHASE

1. A pattern to be classified is presented to the learning matrix type $Max$ and the resulting vector is known as $y_{max}$.

2. The same pattern is presented to the learning matrix type $Min$ and the resulting vector is known as $y_{min}$.

3. The vector $y_{min}$ is negated to obtain the $\sim y_{min}$ vector.

4. Apply the OR binary operation between the components of the $y_{max}$ and $\sim y_{min}$. The resulting vector is known as total class vector $y$.

5. The first $k$ components from $y$, corresponding to the first class, are added; the second $k$ components from $y$, corresponding to the second class, are added too, and so on, until we obtain as many values as classes we have.

6. The greater value(s) is(are) taken as the correct class(es).

7. For those cases in which there are more than one correct class elected, it will not be possible determine to which class the pattern belongs, so it will be considered as incorrect classification.

Example 3. Let the input patterns be divided in three different classes, $x^1, x^2$ to class 1 $x^3, x^4$ to class 2 and $x^5, x^6$ to class 3:

$$x^1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, x^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, x^3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, x^4 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, x^5 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, x^6 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and the corresponding output patterns $y^\mu$ in one-hot and $yz^\mu$ in zero-hot codifications. Then the fundamentals sets are expressed as follow:

$$\{(x^1, y^1), (x^2, y^2), (x^3, y^3), (x^4, y^4), (x^5, y^5), (x^6, y^6)\}$$

$$\{(x^1, yz^1), (x^2, yz^2), (x^3, yz^3), (x^4, yz^4), (x^5, yz^5), (x^6, yz^6)\}$$
Once we have the fundamental sets we must apply the vector partition operator to each one, in this case let \( q = 3 \) be the number of partitions. In order to apply the vector partition operator we need to verify if it is possible to partition the vector in this manner:

\[
\frac{n}{q} = \frac{6}{3} = 2 \in \mathbb{Z}^+
\]

as \( 2 \in \mathbb{Z}^+ \) it is possible to divide the vector, therefore the new fundamental sets are given as:

\[
\{ (x^{11}, y^1), (x^{21}, y^2), (x^{31}, y^3), (x^{41}, y^4), (x^{51}, y^5), (x^{61}, y^6),
(x^{12}, y^1), (x^{22}, y^2), (x^{32}, y^3), (x^{42}, y^4), (x^{52}, y^5), (x^{62}, y^6),
(x^{13}, y^1), (x^{23}, y^2), (x^{33}, y^3), (x^{43}, y^4), (x^{53}, y^5), (x^{63}, y^6) \}
\]

\[
\{ (x^{11}, yz^1), (x^{21}, yz^2), (x^{31}, yz^3), (x^{41}, yz^4), (x^{51}, yz^5), (x^{61}, yz^6),
(x^{12}, yz^1), (x^{22}, yz^2), (x^{32}, yz^3), (x^{42}, yz^4), (x^{52}, yz^5), (x^{62}, yz^6),
(x^{13}, yz^1), (x^{23}, yz^2), (x^{33}, yz^3), (x^{43}, yz^4), (x^{53}, yz^5), (x^{63}, yz^6) \}
\]

now the heteroassociative multimemories type \( \text{Max} \) and \( \text{Min} \) are created.

\[
V^1 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \\ 1 & 1 \end{pmatrix}, \quad V^2 = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 2 & 1 \\ 2 & 1 \\ 1 & 2 \\ 1 & 1 \end{pmatrix}, \quad V^3 = \begin{pmatrix} 2 & 2 \\ 2 & 1 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{pmatrix}
\]

\[
\Lambda^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \Lambda^2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Lambda^3 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}
\]

Once we have trained the ABMMC, in order to classify a pattern, be it known or unknown, it is necessary to do the following procedure. Let \( x^4 \) be the input pattern to be classified, then the pattern is presented to both alpha-beta heteroassociative multimemories type \( \text{Max} \) and \( \text{Min} \), obtaining the following vectors:

\[
y_{\text{max}} = (V^l A \beta x^4 l) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}
\]
\[ y_{\text{min}} = (A^l \nabla_{\beta} x^4 l) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \]

then the vector \( y_{\text{min}} \) is negated

\[ \tilde{y}_{\text{min}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \]

finally, the OR binary operation is made between \( y_{\text{max}} \) and \( \tilde{y}_{\text{min}} \) and the class is chosen

\[
\begin{align*}
0 & \text{ or } 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \{ 0 + 0 = 0 \text{ class 1} \\
0 & \text{ or } 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \{ 0 + 1 = 1 \text{ class 2} \\
1 & \text{ or } 1 & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} & \{ 0 + 0 = 0 \text{ class 3} \\
0 & \text{ or } 0 & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \{ 0 + 0 = 0 \text{ class 3} \}
\end{align*}
\]

Since class 2 has the maximum sum, 1, the pattern presented, \( x^4 \), is assigned to class 2. Indeed, \( x^4 \) corresponds to class 2, therefore we have correct classification.

## 4 Experimental Results

This section reports an experimental study of the ABMMC algorithm. In particular, we chose two tasks relevant to Bioinformatics: promoters recognition and splice-junction sequences identification in strings that represent nucleotides. The databases were obtained from the Machine Learning Repository of the University of California in Irvine [31].

Particularly, the promoters and splice-junction samples are taken from the “E. coli promoter gene sequences (DNA) with associated imperfect domain theory” database and “Primate splice-junction gene sequences (DNA) with associated imperfect domain theory” database, respectively.

The promoter database has 106 instances divided into two classes, promoters and non-promoters, 53 instances to each one. The sequences are formed by 57 nucleotides and its binary codification is shown in table 2.

On the other hand, the splice-junction database has 3190 instances divided into two relevant classes, exon-intron (EI) and intron-exon (IE). The distribution for the classes is: 25% (767) for EI, 25% (768) for IE and the remaining
50% (1655) corresponds to non-significant instances. The genetic sequences are formed by 60 nucleotides and their binary codification is shown in table 2 too. Since we just want to differentiate between EI and IE sequences, the remaining 50% of the instances are considered as non-significant.

Given that the alpha-beta heteroassociative memories use binary operations, the patterns used in the learning and recalling phase must be binary patterns too. So it was necessary to create a correspondence table between the DNA sequences and binary patterns. It was evident, by the results shown in this work and in [32], that the one-hot codification is the optimal.

This codification is very useful, especially in the following situations: where in the DNA sequences there are some positions in which it is necessary to indicate that such position could take more than one value. For example, for the sequence ATCG, it is possible that the third position could be represented either for “C” or “T”. In this case in order to create a binary sequence that represents those values we just have to apply the binary operator “OR” with the corresponding “C” and “T” sequences. This is possible due to the good performance of the alpha-beta memories in the presence of altered patterns (see table 2).

<table>
<thead>
<tr>
<th>Nucleotide Code</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1000</td>
</tr>
<tr>
<td>T</td>
<td>0100</td>
</tr>
<tr>
<td>C</td>
<td>0010</td>
</tr>
<tr>
<td>G</td>
<td>0001</td>
</tr>
<tr>
<td>D</td>
<td>1011</td>
</tr>
<tr>
<td>N</td>
<td>1111</td>
</tr>
<tr>
<td>S</td>
<td>0101</td>
</tr>
<tr>
<td>R</td>
<td>1001</td>
</tr>
</tbody>
</table>

Table 2. Nucleotide Conversion

In order to get an estimate of how well the algorithm learned both, the concept of promoter and the EI or IE splice-junction sequence, a series of experiments was made. Then the results were compared with some other algorithms that work under the same conditions.

4.1 DNA Promoter Sequence Classification

There are two other works to which we could compare our results, J. Ortega [33] and Baffes and Mooney [34], both of them made their experiments under the following conditions: 85 instances were randomly taken to build the training set and the remaining 21 were left in the test set. This procedure was repeated 100 times. The table 3 is a comparative between [33], [34], and the ABMMC, where N/D stand for Not-Determined.

It is evident that ABMMC overcome, without problems, the performance of the other algorithms, even when learning with only 80 or 70 instances
Table 3. Algorithms Comparison

<table>
<thead>
<tr>
<th>Learning J. Ortega Baffes &amp; Mooney</th>
<th>Alpha-Beta Multimemories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td>[33]</td>
</tr>
<tr>
<td>85/106</td>
<td>92.5%</td>
</tr>
<tr>
<td>80/106</td>
<td>N/D</td>
</tr>
<tr>
<td>70/106</td>
<td>N/D</td>
</tr>
</tbody>
</table>

out of the 106 available (the rows labeled "80/106" and "70/106"). For each case, the procedure was repeated 100 times, taking into account the 20 best results, and calculating the maximum, minimum, and average percentage. The results are shown in table 4.

Table 4. Algorithms Comparison

<table>
<thead>
<tr>
<th>ABMMC</th>
<th>85 training set</th>
<th>80 training set</th>
<th>70 training set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Maximum</td>
<td>98%</td>
<td>98%</td>
</tr>
<tr>
<td></td>
<td>Minimum</td>
<td>97%</td>
<td>96%</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>97.45%</td>
<td>96.7%</td>
</tr>
</tbody>
</table>

4.2 DNA Splice-Junction Sequences Classification

Our results are compared with R. Rampone [35] who presented an algorithm named BRAIN (Batch Relevance-based Artificial Intelligence). Two methodologies are used to compare the algorithms results: Hold-Out and 10-Fold Cross-Validation.

4.2.1 Hold-Out

The table 5 shows the result obtained in the experimental phase of the BRAIN algorithm where 2000 instances were taken for the training set and 1186 were left in the test set, leaving 4 repeated instances out of the experiment.

The ABMMC algorithm is assessed under the following conditions: 2000 instances were taken randomly to create the training set and the remaining 1190 were left in the test set. The procedure was repeated 20 times and the best result was chosen. In this case, the repetitions eliminated by Rampone are taken into account.

Table 5. BRAIN Performance with Hold-Out Methodology

<table>
<thead>
<tr>
<th>Classes</th>
<th>Instances for Training set</th>
<th>Instances for Test set</th>
<th>Number of Errors</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>464</td>
<td>303</td>
<td>41/1186</td>
<td>3.4%</td>
</tr>
<tr>
<td>IE</td>
<td>485</td>
<td>280</td>
<td>59/1186</td>
<td>4.9%</td>
</tr>
</tbody>
</table>
Table 6. ABMMC Performance with Hold-Out Methodology

<table>
<thead>
<tr>
<th>Classes</th>
<th>Instances for Training set</th>
<th>Instances for Test set</th>
<th>Number of Errors</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>464</td>
<td>303</td>
<td>38/1190</td>
<td>3.1%</td>
</tr>
<tr>
<td>IE</td>
<td>485</td>
<td>280</td>
<td>43/1190</td>
<td>3.6%</td>
</tr>
</tbody>
</table>

It is clear that ABMMC surpasses the outcome shown by BRAIN. Notice that, since the third class is non-significant, the instances which fall into that class are ignored for comparison purposes.

4.2.2 Cross-Validation

From the 3190 instances of the database, 1000 samples were randomly taken to form a subset to which the 10-Fold Cross-Validation methodology was applied. The table 7 shows the performance of the ABMMC compared against BRAIN and some others Machine Learning algorithms mentioned in [35].

Table 7. ABMMC Performance with Hold-Out Methodology

<table>
<thead>
<tr>
<th>Machine Learning Algorithm</th>
<th>EI Error Rate</th>
<th>IE Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRAIN</td>
<td>0.050</td>
<td>0.040</td>
</tr>
<tr>
<td>ABMMC</td>
<td>0.068</td>
<td>0.078</td>
</tr>
<tr>
<td>KBANN</td>
<td>0.076</td>
<td>0.085</td>
</tr>
<tr>
<td>BackProp</td>
<td>0.057</td>
<td>0.107</td>
</tr>
<tr>
<td>PEBLS</td>
<td>0.082</td>
<td>0.075</td>
</tr>
<tr>
<td>Perceptron</td>
<td>0.163</td>
<td>0.174</td>
</tr>
<tr>
<td>ID3</td>
<td>0.106</td>
<td>0.140</td>
</tr>
<tr>
<td>COBWEB</td>
<td>0.150</td>
<td>0.095</td>
</tr>
<tr>
<td>Near. Neighbor</td>
<td>0.116</td>
<td>0.091</td>
</tr>
</tbody>
</table>

The ABMMC overcomes all the algorithms except BRAIN in the classification of the significant classes IE and EI. It is important to notice that the BRAIN algorithm was created especially to be applied to the “Primate splice-junction gene sequences (DNA) with associated imperfect domain theory” database. On the other hand, the ABMMC is for general purposes; that is, it was not made to be applied in any particular database. Hence, as a future work, we propose the design of a similarity algorithm that helps in specific classifications.

5 Conclusions and Future Work

With the present work, three extensions to the original model of alpha-beta associative memories are presented. First, an extension to the original heteroassociative memory algorithm, which enables it to correctly recall the
whole fundamental set. This is then used to build the alpha-beta heteroassociative multimemories, foundation of the new pattern classifier, the alpha-beta multimemories classifier.

This new classifier is shown, through an experimental analysis, to exhibit a competitive classification performance in the tasks of promoter identification and splice-junction zone localization, even surpassing most of the best classifiers in these tasks, currently present in scientific literature.

This competitive performance presents the opportunity to apply the AB-MMC algorithm to other tasks in Bioinformatics, with foreseeable competitive results.

On the other hand, since not all the comparisons were favorable, it remains as future work to improve the classifier and better tune it to the tasks and databases on which it was tested here.

Acknowledgements. The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, COFAA, SIP, and CIC), the CONACyT, and SNI for their economical support to develop this work.

References


Mining Clinical, Immunological, and Genetic Data of Solid Organ Tansplantation

Michele Berlingerio\textsuperscript{1,2}, Francesco Bonchi\textsuperscript{2}, Michele Curcio\textsuperscript{3}, Fosca Giannotti\textsuperscript{2}, and Franco Turini\textsuperscript{4}

\textsuperscript{1} IMT School for Advanced Studies
Via San Micheletto, 3 - Lucca, Italy
\textsuperscript{2} Pisa KDD Laboratory ISTI - CNR, Area della Ricerca di Pisa
Via Giuseppe Moruzzi, 1 - Pisa, Italy
\textsuperscript{3} Unità di Immunoeumatologia 2, Azienda Ospedaliero Universitaria Pisana
Via Paradisa, 2 - Pisa, Italy
\textsuperscript{4} Pisa KDD Laboratory - Computer Science Dept., University of Pisa
Largo Pontecorvo, 3 - Pisa, Italy

Summary. Clinical databases store large amounts of information about patients and their medical conditions. Data mining techniques can extract relationships and patterns implicit in this wealth of data, and thus be helpful in understanding the progression of diseases and the efficacy of the associated therapies. In this perspective, in Pisa (Italy) we have started an important data collection and analysis project, where a very large number of epidemiological, clinical, immunological and genetic variables collected before the transplantation of a solid organ, and during the follow-up assessment of the patients, are stored in a datawarehouse for future mining. This on-going data collection involves all liver, kidney, pancreas and kidney-pancreas transplantations of the last five years of one of the largest (as to number of transplantations) centers in Europe. The project ambitious goal is to gain deeper insights in all the phenomena related to solid organ transplantation, with the aim of improving the donor-recipient matching policy used nowadays. In this chapter we report in details two different data mining activities developed within this project. The first analysis involves mining genetic data of patients affected by terminal hepatic cirrhosis with viral origin (HCV and HBV) and patients with terminal hepatic cirrhosis with non-viral origin (autoimmune): the goal is to assess the influence of the HLA antigens on the course of the disease. In particular, we have evaluated if some genetic configurations of the class I and class II HLA are significantly associated with the triggering causes of the hepatic cirrhosis. The second analysis involves clinical data of a set of patients in the follow-up of a liver transplantation. The aim of the data analysis is that of assessing the effectiveness of the extracorporeal photopheresis (ECP) as a therapy to prevent rejection in solid organ transplantation. For both analyses we describe in details, the medical context and goal, the nature and structure of the data. We also discuss which kind of data mining technique is the most suitable for our purposes, and we describe the details of the knowledge discovery process followed and extracted knowledge.
1 Introduction

With recent proliferation of information systems in modern hospitals and health care institutions, there is an increasing volume of medical-related information being collected. Appropriate tools are needed to extract relevant and potentially fruitful knowledge from this wealth of medical data.

Traditionally, statistical data analysis was the final phase of experimental design that, typically, included a careful selection of patients, their features and the definition of the hypothesis to test. With the introduction of data warehouses, such a selective approach to data collection is altered and data may be gathered with no specific purpose in mind. Yet, medical data stored in warehouses may provide a useful resource for potential discovery of new knowledge. The activity of analyzing, for the purpose of knowledge discovery, data that has been collected with no clear pre-defined objective in mind, is usually named Data Mining.

Data Mining is an emerging technology aimed at unlocking the knowledge lying dormant in huge databases, thus closing the gap between data generation and data comprehension. In medicine, overcoming this gap is particularly crucial since medical decision making needs to be supported by arguments based on basic medical knowledge as well as knowledge, regularities and trends extracted from data. There are two main aspects that define the significance of and the need for intelligent data analysis in medicine:

- The first aspect concerns the support of specific knowledge-based problem solving activities (diagnosis, prognosis, monitoring, treatment planning, etc.) through the intelligent analysis of individual patients’ raw data. Data are mostly numeric and often quite noisy and incomplete. The aim is to glean, in a dynamic fashion, useful abstractions of the patient’s (past, current, and hypothesized future) situation which can be matched against the relevant (diagnostic, prognostic, monitoring, etc.) knowledge for the purposes of the particular problem solving activity.

- The second important aspect concerns the discovery of new medical knowledge that can be extracted through data mining of representative collections of example cases, described by symbolic or numeric descriptors. The available datasets are often incomplete (missing data) and noisy (erroneous). Of particular value to medicine is the requested accuracy and interpretability of the results of data mining. The interpretability may be achieved by representing the results of data mining graphically or by symbolically expressed rules or relationships. To increase the chances of getting useful and interpretable results, data mining can benefit from medical experts who may specify additional (background) knowledge, interact with the mining process, and evaluate its results. Only the accurate patterns and relationships that are expressed at the right level of abstraction in the vocabulary used by medical experts may be of use for a practitioner who will decide whether to adopt and use the extracted knowledge in daily decision making.
In this perspective, we have started an important data collection and mining project, where a very large number of epidemiological, clinical, immunological and genetic variables collected before the transplantation of a solid organ, and during the follow-up assessment of the patients, are stored in a datawarehouse for future mining. In this chapter we report in details two different data mining activities developed within this project. For both analyses we describe in details, the medical context and goal, the nature and structure of the data. We also discuss which kind of data mining technique is the most suitable for our purposes, and we describe the details of the knowledge discovery process followed and extracted knowledge. We believe that this experience can be repeated in other medical data analyses where the data exhibits similar characteristics and structure.

2 The Health Mine Project

While the immunosuppressive drugs developed in the past two decades have improved the short-term survival of organ allografts, the effects of these regimens on long-term outcome has not yet been determined. A significant shortcoming of current anti-rejection therapies is that recipients require life-long treatment on an immunosuppressive regimen and are left at greater risk of serious side effects. Thus, it is of significant importance to properly assess the best donor-recipient pair in the transplant context. Human Leukocyte Antigens (HLA, described in deeper details later in Section 4), also known as histocompatibility antigens, are molecules found on all nucleated cells in the body. Histocompatibility antigens help the immune system to recognize whether or not a cell is foreign to the body. Hence the success of an organ transplantation is strongly connected to the HLA systems of the donor-recipient pair. Beyond this important role, the HLA system seems to influence also the clinical course of, for example, hepatic cirrhosis, both on viral and autoimmune basis. However, not only different antigens have different importance w.r.t. hepatic cirrhosis, but, to make things more complicated, other not yet well characterized factors could play an important role. It is thus important to assess the relationships that can hold between HLA patterns, together with other medical and non-medical factors, and the success of a solid organ transplantation.

The “Unità di Immunoematologia 2” of the “Azienda Ospedaliero Universitaria Pisana”, a high-specialized Analysis Laboratory, is in charge of studying the patients to be treated with solid organ transplantation. The strong synergy between the laboratory and the transplant group has lead in the last years to excellent results of national relevance. This also thanks to several research projects aiming at understanding post-transplant reject issues. As said before, although the application of last generation immunosuppressive therapies, a minimal percentage of the patients fall into reject. Unfortunately, understanding the the causes and the processes leading to an allograft reject is a difficult task due to the large variety of variables involved,
and to the fact that typically the physicians miss the analytical instruments needed to explore all the possible combinations of such variables.

In this context we have started a collaboration between the “Unità di Immunoematologia 2” and the Knowledge Discovery and Delivery Laboratory of the ISTI - CNR (National Council of Research), aimed at analyzing by means of data mining techniques the information collected about patients who had a solid organ transplant, with the final goal of extracting knowledge useful for understanding the genetical mechanisms involved in the transplant success. This collaboration as been named the Health Mine project, and it as been structured in three subprojects, reflecting different speed in the data collection phase, and different challenges in the data analysis, thus shaping short term and long-term objectives:

1. HLA and liver diseases,
2. Photopheresys assessment,
3. HLA and kidney transplantation.

**HLA and liver diseases.** Here the medical problem is to find possible associations between HLA alleles combinations and serious liver diseases that lead the patients to organ transplantation. The main objective of this analysis is to find patterns involving DNA, gender, age and other variables, which are more frequent in the patients with liver diseases than in the healthy population. In this way we can make hypotheses on possible associations between particular genetic and demographics characteristics and particular kinds of cirrhosis. In this analysis we used Frequent Pattern Discovery techniques, as further explained in Section [4](#).

**Photopheresys assessment.** In the last few years, a new type of therapy, the extracorporeal photopheresys (ECP) has started emerging as a helpful therapy when the reject is detected. This treatment can reduce the amount of lymphocytes that sustain the reject process. The objective of this data mining activity is to assess the efficacy of the photopheresys in patients with liver transplant, by analyzing at the same time several kinds of genetical and biochemical variables, in order to individuate which of them can positively influence the success of the photopheretic therapy.

**HLA and kidney transplantation.** The chimerism and the microchimerism are phenomena that can occur in the recipient after a transplant. In poor words, they consist in the acquisition, by the recipient’s body, of some cells of the donor, due to the organ transplantation. In particular, in this analysis we focus on kidney transplantation, and the objectives are:

- to create a database containing all the clinical variables collected during the follow-up of the recipients,
- to set-up a technique for assessing both from the qualitative and the quantitative point of view the systemic microchimerism by using the HLA polymorphism and the Real-Time technology,
• to assess the relationship among epidemiological, clinical, genetical and immunological variables w.r.t. the study of the polymorphism of the minor histocompatibility antigens.

In the next section we discuss the challenges related to the data collection phase which is still in progress. While in section 4 and 5 we present the two analysis for which the data collection and mining has been concluded in terms of applied methodologies, results and patterns evaluation.

3 Data Collection

The data collection phase in the medical domain is inherently much more challenging that in other domains. It is the nature of the data itself that makes it on the one hand very difficult to be collected (and in reasonably size), and on the other hand very precious for the kind of knowledge that we can acquire by analyzing it. In particular, the reasons behind the hardness of medical data collection are various:

• **Privacy and legal constraints**: due to the highly sensitive nature of the data, the outsourcing of the data mining analysis to researchers external to the medical institution is usually quite complex or in some cases almost impossible. On the one hand the data can hardly be moved outside the medical institution, on the other hand external researcher can be hardly put at work inside the medical institution.
• **Ownership of the data**: it is very difficult to access the medical data owned by different laboratories, unless they do not cooperate in the same research project, and sometime even if they do.
• **Money-cost of data acquisition**: most of the data come from results of expensive chemical-clinical analyses that are not always possible to perform.
• **Time-cost of data acquisition**: there are no machines that take the blood from the refrigerator, perform automatically the needed analyses, report the results in a standard form, sort, merge and clean the data, and so on. All these operations have to be done manually by people that are not full-time dedicated to this process. Sometimes already existing information are not stored digitally, but physically written on paper, thus even more difficult to access, understand and acquire.

For the reasons above, the most involving data collection, i.e., the one for the third subproject (*HLA and kidney transplantation*) is still going on, while for the other two subprojects is finished.

For the first analysis the database contained 2 datasets: the first one containing data from 534 patients with autoimmune or viral cirrhosis, collected since 1996. For each of the patients information about their age, gender, geographic origin and the alleles corresponding to the six HLA antigens corresponding to loci A, B and DRB1, plus some information about the disease are collected. In the second dataset, used as control population, we had data
of 4718 healthy individuals (coming from the same geographic area), collected since 2001, for which we had the same information (except, obviously, for the disease-related information).

The second database, collected since 2001, contained information about 127 patients that received a liver transplant and, in order to prevent the allograft rejection, were subjected to the ECP therapy for a period of about 1 year. For each patient we had several observations of the same set of variables, taken at different times.

The datasets in analysis are unique in terms of kind of information contained, global amount of patients and variables taken into account. Another unique feature is the various kind of information contained in the datasets. In fact, the first one was aimed at recording all the patients that were starting the entire process towards the organ transplantation. For these patients, the information needed was the HLA characterization, the blood group, the kind of diseases and some personal information like age, gender, and so on. For these patients, only one record containing one single value for each variable was needed. In terms of type of information, this corresponds to the market basket analysis case: several baskets (patients) with several items (variable), a single instance for each item.

On the other hand, the second database contained a completely different kind of information. Basically, the medical context in which the data was collected was different: here the physicians kept track of the response of the patients to the ECP treatment. Thus, in addition to the personal and genetical information, that do not change during the time, we had several observations for the same set of variables for each patients. In this case the temporal dimension played an important role, because the focus is not on a particular pattern at a specific time, but rather on the "trend" of some variables, how frequent a kind of response to the treatment can be.

As one can see, these different data need to be treated in different ways, and the next two sections explain the analyses performed on such data, together with the Data Mining techniques used for this purpose.

4 Mining HLA Patterns Associated with Liver Diseases

As stated in Section 2 not only different antigens have different importance w.r.t. hepatic cirrhosis, but, to make things more complicated, other not yet well characterized factors could play an important role. It is thus important to assess the existence of associations between haplotypic settings (possibly mixed with clinical and demographic information) and hepatic cirrhosis. Algorithms developed for frequent patterns discovery can help us to achieve this goal. Thanks to their ability in handling a very large number of variables and the associated exponential search space, such techniques can discover interesting haplotypic patterns beyond the capabilities of traditional analysis methods.
Fig. 1. Gene map of the human leukocyte antigen (HLA) region. The HLA region spans 410^6 nucleotides on chromosome 6p21.1 to p21.3, with class II, class III and class I genes located from the centromeric (Cen) to the telomeric (Tel) end.

This section presents the analysis done in this direction, both from the medical and from the data mining techniques points of view.

4.1 Background and Motivations

The Major Histocompatibility Complex (MHC) is a genetic region (i.e., a set of genes), that is essential for a correct autoimmune response. Such genetic region, located on the short arm of the sixth chromosome, includes the HLA system. The main function of HLA genes consists of discriminating cells that are your own (self) from those that are foreign (non-self). The genetical organization of the HLA system is rather simple (see Figure 1). There are three classes of HLA antigens: six major antigens, two for each locus HLA-A, HLA-B and HLA-C, which are the most important among class I antigens; other four groups of antigens, named HLA-DR, HLA-DQ, HLA-DM and HLA-DP, which form the class II antigens; and finally the HLA class III region, which contains many gene encoding proteins that are unrelated to cell-mediated immunity but that nevertheless modulate or regulate immune responses in some way.

Despite the rather simple organization, the nomenclature defining HLA antigens is rather complicated, because the definition system must take in account the high level of allelic polymorphism, i.e., the high number of different possible values (alleles) for a locus (the position or the site of a gene), present in human population. Due to the extreme polymorphism of the HLA loci, only a few individuals out of the family are identical with respect to this system. For these reasons, the HLA was identified as the genetic locus whose products are directly responsible of the rapid rejection in solid organ and marrow transplantation. Beyond this important role, the HLA system seems to influence also the clinical course of hepatic cirrhosis. In fact, although several infectious factors could influence the clinical course of the
hepatic cirrhosis (e.g., HCV, HBV), the genetical setting of the host patient seems to play a central role. The first genes that have been studied, due to their important function in controlling the immune response, were the class I and class II HLA gene. The association between the class II HLA molecules (HLA-DRB1, HLA-DQB1) and, mainly, the chronic infection by HCV and HBV has firstly been reported in \cite{27,26}. However, the HLA has also an important role in determining the susceptibility to autoimmune based diseases. Most of the diseases associated with the HLA system are, in fact, pathologies related to an autoimmune response against self antigens: in practice, in some genetically predisposed individuals could start some uncontrollable immune phenomena, direct against self antigens that determine a physiologic damage to the target organ.

While some associations between liver diseases due to viral infections and HLA antigens are known in literature, it is still not clear which role is played by which antigens in the development of autoimmune hepatic cirrhosis. It is thus important to assess the existence of associations between haplotypic settings (possibly mixed with clinical and demographical information) and hepatic cirrhosis. Algorithms developed for frequent patterns discovery can help us achieve this goal. Thanks to their ability in handling a very large number of variables and the associated exponential search space, such techniques can discover interesting haplotypic patterns beyond traditional analysis methods capabilities.

In this section we report a frequent pattern discovery analysis on genetical data of patients affected by terminal hepatic cirrhosis with viral origin (HCV and HBV) and patients with terminal hepatic cirrhosis with non-viral origin (autoimmune), conducted with the aim of assessing the influence of the HLA antigens on the course of the disease. In particular, we have evaluated if some genetical configurations of the class I and class II HLA are significantly associated with the triggering causes of the hepatic cirrhosis. The analysis has been performed on two datasets: the first one contains data of 534 patients with terminal cirrhosis that led to liver transplantation, while the second one used as a control population, contains the data of 4718 healthy individuals coming from the same geographic area (this dataset has been previously used in \cite{22}).

The frequent pattern analysis has led to the discovery of some associations already known in the medical literature, and of some not previously known interesting ones, which are object of further biological investigation. The main contribution of this analysis is however methodological: it has proven the adequateness of the frequent pattern discovery methods on this kind of data.

4.2 Frequent Pattern Discovery

The collation of large electronic databases of scientific and commercial information has led to a dramatic growth of interest in methods for discovering
structures in such databases. These methods often go under the general name of data mining. However, recently two different kinds of structures sought in data mining have been identified: models and patterns. The first of these, models, are high level, global, descriptive summaries of data sets. Patterns, on the other hand, are local descriptive structures. Patterns may be regarded as local models, and may involve just a few points or variables; that is, they are descriptions of some small part of the data, instead of overall descriptions. Accordingly, Pattern Discovery has a distinguished role within data mining technology. In particular, since frequency provides support to any extracted knowledge, it is the most used and maybe the most useful measure of interest for the extracted patterns. Therefore during the last decade a lot of researchers have focussed their studies on the computational problem of Frequent Pattern Discovery, i.e., mining patterns which satisfy a user-defined minimum threshold of frequency [2, 13].

The simplest form of a frequent pattern is the frequent itemset.

Definition 1 (Frequent Itemset Mining). Let \( I = \{x_1, ..., x_n\} \) be a set of distinct items, an itemset \( X \) is a non-empty subset of \( I \). A transaction database \( D \) is a bag of itemsets \( t \in 2^I \), usually called transactions. The support of an itemset \( X \) in a database \( D \), denoted \( sup_D(X) \), is the number of transactions which are superset of \( X \). Given a user-defined minimum support, denoted \( \sigma \), an itemset \( X \) is called frequent in \( D \) if \( sup_D(X) \geq \sigma \). The Frequent Itemset Mining Problem requires to compute all the itemsets which are frequent in a transaction database:

\[
\mathcal{F}(D, \sigma) = \{\langle X, sup_D(X) \rangle | X \in 2^I \land sup_D(X) \geq \sigma\}
\]

The identification of sets of items, products, symptoms, characteristics, and so forth, that often occur together in the given database, can be seen as one of the most basic tasks in data mining. Although frequent itemsets are per se meaningful, the original motivation to extract them was given by the well known association rules analysis [11], where the analyst is interested in finding rules describing customers behavior in buying products. Their direct applicability to business problems together with their inherent understandability, even for non data mining experts, made association rules a popular mining method, and frequent itemsets mining one of the hottest research themes in data mining. However frequent itemsets are meaningful not only in the context of association rules mining: they can be used as basic elements in many other kind of analysis, and in particular, they can be used to build global models, ranging from classification [20, 19] to clustering [25, 32].

In the biological domain where our analysis applies, we are not interested in all possible frequent itemsets. Intuitively, we search for patterns describing HLA configurations and their association with the diseases, thus we are mainly interested in itemsets covering as many as possible of the variables regarding HLA loci and the other variables. In practice, if we got a pattern (a set of characteristics) \( \{a, b, c\} \) which has the same frequency of a larger
pattern, e.g., \{a, b, c, d, e\}, then we are only interested in the larger one. This practical consideration justifies the need for closed frequent itemsets.

Closed itemsets were first introduced in \cite{23} and received a great deal of attention especially from an algorithmic point of view \cite{34,31}. They are a concise and lossless representation of all frequent itemsets, i.e., they contain the same information without redundancy. Intuitively, a closed itemset groups together all its subsets that have the same support; or in other words, it groups together itemsets which identify the same group of transactions.

**Definition 2 (Closure Operator).** Given the function \(f(T) = \{i \in I \mid \forall t \in T, i \in t\}\), which returns all the items included in the set of transactions \(T\), and the function \(g(X) = \{t \in D \mid \forall i \in X, i \in t\}\) which returns the set of transactions supporting a given itemset \(X\), the composite function \(c = f \circ g\) is the closure operator.

**Definition 3 (Closed Itemset).** An itemset \(I\) is closed if and only if \(c(I) = I\). Alternatively, a closed itemset can be defined as an itemset whose supersets have a strictly smaller support. Given a database \(D\) and a minimum support threshold \(\sigma\), the set of frequent closed itemsets is denoted:

\[
Cl(D, \sigma) = \{\langle X, \text{sup}_D(X)\rangle \in C_{\text{freq}} \mid \nexists Y \supset X \text{s.t.} \langle Y, \text{sup}_D(X)\rangle \in C_{\text{freq}}\}.
\]

In particular we adopted the DCL_Closed software \cite{21} for mining frequent closed itemsets. In the next section we describe in details the analysis we performed.

### 4.3 Mining HLA

As stated before, the objective of our analysis is to find possible associations between combinations of frequent HLA alleles and serious liver diseases that led the patient to liver transplantation.

We have worked on two databases: the first one containing data from 534 patients with autoimmune or viral cirrhosis. For each of them we had the data about their age, gender, geographic origin and the alleles corresponding to the six HLA antigens corresponding to loci A, B and DRB1, plus some characterization and information about the disease. In the second database, used as control population, we had data on 4718 healthy individuals (coming from the same geographic area) for which we had the same information (except, obviously, for the disease-related information).

Before the mining phase, we have performed some pre-processing on the patients database, in order to make it suitable for the frequent closed itemset discovery algorithm. During this first phase we have removed noisy data and patients for which too much information was missing. At the end of this phase the database of patients was reduced to 410 individuals. Since the algorithms work on integer numbers, we have then mapped all the available information to integers: an integer number (an item) as been assigned to each single possible value of each variable. In order to make it more significant, we have
discretized the age information in buckets: the first bucket ranging from 1 to 30 years, and the following ones having a size of 5 years.

An important problem addressed in this phase has been that of homozygous alleles, i.e., the situation in which the two alleles for a given locus are identical. Since we are dealing with information structured in transactions, i.e., plain sets and not bags, no duplicate items are admitted. Therefore, in order not to lose information about homozygous alleles, we have created a special item for each locus indicating the presence of homozygous alleles.

Following the physicians’ indications we have then divided the pathologies in two main groups, with a total of three item codes:

- Hepatic cirrhosis due to viral infection:
  - one item code for HCV+;
  - one item code for HBV+;
- Hepatic cirrhosis with non viral origin (mainly autoimmune): one unique item code for cryptogenetic cirrhosis, primary biliary cirrhosis, necrosis VBI and VBE, alcoholic cirrhosis, HCC, etc.

Example 1. Table 1 shows a sample of the input file accepted by the algorithm. The file is composed by several transactions (sets of integers), that can have different length: a patient can have different types of cirrhosis at the same time (this is the case of the patient in the third transaction, which has two disease codes, 1002 and 1004), or it could have some genetic variables missing. The first transaction in Table 1 regards a female patient, up to 30 years old, with autoimmune cirrhosis (code 1004) and the following HLA characterization:\n\n\[A_1 = 24, A_2 = 25, B_1 = 8, B_2 = 18, DRB1_1 = DRB1_2 = 1.\] Here the DRB1 is a homozygous locus, i.e. it has the same value in each of the two chromosomes; this is represented by the presence of the item code 2499.

We ran the software on the input database and produced the frequent closed itemsets. After the mining phase we performed a post-processing phase, in which the extracted patterns were automatically selected w.r.t. their interestingness. Here by interesting pattern we mean a pattern with an exceptionally high frequency in the patients database w.r.t. the frequency of the same pattern, without disease information, in the control database. In fact, this situation would correspond to a possible association between a specific pattern and a specific class of diseases.

Table 2 reports some of the results obtained with this method. In the second column we have the pattern which if taken together with the disease code in the third column, constitutes a frequent closed itemset in the patients

<table>
<thead>
<tr>
<th>Table 1. A sample of the input database</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 30 2024 2025 2208 2218 2401 2499 1004</td>
</tr>
<tr>
<td>1 30 2001 2002 2203 2297 2403 2499 1004</td>
</tr>
<tr>
<td>2 35 2024 2031 2214 2251 2414 2404 1004 1002</td>
</tr>
</tbody>
</table>
Table 2. Some interesting patterns discovered

<table>
<thead>
<tr>
<th>ID</th>
<th>Pattern</th>
<th>Disease</th>
<th>patients DB</th>
<th>control DB</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sex=F; A = 1; B = 35</td>
<td>autoim.</td>
<td>1.463%</td>
<td>0.169%</td>
<td>8.656</td>
</tr>
<tr>
<td>2</td>
<td>Age in [56,60]; A = 2; B = 11</td>
<td>autoim.</td>
<td>1.219%</td>
<td>0%</td>
<td>+∞</td>
</tr>
<tr>
<td>3</td>
<td>A1 = 1; A2 = 2; B = 18; DRB1 = 11</td>
<td>autoim.</td>
<td>1.463%</td>
<td>0.169%</td>
<td>8.656</td>
</tr>
<tr>
<td>4</td>
<td>Sex=M; A = 1; B = 18; DRB1 = 11</td>
<td>autoim.</td>
<td>1.463%</td>
<td>0.169%</td>
<td>8.656</td>
</tr>
<tr>
<td>5</td>
<td>Age in [51,55]; A = 1; B = 35; DRB1 = 11</td>
<td>any</td>
<td>1.219%</td>
<td>0.233%</td>
<td>5.231</td>
</tr>
<tr>
<td>6</td>
<td>Sex=M; A = 2; DRB11 = DRB12</td>
<td>autoim.</td>
<td>1.463%</td>
<td>0.254%</td>
<td>5.759</td>
</tr>
<tr>
<td>7</td>
<td>Sex=M; A = 2; B = 51; HCV+, autoim.</td>
<td>HCV+, autoim.</td>
<td>2.339%</td>
<td>0.701%</td>
<td>3.476</td>
</tr>
<tr>
<td>8</td>
<td>Sex=M; A = 2; B = 51; DRB1 = 11</td>
<td>autoim.</td>
<td>1.463%</td>
<td>0.445%</td>
<td>3.287</td>
</tr>
<tr>
<td>9</td>
<td>Age in [56,60]; A = 1; B = 18; DRB1 = 11</td>
<td>HBV+</td>
<td>1.219%</td>
<td>0%</td>
<td>+∞</td>
</tr>
<tr>
<td>10</td>
<td>Sex=F; Age in [41,45]; A = 2; B = 18; DRB1 = 11</td>
<td>any</td>
<td>1.219%</td>
<td>0.042%</td>
<td>29.023</td>
</tr>
<tr>
<td>11</td>
<td>Age in [56,60]; A = 2; DRB1 = 7</td>
<td>HCV+, autoim.</td>
<td>1.951%</td>
<td>0%</td>
<td>+∞</td>
</tr>
</tbody>
</table>

database, whose relative frequency is given in the fourth column. In the fifth column we have the relative frequency of the pattern (without the disease information) in the control database. In the last column is reported the ratio between the relative frequency in the patients and the relative frequency in the control database. The higher this ratio, the more interesting the pattern.

The patterns exhibiting a surprisingly high ratio, automatically selected in our post-processing phase, have been then evaluated by the domain experts.

4.4 Patterns Evaluation

In the following we report some bio-medical interpretation of a few patterns which have been considered particularly interesting by the domain experts.

Patterns 3 and 4: these two patterns, sharing the common haplotype A=1, B=18, DRB1=11, exhibit a very high ratio w.r.t. the control population. This could be explained by assuming that, as reported for some autoimmune diseases (Grave’s disease, Hashimoto’s disease), particular alleles (HLA-DRB1 = 11 in our case) have a higher capability in presenting the antigens to T cells.

Patterns 7 and 8: these two patterns share the subpattern A=2, B=51 with autoimmune cirrhosis. In the biological literature the association among A=2, B=51 and the hepatic cirrhosis with autoimmune origin was already known (see [29] and [30]). The discovery of these patterns confirms the adequateness of the proposed methodology.

Pattern 11: describes a set of patients 56 to 60 years old, with A = 2, DRB1 = 7, with hepatic cirrhosis from viral HCV+ infection on hepatic cirrhosis with autoimmune origin. Such a pattern has a relative frequency of 1,951% in the patients database, while it never appears in the much larger control database. This situation caught our attention and we further analyzed it. Surprisingly, we discovered that while no individual older than 55 is present in the healthy population, we found a 2,607% of healthy individuals with the same haplotype (A = 2, DRB1 = 7), but younger than 55. This could point out that patients with such haplotype are not capable to eliminate the HCV+ virus but they are predisposed to the
development of a chronic cirrhosis, leading to transplantation or to death. This would explain why over a certain age threshold, no individual is present in the healthy population.

The obtained results showed that the HLA antigens connected to high level hepatic damage are different in accordance to the cause of the disease. The pattern discovery techniques we used proved their capability in bringing to light particular associations in the haplotypic setting that could remain hidden to the traditional data analysis methods used in biomedicine.

Another interesting aspect acknowledged by the domain experts is that the HLA pattern discovered are, indeed, haplotypes, while the input data were containing genotypes (genotype is the entire allelic combination of an individual, while the haplotype is the allelic sequence inherited as a block from one of the parents). This fact, on one hand strengthens the biological significance of the patterns obtained, on the other hand, it suggests that frequent pattern discovery techniques can be applied to the haplotype inference problem, i.e., the problem of reconstructing haplotypes of individuals, given their genotypes. This problem has been widely studied in bioinformatics, and to date is still a hot algorithmic problem [5, 11, 10, 12].

5 Mining Temporal Patterns Assessing the Effectiveness of a Therapy

A typical structure of medical data is a sequence of observations of clinical parameters taken at different time moments. In this kind of contexts, the temporal dimension of data is a fundamental variable that should be taken in account in the mining process and returned as part of the extracted knowledge. Therefore, the classical and well established framework of sequential pattern mining is not enough, because it only focuses on the sequentiality of events, without extracting the typical time elapsing between two particular events. Time-annotated sequences (TAS), is a novel mining paradigm that solves this problem. Recently defined in our laboratory together with an efficient algorithm for extracting them, TAS are sequential patterns where each transition between two events is annotated with a typical transition time that is found frequent in the data. In principle, this form of pattern is useful in several contexts: for instance, (i) in web log analysis, different categories of users (experienced vs. novice, interested vs. uninterested, robots vs. humans) might react in similar ways to some pages - i.e., they follow similar sequences of web access - but with different reaction times; (ii) in medicine, reaction times to patients’ symptoms, drug assumptions and reactions to treatments are a key information. In all these cases, enforcing fixed time constraints on the mined sequences is not a solution. It is desirable that typical transition times, when they exist, emerge from the input data. TAS patterns have been also used as basic brick to build a truly spatio-temporal trajectory pattern mining framework [9].
In this section we report a real-world medical case study, in which the \textit{TAS} mining paradigm is applied to clinical data regarding a set of patients in the follow-up of a liver transplantation. The aim of the data analysis is that of assessing the effectiveness of the extracorporeal photopheresis (ECP) as a therapy to prevent rejection in solid organ transplantation.

For each patient, a set of biochemical variables is recorded at different time moments after the transplantation. The \textit{TAS} patterns extracted show the values of interleukins and other clinical parameters at specific dates, from which it is possible for the physician to assess the effectiveness of the ECP therapy. The temporal information contained in the \textit{TAS} patterns extracted is a fruitful knowledge that helps the physicians to evaluate the outcome of the ECP therapy even during the therapy itself.

We believe that this case study does not only show the interestingness of extracting \textit{TAS} patterns in this particular context but, more ambitiously, it suggests a general methodology for clinical data mining, whenever the time dimension is an important variable of the problem in analysis.

The interestingness of the case study by the medical perspective lies in the uniqueness and relevance of the dataset under analysis. While by the data analysis perspective, the interestingness lies (i) in the structure of the data under analysis, (ii) in the importance of the temporal dimension, and (iii) in the repeatability of the experience to other medical data analyses where the data exhibit the same structure. In fact, a typical structure of medical data is a sequence of observations of clinical parameters taken at different time moments. Here the time dimension is crucial: the focus is not only on the observed values, nor only in the sequence they compose, but the typical time that elapses among two events is also very important. For instance, the time elapsed from the start of a certain therapy, to the appearance of a certain clinical phenomenon.

The main contribution of this section is to provide a methodological approach to this kind of data, by means of Time-Annotated Sequences (\textit{TAS}) mining \cite{7, 8}.

The rest of the section is organized as follows. In section 5.1 we describe the biological problem studied in this section. In Section 5.2 we introduce the \textit{TAS} paradigm and how this can be used for mining time-annotated data. Section 5.3 describes the real-world case study in which the \textit{TAS} paradigm is applied to the photopheresis dataset.

5.1 Extracorporeal Photopheresis as a Therapy against Allograft Rejection

In this subsection we describe the biological background to the problem presented in this section.

Originally introduced for treatment of cutaneous T-cell lymphomas \cite{6} and autoimmune diseases \cite{16}, extracorporeal photopheresis (ECP) has been reported to be effective to reverse acute heart, lung, and kidney allograft
rejection episodes, as well as to treat acute and chronic graft-versus-host disease (GVHD) [14]. ECP is performed through a temporary peripheral venous access. It has been speculated that ECP modulates alloreactive T-cell responses by multiple mechanisms: induction of apoptosis, inhibition of antigen-driven T-cell proliferation, and reduced expression of cell surface receptors. To date, the body of evidence supporting the use of ECP in the treatment of solid organ graft rejection stems from experiences with heart, lung, and renal transplant recipients.

In the setting of liver transplantation (LT), acute graft rejection is nearly always amenable to reversal with steroid pulses. Severe graft rejection episodes or those not responding to steroid treatment may be reversed by means of lymphocytolytic antibody therapy. However, the treatment of rejection is not devoid of complications, namely those related to high-dose immunosuppression and steroid pulses. Therefore, the use of ECP for allograft rejection in LT recipients might represent a valid alternative to overcome the side effects associated with current treatment modalities.

In [28], ECP showed no added morbidity or mortality, was well tolerated, and no procedure-related complications were observed. Its efficacy to reverse rejection was clinically remarkable. The use of ECP allowed reduction in immunosuppression in three out of five patients. Noteworthy, ECP was not associated with HCV or HBV flares in the present experience. Such data need long-term confirmation but they may contribute to an expanded application of ECP for patients grafted for viral cirrhosis, thereby helping to reduce the use of steroids in this category of patients. In conclusion, the results in [28] suggest that ECP may represent a valuable alternative to treat graft rejection in selected recipients. Emerging issues awaiting clarification concern indications to ECP, timing and length of treatment, and its cost-effectiveness ratio.

In the case study described in the following, we analyze the dataset collected in [28]. Since prior to [28] only anecdotal reports describe the use of ECP in immunosuppressive protocols for LT recipients, and only one case of a LT recipient treated by ECP has appeared in the international literature [18], we can conclude that this is a unique dataset of this kind.

5.2 From Sequential Patterns to TAS

In this section we summarize the main aspects of the Sequential Pattern Mining (introduced first in [3]), how it has evolved in the TAS paradigm, and how this can be used as a general paradigm for time-annotated data.

Sequential Pattern Mining

Frequent Sequential Pattern mining (FSP) deals with the extraction of frequent sequences of events from datasets of transactions; those, in turn, are time-stamped sequences of events (or sets of events) observed in some
business contexts: customer transactions, patient medical observations, web sessions, trajectories of objects moving among locations.

The frequent sequential pattern (FSP) problem is defined over a database of sequences $D$, where each element of each sequence is a time-stamped set of objects, usually called items. Time-stamps determine the order of elements in the sequence. E.g., a database can contain the sequences of visits of customers to a supermarket, each visit being time-stamped and represented as the set of items bought together. Then, the FSP problem consists in finding all the sequences that are frequent in $D$, i.e., appear as subsequence of a large percentage of sequences of $D$. A sequence $\alpha = \alpha_1 \rightarrow \ldots \rightarrow \alpha_k$ is a subsequence of $\beta = \beta_1 \rightarrow \ldots \rightarrow \beta_m$ ($\alpha \preceq \beta$) if there exist integers $1 \leq i_1 < \ldots < i_k \leq m$ such that $\forall 1 \leq n \leq m \alpha_n \subseteq \beta_{i_n}$. Then we can define the support $\text{sup}(S)$ of a sequence $S$ as the number of transactions $T \in D$ such that $S \preceq T$, and say that $S$ is frequent w.r.t. threshold $\sigma$ is $\text{sup}(S) \geq \sigma$.

Recently, several algorithms were proposed to efficiently mine sequential patterns, among which we mention PrefixSpan [24], that employs an internal representation of the data made of database projections over sequence prefixes, and SPADE [33], a method employing efficient lattice search techniques and simple joins that needs to perform only three passes over the database. Alternative methods have been proposed, which add constraints of different types, such as min-gap, max-gap, max-windows constraints and regular expressions describing a subset of allowed sequences. We refer to [35] for a wider review of the state-of-art on sequential pattern mining.

The $\mathcal{TAS}$ mining paradigm. As one can notice, time in FSP is only considered for the sequentiality that it imposes on events, or used as a basis for user-specified constraints to the purpose of either preprocessing the input data into ordered sequences of (sets of) events, or as a pruning mechanism to shrink the pattern search space and make computation more efficient. In either cases, time is forgotten in the output of FSP. For this reason, the $\mathcal{TAS}$, a form of sequential patterns annotated with temporal information representing typical transition times between the events in a frequent sequence, was introduced in [8].

**Definition 4 ($\mathcal{TAS}$).** Given a set of items $I$, a temporally-annotated sequence of length $n > 0$, called $n$-$\mathcal{TAS}$ or simply $\mathcal{TAS}$, is a couple $T = (\mathfrak{s}, \mathfrak{t})$, where $\mathfrak{s} = \langle s_0, \ldots, s_n \rangle, \forall 0 \leq i \leq n s_i \in 2^I$ is called the sequence, and $\mathfrak{t} = \langle \alpha_1, \ldots, \alpha_n \rangle \in \mathbb{R}^n_+$ is called the (temporal) annotation. $\mathcal{TAS}$ will also be represented as follows:

$$T = (\mathfrak{s}, \mathfrak{t}) = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n$$

**Example 2.** In a weblog context, web pages (or pageviews) represent items and the transition times from a web page to the following one in a user session represent annotations. E.g.:

$$\langle \langle \{/'\} \rangle, \{/'papers.html'\}, \{/'kdd.html'\} \rangle, \langle 2, 90 \rangle \rangle = \{/'\} \xrightarrow{2} \{/'papers.html'\} \xrightarrow{90} \{/'kdd.html'\}$$
represents a sequence of pages that starts from the root, then after 2 seconds continues with page ‘papers.html’ and finally, after 90 seconds ends with page ‘kdd.html’. Notice that in this case all itemsets of the sequence are singletons.

Similarly to traditional sequence pattern mining, it is possible to define a containment relation between annotated sequences:

**Definition 5 (τ-containment (≤_τ)).** Given a n-TAS \( T_1 = (s_1, α_1) \) and a m-TAS \( T_2 = (s_2, α_2) \) with \( n \leq m \), and a time threshold \( τ \), we say that \( T_1 \) is \( τ \)-contained in \( T_2 \), denoted as \( T_1 \leq_τ T_2 \), if and only if there exists a sequence of integers \( 0 \leq i_0 < ... < i_n \leq m \) such that:

1. \( \forall 0 \leq k \leq n \cdot s_{1,k} \subseteq s_{2,i_k} \)
2. \( \forall 1 \leq k \leq n \cdot |α_{1,k} - α_{*,k}| \leq τ \)

where \( \forall 1 \leq k \leq n \cdot α_{*,k} = \sum_{i_{k-1} < j \leq i_k} α_{2,j} \). As special cases, when condition 2 holds with the strict inequality we say that \( T_1 \) is strictly \( τ \)-contained in \( T_2 \), denoted with \( T_1 <_τ T_2 \), and when \( T_1 \leq_τ T_2 \) with \( τ = 0 \) we say that \( T_1 \) is exactly contained in \( T_2 \). Finally, given a set of TAS \( D \), we say that \( T_1 \) is \( τ \)-contained in \( D \) (\( T_1 \leq_τ D \)) if \( T_1 \leq_τ T_2 \) for some \( T_2 \in D \).

Essentially, a TAS \( T_1 \) is \( τ \)-contained into another one, \( T_2 \), if the former is a subsequence of the latter and its transition times do not differ too much from those of its corresponding itemsets in \( T_2 \). In particular, each itemset in \( T_1 \) can be mapped to an itemset in \( T_2 \). When two itemsets are consecutive in \( T_1 \) but their mappings are not consecutive in \( T_2 \), the transition time for the latter couple of itemsets is computed summing up the times of all the transitions between them, which is exactly the definition of annotations \( α_{*} \). The following example describes a sample computation of \( τ \)-containment between two TAS:

\[
T_1: \quad \{a\} \xrightarrow{4} \{b\} \xrightarrow{9} \{c\}
\]

\[
T_2: \quad \{a\} \xrightarrow{3} \{b, d\} \xrightarrow{7} \{f\} \xrightarrow{4} \{c\}
\]

\[
3 + 4 + 7 = 14
\]

**Fig. 2.** Example of \( τ \)-containment computation

**Example 3.** Consider two TAS:

\( T_1 = (\langle \{a\}, \{b\}, \{c\} \rangle, \langle 4, 9 \rangle) \)

\( T_2 = (\langle \{a\}, \{b, d\}, \{f\}, \{c\} \rangle, \langle 3, 7, 4 \rangle) \)

also depicted in Figure[2] and let \( τ = 3 \). Then, in order to check if \( T_1 \leq_τ T_2 \), we verify that:
• $\overline{\tau_1} \subset \overline{\tau_2}$: in fact the first and the last itemsets of $T_1$ are equal, respectively, to the first and the last ones of $T_2$, while the second itemset of $T_1$ ($\{b\}$) is strictly contained in the second one of $T_2$ ($\{b,d\}$).

• The transition times between $T_1$ and its corresponding subsequence in $T_2$ are similar: the first two itemsets of $T_1$ are mapped to contiguous itemsets in $T_2$, so we can directly take their transition time in $T_2$, which is equal to $\alpha_{s,1} = 3$ (from $\{a\} \xrightarrow{3} \{b,d\}$ in $T_2$). The second and third itemsets in $T_1$, instead, are mapped to non-consecutive itemsets in $T_2$, and so the transition time for their mappings must be computed by summing up all the transition times between them, i.e.: $\alpha_{s,2} = 7 + 4 = 11$ (from $\{b,d\} \xrightarrow{7} \{f\} \text{ and } \{f\} \xrightarrow{4} \{c\}$ in $T_2$). Then, we see that $|\alpha_{s,1} - \alpha_{s,2}| = |4 - 3| < \tau$ and $|\alpha_{s,1} - \alpha_{s,2}| = |9 - 11| < \tau$.

Therefore, we have that $T_1 \preceq \tau T_2$. Moreover, since all inequalities hold strictly, we also have $T_1 \prec \tau T_2$.

Now, frequent sequential patterns can be easily extended to the notion of frequent TAS:

**Definition 6 (Frequent TAS).** Given a set $D$ of TAS, a time threshold $\tau$ and a minimum support threshold $\sigma$, we define the $\tau$-support of a TAS $T$ as $\text{supp}_{[\tau,D]}(T) = |\{T^* \in D \mid T \preceq \tau T^*\}|$ and say that $T$ is frequent in $D$, given a minimum support threshold $\sigma$ if $\text{supp}_{[\tau,D]}(T) \geq \sigma$.

It should be noted that a frequent sequence $\vec{s}$ may not correspond to any frequent TAS $T = (\vec{s}, \vec{\alpha})$: indeed, its occurrences in the database could have highly dispersed annotations, thus not allowing any single annotation $\vec{\alpha} \in \mathbb{R}_+^n$ to be close (i.e., similar) enough to a sufficient number of them. That essentially means $\vec{s}$ has no typical transition times.

Now, introducing time in sequential patterns gives rise to a novel issue: intuitively, for any frequent TAS $T = (\vec{s}, \vec{\alpha})$, we can usually find a vector $\vec{\tau}$ of small, strictly positive values such that $T' = (\vec{s}, \vec{\alpha} + \vec{\tau})$ is frequent as well, since they are approximatively contained in the same TAS in the dataset, and then have very similar $\tau$-support. Since any vector with smaller values than $\vec{\tau}$ (e.g., a fraction $\vec{\tau}/n$ of it) would yield the same effect, we have that, in general, the raw set of all frequent TAS is highly redundant (and also not finite, mathematically speaking), due to the existence of several very similar - and then practically equivalent - frequent annotations for the same sequence.

**Example 4.** Given the following toy database of TAS:

\[
\begin{align*}
& a \xrightarrow{1} b \xrightarrow{2.1} c \quad a \xrightarrow{1.1} b \xrightarrow{1.9} c \\
& a \xrightarrow{1.2} b \xrightarrow{2} c \quad a \xrightarrow{0.9} b \xrightarrow{1.9} c
\end{align*}
\]

if $\tau = 0.2$ and $s_{\text{min}} = 0.8$ we see that $T = a \xrightarrow{1} b \xrightarrow{2} c$. In General, we can see that any $a \xrightarrow{\alpha_1} b \xrightarrow{\alpha_2} c$ is frequent whenever $\alpha_1 \in [1, 1.1]$ and $\alpha_2 \in [1.9, 2.1]$.

This problem is solved by the TAS mining software developed in [8], by finding “dense regions” of annotations, and thus grouping together TAS patterns
accordingly. The output of this process is a set of \( \mathcal{ZAS} \) patterns where the annotations are no longer points in \( \mathbb{R}^n_+ \), but instead are intervals: e.g.,

\[
a_{[1,1.1]} \rightarrow b_{[1.9,2.1]} \rightarrow c
\]

For more details see [7, 8].

5.3 Case Study: Mining \( \mathcal{ZAS} \) from Photopheresis Data

In this section we describe the results obtained by applying the \( \mathcal{ZAS} \) paradigm to medical data, resulting from the application of the Photopheresis therapy to patients who had a liver transplant.

Dataset. The dataset under analysis contains information about 127 patients that had a liver transplant and, in order to prevent the allograft rejection, were subjected to the ECP therapy for a period of about 1 year.

Each of them has from few to about 40 observations along the therapy. For each observation 38 different continuous variables are recorded, including information on Red Blood Cells (RBC), Hemoglobin (HB), Hematocrit (HCT), Platelet (PLT), White Blood Cell (WBC), Neutrophils, Monocytes, Lymphocytes, Cluster Of Differentiation 3, 4, 8, 16, 19, 25, 40, Interferon Gamma (INF Gamma), Interleukins 2, 4, 5, 10, 12, Tumor Necrosis Factor (TNF Alfa), Intercellular Adhesion Molecule (ICAM-1), Human Leukocyte Antigen (HLA), Aspartate Aminotransferase (ASAT or GOT, Glutamic Oxalacetic Transferase), Alanine Aminotransferase (ALAT or GPT, Glutamic Pyruvic Transaminase), Gamma Glutamyl Transpeptidase (GGT), Alkaline Phosphatase, Lactate DeHydrogenase (LDH), Creatine PhosphoKinase (CPK), Bilirubin, Serum bilirubin, Human C Virus (HCV), and Prothrombin Time (PT). Unfortunately, the data contains many missing values, thus making many observations, and patients, useless for the analysis objectives.

The dataset resulting from the removal of incomplete or useless data, contains complete information for 50 patients. The dataset also exhibits very peculiar characteristics, such as extremely diverse ranges for each variable for each patient, and non-standard distributions of the values.

For the above reasons, we decided to follow three different approaches, that needed three different kinds of preprocessing, and lead to three different types of results, that are shown below. Obviously, before being able to apply \( \mathcal{ZAS} \) mining to this dataset, each continuous variable must be discretized in order to produce the vocabulary of items. For instance, an item could be “Interleukin 12 \( \in [25,50] \)”. For each analysis we received a feedback from the physicians, who suggested how to proceed to the subsequent stages, by moving the focus from some interesting patterns to others, by including other variables and so on. Tables 3, 4 and 5 show some of the results obtained for the approaches 1, 2 and 3, respectively.
In each table we have 7 columns: in the first one we have the ID of the TA found; in the second column we find the values of the involved variables, encoded and discretized for being processed by the TA miner; columns 3 and 4 show the support of the TA found, both in percentage (0.0 to 1.0) and as an absolute value (number of patients presenting the pattern); in column 5 and 6 we have the lower and upper coordinates, respectively, of the pattern (i.e., the minimal and the maximal elapsed time along the sequence); in the last column we have the density of the pattern (i.e., the minimal number of occurrences of the pattern).

Each TA found can have several time annotations, corresponding to the same sequence found with different frequent time-stamps. For example, the TA n.1 in Table 3 should be read (according to our encoding and discretization of the values) this way: an increment by $10^3 - 10^5$ times of the interleukin 4 was followed by a decrement by at least 85% of the same variable, after 13 to 14 days (according to the first frequent temporal annotation) or after 1 to 2 days (according to the second frequent temporal annotation). TA n.6 in Table 3 contains a sequence of length 3 and should be read: a quite stable (around 105%) value of the interleukin 5 followed by (almost) the same value after 91 to 93 or 0 to 2 days, followed again by (almost) the same value after 0 to 1 day or 35 to 36 days.

First Analysis

Data preparation. In the first analysis, for each variable and for each observation, we focus on the variation w.r.t. the value of the same variable in the previous observation. This has been done with the hope of finding association patterns of “trends”: for instance a small decrement of variable $V_1$ appearing together with a strong increment in variable $V_2$ is frequently followed after a week by a strong increment of variable $V_3$ with $V_1$ staying almost constant.

For this first analysis we only considered the values of the interleukins (2, 4, 5, 10, 12), on explicit request of the physicians.

Results. Table 3 shows some of the results obtained by means of the approach 1. As one can see, the focus in the first approach was to find some interesting patterns involving the interleukins: in this way we had a first feedback for the adequateness of the TA paradigm. As an example, in TA n.1, we found an interesting sequence of increase and decrease of the same variable in 70% of the patients, which is quite a large support. The pattern was supported with two different frequent temporal annotations: the first one finding the elapsed time around 14 days, and the second one very early in the therapy (1 to 2 days after the beginning).

Unfortunately, these patterns were not enough meaningful for the physicians, because the patterns do not say anything about the initial values of the variables, nor about the exact day when the increase or decrease happened. For these two reasons, we decided to change the focus as done in the second analysis.
Table 3. Some of the results of the first analysis

<table>
<thead>
<tr>
<th>ID</th>
<th>Pattern</th>
<th>Support (%)</th>
<th>Support (abs)</th>
<th>Interval</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(100000IL4) (15IL4)</td>
<td>70</td>
<td>33</td>
<td>13 14</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>(15IL4) (100000IL4)</td>
<td>70</td>
<td>33</td>
<td>7 8</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>(15IL4) (105IL5)</td>
<td>61</td>
<td>29</td>
<td>8 9</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>(105IL5) (105IL5)</td>
<td>63</td>
<td>30</td>
<td>21 22</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>(100000IL4) (15IL4 45IL10)</td>
<td>17</td>
<td>8</td>
<td>15 16</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>(105IL5) (105IL5) (105IL5)</td>
<td>42</td>
<td>20</td>
<td>91 0 93 1</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>(105IL5 60IL10) (100000IL4)</td>
<td>31</td>
<td>15</td>
<td>59 60</td>
<td>5</td>
</tr>
</tbody>
</table>

Second Analysis

Data preparation. In the second analysis we focused, for each variable, on its variation w.r.t. the clinical “normal” value, without any reference to the variation w.r.t. the previous observation. In order to keep track not only of the elapsed time among the observations, we added within all the observations (itemsets) also the discretized information about the day when the observation was taken. The discretization steps for the date follow the ECP therapy milestones: i.e., 7, 14, 30, 90, 180, 270 and 365 days. In this analysis, we considered also other variables: GOT, GPT, GGT, Alkaline Phosphatase, HCV and Bilirubin.

Results. Table 4 reports some of the patterns extracted during the second analysis. Adding more variables and changing the items meaning, we obtained more detailed and meaningful results. For example, the ZAS n.1 in Table 4 shows a very low value of the interleukin 12, followed by an even lower value, after 7 to 10 days. However, the main limit of this analysis is the lack of focus on interesting, surprising patterns, due to the strong prevalence of “normal” values that hide all the other patterns. Based on this consideration we moved to another analysis.

Third Analysis

Data preparation. In the third analysis, with the aim of making interesting patterns arise, we changed the discretization by creating for each variable 7 (almost) equally populated bins. Moreover, after applying a couple of runs of the ZAS miner, we also decided to remove the classes with the values in the most normal ranges, in order to make the unusual values come out more easily.

Results. Table 5 shows some of the results obtained during the third analysis. With this approach, we divided the values for each variable in several classes, putting (almost) the same amount of occurrences per class. We also removed most of the normality range values to get more interesting patterns.
Table 4. Some of the results of the second analysis

<table>
<thead>
<tr>
<th>ID</th>
<th>Pattern</th>
<th>Support (%)</th>
<th>Support (abs)</th>
<th>Interval</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(50IL12) (25IL12)</td>
<td>71</td>
<td>35</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>(50IL12) (1000IL10)</td>
<td>61</td>
<td>30</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>(50IL12) (50IL12)</td>
<td>77</td>
<td>38</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>(50IL12) (1000IL4)</td>
<td>69</td>
<td>34</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>(1000BIL) (1000IL10)</td>
<td>42</td>
<td>21</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>(1000BIL) (50IL12)</td>
<td>44</td>
<td>22</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>(225AP) (50IL12)</td>
<td>57</td>
<td>28</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>(300AP) (25IL12)</td>
<td>51</td>
<td>25</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>(25IL12) (1000AP)</td>
<td>26</td>
<td>13</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>(50IL12) (225AP)</td>
<td>53</td>
<td>26</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>(25IL12) (1000AP)</td>
<td>26</td>
<td>13</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>(50IL12) (225AP)</td>
<td>53</td>
<td>26</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>(1000IL10) (1000AP)</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

As also done in the second analysis, we also put the date information itself as a variable (DD). In this way we obtained very meaningful patterns, showing interesting values of the variables and the date when the values appeared in the medical observations. For example, Table 4 showed unusual values of the interleukins. During the pattern evaluation, the physicians guessed that this unusual behaviour could represent an early warning for allograft rejection. Checking the patients that support the two patterns, we found out that physicians’ guess was actually correct.

Table 5. Some results of the third analysis

<table>
<thead>
<tr>
<th>ID</th>
<th>Pattern</th>
<th>Supp. (%)</th>
<th>Supp. (abs)</th>
<th>Interval</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(7DD 113-337GPT) (29-45IL10)</td>
<td>38</td>
<td>19</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>(45-672,5IL10) (11,8-19,7IL10 90DD)</td>
<td>18</td>
<td>9</td>
<td>36</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>(0-30IL12) (94-131IL12 90DD)</td>
<td>18</td>
<td>9</td>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>(25,7-54,6IL4) (1-4,2IL4 90DD)</td>
<td>22</td>
<td>11</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>(7DD 4,5-41,8BIL) (0,1-1IL4)</td>
<td>38</td>
<td>19</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>(4,5-41,8BIL) (14DD 0,1-1IL4)</td>
<td>18</td>
<td>9</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>(25,7-54,6IL4) (1,1-1,96BIL 90DD)</td>
<td>32</td>
<td>16</td>
<td>36</td>
<td>37</td>
</tr>
<tr>
<td>8</td>
<td>(0,1-1IL4) (1,1-1,96BIL 90DD)</td>
<td>26</td>
<td>13</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>(7DD 4,5-41,8BIL) (14DD 0,1-1IL4)</td>
<td>18</td>
<td>9</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
This third analysis raised much interest in the physicians and the biologists. In the future analyses we will take in consideration more variables, and hopefully the physicians will provide us with a larger number of patients. We are also studying how to cross-over these patterns with other models of extracted knowledge regarding information that does not change with time, e.g., some particular genetic patterns of the patients. One possible objective is to develop a prediction model able to rise early warnings for patients for whom the ECP therapy is not working adequately, and thus could result in allograft rejection.

6 Conclusions and Future Work

In this chapter we have described two pattern discovery analyses that we conducted on clinical data of patients in the follow-up of a liver transplantation. The two pattern discovery analyses used different techniques for different objectives.

The first pattern discovery analysis was conducted on genetical data of patients with serious liver diseases, with the aim of discovering associations holding between some genetic configuration and the arising of some serious form hepatic cirrhosis. In particular we have focussed on the HLA genetic region, since it is supposed to play an important role in both viral and autoimmune hepatic cirrhosis. For this kind of analysis, associative frequent patterns where the adopted technique.

The aim of the second data analysis was to assess the effectiveness of the extracorporeal photopheresis as a therapy to prevent rejection in solid organ transplantation. To this aim the discovery of associative frequent patterns, describing trends of different biochemical variables along the time dimension is a key step.

Both analyses lead to the discovery of some patterns already known by the physicians, and other novel ones: the feedback we received by the physicians and biologists was very positive. However, the main contribution of these analyses is methodological: they proved adequateness of pattern discovery techniques in medical data mining, and they described a general methodology that can be replicated in other analyses in the medical domain, whenever the data exhibits similar characteristics. In our forthcoming analyses we are going to apply various pattern discovery techniques to different datasets describing different aspects related to solid organ transplantation. The common ambitious goal is to gain deeper insights in all the phenomena related to solid organ transplantation, with the aim of improving the donor-recipient matching policy used nowadays. Among these analyses, one that is felt particularly promising by the physicians, is aimed at extracting both qualitative and quantitative knowledge about the phenomenon of the chimerism in the blood of the recipient in the immediate follow-up of the transplantation. Also in this case, the data is formed by a set of biochemical variables recorded at different time moments, and the typical time elapsing between two relevant
events is a very important piece of knowledge. Thus, there are all the conditions to apply again the TASM mining paradigm. The patterns extracted then can be used as basic bricks to build more complex models, for instance by enriching them with genetical information. Other analyses we plan to perform by means of pattern discovery techniques, are:

- analyze the polymorphism of the major histocompatibility complex genes in the donor-recipient pairs;
- analyze pairs the polymorphism of the minor histocompatibility complex genes in the donor-recipient;
- assess the influence of the genetic polymorphism of the IL-10 gene [17];
- analyze the genetic polymorphism of genes associated with the chronic reject (ICAM-1, VEGF) [15].

Acknowledgments

We wish to thank the biologists and physicians of U.O. Immunoematologia, Azienda Ospedaliero-Universitaria Pisana, for providing us the data and the mining problem described in this section. In particular we are indebted with Irene Bianco, Michele Curcio, Alessandro Mazzoni and Fabrizio Scatena. We must also thank Mirco Nanni and Fabio Pinelli for the TASM mining software. This collaboration between physicians, biologists and computer scientist is carried out within the project “La Miniera della Salute”, supported by “Fondazione Cassa di Risparmio di Pisa”.

References


Substructure Analysis of Metabolic Pathways by Graph-Based Relational Learning

Chang hun You, Lawrence B. Holder, and Diane J. Cook

School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164-2752
{changhun,holder,cook}@eecs.wsu.edu

Systems biology has become a major field of post-genomic bioinformatics research. A biological network containing various objects and their relationships is a fundamental way to represent a bio-system. A graph consisting of vertices and edges between these vertices is a natural data structure to represent biological networks. Substructure analysis of metabolic pathways by graph-based relational learning provides us biologically meaningful substructures for system-level understanding of organisms.

This chapter presents a graph representation of metabolic pathways to describe all features of metabolic pathways and describes the application of graph-based relational learning for structure analysis on metabolic pathways in both supervised and unsupervised scenarios. We show that the learned substructures can not only distinguish between two kinds of biological networks and generate hierarchical clusters for better understanding of them, but also have important biological meaning.

1 Introduction

A biological organism has one ultimate goal: to continue the life of its species and itself in the natural environment. This goal requires two important activities, maintaining low entropy in the environment and reproducing oneself. Biological organisms need to process various functions to maximize free energy and minimize entropy. These two basic processes are preformed by a variety of interactions in an organism.

Fundamentally the organism is a system itself as well as a property of a biotic ecosystem. A biological organism is not just composed of various objects, but also has dynamic and interactive relationships among them. A system-level understanding is a more efficient way to approach the organisms. With advances in computer science, bioinformatics plays a central role in life science. Traditional bioinformatics has been focused on molecular-level research. Genomics and proteomics, main areas in molecular-level research, have studied function and structure of macromolecules in organisms, and produced a huge amount of results. However almost every biomolecule plays its role only in
harmony with other components of the cytoplasmic environment. Molecular-level understanding is definitely a fundamental step, but it is not the final step. It is time to steer our steps to system-level approaches to bio-systems.

A biological network is a fundamental and indispensable way to describe a complex system in terms of both the structure and its dynamics. The final goal of systems biology is to model and simulate the biological networks for better understanding of bio-systems and contribution for drug discovery. An efficient way to model unrecognized biological networks is to discover patterns in existing biological networks. Biological networks contain various biomolecules and their relationships. The patterns of relationships in biological networks are crucial to understanding organisms and to modeling them at the system-level. Structure analysis of biological networks is a primary movement in systems biology.

Logic-based and graph-based approaches, as subfields of multi-relational data mining, are applied to mine patterns in biological networks. Logic-based data mining, also called inductive logic programming, represents networks using logic [22]. But this approach requires complicated syntax and the definition of prior knowledge. A graph has been widely used to represent a variety of relational data such as computer networks, social networks, and biological data. A biological network is another appropriate domain to be represented as a graph. Graph-based relational learning can be applied to find the meaningful patterns in the biological network that is represented as a graph.

In this paper, we review systems biology and some multi-relational data mining approaches applied to biological networks, and we describe the knowledge discovery approach used in the SUBDUE graph-based relational learning system. We then show the application of SUBDUE to metabolic pathways, which are downloaded from the KEGG PATHWAY database and represented as a graph. The goal of this research is to show that the learned substructures describe the system-level features in metabolic pathways and convey important biological meaning. Structure analysis on the same metabolic pathways from different species finds the substructure showing the unique features for specific species. Supervised learning shows that the learned substructures can identify what is unique about a specific type of pathway, which allows us to understand better how pathways differ. Unsupervised learning produces hierarchical clusters that describe what is common about a specific group of pathways, which provides us better understanding of common structure in pathways. Ultimately, these substructures can improve both our ability to understand recognized pathways and categorize unrecognized ones.

2 Systems Biology and Biological Networks

Systems biology is a novel stream of life science focusing on a comprehensive bio-system including integrated and interacting biological networks which are relations of genes, proteins and other biomolecules [15]. A system should be studied in a system-level manner including using comprehensive
Fundamentally an organism is a system itself as well as a property of a bio-ecosystem. The organism has a systematically well-organized structure consisting of multi-level compositions such as tissue, organ and organ system, all of which are based on a cell as a functional and structural basic unit. Each constituent is cooperating with others interactively as well as organized systematically. Even the basic unit, the cell, is also a system itself. A cell has a variety of biomolecules that are working with interactive relationships among them.

A huge amount of biological data has been generated by long-term research. Each result cannot allow us to understand a whole biological system, because any molecule or constituent in the organism never works alone. They always interact with others in the system. For this reason an organism should be explored as a system.

A biological system can be described as a large biological network, which consists of numerous small networks. Each network has various biomolecules and their relationships. Generally a cellular system is represented by three kinds of biological networks: metabolic pathways, protein-protein interactions and gene regulatory networks. Our research is currently focused on the metabolic pathways.

**Fig. 1.** TCA cycle biological network of *Homo Sapiens* [20]
Figure 1 shows a metabolic pathway called the TCA cycle which is a metabolic pathway for the production of ATP (a fundamental energy molecule in a cell). A rectangle represents an enzyme (protein) or a gene, and a circle represents a chemical compound. Each arrow describes a relationship between these molecules. In marked area (A), a compound, (S)-Malate (L-Malic acid), as a substrate is changed to another compound, Fumarate, as a product by an enzyme, ec:4.2.1.2. This is a basic biochemical reaction. The metabolic pathway is a complex network of biochemical reactions. A fundamental step to study metabolic pathways is the identification of structures covering a variety of biomolecules and their relationships. Dynamics and control methods of metabolic pathways are also included, because biological systems are interactive and well-controlled optimized systems [15].

Our current research is focused on identifying the structure. Our ultimate goal is to make a blueprint for system-level understanding and its application based on an understanding of the structure, dynamics and control of metabolic pathways.

The KEGG PATHWAY is a widely known database which contains information on various kinds of pathways including pathway image files (figure 1) [14]. The KEGG PATHWAY database has 84,441 pathways generated from 344 reference pathways (on December, 2008). It has six fundamental categories of pathways: Metabolism, Genetic information processing, Environmental information processing, Cellular processes and human diseases and Drug development. This database contains not only various information on pathways, but also plentiful information of their components as linked databases. It also has the KGML (KEGG Markup Language) as an exchange format for KEGG pathways, based on XML.

3 Related Work

Biological networks consist of various molecules and their relationships. Each molecule can have its own properties and can also influence relationships with other molecules. For this reason, traditional data mining, focusing only on the properties, is not applicable to biological networks. Multi-relational data mining is focused not only on properties of molecules but also their relationships [7]. To apply multi-relational data mining, it is necessary to represent the data along with its multiple relations. First-order logics and graph representations are used for representation of multi-relational data. These representation methods lead to two general approaches of multi-relational data mining: logic-based data mining and graph-based data mining.

In this section, we introduce several approaches to analyze biological networks. Then, we will introduce the multi-relational data mining approach as a method to analyze biological networks including inductive logic programming as a logic-based data mining method and two categories of graph-based data mining: graph-based relational learning and frequent subgraph mining.
Graph-based relational learning, as our main focus, will be described in section 4.

3.1 Analysis of Biological Networks

Aittokallio and Schwikowski survey recent works of graph-based methods for analysis of biological networks [1]. They categorize graph-based approaches into three levels: global structural property, local structural connectivity and hierarchical functional description. They also introduced some recent approaches including integrating data from multiple source, graph-based software tools for network analysis and network reconstruction by reverse engineering.

Cheng et al. [3] show an approach of mining bridges and motifs from biological networks. They use a statistical method to detect structural bridges and motifs, and compare their bridges and motifs with functional units in the biological networks. They suggest the structures of the discovered bridges and motifs are significantly related to their function. Hwang et al. [13] introduce an approach of detecting novel functional patterns in protein-protein interaction networks using graph properties and signal transduction behavior. They model the dynamic relationships between two proteins using a signal transduction model to represent functional similarity between proteins. Huan et al. [12] try to discover spatial patterns in protein structures using the subgraph mining approach. They use Delaunay tessellation to represent three-dimensional structure of proteins, where Delaunay tessellation is defined based on the coordinates of molecules in the proteins. If two Voronoi cells share a common face, two points are connected by an edge. In this way, they represent the protein structure as a graph, and apply the frequent subgraph mining approach.

Mathematical modeling abstractly describes a system using mathematical formulae [20, 27]. Most of these approaches, as a type of quantitative analysis, model the kinetics of pathways and analyze the trends in the amounts of molecules and the flux of biochemical reactions. But most of them disregard relations among multiple molecules.

Our research applies graph-based data mining to learn the patterns in the dynamics of biological networks [29, 30]. We introduce a dynamic graph containing a sequence of graphs that represent the time-based versions of biological networks to represent the structural changes of biological networks. Our approach first discovers graph rewriting rules between two sequential graphs to represent how one network is changed to another, and then discover more general transformation rules in the set of the graph rewriting rules to describe how biological networks change over time.

There are many computational approaches to analyze biological networks. Biosystems are organized as networks and perform their function in relations to molecules and systems. For this reason, we need to focus on the structural properties of biological networks rather than the molecules themselves.
3.2 Logic-Based Data Mining

Logic-based data mining is a multi-relational data mining technique using first order logic to represent data. Inductive Logic Programming (ILP), a typical logic-based data mining technique, is an approach to induce hypotheses (rules, concepts, or knowledge) from observations (examples, instances, or experiences) by using logic to represent hypotheses and observations [7]. ILP has been applied in several ways to biological domains such as genomics and proteomics. ILP in company with other approaches has been applied to metabolic pathways.

Support Vector Inductive Logic Programming (SVILP) is the intersection approach between Support Vector Machines (SVM) and ILP. By using logic to represent background knowledge, the SVILP technique has been applied to the prediction of toxicity of given materials [23]. Stochastic methods are also applied to logic programming to model metabolic pathways. This approach models rates of biochemical reactions using probabilities in addition to representation of metabolic pathways by logic [21]. A last approach is a cooperation between induction and abduction. This approach generates hypotheses from not only abductive reasoning from experimental data (concentrations of metabolites), but also inductive reasoning from general rules (from the KEGG database) to model inhibition in metabolic pathways [25]. An inhibitor is a chemical compound to control biochemical reactions.

Several ILP-related techniques have been successfully applied to metabolic pathways. Logic programming can efficiently represent relational data, but prior rules and examples may be necessary to represent entire pathways.

3.3 Frequent Subgraph Mining

The graph is an abstract data structure consisting of vertices and edges which are relationships between vertices. Graph-based data mining denotes a collection of algorithms for mining the relational aspects of data represented as a graph. Graph-based data mining has two major approaches: frequent subgraph mining and graph-based relational learning. Frequent subgraph mining is focused on finding frequent subgraphs in a graph. There are two well-known approaches to bioinformatics domains. Frequent SubGraph discovery, FSG, finds all connected subgraphs that appear frequently in a set of graphs. FSG starts by finding all frequent single and double edge graphs. During each iteration FSG expands the size of frequent subgraphs by adding one edge to generate candidate subgraphs. Then, it evaluates and prunes discovered subgraphs with user-defined constraints [19].

Graph-based Substructure Pattern Mining, gSpan, uses the depth-first search and lexicographic ordering. First gSpan sorts the labels, removes infrequent vertices and edges, and it relabels the remaining vertices and edges. Next it starts to find all frequent one-edge subgraphs. The labels on these edges and vertices define a code for each graph. Larger subgraphs map themselves to longer codes. If the code of B is longer than A, the B code is a
child of the A code in a code tree. If there are two not-unique codes in the tree, one of them is removed during the depth-first search traversal to reduce the cost of matching frequent subgraphs [28]. There are a few approaches of frequent subgraph mining applied to metabolic pathways. Pathway Miner, a simplified graph-mining approach on metabolic pathways, proposes a simplified graph representation consisting of just enzyme relationships. In this way, the approach may avoid the NP-hard subgraph isomorphism problem and find frequent patterns quickly [17]. However, the over-simplified representation makes this approach focus on just enzyme relationships, missing some important features of metabolic pathways.

Mining coherent dense subgraphs uses the correlation of graphs which represent gene regulatory networks [11]. This approach compresses a group of graphs into two meta-graphs using correlated occurrences of edges for efficient clustering. This approach also loses some biological characteristics of gene regulatory networks, because its representation of a graph is derived only from the similarity of the gene expression patterns between two genes, not representing how the practical biological interactions exist.

## 4 Graph-Based Relational Learning

Graph-based relational learning is focused on novel and meaningful, but not necessarily most frequent, substructures in a graph representation of data. We use the SUBDUE graph-based relational learning approach to discover patterns which not only abstract instances of the patterns by compression, but also provide better understanding of the data [5]. SUBDUE can perform unsupervised learning and supervised learning by substructure discovery based on Minimum Description Length (MDL). Using background knowledge given as predefined substructures can guide graph-based relational learning to find more meaningful substructures. SUBDUE has been applied to a variety of areas such as Chemical Toxicity [4], Molecular Biology [24], Security [9] and Web Search [6].

SUBDUE accepts input data which is represented as a graph with labeled vertices and labeled, directed or undirected edges between vertices. The objects and attribute values of the data are usually represented as vertices, while attributes and relationships between objects are represented as edges. Figure 3 shows a graph representation of a KEGG biological network. There are five ‘Entry’ vertices which represents Enzyme or Compound. Each Entry has two attributes: name and type. Relationships are given as directed and labeled edges from Entry to its attributes. More detail on this graph representation will be provided later.

### 4.1 Discovery Algorithm

The SUBDUE discovery algorithm is based on a beam search as shown in figure 2. The algorithm starts with three parameters: input graph, beam length
Subdue\((graph G, beam B, limit L)\)
\[Q = \{v \mid v \text{ is a vertex in } G \text{ with a unique label}\}\]
\(bestSub = \text{first substructure in } Q\)
repeat
   foreach substructure \(S \in Q\) do
      add \(\text{Extend}(S)\) into \(\text{extSubs}\)
      foreach \(\text{newSub} \in \text{extSubs}\) do
         \(\text{Evaluate}(\text{newSub})\)
         add \(\text{newSub}\) in \(Q'\) (length < \(B\))
      if \(bestSub \in Q'\) better than \(bestSub\) then
         \(bestSub = bestSub \in Q'\)
      \(Q = Q'\)
   until \(Q\) is empty or \(\text{Num.OfSubs.Extended} > L\)
return \(bestSub\)

\(\text{Extend}(S)\): extend Sub. \(S\) by one edge in all possible ways
\(\text{Evaluate}(S)\): evaluate Sub. \(S\) using MDL

Fig. 2. SUBDUE’s discovery algorithm

and limit value. The beam length limits the length of the queue which contains extended substructures, and the limit value restricts the total number of substructures considered by the algorithm. The initial state of the search is the set of substructures representing each uniquely labeled vertex and their instances. The \(\text{Extend}(S)\) function extends each instance of a substructure in the \(Q\) in all possible ways by adding a single edge and a vertex, or by adding a single edge if both vertices are already in the substructure. The substructures in the \(Q'\) are ordered based on their ability to compress the input graph as evaluated by \(\text{Evaluate}(S)\), using the minimum description length (MDL) principle. This search (repeat loop) terminates when the number of substructures considered reaches the limit value, or the algorithm exhausts the search space. Then it returns the best substructure.

SUBDUE can be given background knowledge in the form of predefined substructures. SUBDUE finds the instances of these substructures and compresses them. Using this approach, we can verify whether the patterns learned from a graph belong to another graph.

4.2 MDL and Heuristic Methods

The discovery algorithm of SUBDUE fundamentally is guided by the minimum description length principle. The heuristic evaluation by the MDL principle assumes that the best substructure is the one that minimizes the description length of the input graph when compressed by the substructure \(S\). The description length of the substructure \(S\) is represented by \(DL(S)\), the description length of the input graph is \(DL(G)\), and the description length
of the input graph after compression is $DL(G|S)$. SUBDUE’s discovery algorithm tries to minimize $DL(S) + DL(G|S)$ which represents the description length of the graph $G$ given the substructure $S$. The compression of the graph can be calculated as

$$Compression = \frac{DL(S) + DL(G|S)}{DL(G)}$$

where description length $DL()$ is calculated as the number of bits in a minimal encoding of the graph. Cook and Holder describe the detailed computation of $DL(G)$ in [5].

The discovery algorithm of SUBDUE is computationally expensive as other graph-related algorithms. SUBDUE uses two heuristic constraints to maintain polynomial running time: Beam and Limit. Beam constrains the number of substructures by limiting the length of the $Q'$ in figure 2. Limit is a user-defined bound on the number of substructures considered by the algorithm.

4.3 Unsupervised Learning

Once the best structure is discovered, the graph can be compressed using the best substructure. The compression procedure replaces all instances of the best substructure in the input graph with a pointer, a single vertex, to the discovered best substructure. The discovery algorithm can be repeated on this compressed graph for multiple iterations until the graph cannot be compressed any more or on reaching the user-defined number of iterations. Each iteration generates a node in a hierarchical, conceptual clustering of the input data. On the $i$th iteration, the best substructure $S_i$ is used to compress the input graph, introducing a new vertex labeled $S_i$ to the next iteration. Consequently, any subsequently discovered subgraph $S_j$ can be defined in terms of one or more $S_i$, where $i < j$. The result is a lattice, where each cluster can be defined in terms of more than one parent subgraph.

4.4 Supervised Learning

The SUBDUE discovery algorithm has been extended to perform graph-based relational concept learning, or supervised learning [8]. The main approach of supervised learning is to find a substructure that appears often in the positive examples, but not in the negative examples. The substructure value is increased when positive examples are covered by the substructure, but is decreased where negative examples are covered. Positive examples not covered by the substructure and negative examples covered by the substructure are considered errors. The substructure value is calculated by

$$value = 1 - \text{error}$$

where the error is calculated by
\[
\text{error} = \frac{\#\text{PosEgsNotCvd} + \#\text{NegEgsCvd}}{\#\text{PosEgs} + \#\text{NegEgs}}
\]

\#\text{PosEgsNotCvd} is the number of positive examples not containing the substructure, and \#\text{NegEgsCvd} is the number of negative examples containing the substructure. \#\text{PosEgs} is the number of positive examples remaining in the experimental set, of which the positive examples that have already been covered in a previous iteration were removed, and \#\text{NegEgs} is the total number of negative examples, which is constant, because negative examples are not removed.

SUBDUE’s supervised learning uses two approaches to minimize error. First, by using the definition of description length SUBDUE tries to find a substructure \(S\) minimizing \(DL(G^+|S) + DL(S) + DL(G^-) - DL(G^-|S)\), where the last two terms represent the incorrectly compressed negative example graph. This approach will lead the discovery algorithm toward a larger substructure that characterizes the positive examples, but not the negative examples.

In addition to the compression-based evaluation, SUBDUE can use a set-cover approach based on the error measure. At each iteration SUBDUE adds a new substructure to the disjunctive hypothesis and removes covered positive examples. This process continues until either all positive examples are covered or no substructure exists discriminating the remain positive examples from the negative examples.

5 Substructure Analysis in Metabolic Pathways

Our goal is the application of the SUBDUE graph-based relational learning system to the KEGG metabolic pathways to find better understanding and biologically meaningful substructures. These substructures can distinguish two pathways or provide the common features in several pathways. Research shows that topological features of biological networks are closely related to biological functions [13, 2].

A simple way to apply supervised or unsupervised learning to the pathways is based on molecules, such as genes, proteins and other macro molecules. Because each molecule has a specific structure and other biochemical features, we can easily distinguish two groups or find the common features in a group. But our research is focused on the pattern of the relationship between molecules for system-level understanding of pathways. The pattern of relationship can be shown in a variety of forms, such as biochemical reaction, enzyme activity and signal transduction.

This section first introduces our graph representation (section 5.1). As a preliminary task, we describe substructure analysis on individual metabolic pathways (section 5.2). Then we represent our main experiments in this research: supervised learning (section 5.3) and unsupervised learning (section 5.4) on groups of metabolic pathways. The ultimate goal of our exploration
is to show that the substructures found by graph-based relational learning are biologically important and meaningful.

5.1 Graph Representation

Input graphs for SUBDUE are converted from KGML files. KGML is a standard data format to express and distribute a biological network from KEGG. There are three major entities in KGML: Entry, Relation and Reaction. Entry represents various biomolecules in the metabolic pathway, such as enzyme, gene, compound and so on. Relation denotes a relationship between two or more enzymes, genes and maps. The maps denote the types of the Entry nodes linked to the other pathways [26]. The names of these Entry nodes represent the name of the linked pathways. Reaction is a biochemical reaction between two or more compounds catalyzed by one or more enzymes. Detailed information on KGML is described in [26]. In biochemical semantics, Entries are nodes of metabolic pathways, and Relations and Reactions are relationships between two or more Entries.

Fig. 3. A graph representation of a metabolic pathway

In our graph representation, Relations and Reactions are also represented as vertices in order to describe the properties of Relations and Reactions. Vertices representing major entities have two satellite vertices which are connected to their main vertex by edges, labeled as Name and Type, to explain its property. A name vertex linked by the Name edge denotes the KEGG ID, and a type vertex linked by the Type edge describes the property of the entity vertex. A Relation represents the association between two or more Entries (genes or enzymes) by an edge whose label represents a direction from one
Entry to another. Reaction also has connections with two compounds (described as a substrate and a product) and an enzyme (as a catalyst). Figure 3 shows an example of our graph representation, which has five Entries: three compounds and two enzymes. There is also a Relation between two enzymes, and two Reactions sharing a compound and having relationships with two other compounds.

Our research uses two kinds of graph representation: named and unnamed graph. The graph in figure 3 is the named graph, which includes the KEGG IDs. The unnamed graph is the same graph only excluding any unique IDs of each Entry and Reaction; all vertices and edges regarding “name” are removed from the graph in figure 3.

Unique IDs can pose potential problems when SUBDUE searches for substructures that distinguish two groups of graphs, which contain a group of pathways as positive examples and another group of pathways as negative examples. For example, when we try to distinguish two metabolic pathways $G_1$ and $G_2$, an enzyme which exists only in $G_1$, but not in $G_2$, is sufficient to distinguish. Because our research is focused on the pattern of metabolic pathways, a specific name is not useful for finding the pattern. For this reason, our supervised (section 5.3) and unsupervised (section 5.4) learning on groups of pathways use the unnamed graphs (the first phase). The named graphs are used at the second phase to verify the biological meaning of the patterns. The following sections describe this process in more detail.

5.2 Substructure in Metabolic Pathway

This section shows a brief example substructure analysis on metabolic pathways. Here we have two metabolic pathways. SUBDUE tries to find the substructures that exist in one pathway, but not in another. In this experiment, we show that the species-specificity is related to the structure of metabolic pathways. Species-specificity is one of the most important concepts in biology. Basically species-specificity is derived from protein structure and gene sequence.

We arrange two glycolysis pathways from Human and E.coli: hsa00010 and eco00010. Glycolysis is a representative energy generating process in almost every cell. We seek a slightly structural difference between these pathways from two species. A named graph of hsa00010 has 1047 vertices and 1208 edges, and the one of eco00010 has 1002 vertices and 1132 edges. The former is a positive example and the latter is a negative example for supervised learning of SUBDUE. Then we run SUBUDE to find substructures that exist in hsa00010, but not in eco00010.

As a result, the best substructure is found as shown in figure 4. While this substructure is discovered in eco00010 as well as hsa00010, it is still possible to show how the two pathways differ. The instances of the best substructure in figure 4 are found 10 times in hsa00010 and 6 times in eco00010. After
inspecting locations of these instances on each pathway, we can identify structural differences between two pathways.

Five cases out of 10 instances in hsa00010 are also found in eco00010. The other five instances are found only in hsa00010. These five instances show unique parts exist only in hsa00010. Figure 5 shows an example of one of these five instances. Figure 5 (A) and (B) show the part of the glycolysis pathways in human and E.coli respectively. Connected circled entities denote the instance of the best substructure. The marked instance includes a reaction (R00710) catalyzed by an enzyme (ec:1.2.1.5) and three relations related with the enzyme. The reaction R00710 exists only in hsa00010, but not in eco00010. We can confirm this concept in figure 5. Rectangles and circles denote enzymes and compounds. There is no symbol of relation in this figure. Grayed rectangles represent existing enzymes in the species and white rectangles denote only conceptual views (practically do not exist). As shown in this figure eco00010 does not include ec:1.2.1.5 and ec.1.1.1.2, which exist in hsa00010. The process of Acetaldehyde (NAD+ oxidoreductase) does not exist in eco00010. SUBDUE discovers this substructure existing in hsa00010, but not in eco00010. In this way structure analysis by SUBDUE can characterize the unique features of metabolic pathways.

We execute the same method on other experiment sets: 00020 (TCA cycle), 00051 (Fructose and mannose metabolism), 00061 (Fatty acid biosynthesis) and 00272 (Cysteine metabolism) from human and E.coli. The results are shown in table 1. The first column shows the name of the pathway. The second column shows the number of instances of the best substructure found in hsa pathways, and the third column shows the number of instances of the best substructure found in eco pathways.

The best substructure of the 00020 experiment is found in both pathways as in the 00010 experiment. But in the other three experiments the best substructures are found only in the hsa pathway. The best substructure existing only in one pathway precisely shows the unique features of the pathway. In
Fig. 5. Part of Glycolysis pathways in human (A) and E.Coli (B) [26]. Connected three circled entities represents an instance of the best substructure in figure 4.

Table 1. Results of structure analysis on pathways

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Number of instance in hsa</th>
<th>Number of instances in eco</th>
</tr>
</thead>
<tbody>
<tr>
<td>00010</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>00020</td>
<td>66</td>
<td>47</td>
</tr>
<tr>
<td>00051</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>00061</td>
<td>171</td>
<td>0</td>
</tr>
<tr>
<td>00272</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

the case of the best substructure existing in both pathways, the number or the location of the instances can also indicate distinguishing features of the pathways: how many instances are in the pathway or what process is included
into the pathway. In this way substructure analysis allows us to reveal the system-level features of metabolic pathways.

5.3 Supervised Learning in Metabolic Pathways

The main goal of supervised learning is to distinguish between a biological network in a species group and a different network in the same group. This task provides us the unique substructures in the specific group of pathways to understand better how pathways differ. The distinguishing pattern of relationships between molecules, with limited consideration of the features of each molecule, can also play an important role in system-level understanding of organisms.

As described previously supervised learning uses the unnamed graphs for substructure discovery (the first phase) and the named graphs for verification of the substructure (the second phase). Graphs of metabolic pathways are divided into two groups: positive and negative examples. SUBDUE searches for patterns which exist in positive examples, but not in negative examples. We then use SUBDUE to find the erased unique IDs or unfound (in the first phase) vertices and edges in a group of named graphs using the best substructure from the first phase as predefined substructures. The second phase takes aim at verifying the biological meaning of the discovered substructures. Linked databases of the KEGG PATHWAY are also used to identify biological meaning of the final substructures.

The discovery algorithm uses the set-cover approach and it is iterated for the number of positive examples. We use the heuristic constraint for polynomial running time as described in section 4.2. Our heuristic Limit, \( L \), is calculated by

\[
L = V + B(E\gamma - 1) \tag{1}
\]

where \( V \) is the number of initial vertices, \( B \) is Beam length, \( E \) is the number of unique labeled edges, and \( \gamma \) is a heuristic constant. \( V \) and \( E \) are determined from the input graph (positive examples). \( B \) is set as 4 because this value is generally used in the successful application of SUBDUE to various domains. When we assume that the substructure learned by SUBDUE has the same number of edges as the number of unique labeled edges in the metabolic pathway, \( \gamma \) is 1.0. We try different \( \gamma \) values, and determine which value gives the best substructure in the shortest time. After several experiments, we found 1.5 (supervised learning) and 1.0 (unsupervised learning) as the best choice for \( \gamma \) in this domain.

Results of supervised learning

Supervised learning in SUBDUE tries to find the substructures that exist in the positive examples, but not in the negative examples. The choice of which set of pathways is the positive example might affect the classification result. Since our goal is the better classification between two groups, we run two...
cases. First (A+), we make a positive example set and a negative example set. The second (B+) is vice versa. We present the classification accuracy and running time for both in each experimental set. Assuming that we run two cases in parallel, the maximum accuracy expresses the best case of the classification and the maximum running time represents the worst case of the running time.

Table 2. Results of supervised learning

<table>
<thead>
<tr>
<th>Set</th>
<th>Ex. (A / B)</th>
<th>Size (V+E)</th>
<th>Time A+ (s)</th>
<th>Acc. A+ (%)</th>
<th>Time B+ (s)</th>
<th>Acc. B+ (%)</th>
<th>Time Max (s)</th>
<th>Acc. Max (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00300_00310:euk</td>
<td>9/16</td>
<td>14,715</td>
<td>1.26</td>
<td>44.00</td>
<td>1.17</td>
<td>64.00</td>
<td>1.26</td>
<td>64.00</td>
</tr>
<tr>
<td>00520_00530:euk</td>
<td>14/17</td>
<td>15,689</td>
<td>2.19</td>
<td>83.87</td>
<td>1.44</td>
<td>67.74</td>
<td>2.19</td>
<td>83.87</td>
</tr>
<tr>
<td>00010_000900:euk</td>
<td>17/16</td>
<td>38,804</td>
<td>79.00</td>
<td>100.00</td>
<td>8.84</td>
<td>100.00</td>
<td>79.00</td>
<td>100.00</td>
</tr>
<tr>
<td>00010_00061:euk</td>
<td>17/15</td>
<td>56,914</td>
<td>27.72</td>
<td>100.00</td>
<td>54.12</td>
<td>100.00</td>
<td>54.12</td>
<td>100.00</td>
</tr>
<tr>
<td>00230_00240:euk</td>
<td>17/17</td>
<td>75,086</td>
<td>49.38</td>
<td>100.00</td>
<td>50.79</td>
<td>100.00</td>
<td>50.79</td>
<td>100.00</td>
</tr>
<tr>
<td>00010_00230:euk</td>
<td>17/17</td>
<td>75,786</td>
<td>57.62</td>
<td>100.00</td>
<td>54.12</td>
<td>100.00</td>
<td>54.12</td>
<td>100.00</td>
</tr>
<tr>
<td>00300_00310:45</td>
<td>33/42</td>
<td>41,569</td>
<td>11.22</td>
<td>44.00</td>
<td>18.50</td>
<td>56.00</td>
<td>18.50</td>
<td>56.00</td>
</tr>
<tr>
<td>00520_00530:45</td>
<td>39/40</td>
<td>42,092</td>
<td>17.48</td>
<td>64.56</td>
<td>18.71</td>
<td>54.43</td>
<td>18.71</td>
<td>64.56</td>
</tr>
<tr>
<td>00010_000510:45</td>
<td>44/41</td>
<td>82,767</td>
<td>129.71</td>
<td>100.00</td>
<td>337.25</td>
<td>100.00</td>
<td>337.25</td>
<td>100.00</td>
</tr>
<tr>
<td>0010_000900:45</td>
<td>44/41</td>
<td>88,041</td>
<td>109.42</td>
<td>100.00</td>
<td>130.19</td>
<td>100.00</td>
<td>130.19</td>
<td>100.00</td>
</tr>
<tr>
<td>00010_00020:45</td>
<td>44/39</td>
<td>110,701</td>
<td>302.62</td>
<td>100.00</td>
<td>876.96</td>
<td>50.60</td>
<td>876.96</td>
<td>63.86</td>
</tr>
<tr>
<td>00251_00252:45</td>
<td>45/45</td>
<td>116,621</td>
<td>354.47</td>
<td>61.11</td>
<td>226.82</td>
<td>53.33</td>
<td>226.82</td>
<td>61.11</td>
</tr>
<tr>
<td>00010_00061:45</td>
<td>44/45</td>
<td>117,582</td>
<td>247.18</td>
<td>100.00</td>
<td>305.51</td>
<td>46.99</td>
<td>305.51</td>
<td>100.00</td>
</tr>
<tr>
<td>00010_00251:45</td>
<td>44/45</td>
<td>129,187</td>
<td>410.27</td>
<td>94.38</td>
<td>503.64</td>
<td>61.80</td>
<td>503.64</td>
<td>94.38</td>
</tr>
<tr>
<td>0010_00230:45</td>
<td>44/45</td>
<td>179,393</td>
<td>1322.95</td>
<td>76.40</td>
<td>876.96</td>
<td>91.01</td>
<td>876.96</td>
<td>91.01</td>
</tr>
<tr>
<td>00230_00240:45</td>
<td>44/45</td>
<td>183,701</td>
<td>368.12</td>
<td>100.00</td>
<td>2349.60</td>
<td>60.00</td>
<td>2349.60</td>
<td>100.00</td>
</tr>
<tr>
<td>00520_00530:150</td>
<td>137/136</td>
<td>150,363</td>
<td>874.79</td>
<td>53.85</td>
<td>1236.14</td>
<td>53.41</td>
<td>1236.14</td>
<td>53.85</td>
</tr>
<tr>
<td>00300_00310:150</td>
<td>136/143</td>
<td>157,267</td>
<td>441.32</td>
<td>48.75</td>
<td>587.7</td>
<td>53.41</td>
<td>587.7</td>
<td>53.41</td>
</tr>
<tr>
<td>00010_000900:150</td>
<td>149/104</td>
<td>188,269</td>
<td>160.45</td>
<td>95.21</td>
<td>1117.41</td>
<td>100.00</td>
<td>1117.41</td>
<td>100.00</td>
</tr>
<tr>
<td>00010_00061:150</td>
<td>149/140</td>
<td>371,032</td>
<td>305.51</td>
<td>100.00</td>
<td>4013.80</td>
<td>46.99</td>
<td>4013.80</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Table 2 shows the experimental sets and the results for supervised learning. The first column shows the name of the set which consists of three parts: A, B and source group. A and B represent two groups of pathways, and the source group represents the species set. The Eukaryote set consists of all eukaryote species (17) in the KEGG PATHWAY database. The 45 set has 45 species, and the 150 set has 150 species. The second column provides the number of pathways in each group. This number is less than or equal to the number of each source set, since the metabolic pathway may not yet be constructed (or not presented) in the specific species. For example all 17 species of the eukaryote cell have the 00010 network. But, *Encephalitozoon cuniculi* (fungi) and *Danio rerio* (Zebra fish) do not have the 00061 network. The third column shows the total size of the graphs, which is calculated as $\text{size}(G) = |V| + |E|$, where a graph $G = (V, E), V$ is the set of vertices and $E$ is the set of edges. The 4th and the 6th columns show the running
time (seconds), and the 5th and the 7th columns show the accuracy. Accuracy is calculated as \((TP + TN)/(|A| + |B|)\), where \(TP\) is the number of the positive examples containing at least one of the best patterns from any iteration, and \(TN\) is the number of the negative examples containing none of the best patterns from any iteration. The 4th and the 5th columns (A+) represent experiments that have A as positive and B as negative examples. The 6th and the 7th columns (B+) represent the reverse experiments. The last two columns show the maximum running time and accuracy in each set. We use a convention \(A(+)B(-):src\) when we denote each experimental case. For example, \(00300(+)00310(-):euk\) represent A+ case of 00300_00310:euk experiment.

Supervised learning uses 1.5 as the \(\gamma\) constant in the heuristic equation (1). Higher \(\gamma\) may sometimes afford better substructures which show more accuracy. But computational expense is usually not worth the small increase in accuracy. We make a compromise on this \(\gamma\) constant between running time and accuracy of classification. Each set has 11 ∼ 13 initial unique vertices and 8 ∼ 11 unique edges, so Limit, L, can be calculated as 55 ∼ 73.

Each case shows the different result in terms of running time and accuracy, dependent on what is used as the positive examples. The average accuracy of all experiments consisting of A+ and B+ is 71.76%. The average of maximum accuracy is 82.3%. There are discrepant cases between \(A+\) and \(B+\). 7 sets out of 20 show that one case is better than the average and another is worse. For instance, the 00010_00510:45 set has 100.00% accuracy in \(A+\) case and 44.00% in \(B+\). However, SUBDUE finds the substructures well to distinguish between two groups of pathways with more than 60% accuracy (17 sets out of 20). Figure 6 shows the running time with the graph size: \(A+, B+,\) and MaxRun (the maximum running time). SUBDUE’s running time increased polynomially with the size of the graph.
Verification of the substructures

The goal of supervised learning is to find the patterns which are not only able to distinguish between two sets of examples, but are also biologically meaningful. The pattern found by SUBDUE can differentiate well between two examples. We verify biological meaning of these patterns by using the linked database of KEGG PATHWAY [26]. Figure 7 shows the best substructure, which is found in 40 instances of 40 examples in the first iteration of the 00010(+)_00900(-):45 experiment. This substructure which covers 90.9% of the positive examples (40 out of 44) is related to two reactions. Because the edge E_to_Rct represents a relationship between reaction and enzyme (gene), the entry should be the enzyme or the gene.

![Fig. 7. First best pattern from supervised learning on 00010(+)_00900(-):45 set](image)

In the second phase (verification phase) SUBDUE runs on the named graph of the same example set 00010(+)_00900(-):45 with the first best pattern (figure 7) as the predefined substructure. SUBDUE can find clearly all forty instances in the named graph. The second phase adds more vertices and edges which are erased in the unnamed graph or are not found at the first phase. The substructure in figure 8, which is the updated pattern from the result of the first phase, is the final result of this experiment. The vertices and edges marked by “[]” are included from the original substructure learned in the first phase. With this completed substructure, we can refer to linked databases in the KEGG PATHWAY database.

SUBDUE supervised learning finds the substructure representing that an enzyme catalyzing two reactions, which share the same substrate and product. Generally an enzyme catalyzes a reaction, but some enzymes can be related to two or more reactions. Figure 8 shows two reaction vertices are connected to an entry (enzyme) vertex by two E_to_Rct edges, which denote...
links between an enzymes and a reaction. The two reactions include a shared substrate (linked by a S_to_Rct edge) and product (linked by a Rct_to_P edge). The S_to_Rct edge denotes a link from a substrate to a reaction, and the Rct_to_P edge represents a link from a reaction to a product. SUBDUE finds that this substructure exists only in 00010 examples, not in 00090 examples.

In this substructure a0e:aq_1065 which is the gene name, represents the enzyme ec:1.2.1.12 (glyceraldehyde-3-phosphate dehydrogenase). This enzyme catalyzes two reactions, R01061 and R01063, which are oxidoreductase reactions of \( \text{NAD}^+ \) and \( \text{NADP}^+ \). \( \text{NAD}^+ \) and \( \text{NADP}^+ \) are coenzymes that function as carriers of hydrogen atoms and electrons in some oxidation-reduction reactions, especially ATP (Adenosine TriPhosphate: energy material) related reactions. In our experiment the learned substructure is found only in the positive examples (Glycolysis), not in the negative examples (Terpenoid biosynthesis). Glycolysis is an energy generating process which degrades a molecule of glucose in a series of enzyme-catalyzed reactions to yield two molecules of the Pyruvates and ATPs. The conclusion of verification shows that the substructure found by SUBDUE can distinguish between two metabolic pathways and has an understandable biological meaning.

Two different metabolic pathways have unique relations as well as unique biochemical molecules. This research is focused on the unique relations. In case of 00010(+) 00900(-):45, an enzyme has relations with two reactions at the same time. The enzyme has an important feature called substrate specificity, which indicates that an enzyme can be active only when binding with a specific compound. For this reason, the enzyme, ec:1.2.1.12, catalyzes two reactions which have a common relation with the compound, cpd:C00118. In addition that identification of the unique biomolecules in each biological network is a fundamental step, but discovery of the unique relations is also important to classify metabolic pathways. The pattern of relations in the metabolic pathway can be a guide to model an unrecognized metabolic pathway.

5.4 Unsupervised Learning in Metabolic Pathways

Unsupervised learning tries to find common substructures in a set of different pathways of one species. The ultimate purpose of applying unsupervised learning to metabolic pathways is to provide a better understandable blueprint of metabolic pathways by using hierarchical topologies. This experiment allows us to understand what common structures the different networks have. The common patterns of relationships in metabolic pathways can contribute to biological network research accompanied with traditional bioinformatics.

Like supervised learning, unsupervised learning also employs the unnamed graphs in the first phase and the named graphs in the second phase. In the first phase SUBDUE discovers the substructures and generates hierarchical
clusters using iterative discovery. Then SUBDUE adds eliminated unique IDs or unfound vertices and edges from the first phase. This process uses the substructures discovered from the first phase as predefined substructures in the second phase. The second phase also tries to verify the biological meaning of the discovered substructures by referring to the linked databases of the KEGG PATHWAY, which are used to identify biological meaning of the final substructures. The same heuristic equation (1) is used to compute the limit, L, as in supervised learning.

**Table 3. Results of unsupervised learning**

<table>
<thead>
<tr>
<th>Set</th>
<th>Number of examples</th>
<th>Size (V+E)</th>
<th>Running time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ath</td>
<td>100</td>
<td>68,585</td>
<td>267.81</td>
</tr>
<tr>
<td>dme</td>
<td>92</td>
<td>58,166</td>
<td>204.52</td>
</tr>
<tr>
<td>eco</td>
<td>102</td>
<td>78,252</td>
<td>418.42</td>
</tr>
<tr>
<td>rno</td>
<td>96</td>
<td>61,409</td>
<td>183.60</td>
</tr>
<tr>
<td>sce</td>
<td>86</td>
<td>63,078</td>
<td>249.69</td>
</tr>
<tr>
<td>mmu</td>
<td>106</td>
<td>81,634</td>
<td>556.81</td>
</tr>
<tr>
<td>hsa</td>
<td>110</td>
<td>90,157</td>
<td>598.99</td>
</tr>
</tbody>
</table>

**Fig. 9. Running time with graph size**

**Results of unsupervised learning**

Table 3 shows the experimental sets used in unsupervised learning. Set represents the name of the species [26]. The number of examples denotes the number of metabolic pathways which the species has in the KEGG PATHWAY database. The 110 metabolic pathways in hsa (Homo Sapiens) is the largest number in the KEGG, when we include just metabolic pathways, not regulatory networks. Other species have fewer metabolic pathways, because...
they do not exist or are yet to reconstructed. Size is the size of the graph as described above. The last column shows the running time. Each run iterates 10 times to construct hierarchical clusters. Unlike supervised learning, this experiment uses MDL as the evaluation method. Unsupervised learning uses 1.0 as the $\gamma$ constant in the heuristic equation (1). Each set has $14 \sim 16$ initial unique labeled vertices and $8 \sim 11$ unique labeled edges. Limit, L, is calculated as $42 \sim 54$. SUBDUE runs in polynomial time with the size of the graph as shown in figure 9.

**Verification of the substructures**

The purpose of SUBDUE unsupervised learning is to find the common substructures, which describe the regular features in a group of metabolic pathways. Moreover, hierarchical clusters of the common substructures show a blueprint of metabolic pathways. We provide hierarchical clusters learned by SUBDUE and verify them using the linked databases of KEGG PATHWAY. Partial hierarchical clusters of substructures learned from the dme (fruit fly) set are shown in figure 10. SUB$_{i}$ denotes the best substructure in the $i$-th iteration.

The hierarchical clusters show that the substructures in the upper level are contained in lower level. For example, SUB$_8$ includes two SUB$_1$, one SUB$_3$ and one SUB$_4$. The general substructures are used to compose more specific substructures. This is how SUBDUE shows the common relational patterns of the metabolic pathways and how the patterns relate to each other hierarchically. SUB$_8$ is discovered in three metabolic pathways of fruit fly (dme). This substructure is not only the common feature in these three pathways, but also the distinct property from other pathways.

**Fig. 10.** Partial Hierarchical Clusters of metabolic pathways in fruit fly
SUB_1 shows a basic reaction that is found in 972 instances of 90 examples. SUB_3 is found in 3,659 instances of 47 examples at the third iteration. SUB_4, found in 1,136 instances of 21 examples, represents a relation with the ECrel property. The ECrel relation is an enzyme-enzyme relation where two enzymes catalyze successive reaction steps [26]. SUB_8 discovered in 264 instances of 3 examples includes one relation of two enzymes which catalyze two successive reactions. Moreover, SUB_8 has an additional meaning to SUB_4 such that the “link” edge connects to a compound which is a product of the first reaction of this relation and a substrate of the second reaction at the same time [26].

SUB_8 includes an enzyme-enzyme relation which relates three consecutive chemical compounds. Figure 11 shows an example of SUB_8 which is found in the dme00052, Galactose metabolic pathway of the fruit fly. Figure 11 is a fully updated substructure thorough the second phase. Like the previous case, the nodes and edges checked with “[ ]” are found in the first phase; others were added in the second phase. This substructure shows a relation between two enzymes which shares a compound as a substrate by one and a product by another. The enzyme-enzyme relation has a relationship with two reactions: R01092 and R01105 [26]. R01092 is catalyzed by the enzyme of the gene, dme:CG5288-PA, and R01105 is catalyzed by the enzyme of the gene, dme:CG9092-PA. The substrate of R01092 is the C05796 compound (Galactin). The product of this reaction is C00124 (D-Galactose), which is also the substrate of R01092. R01092 produces C00446 (alpha-D-Galactose 1-phosphate) as the product compound. The relation in this substructure has the link as a pointer to C00124 because this compound is the shared metabolite in two reactions catalyzed by two enzymes connected within this relation.

SUB_1 and SUB_4 are found in all experimental sets (species), and SUB_8 is commonly found in ath, dme, eco and sce. A hierarchical clustering presents the common relations, which shows how biological molecules work interactively with others in the different species. This provides system-level understanding of metabolic pathways.
6 Conclusion

Systems biology views an organism as a system. System-level understanding indispensably involves integrating heterogeneous data and a variety of relations among the entities. The biological network is a crucial way to describe the biological system. Biological networks include various biomolecules and assorted relationships among molecules. Structure analysis of metabolic pathways allows us to understand how biomolecules interact with others. The research on the relations can play a contributive role in systems biology.

This research shows several methods of structure analysis on metabolic pathways. Substructure discovery on the same metabolic pathways from two species reveals the unique features of the pathways related to the species. Even in the cases that SUBDUE cannot find a unique substructure distinguishing two pathways, the number or the location of the instances of the substructure is able to distinguish them; how many specific relations or what specific relations are included into the pathway. Supervised learning shows the substructures that can identify what is unique about a specific type of pathway, which allows us to understand better how pathways differ. Unsupervised learning generates hierarchical clusters that reveal what is common about a specific type of pathways, which provides us better understanding of the common structure in pathways.

Moreover, our results show that the substructures discovered by SUBDUE have understandable biological meaning. These substructures, when considered as building blocks, can be used to construct new metabolic pathways. Ultimately, we can consider these substructures as guides to define a graph grammar for metabolic pathways that would improve both our ability to generate new networks and our comprehension of pathways [18]. These building blocks of metabolic pathways open our sights to an advanced application: drug discovery. The substructure of metabolic pathways learned by SUBDUE allows us to identify the target place of the drug in pathways. In addition a graph grammar of relational patterns on metabolic pathways can guide us to simulate the drug interaction on pathways.

Our future works include graph-based relational learning on graphs representing dynamics of biological networks and association with other methodologies for efficient learning on biological networks.

References


Design of an Online Physician-Mediated Personal Health Record System

Jake Y. Chen1,2,3,*, Shailaja Taduri2, and Frank Lloyd3,4

1 Indiana University School of Informatics, Indianapolis, IN 46202
jakechen@iupui.edu
2 Department of Computer and Information Science, Purdue University School of Science, Indianapolis, IN 46202
3 Indiana Center for Systems Biology and Personalized Medicine, Indianapolis, IN 46202
4 The Center for Surgical Sciences and Trauma, Methodist Hospital, Indianapolis, IN 46206

Abstract. Electronic Medical Record (EMR) has been adopted by most hospitals in the United States to keep track of patient medical information today. However, HIPAA (Health Insurance Portability and Accountability Act) regulations, technical information system design challenges, and social-economic factors—all contributed to significant barriers to granting patients access to his own EMR maintained by hospitals with the personal health records (PHR) maintained by patients. In this chapter, we describe a new Personal Health Office Framework (PhoF), which is designed to enable patients effectively manage their own PHR integrated with a portion of their EMR with the guidance of health-care providers. PhoF is characterized with a unique online physician-approval process, which can allow a computer agent to retrieve specific information securely on a particular patient maintained with the approving physician’s hospital EMR. The mediated access to medical treatment information enables a physician to administer patient self-education and self-care for online patients and therefore may help reduce medical errors and health care cost. A prototype system with web user interface is described and several case studies are presented to show the how the concept works in practice.

Keywords: electronic medical records, personal health records, computer agents, physician-mediation.

1 Introduction

Providing the most cost-effective care and quality of service to patients with continued information technology innovation is a major theme in the United States health care industry today. In many hospitals and clinics throughout the country, digital transcription of physician’s notes, electronic medical records, and automated patient scheduling systems are becoming commonplace. Each time a patient visits his/her health care provider, an electronic record of the visit can now be made, compiled into, and maintained as part of the provider’s medical information system

* Corresponding author.
(IS). This type of centralized medical information system serves as an emerging central platform for managing and planning quality health care by hospital administrators, physicians, nurses, and other medical researchers. It also serves as a legal recording system by which the patient’s insurance company verifies medical billing. The system replaces the traditional paper-based system as an effective platform of choice, known to help reduce medical errors, reduce billing errors, and increase efficiency.

Along this trend, there has been a similar drive to provide access and self-management of individual electronic medical records to patients and health-conscious consumers in the United States [1]. There are two types of electronic records in the markets today available to patients. The first type, Electronic Medical Records (EMR), contains a patient’s hospital-related health care record in digital format stored in hospital/clinic’s information systems. EMR, while managed electronically by health-care providers, include data from hospitals, pathology labs, clinics, and insurance companies. The effective implementation of EMR systems have been shown to improve health care efficiency and safety while reducing costs [2]. As a good example of the latest EMR software systems, Medicware (www.medicware.com) aims to promote patient care and improve treatment efficiency by enabling providers to manage their patient’s complete clinical experience, which includes office visits, medical examination results, health maintenance, referrals, medications and prescriptions.

The second type, Personal Health Record (PHR), contains a personal’s health log created and maintained by individuals. A good PHR record should provide complete and accurate summary of health and medical history of an individual including primary care providers, health insurance, allergies, current medications, illness and hospitalizations, surgical history, laboratory test results, as well as personal health information such as medication, allergies, diet, weight, mental status, and exercising status. For example, the American Health Information and Management Association (AMIMA) has been educating patients on the need to make use of PHR to improve health management. Electronic version of a PHR can be internet-based or PC-based software, portable yet connected to backend information systems. Various industry partners such as CapMed, ihealthRecord, Dr.iNet, Medic Alert has just begun to offer PHR solutions to the customers as standalone software and data entry systems, which consumers can interact with.

In spite of the progress in digitizing medical and health information for individuals concerned with their own health or disease issues, the choices available for patients to access and control their medical and health information are still extremely limited. On one hand, even in hospitals where information systems are available for hospital staffs and physicians, majority of them still offer paper-based records. This is partly because EMR systems were developed primarily to help hospital workers gain integrated view of disparate data sources such as laboratory systems, pharmacy systems, and physician notes. It’s challenging to develop user interfaces to make such system easy to understand by non-medical users [3]. Moreover, privacy and confidentiality are perceived to be a significant problem, even though a protected electronic medical record generally offers much better security than a paper-based record [4]. All EMR systems specifically developed for physicians have to be compliant with the HIPAA (Health Insurance Portability and Accountability Act) legal rules and regulations in the United States. On
the other hand, while PHR enables personalized self-management of health-related information, it has been kept independently of hospital EMR systems. This division between EMR and PHR has created challenges for individuals to access vast amounts of information already available from different hospital systems for self-care and error-checking. It also restricted the ability of medical professionals to improve quality of services and for patients and health-care professionals to keep disease care communication channel open outside of the health care service environment. Therefore, it has become critical that a new health and disease electronic data management paradigm should emerge.

2 Background

Creating and maintaining an accurate PHR system connected to backend EMR systems from different sources for health-conscious individuals have the clear benefits of enabling patient self-education and error-prevention. It also remains an active challenging research topic for the following reasons:

First, the current process of obtaining EMR by individual patients is still a slow and manual process, because the care providers moving information from EMR into PHR need to abide by HIPAA rules. Every patient has the right to receive copies of personal health records, if the patient completes and submits an “authorization for release of information” form to the service provider (for an example, visit http://www.myphr.com). The health care providers need to document the entire authorization process and keep the patient’s EMR records private and confidential from each other unless they are explicitly released by patients. [5] The manual authentication and approval process takes time and may curb most patients’ desire to explore their EMR for health-related questions.

Second, current EMR systems have not been designed with individual patients as end users in mind. To consider access of EMR by patients, many information systems design choices, e.g., privacy vs. accessibility, functionality vs. usability, guided vs. Un-guided, have to be carefully made. To protect patient privacy, confidentiality and integrity of PHR, appropriate security measures have to be incorporated. Accessibility of PHR with a patient’s consent to other care provider’s EMR would also be ideal. Some patients may need to access their EMR from their PHR system, which is extremely difficult to perform because the lack of data standards and software interoperability among different EMR and PHR software systems. Self-service may be preferred by the system designers because the cost of human intervene is reduced; however, as the quality of software-based services are not good enough to help patient users achieve satisfaction, guided online service may be preferred. Therefore, involvement of medical professionals, physicians and nurses, are necessary features to consider building in order to promote instead of demote communications between patients and physicians. The system architect needs to make make such decisions to find a balance between feature offering and system usability within patient groups.

Third, social-economic factors, policy, and life-style choices may all contribute to the interruption of integration between EMR and PHR. Individuals with issues of mental health, ADHD, drug and alcohol problems, for example, may be less likely to manage their personal records well than professionals with diabetic
concerns. The service providers must take these factors into great consideration when designing new systems. Economic factors such as the affordability to use the designed systems should also be taken into account. Due to technology uncertainty, policy makers may also institute rules that further constraints the current EMR systems from being accessed outside of its designed users—doctors and health care professionals. On the other hand, each individual has a different life-style choice when it comes to maintaining personal health. Some are very health-conscious and be more willing to accept temporary function deficiency of a developing system than others. Some patients are keenly focused on only certain health conditions such as tracking blood sugar and cholesterol levels and others may have different preferences. It’s not clear whether an initial system launched may be well received immediately among the anticipating end user community.

To address these challenges, several exploratory systems have been reported. PING (Personal Internetworked Notary and Guardian) is a framework proposed to address the suggested need.[6] It is designed to be a secure and scalable software platform for archival PHRs. The architecture of PING is based on http protocols, XML data structure, and public-key cryptography to grant patients secure control over their medical data online. The PING architecture also includes a role-based authorization scheme to assign access privileges, allowing different clinical researchers, physicians, and medical professionals to work together without compromising patient’s privacy. In addition to the basic architecture, it also comes with software tool to populate the patient's record migrated from EMRs. However, it remains largely a research prototype and few real-world applications have been developed with this architecture, largely due to rapid development of web-based software system fueled by the Internet boom.

MyPHR.com (http://www.myphr.com) takes a web portal approach to building consumer-oriented PHRs connected to EMRs. The system is developed by the American Health Information Management Association. The Internet-based PHR portals such as MyPHR.com provide lots of information on how to collect an individual’s own medical records from EMRs through self-guided tutorials. Sophisticated software features are also being developed to support personal online tracking of health vital measures and medications. However, studies showed such PHRs have too limited functionality and relevant medical history to be useful on daily basis to most individuals. [7]

IHealthRecord (www.ihealthrecord.org) is a recent successful model that tries to make up for the deficiency of simple online PHR portals. The IHealthRecords system is co-developed by the joint work between the private Medem network and leading US medical societies. The most distinctive feature of the system is that it promotes physician and patient communication by extending health management outside of the confines of hospitals. Physicians and patients may send messagings and alert to each other of upcoming events to improve medical office efficiency and patient revenues for the physicians, and to allow sharing of patient health information with their physicians during emergency. While the concept is exciting, the architecture and information model are controlled by the Medem network—not based on open standards. Therefore, it is not clear how the individuals may port their health information at IHealthRecord to other future PHR portals should he/she desires so. The information that patient can pull from
the hospital’s EMRs are largely non-existent. Therefore, it is still not a true integrated EMR/PHR system.

CapMed PHR (http://www.capmed.com/solutions/applications.asp) provides innovative software/hardware solutions, e.g., PHR USB flashdrive, PHR CDs, and online PHRs. These solutions are extremely portable, easily interoperable, and easier to manage than paper-based system. Patients may use these tools to store medical emergency information, create family medical history, track lab values and vital signs, track medications, and log communications with medical providers. While CapMed does sufficiently good job helping individuals manage nearly all aspects of their medical/health care needs through extensive PHR, the role of medical professionals, especially physicians, are not emphasized. Therefore, the EMR that such systems may connect to is quite limited in functionality, because the bulk of medical treatment data would be too much to leave to self-guided individuals concerned about the the accuracy/completeness of their own health data entered each day.

Apparently, there is still plenty of significant development opportunities for PHRs integrated with hospital EMR data. Combining the successful design principles of the above exemple systems, we believe future system should have the following characteristics:

- **The system should be secure and offer multiple level of access control.** Privacy should be placed in the highest priority, due to the sensitive nature of the data. Therefore, multi-user concurrency-controlled system that allow asynchronous secure management of data should be chosen over simple systems that may compromise the data privacy.
- **The system should allow 360° self-management of patient’s health and disease.** This include comprehensive tracking of the patient’s contact information, medication, allergies, medical history, emergency contacts, hospital visits, fitness/diet logs, and comprehensive journals related to personal health concerns.
- **The system should promote the interactions between physicians and patients.** Patients may not have the best expertise to sift through vast amount of health information (such as those available on WebMD.com or NIH web sites); therefore, guidance and interactions with expert physicians or other medical/health professionals are necessary. Physicians may not have time to call/write/email patients around the clock, and therefore would like to provide high quality of service by referring patient preparation and self-education through electronic resources that physicians themselves trust.
- **The system should be convenient to use by most individuals.** Therefore, the system should adopt open-standards to maximize the chance of interoperability with other available resources (such as medical theraurus lookup web services). The system should also allow flexible download of content from software system to/from portable media (USB or CD). The systems should also allow bulk supervised filtering of medical data from EMRs to PHRs, and vice versa. This would bring increased usage of the data. For chronic disease patients, the convenience can be compared to
being able to control one’s finances everyday through online banking and personal financial planning.

3 Framework Design

We show a novel design of a PHR information system framework which we call Personal Health Office (PHO), based on our experience and analysis of existing PHR systems. The PHO framework aims to enable patients manage their personal health records that are pre-populated with their own EMRs retrieved from the health care provider’s EMR systems. Using this design, the framework allows the patients to gain access to their own sensitive EMR in a secure architecture outside of the transactional databases of the hospitals, expedited by their physicians. The framework also enables the extraction, transformation, and loading (ETL) of conventional EMR data to interactive graphical charts and allows the patients to annotate on the transformed data. Moreover, the framework provides secure and easy online management of comprehensive PHR functions by the patients with health blogs and notes on EMRs. The framework can be extended with many features with standard web technology and robust relational database engine. Furthermore, the framework gives health care service providers and physicians an opportunity to validate transcription errors caught and corrected by patients themselves. Lastly, the framework supports the extraction and integration of data from various public sources such as hospital directories and physician directories in centralized data repositories outside of the hospital EMR or personal PHR systems. An overall design of this PHO architecture is shown in Fig. 1.

In our PHO framework (Fig. 1), public database contents (top left) and personalized medical records (top right) are separately retrieved but combined together. The public database contents include bulk downloaded information not only including what is shown in the figure but also information such as physician specialty database, drug database, drug safety, and health tips, commonly in various existing relational database formats. The personalized medical records information from the service provider’s EMR information systems commonly take the form of un-structured PHR documents. These documents can be retrieved using automated bulk web content downlladers, web services, or web page information extracting tools. The parsed PHR data can then be transformed into chronological health events and loaded into the PHO database managed by Oracle database mangement system (DBMS). Application modules such as the following are then built to support various content presentations (charting, commenting, etc):

- **Data Extraction, Loading and Transformation (ETL) Engine.** At the 1st and information extraction layer, this module is responsible for extracting information from document collections. It is used to extract the data originally stored in EMR systems and public systems, transform into various object representations in the PHO database.
  
- **Relational DBMS engine.** At the 2nd and database layer, this module is used to manage the data and meta-data necessary for holding the personal health event information. It also serves as a control system between different PHO web system components.
Design of an Online Physician-Mediated Personal Health Record System

Fig. 1. The Personal Health Office (PHO) architecture framework

- **Encryption/Decryption Engine.** At the 3rd and interaction layer, this module is responsible for securely encoding and decoding of the patient PHR data during secure storage and transmission sessions.

- **Health Blog Engine.** At the 3rd and interaction layer, this module is responsible for supporting patient’s personal health blog record management.

- **Personal Profile Engine.** At the 3rd and interaction layer, this module enables the storage and retrieval of system user’s personal profile such as medication, allergies, contact information, and current disease treatments.

- **Authorization Engine.** At the 3rd and interaction layer, this module helps manage patient’s access permission requests, perform physician’s authorization to approve or deny patient requests to access their hospital records, and process these access requests using secure protocols.

- **Charting Engine.** At the 3rd and interaction layer, this module is responsible for retrieving the patient’s health events and converting them to easy-to-comprehend graphical health charts.

- **Annotation Engine.** At the 3rd and interaction layer, this module is responsible for inserting patient’s annotations on his personal health charts in the database. A chronological annotation log can be maintained in the chosen database. It also retrieves the saved annotations for reviews.

- **Authentication Engine.** At the 3rd and interaction layer, this module performs user logon authentication of the PHO web system. It checks the credentials of the user to access approved features.
This design has several advantages compared with existing systems. First, the design incorporates multiple data sources, which can be derived from bulk public directory information with hospital EMRs and personal PHR health self-management data; therefore, it can be used to develop comprehensive data-driven patient-centric PHR applications. Second, the design converges on robust database management system which underpins a set of functional modules; therefore, the system is scalable and adaptable to changing individual user needs. Third, the management of health data has extensive built-in security and access control both at the database layer (Oracle DBMS functions) and the application layer (encryption/decryption, authentication and authorization engines). Fourth, the framework is very flexible and extensible; one can add multi-media presentations such as treatment educational videos, mp3 audios for latest treatment/health tips, and annotated image from pathology/radiology labs.

4 Usage and Functional Interactions

In the previous section, we showed a framework design which shows all the components the PHO system should have in order to address current challenges in developing PHO systems. In this section, we further show how the primary users of this system could interactively use the PHO software. In Fig. 2, we show such a system relies on a web server that serves various user requests via secure http protocols. The system sets different types of access privileges to three different types of users, i.e., patients, physicians, and web administrators. Depending on the type of user roles, the PHO system customizes user-specific interfaces and functional features.

Key to this functional interaction paradigm is a new concept, Physician-mediated Patient EMR Review. This new concept addresses the difficult problem of enabling individual access to hospital EMRs, which traditionally is paper-based and slow, because it needs to involve hospitals and HIPAA regulations. The concept is an attempt to solve the problem, which is that hospitals own the information content located in their protected EMRs yet don’t have all the right to access the content, while individual users have the rights to access their own records yet don’t own the information/software located in the hospital EMR. The solution lies in introducing physicians into the interaction paradigm for two reasons: first, physicians are pre-authorized to access hospital EMRs related to their own patients legally; second, physicians are motivated to help their own patients access their own records conveniently, if such help could bring satisfied patients, reduced medical errors, increased patient cooperation, and increased efficiency while receiving patients in the clinic. The concept works in the following five steps:

1. **Request.** A patient logs in his PHO system and issues multiple requests to his/her physicians at different hospitals for permissions to access his/her own hospital EMR data. He then logs out and waits for approvals.
2. **Approve.** One of the patient’s physicians logs into the third-party PHO system (not inside hospital EMR system) later and finds several of patients’ requests for accessing to different types of their medical records. The physician either approves or denies access to his/her requesting patients.
3. **Retrieve.** The PHO system documents the consenting process and process the data collection request, using the approving physician’s credentials to automatically retrieve records in batch on behalf of all approved patients from all hospital physicians who approved access.

4. **Access.** The patient checks the status of submitted requests. He will find some requests get approved, some pending, while others get rejected with rejection reasons stated in the commenting fields. The retrieved records may be completely reorganized by the PHO system to improve patient self-education and ease of data interpretations.

5. **Annotate.** The patient reviews his/her personal health charts, event logs, and medical treatment transcripts about themselves. He/She adds comments to the health log events for his/her own health record keeping.

6. **Share.** The patients, after filtering private information, may choose to share some information online with others, including his/her peers and physicians, to promote health self-education in his community.

The benefits of this physician-mediated patient EMR review paradigm are potentially huge. The patients can quickly gain access to their EMR and integrate accessed records into their PHR immediately. The patients can also discover potential inaccuracies (such as personal information or current medications), which allows the hospitals to correct. The hospitals don’t need to worry about developing increasingly complex EMR system to yet another audience—individual
patients—that drives up cost; instead, they can be relieved from HIPAA patient consenting paperwork, medical information filtering, and medical-error related liabilities. The physicians can also take a central role in efficient patient self-education, record review, and guided medical assessment—all contributing to the improving the quality of the services offered.

We break down the functional interactions based on three user types: patients, physicians, and web administrators. They interact with the PHO software in different ways described as the following.

- **Patient Functions.** Patients could gain privileged access to the PHO web system (Logon), view the status of the requests submitted to the physician (Show Request Status), request access permission to his/her own records stored in the hospital’s EMR (Request Access), view personal profile information such as name, date of birth, current medications, and allergy information (Access Profile), view their personal health record linked to EMR after physician approval (Access PHR), add comments and timestamps of such comments to health records collected from different hospital record systems (Annotate PHR), view and add his daily health online blogs (Add Daily Health Blog), and locate his physician in order to request access permission online or schedule follow-up visits/consultation sessions (Local Physicians).

- **Physician Functions.** Physicians could gain access to the PHO web system and remote hospital EMR web access (Logon), use his hospital access privilege to grant or reject access permissions to patients who requests access to their own records permissions (Grant/Deny Access), review patient PHR that provides a different view of patient PHR from the traditional hospital EMR (Review Patient PHR), review patient’s shared health event annotations to help improve their in-hospital health care (Review Patient Annotations), update their contact information (Update Contact Info), and communicate with patients for clinical trial information (Send Clinical Trials).

- **Web Administrator Functions.** They can load patient basic information database from different sources into the standalone PHO online system (Register Patient), load physician directory and specialty information database into the standalone PHO online system (Register Physician), associates each patient with the designated physician based on the patient’s health care provider information retrieved by his hospital EMR (Match Patient-Physician), and manage hospital EMR description including data format and source link (Save Hospital Profiles).

5 Prototyping and Case Study

We present a case study to show how the new framework design could be implemented as a research prototype to guide future large-scale implementations. To do so, we introduce a hypothetical patient, Henry Wayne, a diabetic patient of hypothetical physician Dr. Alexander Anthony at a hospital. Alexander’s medical
treatment history is presumed to be stored in the hospital’s online EMR system. We show a prototype implementation of the framework design in Oracle 10g database at the backend and Apache/Tomcat web server at the frontend using https protocol.

5.1 User Logons

This screen is designed as a secure access point to Personal Health Office online system. The users of the system are patients and physicians (see Fig. 3). When the user logs in as a patient, the user is authenticated by the authentication engine and the appropriate menu is displayed based on the user access level.

5.2 EMR Access Request and Approvals

The first time the patient Henry Wayne requests any EMR access, he needs to identify his physician. In Fig. 4, we show that the patient may search physician from the physician directory, and the Authorization Engine will retrieve a list of primary physicians for the patient and displays to the patient. The patient then selects pre-associated physician “Dr. Alexander Anthony” from the PHO system. Once a selection of a physician is submitted, the patient can wait for the outcome from his selected doctor.

There are two different user interfaces, one for the patient and the other for the physician, that may be used to check/act on the status of patient EMR online.
access (See Fig. 5). In the example shown, patient Henry is still waiting for Dr. Anthony to approve his requests. Therefore, Henry sees the status “pending” while Dr. Anthony sees two options for Henry’s request “to Approve” or “to Reject”, along with approved request from another patient (John Smith) done earlier. If Henry’s request is rejected, the next time he checks back into the system, he would see the request status to go from “Pending” to “Reject”.

<table>
<thead>
<tr>
<th>Request Status</th>
<th>Patient Name</th>
<th>Physician Name</th>
<th>Request Status</th>
<th>Requested Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Henry Wayne</td>
<td>Alexander Anthony</td>
<td>Pending</td>
<td>April 17, 2007 06:09 PM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patient's Requests</th>
<th>Patient Name</th>
<th>Physician Name</th>
<th>Request Status</th>
<th>Requested Date</th>
<th>Approve/Reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>Alexander Anthony</td>
<td>Approved on April 14, 2007 05:14 PM</td>
<td>April 14, 2007 05:04 PM</td>
<td>Deny</td>
<td></td>
</tr>
<tr>
<td>Henry Wayne</td>
<td>Alexander Anthony</td>
<td>Pending</td>
<td>April 17, 2007 06:09 PM</td>
<td>Approve</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Fig. 5. Different User Interfaces for Patients (panel a) and Physicians (panel b) Who Check the Status of Patient EMR Access Requests

5.3 Access Personal Health Charts

After permission to retrieve his/her EMR from the hospital is granted by authorized physicians, patients can then access their personal health record. To show how the transformation was made to make the medical record easy to interpret, we show one of the many health events retrieved visually using charts. Fig. 6 below shows a pie chart of lab tests frequency distribution.

Fig. 6. A Personal Health Chart Generated from EMR Data Collected and Re-organized into the PHO System
The Charting Engine can generate different types of charts. The charts may be viewed chronically by selecting the “yearly” option or in its entirety by selecting ‘All’. In addition, in the current implementation, by default, the patients grants their primary physicians automatic approval to view their health charts and comments made.

5.4 Annotate Personal Health Records

There are two types of annotation we will demonstrate. In Fig. 7, panel a, we show a chronicle of blood sugar level changes for the patient. Note the data is integrated first with public database in which the normal range of blood sugar level may be looked up. This lookup content is used to automatically annotate each chronicle event regardless of the level “normal”, “high”, or “low”. When clicked on ‘Saved Notes’ link, the review comments saved by the patient will be displayed. The second type of annotation is provided by user as comments. Next to three blood sugar measurements, there are “Saved Notes” in the Patient’s Comments section. Clicking on the link on September 27, 2006, a patient will see or update his own notes, which can be saved again for future use.

<table>
<thead>
<tr>
<th>Event Date</th>
<th>Blood Sugar Level (Target Range)</th>
<th>Normal/High/Low</th>
<th>Patient’s Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 05, 2006</td>
<td>88 (70-109)</td>
<td>Normal</td>
<td>Saved Notes</td>
</tr>
<tr>
<td>January 06, 2006</td>
<td>102 (70-109)</td>
<td>Normal</td>
<td></td>
</tr>
<tr>
<td>February 01, 2006</td>
<td>183 (70-109)</td>
<td>High</td>
<td>Saved Notes</td>
</tr>
<tr>
<td>September 25, 2006</td>
<td>161 (70-109)</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>September 26, 2006</td>
<td>132 (70-109)</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>September 27, 2006</td>
<td>143 (70-109)</td>
<td>High</td>
<td>Saved Notes</td>
</tr>
<tr>
<td>September 28, 2006</td>
<td>165 (70-109)</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>October 07, 2006</td>
<td>187 (70-109)</td>
<td>High</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7. A Health Event Chronicle from PHO the System Showing Blood Sugar Level (a) and Patient’s Comments (b)

5.5 Sharing PHR with Physicians

By default in this implementation, patients automatically grant their primary physicians permission to view their health charts and comments made in the PHO. Therefore, a physician may be able to view each of his patient’s review comments, in addition to easy-to-understand subsets of the EMR that the physician can already access. This feature facilitates an environment where the physician can
study patient medical lab tests and take appropriate decisions on prescribing new tests for the disease that the patient is diagnosed for. Another component is to facilitate broadcast of clinical trial messages to all the patients of the physician. In Fig. 8, we show an example user interface to select a patient from a physician’s account. After the physician clicks on the “View Charts” button, the system displays the same sets of charts and tables, except that physician’s view is read-only.

<table>
<thead>
<tr>
<th>Patient Name</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Charts</td>
<td>View Patients Review Comments</td>
</tr>
</tbody>
</table>

**Fig. 8.** User Interface for Physician to Choose Patients who Shares PHR with him/her

### 6 Discussion

The physician-mediated personal health office software system is a new method, which aims to benefit patients, physicians, and information technology administrators significantly with the help of PHO software agents. The PHO software agents can help integrate public health data with individual’s medical data from EMRs and personal health history data. The system has built-in security and access control, with robust relational database management system in the backend, and secure web server applications in the frontend. With physician as mediator, especially when future PHO applications are integrated with physician’s daily information management (including email), the small overhead can translate into potentially huge gains for patients, who are excited about managing PHR and EMR together, understanding their own health problems better, and being able to self-care outside of the hospital confinements.

Adopting and enforcing accepted standards is an important topic not addressed here. This requires first and foremost the development of a complete data model for representing the structure of the domain covered in this application. For example, there should be a many-many relationship between patients and physicians as it exists in real world. With this feature, if a patient’s access request is delayed or unfairly denied by one physician, the patient would be able to request another physician to view health information pertaining to a different category. Also important is the mapping of medical terms from different hospitals, patients, and their EMRs/PHRs onto the presentation layer using standards such as the UMLS [8].

To make the prototype increasingly significant and useful, we would be transferring the concepts, design models, and prototype programming codes into the Open Source community. It is necessary that the community of developers are involved in this project. We believe the sharing and social networking potential of
Design of an Online Physician-Mediated Personal Health Record System

making individual’s integrated PHRs available online, at the individual’s control, could bring revolutionary changes to the way we view and use our health information in the future.

References

Completing the Total Wellbeing Puzzle Using a Multi-agent System

Maja Hadzic¹, Meifania Chen¹, and Rick Brouwer²

¹ Curtin University of Technology, Digital Ecosystems and Business Intelligence Institute (DEBI)  
GPO Box U1987 Perth,  
Western Australia 6845, Australia  
² Total Wellbeing Medical and Counseling Centre  
Suite 1 / 857 Doncaster Rd  
Doncaster East 3109  
Victoria, Australia  
{m.hadzic,m.chen}@curtin.edu.au, rickb@totalwellbeing.com.au

Abstract. Our research focus is the implementation of agent-based systems within the health domain, more specifically, in the study of total wellbeing. We use an evidence-based total wellbeing ontological model where the total wellbeing is seen as a function of physical health, mental health, emotions, relationships, financial situation and spirituality. We use the TICSA methodology to design a multi-agent system. This multi-agent system is based on the Total Wellbeing Ontology and helps intelligent retrieval, management and analysis of information related to total wellbeing. We hope this system to expose evidence that will support general public in managing their personal wellbeing better, and health professionals in adapting their services to address patients’ needs more systematically and effectively.

Keywords: Total Wellbeing, Ontology-based Multi-agent System, Multi-agent System Design, e-Health, Health Information System.

1 Introduction

The main features of agents are their autonomous, goal-driven, intelligent, proactive, cooperative, collaborative and mobile capabilities. Agents have the ability to act independently from both the user and the rest of the system. They are usually driven by a specific goal which benefits the whole system. Agents are equipped with intelligence that enables them to reason and perform the beneficial actions. Agents are proactive, namely, are able to make decisions and take action on their own initiative.

Even though the agent is able to act autonomously, it has to be sociable; it needs to cooperate and collaborate with other agents of the system. The performance of multi-agent systems is based on collaborative effort of various agent types. The different agents work cooperatively towards a shared goal. Multi-agent systems are designed to reach their full potential through cooperation, coordination of their actions, tasks and results sharing. The collaborative nature of agents enables
the multi-agent system to find solutions to complex problems that are usually impossible to be solved by individual agents.

Additional advantage of agent-based systems is mobility of agents. Agents can be created as being capable of migrating to different places and execute their actions at different places. Addition of mobile agents has the potential to increase dynamics and efficiency of the whole system.

Our research focus is the implementation of agent-based systems within health domain, more specifically, in the study of total wellbeing. Our health and wellbeing is not only determined by our physical health but also by mental health, our emotions, relationships, financial situation and spirituality. A huge body of the information is available on these and related topics. The three main issues hindering effective use of the available information include [1]:

1. size of the available information
2. information resources of autonomous, distributed and heterogeneous nature, and
3. lack of tools to systematically analyse the huge body of available information and derive useful knowledge from it.

Health professionals and general public are faced with numerous difficulties, from more specific problems such as complex nature of the health (as explained in Section 3) to more generic problems such as [1,2]:

1. amount of health information is rapidly increasing as new papers and journals are increasingly being published and added to the existing information resources
2. available information is found in inconsistent structures which is a result of autonomous information resources, and
3. various information resources contain related, overlapping and semi-complementary information but we lack tools to identify these occurrences.

In this paper we present a multi-agent system that will enable efficient and effective use of health-related information. A number of multi-agent systems have been implemented within medical and health domain and these are discussed in Section 2. We focus specifically on the information that is related to the total wellbeing and relationships between different dimensions of the total wellbeing. We explain the complex nature of total wellbeing and discuss related studies in Section 3. In Section 4, we use the TICSA approach [34] to design the multi-agent system. We give our final remarks in Section 5.

2 Multi-agent Systems and Ontologies in Health Domain

Multi-agent systems are increasingly being used in the medical and health domain. Implementation of agent-based systems empowers practitioners, scientist, patients and general public to address their needs in a more relevant way. Some of the existing agent-based systems are designed to use information specific to a medical and health organizations; others use information from the Internet.
Organization-based multi-agent systems are designed for to be used within a specific institution. Usually, these systems help manage and use the existing information. For example, Agent Cities [3] is a multi-agent system that consists of agents that provide medical services. The agents enable the user to access his/her medical record, to make a booking to be visited by a particular kind of doctor and to search for medical centres on the basis of a given set of requirements. AADCare [4] agent-based system is designed as a decision support system for physicians. It has the ability to map patient’s records to the predefined domain knowledge such as disease knowledge, clinical management plans, patient records etc.

Other multi-agent systems retrieve information from the Internet. BioAgent [5] is a mobile agent system where each agent is associated with a given task. For successful completion of its task, the agent needs to travel among multiple locations. At each location, the agent performs a set of actions. Information integration procedure takes place at the end of the trip. Holonic Medical Diagnostic System (HMDS) [6] architecture is a medical diagnostic system i.e. Internet-based diagnostic system for diseases. It is based on the holonic paradigm, multi-agent system technology and swarm intelligence. The patient’s information is kept in comprehensive computer readable patient record called computer readable patient pattern (CRPP). The agents of the holarchy use the information available via Internet to processed CRPP. Another work describes web crawling agents [7] that can be designed to fetch information about diseases on the basis of the information about mutated genes related to these diseases.

Use of the ontologies within biomedical communities is of great significance [8]. The Gene Ontology (GO) (http://www.geneontology.org/) has been used to annotate major repositories for plant, animal and microbial genomes. This has resulted in consistent descriptions of gene products in different databases. The Unified Medical Language System (UMLS) [8] contains 1 million biomedical concepts, 135 semantic types and 54 relationships used to classify the UMLS concepts. Human Disease Ontology [10] captures and represents knowledge about human diseases according the four different dimension: disease types, symptoms, causes and treatments subontologies. Protein Ontology (http://proteinontology.info/) [11] enables capturing of declarative knowledge about protein domain and classification of that knowledge to allow reasoning. It enables access not only to related data but also semi-structured data such as XML or metadata annotations and unstructured information. A large number of biomedical ontologies is available via The Open Biomedical Ontologies (http://obofoundry.org/). These ontologies cover various knowledge domains such as anatomy, biological processes, biochemistry, health and taxonomy.

In our project we focus on making a channel through which dispersed health information will be unified under one umbrella to help identify interdependencies and interrelationship between different aspects of health. Lots of the information is available but, due to the large body of information, some important information may escape the users notice and be neglected.

The Web crawling agent and BioAgent system could be used by our system with some modifications. We can use the same principle of fetching information, agent migration among multiple locations, information retrieval from each location and information integration at the end of the trip. But the information we are
interested in is not regarding genetic causes of human diseases and genome analysis but regarding total wellbeing of individuals. Jointly with the founder of the HMDS idea [6], a system has been developed for the retrieval of information about human diseases [12]. In this paper we look at the problem from a different point of view. We are examining health not only as a function of physical and mental health but also as a function of relational, emotional, spiritual and financial wellbeing as defined by the Total Wellbeing Ontology [45]. No multi-agent system exists yet to meet the needs of such a system.

3 A Holistic Approach to Health

The revolutionary technology development has resulted in rapid introduction of cutting-edge technologies into our societies. We became very dependant on the high technologies and comfort they brought. However, it has been evidenced that this material comfort has failed to bring us better health, greater inner peace and a fuller sense of meaning, purpose and satisfaction [13]. While the lives of individuals may have become better, evidence [13] suggests that general health and wellbeing of our societies became worse. Since 1960:

1. divorce rate has doubled
2. teen suicide rate has tripled
3. recorded violent crime rate has quadrupled
4. prison population has quintupled
5. percentage of the babies born to unmarried parents has increased six fold, and
6. cohabitation (a predictor of future divorce [14]) has increased sevenfold.

Moreover, it appears that the occurring problems are increasing over time, and are gaining a momentum rather than being random events. It has been predicted that depression will be the world’s leading cause of disability by 2020 [15].

Staying ignorant in this situation does not seem like a wise choice. We have to examine our health more closely and become more aware of different factors that are affecting our health both positively and negatively. The total wellbeing is not only defined by our physical health but the end product of the interplay of the physical, mental, emotional, financial, relational and spiritual events of a lifetime [45]. This framework can help us precisely define and measure total wellbeing of individuals in our modern society. In this section we give some background to this approach.

3.1 Physical and Mental Health

A strong association between mental perceptions and physical health has been demonstrated by numerous studies. Continued expose to stress can put a strain on various organs, leading to systems breakdown, compromised immune response and ultimately the deterioration of physical health [16, 17]. It has been reported that some chronic physical illnesses such as asthma [18], diabetes [19] and cardiovascular disease [20] are linked to mental illnesses such as depression. The relationship between physical and mental health is bidirectional. Physically active
people tend to have better mental health [21]. It has been shown that the physically active people have scores for positive self-concept, more self-esteem and more positive "moods" and "affects." For this reason, physical activity has been successfully used as a non-pharmacological treatment for depression [22, 23]. Additionally, physical activity may reduce the symptoms of anxiety, improve social skills and cognitive functioning, and be a beneficial adjunct for alcoholism and substance abuse programs [24].

3.2 Financial, Physical and Mental Health

Studies of the Data from the National Health Interview Survey, the National Survey of Families and Households, the Survey of Income and Program Participation indicate that increases in income significantly improve mental and physical health [27]. However, the same study claims that increases in income increase the prevalence of alcohol consumption which in its turn may be damaging to mental and physical health. Cross-national comparison studies have exposed the evidence for positive relationship between per capita income and health [25]. They report an elevated level of wellbeing from countries high in per capita income. Positive relationship between happiness and per capita income has emerged from a comparison study from seven parts of the world [26]. However, inconsistencies in the data were also evident. For example, Latin America scored almost as high as Western Europe, which has more than double the Latin American per capita income. We conclude that even though financial factors play a large role in total wellbeing, there are other factors which play a significant part in contributing to total wellbeing.

3.3 Social, Physical and Mental Health

Prospective studies exposed the evidence that persons with unhealthy social relationships have increased risk of death [28]. Similar results have been obtained through experimental and quasi-experimental studies of humans and animals. While House et al. [28] report increased mortality rate in isolated individuals and Kawachi & Berkman [29] highlight beneficial roles of social ties play on the mental wellbeing, not all relationships are beneficial. Some relationships are destructive and it is better to avoid them. Again we notice bidirectional relationship between the two different dimensions, namely, between social relationships and health. Rolland [30] studies effect of illness or disability on a couple's relationship. Such situations often put the relationship out of balance and consequently result in dysfunctional relationship patterns.

3.4 Emotions, Physical and Mental Health

The effect of emotions on physical health has also been examined. Evidence had been exposed that the emotional stress such as anxiety has a negative impact on immunity [31]. Another study has found anxiety to be a significant and independent predictor of cardiovascular disease [32]. Dr Colbert [33] defines the destructive emotions in terms of their origin, nature and manifestations. He also explains
negative effect of these emotions on our health. Their toxic effect may result in a variety of illnesses including hypertension, irritable bowel syndrome, arthritis, multiple sclerosis and some types of cancer.

3.5 Physical and Spiritual Health

Powell et al. [34] report positive association between physical health and spirituality. The researchers state that the risk of mortality is reduced by 25% in church/service attendees. They provide additional evidence that the religion or spirituality protects against cardiovascular disease through the healthy lifestyle the doctrine encourages. Dr Wright [35] discusses and explains spiritual roots of numerous diseases, possible spiritual blocks to healing and spiritual pathways to health and wellbeing.

3.6 Mental and Spiritual Health

Dr D’Souza [36] claims that the majority of mentally ill patients rate spirituality as very important and desire their therapist to take their spiritual needs into consideration. From this study it appears that majority of the mentally ill patients are spiritual in one way or the other. One would ask here if the spirituality triggered the mental disorder in the first place. Bergin [37] addresses this issue as he evaluates the effect of spirituality on mental health. He makes it clear that religion can have both positive and negative effect. The negative situations are marked by evil being clothed in religious language, namely, misuse of religion for personal interests and gains. The positive impact is marked by a personalized experience, a true personal conviction or commitment and is manifested in dramatic personal healing or transformation.

The physical, mental, financial, social, emotional and spiritual wellbeing are interconnected and mutually dependent on one another. For this reason, we need to approach total wellbeing as being a function of multiple dimensions. This necessitates analysis of each dimension individually and in relation to other dimensions. We can find no evidence that such multidimensional research has ever been performed. In the following section we describe a multi-agent system that can help the integrated study of the total wellbeing and help us gain a clearer understanding of the interrelationships and interdependencies between the six dimensions.

4 A Holonic Multi-agent System for Total Wellbeing

In this section, we describe a multi-agent system that can be used to intelligently retrieve, manage and use information about total wellbeing. In this system, total wellbeing is seen as a function of physical, mental, relational, emotional, financial and spiritual wellbeing. The TICSA approach, described in [38], can be used to design the multi-agent system. The TICSA methodology consists of the following five steps:

1. Identify Agent Types According to Their Responsibilities
2. Define Agent’s Intelligence
3. Define Agent’s Collaborations
4. Protect the System by Implementing Security Requirements
5. Assemble Individual Agents

4.1 Identify Agent Types According to Their Responsibilities

In this step it is important to identify specific agent types and corresponding function required to enable intuitive flow of problem solving, task and result sharing. A system composed of various agent types with different but complementary functions performs the best. Usually, the different agents work on the different aspects of the overall problem and solve complex problems through their synergetic effort.

In our system, different types of agents identify relevant information within information resources, retrieve this information, store and manage it within a dedicated database, and intelligently retrieve specific information when requested by a user. We use three different types of agents:

1. **Information agents**, to identify and retrieve relevant information from information resources;
2. **Intelligent agent**, to systematically store the retrieved information in accordance with the total wellbeing dimensions and intelligently manage this information;
3. **Interface agent**, to assist the users with formulating queries and present the requested information in meaningful ways.

Additionally, the system can be empowered with Data Analysis agents such as Data Mining agents, which will analyse the retrieved information and identify possible patterns in the data. Successful implementation of data mining algorithms has been illustrated within mental health domain [39, 40, 41, 42, 43]. Due its complex and multi-dimensional nature, a similar approach can be taken when mining total wellbeing knowledge.

4.2 Define Agent’s Intelligence

The agents of the system need to be equipped with the knowledge that will enable them to perform their task intelligently. They need to be able to identify and retrieve relevant information, to systematically store and manage information, to communicate with each other, etc. The knowledge base has been predominantly used to provide agents with intelligence and enable them to perform their action efficiently and effectively. Some researchers prefer use of ontology over knowledge base as the ontology is a more expressive knowledge model [44]. Ontology captures and represents specific domain knowledge through specification of meaning of concepts. This includes definition of the concepts as well as domain-specific relationships between those concepts.

We will use a combination of knowledge bases and ontologies in our system. The Total Wellbeing Ontology is composed of six sub-ontologies: Physical, Mental, Relational, Emotional, Financial and Spiritual Wellbeing [45]. This ontology structure will define the way retrieved information is being stored within the
system. The systematic and meaningful organization of the information will enable efficient and effective retrieval of the information as well as meaningful representation of the information to the user.

Note that the Total Wellbeing Ontology will evolve as more knowledge on this topic becomes available. We aim to combine Data Mining and Multi-agent technologies to design a system that will help to update the Total Wellbeing Ontology on a regular basis.

4.3 Define Agent’s Collaborations

In the first stage of the TICSA design methodology we identified different types of agent in line with their different functions within the multi-agent system. In this stage we will define structural organization and position of agents within the system. We need to create system architecture that will enable the most optimal performance of agents. The agents have to be organized in such way that the problem solving process can easily flow towards its completion and the communication between different agents can be easily established at any stage of the processing cycle. Addressing those two issues effectively has the potential to greatly contribute to the efficient and effective performance of the system. In some cases, a system with a simple organization of agents functions the best. In other cases it may be required to design a more complex system to address the needs of complex problem at hand.

Similar to work described in [12], we propose a Holonic Multi-agent Structure (shown in Figure 1) as a nested hierarchy of six holarchies in which each of the six Total wellbeing Ontology dimensions (Physical, Mental, Relational, Emotional, Financial and Spiritual Wellbeing) is associated with one holarchy. The holarchy agents will have the role of Information agents and will retrieve information specific to their holarchy. They will search for information to reveal interrelationships between the different dimensions. For example, agents of Physical holarchy will retrieve information regarding relationships between Physical dimension and any other dimension defined by the Total Wellbeing Ontology. Note that it will not retrieve information about Physical dimension alone as that is not the purpose of this project.

All the information retrieved by the Information agents of the six holarchies will be forwarded to the Intelligent agent. The Intelligent Agent has three main functions. Firstly, it needs to validate the information received from the Information agents. Some of the retrieved information may come from web pages that contain personal opinions and not scientific results of evidence-based studies. Invalid information may be kept separately for referencing purpose but will not be incorporated into the dedicated database until more evidence is found to support that statement. Second function of the Intelligent agent is to systematically organize the retrieved information within a dedicated database. The agent uses the Total Wellbeing Ontology for this purpose. Third function of the Intelligent agent is to retrieve specific information when requested by the Interface agent.
The Interface agent assists the user in formulating queries. It sends the request to the Intelligent Agent. The Intelligent agent retrieves the specific information. Optionally, it may need to activate Information agents to search for more evidence. Intelligent agent forwards information of interest to the Interface agent. Interface agent uses the Total Wellbeing Ontology to present this information to the user in a meaningful way.

4.4 Protect the System by Implementing Security Requirements

The security criteria must be addressed during the development of a multi-agent system. The aim should be to provide as much security as possible. This can be achieved through precise identification of potential security threats, addressing effectively those issues and implementing them within the multi-agent system.

The six security properties [46] of confidentiality (assuring that the information is only available to authorized agents), authentication (verifying identity of an agent), integrity (guaranteeing that the retrieved information is identical to the source information), access control (identify access rights of an agent), non repudiation (verifying involvement of an agent in certain communication) and availability (assuring ease of access and use of information and resources to authorized agents). We have identified some additional characteristics of agents that can contribute to increased security of the system from the internal point of view. These include:

(1) Compliance. The designer team has created the agents and the overall system, and is best positioned to identify and specifically define the set of rules and regulations that must be closely followed by all agents of the system. The agents that refuse to follow the rules and regulation are putting in danger not only their own existence but, if ignored, are becoming a serious threat to the whole system.
(2) Service. The multi-agent system is composed of various agents with different but complementary capabilities. The expertise complementarity is a feature that holds the agent society together. Agents are dependent on each other for mutually beneficial actions and accomplishing goals of the overall system. The agents should be able to work together and provide service to one another. The agents that refuse to collaborate should be replaced by agents willing to provide service to others.

(3) Commitment. The agents of the system must be completely committed to the overall goal and purpose of the multi-agent system. Some agents may be partially committed and reliance on such agents may be risky. In worst cases, agents may commit to the overall goals and purposes of other malicious multi-agent systems and turn against the multi-agent system they were originally created to work for.

The nine identified properties are critical inside as well as outside the multi-agent system, such as during the interaction of the multi-agent system with the environment. It is necessary to effectively address these issues and efficiently implement them within the multi-agent system. In our multi-agent system, all nine requirements must be addressed. For example, in regard to the integrity requirement, the information provided to the Intelligent agent by the Information agents should be identical to the information found in the information resource from which this information was retrieved. In the same way, information provided to the Interface agent by the Intelligent agents should be identical to the information found in the database (as well as the information found in the original information resource). In analogous way we can address other security requirements.

4.5 Assemble Individual Agents

We have discussed different functions and types of agents, their intelligence, system architecture and security requirements that must be addressed within the system. In this step we bring the different aspects together and create the different agents defined by the previous steps.

It is important for each agent to identify required agent components, design the components and assemble them into an unified system i.e. individual agent. For example, these agent components may be Human Interface (to enable interaction with users e.g. interaction between Interface agent and users), Agent Interface (to enable interaction with other agents e.g. interaction between Interface & Intelligent agents and Intelligent & Information agents), Communication component (to enable exchanging and processing messages during agent’s interactions), Task component (to enable the agents to perform tasks assigned to them), Domain component (Total Wellbeing Ontology), Environment knowledge (to describe environment in which the agents perform their functions), History files (past experiences of agents), and so on.

It is possible that to achieve variety of agents within a multi-agent in three different ways:

(1) using same components but changing the content of these components
(2) changing combination of the used components
(3) changing both content and combination of the used components.
Completing the Total Wellbeing Puzzle Using a Multi-agent System

Interface, Intelligent and Information agents are very different from each other so they can be designed using the third approach. However, the Information agents are very similar to each other for the reason that they perform same tasks but within different domains, as determined by the holarchy to which they belong. They can be designed using the first approach.

5 Conclusion and Future Works

The technology breakthrough, the modern way of living and modern societies have not only resulted in comfortable lives but also in increased stress, pressure and fears which in their turn started degrading our health and wellbeing. More than ever, we need to take a holistic approach in studying and controlling our health and wellbeing. We use an evidence-based ontological model that defines the total wellbeing as a function of the physical, mental, emotional, financial, relational and spiritual wellbeing. We have used the TICSA (Types, Intelligence, Collaboration, Security and Assembly) approach to design a multi-agent system for intelligent retrieval, management and use of information related to the six different dimensions of the total wellbeing and their relationships. This system has the potential to expose the evidence that will change the way we understand, control and manage our health. It will support general public in managing their personal wellbeing better and health professionals in adapting their services to address patients’ needs more systematically and effectively. We are in early implantation stage of the system, and our progress will be reported in the forthcoming papers.

References

The Minimal Model of Glucose Disappearance in Type I Diabetes

Margarita Fernandez¹, Minaya Villasana², and Dan Streja³

¹ Depto. de Cómputo Científico y Estadística, USB, AP 89000; Caracas 1080-A; Venezuela
margarita@cesma.usb.ve

² Depto. de Cómputo Científico y Estadística, USB, AP 89000; Caracas 1080-A; Venezuela
mvillasa@usb.ve

³ David Geffen School of Medicine, UCLA, Los Angeles, USA
dstreja@ucla.edu

Summary. In this chapter we evaluate the ability of the minimal model of glucose disappearance to describe experimental data collected from 9 diabetic patients controlled subcutaneously by an insulin pump. Two versions of the minimal model are used: the nonlinear classic minimal model developed by Bergman et al. (MM) and the linear approximation proposed by Fernandez et al. (LMM). All data windows (n = 13) show residuals that are correlated for both the LMM and MM (p-value < 0.01). The results also show that both the LMM and MM provide an equivalent goodness of fit with $R^2$ values that are statistically equivalent (p-value > 0.05). This study confirms that the minimal model of glucose disappearance, either the classic or linear version, is unable to describe the observed experimental data possibly as a result of the physiological constraints imposed by the minimal model approach on the system dynamics, together with possible errors derived from the unmeasured insulin dynamics. Further testing on more complex models should be performed.

1 Introduction

The aim of the present study has been to evaluate if the minimal model of glucose disappearance is able to describe the experimental data collected from 9 diabetic patients being controlled subcutaneously by an insulin pump.

On previous research the authors have shown the inability of the simple two compartmental model known as the minimal model [3] to follow these experimental data [14]. This model was included in the study to describe the dynamics of glucose, and two input models were also supplied [14]:

- An interstitial insulin absorption model proposed by Shichiri et al. [18] to account for the dynamics of insulin supplied subcutaneously by the insulin pump.
- A model published by Radziuk et al. [17] to describe the rate of appearance of the external glucose following ingestion.

In this respect, the chosen input models have been identified by the authors as possible sources of error and further work has been proposed to improve the results.
obtained [14]. A summary of the limitations of each input model and the proposed improvements is as follows [14]:

- The rate of glucose absorption model could be oversimplified. For glucose loads different from 45 or 89 g, we took as valid the curve that corresponds to the load closest in value to these [14] (e.g., if the actual glucose load was 15 g then the curve obtained for 45 g was chosen as the rate of appearance). This approximation would introduce unavoidable errors until better estimates could be obtained. For this reason, a more recent model presented by Lehmann and Deutsch [16] is being included in the present work instead.

- The insulin absorption model could be too simple to describe the data under study [14]. A recent publication by Willinska and coworkers [19] shows that a model like Shichiri’s model underestimates the post meal peak of plasma insulin, whilst improvement in the model fit was enhanced when two absorption channels were included. This latter model has been used in the present study in substitution to Shichiri’s model.

These new input models are of superior quality in terms of data fitting and it is expected that they will help to improve the ability of the minimal model in following the experimental data.

2 Data

Glucose data collection was achieved through a continuous glucose monitoring sensor (CGMS, Medtronic MiniMed, Northridge, CA). This technology allows 5 min timed testing of interstitial glucose. The instrument is periodically calibrated using patient’s plasma glucose and the data are transformed to accurately and timely reflect patient’s serum glucose. The instrument’s precision has been tested for concentrations ranging from 40 to 400 mg/dL.

Nine patients with longstanding Type I diabetes (C peptide negative) being controlled subcutaneously by an insulin pump participated in the trial. After appropriate instructions, the CGMS was inserted into the subcutaneous abdominal fat tissue and calibrated over a 60 min period, as per standard Medtronic MiniMed operating guidelines. The patients were monitored for 72 hours. At the time of the instructions the patients were asked to record the time of administration of subcutaneous insulin, the insulin bolus injections and basal rates and the amount of carbohydrate ingested. All patients were previously instructed in evaluating the quantity of carbohydrates and administered fast acting insulin.

After 5 days the patients returned, and the data was downloaded into the computer. Time series were obtained of interstitial glucose every five minutes, along with the times at which the patient recorded events such as: bolus insulin injection times and amounts, glucose ingestion and approximate loads, and the times of exercise initiation. The glucose data was transformed in serum glucose values with Medtronic-Minimed software version 3.0.

Subsequently, data windows of 300 min each were extracted from the time series upon the initiation of a meal. If the patient indicated food ingestion or more than two
insulin self-administration during the window, those windows were excluded from the analysis. The 300 min time frame was chosen, since this time seems appropriate for an oral glucose tolerance test (OGTT) to reach its steady state.

3 Background

The integrated mathematical model used in this research has been represented in Fig. 1, where glucose dynamics is described by the well known minimal model of glucose disappearance [3]. This model consists of two compartments that involve plasma glucose concentration $y_1$ and the action of insulin $x$, and is detailed in Section 3.1.

Since in Type I diabetes there is very little insulin production, plasma insulin is obtained through the external administration of insulin (fast acting or monomeric) in the interstitium. Infused insulin stored in the interstitial compartment is represented by $q_x$, $q_y$ in the insulin absorption model and $q_z$ accounts for the subcutaneous distribution pool. The dynamics of subcutaneous insulin absorption is being modelled as detailed later on in Section 3.4 which in turn hands out plasma insulin concentration $y_2(t)$, regarded as an input to the glucose dynamics model. The insulin infusion rate IR, as recorded in the patient’s log of events, is supplied through the interstitial compartment.

The integrated model also includes the rate of appearance of absorbed glucose $R_a(t)$ as an external input following the glucose loads recorded by the patients. This rate of absorption is obtained through a modelling function subsequently described in Section 3.3 and represented as input $u_1(t)$ in the glucose model.

There are two versions of the minimal model that have been used by the authors in order to see if both models perform in a similar way for the data under study. They are the nonlinear classic minimal model developed by Bergman et al. [3] and
the linear approximation proposed by Fernandez et al. [15]. Their equations are described in the next sections.

### 3.1 The Classic Minimal Model

The interaction between glucose and insulin has been modelled using the minimal model (MM) of Bergman et al. [3]. This model includes two compartments and its uniquely identifiable parameterisation is described by:

\[
\dot{y}_1 = -S_G y_1 - x y_1 + p_0 + u_1(t), \quad y_1(0) = y_{10} \tag{1}
\]
\[
\dot{x} = -p_2 [x - S_I (y_2(t) - y_{20})], \quad x(0) = 0 \tag{2}
\]

where \(y_1\) (mg/dL) is glucose concentration in plasma, \(x\) (1/min) is insulin action, \(y_2(t)\) (μU/ml) is insulin plasma concentration, \(y_{10}\) is basal glucose concentration, \(y_{20}\) is basal insulin concentration, \(u_1(t)\) (mg/dL per min) is external glucose, \(p_0\) (mg/dL per min) describes insulin action, \(p_2\) (1/min) is the extrapolated hepatic glucose production at zero glucose concentration and \(S_G\) (1/min) and \(S_I\) (1/min per μU/ml) are parameters of glucose effectiveness and insulin sensitivity respectively.

The model has been written so that the constants \(S_G\) and \(S_I\) are physiologically meaningful. In this case, \(S_G\) measures the effect of glucose on its own disappearance, while \(S_I\) measures the effect that insulin has on the disposal of glucose.

Model output is given by the concentration of glucose in blood \(y_1\). Inputs to this model include the extrapolated hepatic production of glucose given by \(p_0 = S_G y_{10}\) from steady state constraints and two external inputs which in the present study are represented by:

1. The plasma insulin \(y_2(t)\) present at time \(t\), due to the injection of monomeric insulin analogues subcutaneously, where \(y_2(t)\) is obtained from the model of absorption formulated by Willinska and coworkers [19] and described in Section 3.4.
2. The rate at which the external (ingested) glucose is absorbed \(u_1(t) = R_a(t)/V_1 \times 10^2\), where \(V_1\) (ml) is 20% of the patient’s body weight [6]. The rate of absorption \(R_a(t)\) is estimated from the model of glucose absorption proposed by Lehmann and Deutsch [16] as presented in Section 3.3.

Initial values for parameters \(S_G\) and \(S_I\) of the MM were taken from [5] for the normal man after an OGGT/MGTT. The initial value of \(p_2\), the insulin action parameter, was taken as reported for the IVGTT in the normal man [8].

### 3.2 The Linear Minimal Model

This model is formally equivalent to the nonlinear minimal model presented by Bergman et al. in 1979 [3] and has been successfully validated in previous studies, including Type I [10, 14, 9] and Type II [15, 12] diabetic patients. It has the advantage of its linear structure which makes it simpler and a good candidate for the closed loop control of glucose.
This model includes two compartments and it has been reparameterized to better reflect its relationship with the MM as follows:

\[ \dot{y}_1 = -S_G y_1 - x_L + p_0 + u_1(t), \quad y_1(0) = y_{10} \] (3)

\[ \dot{x}_L = -p_2 [x_L - S_L (y_2(t) - y_{20})], \quad x_L(0) = 0 \] (4)

As in the MM, the model output is given by the concentration of glucose in blood \( y_1 \) (mg/dL) (with \( y_{10} \) as the basal value), \( p_2 \) (1/min) describes the insulin action, and \( S_G \) (1/min) is the parameter of glucose effectiveness. Insulin action \( x_L \) is in (mg/dL per min) and insulin sensitivity index \( S_L \) is in (mg/dL per min per \( \mu U/ml \)).

Here inputs to this model also include the extrapolated hepatic glucose production given by \( p_0 = S_G y_{10} \) (mg/dL per min) from steady state constraints and two external inputs: the plasma insulin \( y_2(t) \) (\( \mu U/ml \)) obtained from the Willinska’s model of absorption (with \( y_{20} \) as the basal value) and the rate at which the external glucose is absorbed \( u_1(t) = R_a(t)/V_1 \times 10^2 \) (mg/dL per min) with \( R_a(t) \) estimated from Lehmann and Deutsch absorption model.

The LMM has been previously identified from the IVGTT in the normal man \([15, 10]\). However, the data set under the present study correspond to the dynamics of glucose after an external stimulus such as the oral glucose tolerance test OGGT. Therefore, identification of the LMM model was performed from the dynamic response of Cobelli’s model, following the standard OGGT (100 gr. of glucose load) on the average normal man \([7]\), to get initial estimates that better reflect the experiment under study. Parameter estimates were obtained by means of linear regression methods between the model dynamic response to the test and the variables of interest: plasma glucose and interstitial insulin \([11]\).

### 3.3 Rate of Appearance of External Glucose

It is the authors’ concern that the rate of absorption model could be oversimplified. The rate of absorption used initially in \([14]\) was obtained by digitising the rate of absorption curves given in the article by Radziuk et al \([17]\). The aforementioned curves correspond to an ingestion of 45 and 89 g of glucose load. The data was then interpolated to obtain smooth estimates in time for the rate of absorption. When the patient reported glucose loads different from 45 or 89 g, we took as a valid approximation the curve that corresponds to the load closest in value to these.

Fig. 2 shows the two rates of absorption for the known glucose loads (45 and 89 g). In a sense there is no model as such, but rather a lookup table with only two entries. This severely limits the quality of the approximations as only two possible loads were permitted and all other glucose ingestion loads were approximated by these. This approximation would introduce unnecessary errors in the model. For this reason, a more recent model presented by Lehmann and Deutsch \([16]\) is being included in our work instead.

In their article Lehmann and Deutsch present an integrated model for glucose and insulin interaction in Type I diabetic patients considering the action of external insulin and carbohydrate absorption. From their model we took the model for gastric
emptying of glucose following ingestion of a meal containing $Ch$ millimoles of glucose equivalent carbohydrate. The gastric emptying (the rate of absorption) is given as a function of time. The model depends on the carbohydrate intake as follows: if the load is greater than 10 g, then the model assumes a trapezoidal shape, meaning that there is a period of maximal emptying. If on the other hand the load is less than 10 g, then no such maximal rate is seen.

Lehmann and Deutsch provide a model expressing the load in terms of millimoles of carbohydrate. We have approximated the specific weight of glucose to be 180 in order to get $Ch/180 \times 10^3$ mmols of substance. Thus, by multiplying the gastric emptying function provided in [16] by $180 \times 10^{-3}$ we obtain the rate of absorption in terms of g/hour.

The equations that define the rate of absorption following this model, including a correction made by the authors to accurately account for the total amount of glucose ingested (see [13] for details), with $t$ measured relative to ingestion time (at ingestion $t = 0$) and $V_{\text{max}}$ equal to 120 mmol/h, are as follows:

- If the load is less than 10 g then:

$$R_a(t) = \begin{cases} 
(V_{\text{max}}/T_a)t & : t < T_a \\
V_{\text{max}} - \frac{V_{\text{max}}}{T_d} (t - T_a) & : T_a < t < T_a + T_d \\
0 & : \text{otherwise}
\end{cases}$$

where $T_a = T_d = \text{load}/V_{\text{max}}$.

- If the load is greater than 10 g then:

$$R_a(t) = \begin{cases} 
(V_{\text{max}}/T_a)t & : t < T_a \\
V_{\text{max}} & : T_a < t < T_a + T_d \\
V_{\text{max}} - \frac{V_{\text{max}}}{T_d} (t - T_a - T_{\text{max}}) & : T_a + T_{\text{max}} \leq t < T_{\text{max}} + T_a + T_d \\
0 & : \text{otherwise}
\end{cases}$$

Here $T_a = T_d = 0.5$ h, and $T_{\text{max}}$ is given by the expression

$$T_{\text{max}} = \frac{[\text{load} - V_{\text{max}}(T_a + T_d)]}{V_{\text{max}}}.$$
In Fig. 3 we can see the different functions for glucose absorption obtained by Lehmann and Deutsch model, according to the amount of glucose ingested (say 45, 89, 5 and 9 g respectively). We see that the functional form changes depending on the critical load and that the predicted total amount of glucose ingested (area under the curves) is approximately the same as the corresponding glucose load. This model is an improvement in respect to Radziuk’s model in that it accounts for specific glucose loads even though the functional forms are simpler. Using a rough numerical integrator (lower Reimann sums) for the curves depicted using Radziuk’s estimate, we obtain that the predicted loads are approximately 42 and 87 g, whilst for the model given by Lehmann and Deutsch the curves predict overall a glucose load very close to the actual load reported by the patients.

### 3.4 Insulin Absorption Model

As previously stated, the insulin absorption model used in [14] is a model provided by Shichiri and colleagues [18]. Their subcutaneous insulin absorption model
consists of 3 compartments representing the subcutaneously infused insulin pool, the subcutaneous insulin distribution pool and the plasma insulin space.

In this model, the equations that govern the dynamics for the subcutaneous insulin absorption are given by:

\[
\dot{q}_x = IR(t) - k_{12}q_x, \quad q_x(0) = 0 \tag{5}
\]
\[
\dot{q}_y = k_{12}q_x - (k_{20} + k_{23})q_y, \quad q_y(0) = 0 \tag{6}
\]
\[
\dot{q}_z = k_{23}q_y - k_{30}q_z, \quad q_z(0) = 0 \tag{7}
\]
\[
y_2(t) = q_z/V \tag{8}
\]

where \(IR(t)\) is the insulin infusion rate expressed in \(\mu U/min\); \(q_x\) and \(q_y\) represent the insulin quantities in the subcutaneously infused insulin pool and in the subcutaneous insulin distribution pool, respectively, and are given in \(\mu U\); \(q_z\) is the plasma insulin quantity in \(\mu U\); \(k_{12}, k_{20}, k_{23}, k_{30}\) are rate constants in \(\text{min}^{-1}\); \(y_2(t)\) is the plasma insulin concentration in \(\mu U/ml\) and \(V\) is the plasma insulin volume expressed in ml.

In [18] the constants of this model were estimated by using a nonlinear least squares method to fit the data obtained in ten Type I diabetic subjects treated with Pro(B29) human insulin (Insulin Lispro, U-40, Eli Lilly Co., Indianapolis, IN, U.S.A.), which is a fast acting insulin. To calculate each constant, 0.12 U/kg of Lispro insulin diluted to the concentration of 4 U/ml with saline was subcutaneously injected into the abdominal walls of the patients. This experiment resulted in:

\[
k_{12} = 0.017
\]
\[
k_{30} = 0.133
\]
\[
k_{20} = 0.0029
\]
\[
k_{23} = 0.048
\]

and \(V\) was estimated as \(V = 0.08\) ml per body weight in g.

The initial values for this model have been assumed to be \(q_x(0) = q_y(0) = q_z(0) = 0\) in our simulations. However, the patients’ initial condition in each window is

![Graph](image)

\(\text{(a) Injection of 0.12 (U/kg) at } t = 0\)

\(\text{(b) Injection of 0.12 (U/kg) at } t = 0\) and basal rate of 0.0212 (U/min)

**Fig. 4.** Simulation results for plasma insulin concentration \(y_2(t)\), according to the model by Shichiri et al.
affected by the amount of insulin supplied in the previous window. Therefore, an initial zero level in the insulin compartments is regarded as an approximation. The simulation results can be observed in Fig. 4 for a bolus injection of 0.12 U/kg at \( t = 0 \) and for the same bolus injection plus a basal rate of 0.0212 U/min.

Regarding the validity of this model, a recent publication [19] shows a model like Shichiri’s model underestimates the post meal peak of plasma insulin, whilst improvement in the model fit has been enhanced when two absorption channels were included in a model formulated by Willinska and coworkers [19]. For this reason, this latter model is being used in the present study in substitution to Shichiri’s model.

This model has plasma insulin represented by a single compartment. Insulin in the interstitial compartment is decomposed into 2 compartments to describe the delay in insulin absorption and maintaining two pathways for absorption. The degradation of insulin is assumed saturable and is modelled by a Michaelis-Menten functional type. Following the original nomenclature, the model equations can be formulated as follows:

\[
\begin{align*}
\frac{dQ_{1a}}{dt} &= ku - k_{a1}Q_{1a} - LD_a \\
\frac{dQ_{1b}}{dt} &= (1-k)u - k_{a2}Q_{1b} - LD_b \\
\frac{dQ_3}{dt} &= k_{a1}Q_{1a} - k_{a1}Q_2 \\
\frac{dQ_3}{dt} &= k_{a1}Q_2 + k_{a2}Q_{1b} - k_eQ_3 \\
LD_a &= V_{\text{MAX},LD}Q_{1a}/(K_{M,LD} + Q_{1a}) \\
LD_b &= V_{\text{MAX},LD}Q_{1b}/(K_{M,LD} + Q_{1b}).
\end{align*}
\]

where \( Q_{1a} \) and \( Q_{1b} \) are in mU and stand for mass of insulin administered through continuous infusion and bolus injection respectively, \( Q_3 \) represents insulin mass in plasma expressed in mU, \( u \) is the insulin input in mU/min and \( LD_a \) and \( LD_b \) are
Fig. 6. Plot of the mean residuals for all data windows (n = 13), showing the difference between the LMM and MM approximation, for both the old and new modelling approaches (OIM and NIM respectively). They both show deviations that have a pattern.

the local degradation at injection site for continuous infusion and bolus injection respectively, given in mU/min; they are Michaelis-Menten type functionals defined through the parameters $V_{MAXLD}$ (mU/min), $K_{M,LD}$ (mU) which are saturation level and half maximal value of degradation respectively. Constants $k_e$, $k_{a1}$, and $k_{a2}$ are transfer rates in 1/min and $k$ is a dimensionless parameter. The output of this model is given by the plasma insulin $y_2 = Q_3/V$ (mU/L), with $V$ as the plasma insulin volume.

In Fig. 5 the simulation results of this model can be seen for a bolus injection of 0.12 U/kg and also for this bolus injection together with a basal rate of 0.0212 U/min.

4 Statistical Analysis

Only 13 data windows of 300 min from a total of 33 were selected for analysis. These windows were defined in our previous work [14] as acceptable windows as they fail to meet the following rejection criteria:
1. At least one of the coefficients of variation (CVs) was greater than 100%. The CV was estimated as the ratio between the standard deviation and mean value of each estimated parameter [4].

2. The $R^2$ measured was less than 0.80. This measure is interpreted as the fraction of the total variance of the data that is explained by the model, and the sum of squared residuals (SSR).

4.1 Goodness of Fit

The models were built in MATLAB/Simulink in a PC based environment and parameter estimation was performed using nonlinear least squares methods from MATLAB/Optimization Toolbox.

The goodness of fit was qualitatively assessed by plotting against time the predicted model response and the experimental data. A quantitative measure was also given by means of the $R^2$ value.

The Wilcoxon signed rank test was used to evaluate optimal parameter differences between the $R^2$ values obtained by the old and new input models approaches, for both the MM and LMM models. This test returns the significance for a test of the null hypothesis that the median difference between two samples is zero. The null hypothesis $H_0$ that the medians are not significantly different is accepted when the $p$-value is greater than the $\alpha$ significance level and is rejected otherwise.

4.2 Analysis of Residuals

The residuals were tested for randomness by means of the Anderson-Darling test [2] and plotted against time to detect possible outliers or systematic deviations.

The correlation of errors was also studied by computing the Pearson’s correlation coefficient between $e(1)...e(N-1)$ and $e(2)...e(N)$ for each data set, where $e(i)$ is the $i$-th residual and $N$ is the total number of points, and performing a $t$-test on the transformed coefficients [1].

5 Results

The output of the simulations are displayed in Fig. 7 to Fig. 19. In each figure the first subplot shows the amount of carbohydrate ingested (right panel) and the rate of glucose absorption $RA$ (left panel), the second subplot includes the bolus injection and continuous insulin infusion $IR$ (right) together with the simulated plasma insulin dynamics $y_2$ (left) and finally the third subplot shows the dynamics of plasma glucose concentration $y_1$.

Overall the precision of the model parameters is good (CV $<=$ 100) but the normalised plot of the residuals shows deviations that have a pattern. In Fig. 6 the residuals from simulations using the old glucose [17] and insulin absorption models [18], that we will refer to as the the old input models (OIM) hereafter, are shown in the first plot for both the LMM and MM. All windows (n = 13) for the
Fig. 7. LMM and MM fit to data obtained experimentally from patient 1, for the old (left) and new (right) model inputs.

LMM show residuals that are correlated (p-value < 0.01) and the Anderson-Darling test shows evidence that 9 out of 13 (69%) windows have random residuals (p-value < 0.05), whilst residuals for 4 windows (31%) are not random (p-value > 0.05). For the MM also correlation is found for all windows (p-value < 0.01), and the

Table 1. The $R^2$ values for the LMM and MM after they have been fitted to the experimental data using the old and new input models (OIM and NIM respectively)

<table>
<thead>
<tr>
<th></th>
<th>LMM OIM</th>
<th>NIM</th>
<th>MM OIM</th>
<th>NIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9895</td>
<td>0.9822</td>
<td>0.9927</td>
<td>0.9876</td>
<td></td>
</tr>
<tr>
<td>0.9562</td>
<td>0.9649</td>
<td>0.9537</td>
<td>0.9685</td>
<td></td>
</tr>
<tr>
<td>0.8405</td>
<td>0.9110</td>
<td>0.8463</td>
<td>0.8894</td>
<td></td>
</tr>
<tr>
<td>0.7845</td>
<td>0.7318</td>
<td>0.8611</td>
<td>0.7191</td>
<td></td>
</tr>
<tr>
<td>0.8312</td>
<td>0.8515</td>
<td>0.9870</td>
<td>0.8462</td>
<td></td>
</tr>
<tr>
<td>0.7607</td>
<td>0.8428</td>
<td>0.8100</td>
<td>0.8900</td>
<td></td>
</tr>
<tr>
<td>0.8002</td>
<td>0.6647</td>
<td>0.7907</td>
<td>0.6647</td>
<td></td>
</tr>
<tr>
<td>0.8727</td>
<td>0.8811</td>
<td>0.8687</td>
<td>0.8789</td>
<td></td>
</tr>
<tr>
<td>0.9692</td>
<td>0.9631</td>
<td>0.9719</td>
<td>0.9678</td>
<td></td>
</tr>
<tr>
<td>0.9392</td>
<td>0.9614</td>
<td>0.8635</td>
<td>0.8467</td>
<td></td>
</tr>
<tr>
<td>0.8753</td>
<td>0.5754</td>
<td>0.8710</td>
<td>0.5826</td>
<td></td>
</tr>
<tr>
<td>0.8750</td>
<td>0.5342</td>
<td>0.9044</td>
<td>0.5623</td>
<td></td>
</tr>
<tr>
<td>0.8128</td>
<td>0.8287</td>
<td>0.7868</td>
<td>0.8322</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 8. LMM and MM fit to data obtained experimentally from patient 2, for the old (left) and new (right) model inputs.

Fig. 9. LMM and MM fit to data obtained experimentally from patient 3, for the old (left) and new (right) model inputs.
Fig. 10. LMM and MM fit to data obtained experimentally from patient 4, for the old (left) and new (right) model inputs.

Fig. 11. LMM and MM fit to data obtained experimentally from patient 5, for the old (left) and new (right) model inputs.
Fig. 12. LMM and MM fit to data obtained experimentally from patient 6, for the old (left) and new (right) model inputs.

Fig. 13. LMM and MM fit to data obtained experimentally from patient 7, for the old (left) and new (right) model inputs.
Fig. 14. LMM and MM fit to data obtained experimentally from patient 8, for the old (left) and new (right) model inputs.

Fig. 15. LMM and MM fit to data obtained experimentally from patient 9, for the old (left) and new (right) model inputs.
Fig. 16. LMM and MM fit to data obtained experimentally from patient 10, for the old (left) and new (right) model inputs.

Fig. 17. LMM and MM fit to data obtained experimentally from patient 11, for the old (left) and new (right) model inputs.
Fig. 18. LMM and MM fit to data obtained experimentally from patient 12, for the old (left) and new (right) model inputs.

Fig. 19. LMM and MM fit to data obtained experimentally from patient 13, for the old (left) and new (right) model inputs.
Anderson-Darling test shows evidence that 7 out of 13 windows (54%) have random residuals ($p$-value < 0.05) in contrast with 6 (46%) of them ($p$-value > 0.05).

When using the new modelling approach, with the glucose absorption model of Lehmann and Deutsch [16] and the insulin absorption model from Willinska et al. [19], regarded as the new input models (NIM) hereafter, all windows ($n = 13$) again show residuals that are correlated for both the LMM and MM ($p$-value < 0.01). The Anderson-Darling test shows evidence for the LMM that 7 out of 13 windows have random residuals, whilst 6 have not, and this is 8 out of 13 for the MM ($p$-value < 0.05) with 5 showing a non-random pattern ($p$-value > 0.05).

If we compare the $R^2$ value for the LMM reported in columns 1 and 2 in Table 1 obtained for the old and new modelling approaches we can see that differences between the OIM and NIM are not statistically significant according to the Wilcoxon signed rank test ($p$-value > 0.05). Also, the test did not find evidence that the $R^2$ values for the MM reported in columns 3 and 4 are different for the old and new modelling approaches ($p$-value > 0.05).

This results suggests that in terms of fitting the new input models have no significant effect when compared to the old approach, and this is true for both the LMM and MM.

If we also compare the $R^2$ values of the LMM and MM together for the old modelling approach reported in columns 1 and 3 in Table 1 the test also confirms these values are statistically equivalent ($p$-value > 0.05). The same applies to the new modelling strategy between columns 2 and 4 in Table 1.

According to this analysis both the LMM and MM provide a goodness of fit that is equivalent in terms of the $R^2$ measure, for both the old and new input modelling approaches.

The results obtained highlight the minimal model limitations in following the experimental data despite the fact that new models of superior quality in terms of data fitting have been included. According to this, the lack of insulin measurements together with the minimal model physiological constraints on the system dynamics can be said to be the possible sources of error. However, the fact that residuals still show a systematic pattern could also be explained by the presence of structured noise. A preliminary study has been carried out in this respect but conclusions can not be drawn at the present stage.

Further studies should be carried out for better models to be proposed and/or structured noise to be removed.

### 6 Conclusions

First of all, this study confirms that the minimal model of glucose disappearance, either the classic or linear version (MM and LMM respectively), is unable to describe the observed experimental data possibly as a result of the physiological constraints imposed by the minimal model approach on the system dynamics, together with possible errors derived from the unmeasured insulin dynamics. If including insulin measurements or a complex model of the glucose dynamics would result in a better approximation remains to be tested.
The presence of structured noise could also explain the fact that residuals still show a systematic pattern. Further work should be carried out to test this hypothesis. In the second place, this study also confirms that the LMM is equivalent to the MM in terms of data fitting for the experimental data under study. This result, together with the experiments already reported in the literature, [10][14][9][15][12] they have all contributed to this linear approximation to be further validated.

References

Genetic Algorithm in *Ab Initio* Protein Structure Prediction Using Low Resolution Model: A Review

Md. Tamjidul Hoque¹, Madhu Chetty², and Abdul Sattar¹

¹ IIIS, Griffith University, Nathan, QLD-4111, Australia
² GSIT, Monash University, Churchill, VIC 3842, Australia

Tamjidul.Hoque@gmail.com, Madhu.Chetty@infotech.monash.edu.au, a.sattar@griffith.edu.au

Abstract. Proteins are sequences of amino acids bound into a linear chain that adopt a specific folded three-dimensional (3D) shape. This specific folded shape enables proteins to perform specific tasks. The protein structure prediction (PSP) by *ab initio* or *de novo* approach is promising amongst various available computational methods and can help to unravel the important relationship between sequence and its corresponding structure. This article presents the *ab initio* protein structure prediction as a conformational search problem in low resolution model using genetic algorithm. As a review, the essence of twin removal, intelligence in coding, the development and application of domain specific heuristics garnered from the properties of the resulting model and the protein core formation concept discussed are all highly relevant in attempting to secure the best solution.

1 Introduction

*Ab initio* protein structure prediction (PSP) is an important and very challenging interdisciplinary problem encompassing *biochemistry*, *biophysics*, *structural biology*, *molecular biology* and *computational biology* to give just a couple of examples. Structure prediction, especially in revealing the relationship between sequences and protein folding is the key to combating many diseases and the development of several crucial biotechnological applications and the *ab initio* approach in this regard offers great hope for improving the human condition. More than half of the dry weight of a cell is made up of proteins of various shapes and sizes and protein’s specific folded three-dimensional (3D) shape (Fig. 1) enables it to perform specific tasks. From the computing point of view, the exciting investigations concerning proteins is not necessarily about these molecules carrying out vital tasks but mainly about the process of its acquiring various shapes, i.e. protein folding problem, which enable it to perform the specific tasks. To solve the PSP problem, among other approaches nondeterministic searching approach Genetic Algorithms are found promising [1, 2, 3]. On the other hand, to model and to handle the complexity of the protein folding the low resolution model found [4, 5, 6] to be effective exploring the vast and convoluted search space in a reasonable time.
scale. The low resolution model aids in providing a valuable theoretical insight which is otherwise often very hard to extract in the high resolution model.

In this article, we prefer to provide a review to show how novel techniques can improve GA to handle the low resolution based PSP problem, which is yet too complex to be solved. Thus in Section 2, the conformational complexity of protein structure prediction has been discussed. Section 3 describes the modelling issue of the computational protein structure prediction. Section 4 discusses novel computational techniques to cope the low resolution model. The preference of the face-centred-cube (FCC) lattice configuration for the PSP problem has been advocated in Section 5 and in Section 6 a novel model, named hHPNX model, has been presented which can remove some limitations of the existing HP and HPNX model and thus provides better predictions. Finally, Section 7 draws the conclusions.

## 2 Conformational Complexity

Amongst the 20 different amino acids, any two can join themselves by forming peptide bond thus resulting in an amide plane (Fig. 2). Formation of peptide bonds and properties of amide plane are very important in providing specific shape to a specific polypeptide chain formed from the amino acid concatenation. The amide plane is rigid and dihedral angles, $\phi$ and $\psi$ provide flexibility in mobility about $2\pi$, around the N-C$_\alpha$ and C$_\alpha$-C connecting axis. Each of the amino acids can have large number of torsion angles $\chi$ (see Fig. 2) depending on the length of the side chain, however here we assume two per amino acid. To estimate the complexity and to test the feasibility of an exhaustive search algorithm can be considered by all possible combinations of the shape parameters (e.g., dihedral and torsion discrete angles); if there are $n$ numbers of residues in a particular sequence, the total number of conformations ($C_{Tot}$) can be expressed as:

$$C_{Tot} \approx (\phi_1 \times \phi_2 \times \ldots \times \phi_{(n-1)}) (\psi_1 \times \psi_2 \times \ldots \times \psi_{(n-1)}) (\chi_1 \times \chi_2 \times \ldots \times \chi_{2n})$$

However, in practice for sterical disallowance, due to the shape and size of the atoms and their positioning, some reduction in the degree of freedom is possible, which is commonly depicted by the *Ramachandran plot* [8]. Even though, the
search space remains astronomically large. For example, with tremendous simplification, assume each amino having only three discrete angles with three degrees of freedom, a 50 residue-long protein sequence will have \( \approx 50^3 \times 3 \) possible conformations. Now, typically a computer capable of searching \( \approx 200 \) conformations per second would require \( \approx 5.8661^{61} \) years to confirm the best search result.

Along with the conformational search complexity, in reality, there are also other forces [10] such as hydrophobic interaction, hydrogen bonding and electrostatic forces together with Van der Waals interactions, disulphate bridge, so on serve to influence the final 3D conformation. We discuss the existing conformational investigation techniques in two categories next.

2.1 Non-computational Techniques

Non-computational or experimental techniques such as X-ray crystallography (XC) [11] and nuclear magnetic resonance (NMR) spectroscopy methods are used for PSP. They are very time consuming, expensive and labour intensive [12]. Moreover, the NMR becomes less accurate for longer sequence and the crystallization for XC process may force the protein to have a non-native structure [13].

2.2 Computational Techniques

The computational approaches have the potential to correlate and predict the primary sequence of a protein to its structure thus can overcome the aforementioned difficulties associated with the experimental approaches. So, there has been
significant research interest [7] into application of computational approaches for protein structure prediction. Approaches such as homology modelling [14] (which is based on the similarity comparison of the sequence) and threading [15] (which is a process to thread together likely short sub-conformation of the corresponding subsequence) are based on the database of protein sequences and their corresponding structure. However, as these methods depend on the availability of similar sequence samples in the database, their results may become unconvincing for dissimilar sequences [4, 5] and they become less accurate for longer sequences as the formation of the whole conformation derived from its sub-conformations is less likely to match the native conformation because more dissimilarity is added between similar fragments [16, 17].

Consequently, the *ab initio* (meaning ‘from the origin’) or *de novo* approach predict folded protein’s 3D structure from its primary sequence alone [18] based on intrinsic properties (namely, hydrophobic and hydrophilic) of amino acids. The concept of *ab initio* folding is based on the Anfinsen’s thermodynamic hypothesis, which assumes [19, 20] that the native state of the folded protein is the global free energy minimum. Together with Levinthal Paradox which Cyrus Levinthal postulated [21], in what it is popularly known as that, “the proteins fold into their specific 3D conformations in a time-span far shorter than it would be possible for protein molecules to actually search the entire conformational space for the lowest energy state. However, in contrast protein cannot sample all possible conformations while folding, and hence folding cannot be a random process which leads to conclude that folding pathways must exist”, which motivates the *ab initio* based computation. However, in practice, as *ab initio* approach is computationally intensive, usually short protein sequences have been simulated at the atomic level, mostly using simplified low-resolution model and simple fitness function. Some methods are hierarchical [9, 19, 22] in that they begin with a simplified lattice representation and end with an atomistic detailed molecular dynamics simulation [23, 24]. With further advancement, the energy functions include atom-based potentials from molecular mechanics packages [25] such as CHARMM, AMBER [26] and ECEPP [27]. While *ab initio* is the most computationally demanding of the three computational approaches, it conversely also is the most promising in providing reliability, accuracy, usability and flexibility in checking the functional divergence of a protein by modifying its structure and sequence.

### 3 Models for Structure Prediction

The most appropriate approach for protein modeling would be to simulate the actual folding process which occurs in nature [28], such as molecular dynamics (MD) (which is based on collaborative motion and energy of the molecules in a protein sequence) [29, 30, 31]. However, this is infeasible for two reasons:

* The computation time even for a moderately-sized folding transition exceeds the feasible range even using the current best capable machines applying molecular dynamics principles.
ii) The forces involved in the stability of the protein conformation are currently not modeled with sufficient accuracy.

Thus, to handle the complexities for PSP, models of different resolutions are applied, which help transform the continuous large conformational landscape into a reduced and discrete search landscape, reducing the timescale of protein motion and makes the sampling of the landscape more feasible. Also, the modeling chronology from low to high considers the backbone modeling first and then subsequently the side-chain packing and extended modeling. In low resolution models, more atoms are grouped together, especially from the same amino acid and then treated as a single entity. The most simplified paradigm is the lattice model which focuses only upon hydrophobicity by dividing the amino acids into two parts: hydrophobic (H) and hydrophilic or polar (P) thereby leads to its popular appellation of the HP model [32, 33]. The lattice can have several regular shapes with varying numbers of neighboring residues either in 2D or 3D, such as square, cubic, triangular, face-centered-cube (FCC) [22, 34], or any of the Bravais Lattices. Conversely, the off-lattice model [35, 36] relaxes the regular lattice structure and both lattice and off-lattice normally start with backbone modeling and then increase the resolution, breaking the residues into further smaller constituents or considering the inclusion of side-chains. In the side-chain-only [37] (SICHO) approach, the side chains are initially constructed ahead of the main chain, with the argument being that interactions within proteins are due to different characteristics of the side chain, while the interactions of the main chain are rather more generic. CABS (an acronym for Cα-β and Side group) [38] is a relatively high resolution lattice model which assumes a lattice confined Cα representation of the main chain backbone, with 800 possible orientations of the Cα–Cα vectors. The lattice spacing of the underlying simple cubic lattice is assumed to be 0.61Å. The model assumes four united atoms (interaction centres) per residue: α-carbon, centre of the virtual Cα–Cα bond (serving as a centre of interactions for the peptide bonds), Cβ (see Fig. 2) and where applicable, the centre of mass of the side-group. While the coordinates of the α-carbons are restricted to the underlying lattice, the coordinates of the remaining united atoms are off-lattice and defined by the Cα-coordinates and the amino acid identities. The force-field of this model consists of several potentials that mimic averaged interactions in globular proteins. Finally, the direct all-atom [12, 39] model considers all the atoms including the forces. The finest possible model applies the theories of Quantum Mechanics (QM) with the principal simulation paradigm, especially for the all-atom model, being based upon the thermodynamics hypothesis, namely that the stable structure corresponds to the global free energy minimum. The computation to find the most stable energy-free state is based on MD [12, 30] using the collaborative motion and energy of the molecules involved from the protein and solvent. In MD simulation [40], the system is given an initial thermal energy and the molecules are allowed to move in accordance with MD principles. After a short time delay, typically $10^{-15}$ to $10^{-4}$ seconds, forces are used to calculate the new position of the atoms, which produces the atomic coordinates as a function of time. IBM’s Blue Gene [40, 41] project involved such an effort with peta-FLOP capability ($10^{15}$ floating point
operation per seconds). This is still however, many orders of magnitude lower than the requirement for a realistic solution.

With the objective of successfully building an effective computational strategy to unravel the complexities of the sequence-to-folding relationship, even using the well-established HP model, an efficient and robust solution has still to be developed. In highlighting the various computational intelligence approaches for ab initio PSP, the next section focuses mainly upon the low resolution HP model.

The HP Model

The HP model introduced by Dill [32, 33] is based on the fact that the hydrophobic interactions dominate protein folding. The Hs form the protein core freeing up energy, while the Ps, have an affinity with the solvent and so tend to remain in the outer surface. For PSP, protein conformations of the sequence are placed as a self-avoiding walk (SAW) on a 2D or 3D lattice. The energy of a given conformation is defined as a number of topological neighbouring (TN) contacts between those Hs, which are not sequential with respect to the sequence.

PSP is formally defined as: for an amino-acid sequence \( s = s_1, s_2, s_3, \ldots, s_n \), a conformation \( c \) needs to be formed whereby \( c^* \in C(s) \), energy \( E^* = E(C) = \min\{E(c) \mid c \in C\} \) [42], where \( n \) is the total amino acids in the sequence and \( C(s) \) is the set of all valid (i.e., SAW) conformations of \( s \). If the number of TNs in a conformation \( c \) is \( q \) then the value of \( E(c) \) is defined as \( E(c) = -q \) and the fitness function is \( F = -q \). The optimum conformation will have maximum possible value of \( |F| \). In a 2D HP square lattice model (Fig. 3. (a)), a non-terminal and a terminal residue, both having 4 neighbours can have a maximum of 2 TNs and 3 TNs respectively. In a 2D FCC HP model (Fig. 3. (b)), a non-terminal and a terminal residue both having 6 neighbours can have a maximum of 4 TNs and 5 TNs respectively.

Many of the successful PSP software such as ROSETTA [4, 43], PROTINFO [44, 45], TASSER [46] use various resolution of models embedded into the

![Fig. 3. Conformations in the 2D HP model shown by a solid line. (a) 2D square lattice having fitness = - (TN Count) = -9. (b) 2D FCC lattice having fitness = -15. ‘●’ indicates a hydrophobic and ‘○’ a hydrophilic residue. The dotted line indicates a TN. Starting residue is indicated by ‘1’.](image-url)
Genetic Algorithm in Ab Initio Protein Structure Prediction

hierarchical paradigm [6, 46–49] to cope with the high computational complexity. The low resolution model can be used to determine the backbone of the 3D conformation and can pass it to the next step for further expansion.

4 Search Algorithms

The PSP in HP lattice model has been proven to be NP-complete [50, 51], which implies that neither a polynomial time nor an exhaustive search [52–55] methodology is feasible. Thus the non-deterministic search techniques have dominated attempts, of which there are ample approaches such as, Monte Carlo (MC) simulation, Evolutionary MC (EMC) [56, 57], Simulated Annealing (SA), Tabu search with Genetic Algorithm (GTB) [58] and Ant Colony Optimization [42], though because of their simplicity and search effectiveness, Genetic Algorithm (GA) [1–3, 7, 9, 59, 60] is one of the most attractive [2, 59]. Therefore, we focus on GA and we starts with preliminaries on GA associated with PSP problem in low resolution.

4.1 Underlying Principle of Nondeterministic Search and GA Preliminaries

The algorithm shown in Fig. 4. provides a generic framework for the nondeterministic search approaches.

1. Initial random solution generated randomly or, using domain knowledge.
2. Obtain new solution \( S_{\text{new}} \) from the current single solution \( S_{\text{curr}} \) or pool of solutions using special operator/operation defined by individual approaches.
3. Assess the quality or the fitness \( F \) of \( S_{\text{new}} \).
4. IF \( F \) indicates improved solution accept \( S_{\text{new}} \); ELSE accept/reject based on special criteria.
5. IF END-OF-SOLUTION is not reached THEN go back to Step 2.

**Fig. 4.** Template for a nondeterministic search approach

![Fig. 4. Template for a nondeterministic search approach](image)

**Fig. 5.** An example showing (a) 1-point crossover, (b) mutation by 1 bit flipping
Fig. 6. An example of mutation operation [2]. Dotted lines indicate TN. Residue number 11 is chosen randomly as the pivot. For the move to apply, a 180° rotation alters (a) with $F = -4$ to (b) $F = -9$. ‘$\rightarrow$’ indicates mutation residue.

Nondeterministic approaches can vary base on the steps shown in Fig. 4. For instance, Hill Climbing approach [61] starts (step 1) with a random bit string and then obtains (in step 2) a set of neighboring solutions by single bit flipping of the current solution. Then, the best is keep as the new current solution and the process is repeated until the stop criterion is met. SA uses the same framework, but differs in its acceptance criteria (step 4): When the new solution is not better than the current, the algorithm can still accept it based upon some randomly defined criteria. GA uses a pool of solution (step 2), named population and obtains new solution by crossover (see Fig. 5 (a)) and mutation (see Fig. 5 (b)) operators. In the PSP context, mutation is a pivot rotation (see Fig. 6) which is also followed in crossover operation (see Fig. 7).

GAs optimize the effort of testing and generating new individuals if their representation permits development of building blocks (schemata), a concept formalized in the Schema Theorem [1, 61–70]. In each generation of a GA, the fitness of the entire population is evaluated by selecting multiple individuals from the current population based on their fitness before crossover is performed to form a new population. The $i^{th}$ chromosome $C_i$ is selected based on the fitness $f_i$ with the proportionate selection $\left(\frac{f_i}{\bar{f}}\right)$, where $\bar{f}$ is the average fitness of the population.
Parents then produce off-spring by crossover at a rate $p_c$ for the population of size $Pop_z$, thus forming the next generation. Mutation is applied on the population of generated off-spring at a rate $p_m$ and the selection probability of any off-spring or chromosome is again $\left( f_i / \bar{f} \right)$. A small percentage, typically between 5% and 10% of elite chromosomes (those having higher fitness), are copied to the next generation to retain potential solutions. The remaining chromosomes (if they exist), which are unaffected by crossover, mutation or elitism operations are then moved to the next generation.

Throughout this article, a short sequence will imply a sequence with $n < 25$ ($n$ indicates the number of residues in a sequence or the protein length), a moderate length will imply $25 \leq n < 50$ and long sequences will imply $n \geq 50$.

### 4.2 Incorporating Intelligence into the GA

The fundamental basis of the GA, the schema theorem, supports that schema fitness with above average values in the population will more likely be sustained as generations proceed and as a consequence the similarity [61, 64, 71–73] of chromosomes grows within the population, thus grows *twins* (same or similar chromosomes) leading lower variations within the population. The existence of twins and the requirement for their removal in a GA is not new, as their growth was considered in evaluating the cost of duplicate or identical chromosomes in [72]. It suggested starting each chromosome with different patterns to avoid twins, but if twin growth is inherent in a GA search, then the effect of initialization using different patterns will decline relatively quickly for long converging problems. Also, [61] advocated that if a population comprised all unique members, tests need to be continually applied to ensure identical chromosomes did not breed. If chromosome similarities within population do not grow, then the GA may not converge as the search process effectively remains random rather than stochastic, while if similarities grow, then finding a non-similar chromosome to mate with clearly becomes more scarce because of the inevitable occurrence of twins, and the increasingly high cost of finding dissimilar chromosomes in a lengthy convergence process.

To solve, the need for twin removal was originally highlighted in [73]. The study however, was confined to the detection and removal of identical chromosomes only. Recently, in [71], the notion of *twins* was broadened by introducing *chromosome correlation factor* (CCF) [71] which defines the degree of similarity existing between chromosomes, and it was shown that by removing chromosomes having a similarity value greater than or equal to specific value of CCF during the search process enables the GA to continue seeking potential PSP solutions to provide superior results and helps overcome fallacious effect (see Fig. 8) of the selection procedure.
Fig. 8. The probability of a chromosome $C_k$ (with fitness $f_k$) being selected by roulette wheel selection, is $p_k = f_k / \sum_{i=1}^{n} f_i$. So, for a population of eight chromosomes having fitnesses 8, 6, 6, 6, 6, 6, 4 and 1 for example, the proportionate selection probability of the first chromosome will be $p_1 = (8/43)$, and similarly $p_2 = (6/43)$, ..., $p_8 = (1/43)$. The fallacy is, from the pie-chart, we see the fitness 6 occupies 68% in total (assume chromosomes having the same fitness are identical), so the effective selection probability is, $P_{\text{effective}} = \sum_{i=2}^{6} p_i = 30/43$ or, 70% instead of 14%.

<table>
<thead>
<tr>
<th>Length</th>
<th>Sequences</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>H2(PH)3PH4PH(P3H)2P4H(P3H)2PH4P(HP)3H2</td>
<td>[74]</td>
</tr>
<tr>
<td>60</td>
<td>P2H3PH8P3H10PH3H12P4H6PH2PHP</td>
<td>[74]</td>
</tr>
<tr>
<td>64</td>
<td>H12(PH)2(P2H2)2P2HP2H2PPHP2H2P2(H2P2)2(HP)2H12</td>
<td>[74]</td>
</tr>
<tr>
<td>85</td>
<td>4H4P12H6P12H3P12H3P12H3P1H2P2H2P1H1P1H</td>
<td>[75]</td>
</tr>
<tr>
<td>100a</td>
<td>6P1H1P2H5P3H1P5H1P2H 4P2H2P2H1P5H1P10H1 P2H1P7H1P7H2P1H1P3H6P1H1P2H</td>
<td>[75]</td>
</tr>
<tr>
<td>100b</td>
<td>3P2H2P4H2P3H1P2H1P2H1P4H8P6H2P6H9P1H1P2H1P11H2P3H1P2H1P1 P2H1P1H3P6P3H</td>
<td>[75]</td>
</tr>
</tbody>
</table>

‘H’ and ‘P’ in the sequence indicate hydrophobic and hydrophilic amino acid, respectively.

**Outcome of the Twin Removal**

Simulations were undertaken both with (CCF ≤ 1) and without the twin (WT) removal strategy implemented in the population, with in the former case, the twin removal being performed after the crossover and mutation operations. In every generation, twins were removed in all runs for a range of CCF settings from $r = 1.0$ (identical chromosomes) down to $r = 0.5$ (the least chromosome similarity, i.e., $0.5 \leq \text{CCF} \leq 1.0$) in steps of 0.1. Following twin removal from a population, the gap was filled by randomly generated chromosomes. The default GA parameters [71] for all experiments were set for population size, crossover, mutation and elitism rates as $200, 0.8, 0.05$ and $0.05$, respectively, and the 2D square HP lattice model was applied to the various benchmark sequences (Table 1). The
Table 2. Run results for 5 iterations of PSP for various sequences using GA. Each iteration has maximum generation = 6000, the average fitness of the runs is shown below.

<table>
<thead>
<tr>
<th>Length</th>
<th>WT</th>
<th>r = 1.0</th>
<th>r = 0.9</th>
<th>r = 0.8</th>
<th>r = 0.7</th>
<th>r = 0.6</th>
<th>r = 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>-29.4</td>
<td>-32.6</td>
<td>-33.4</td>
<td>-33.8</td>
<td>-32.2</td>
<td>-32.4</td>
<td>-32.6</td>
</tr>
<tr>
<td>64</td>
<td>-29.4</td>
<td>-34.2</td>
<td>-35</td>
<td>-37</td>
<td>-35.4</td>
<td>-34</td>
<td>-32.2</td>
</tr>
<tr>
<td>85</td>
<td>-42.2</td>
<td>-45</td>
<td>-47</td>
<td>-46.8</td>
<td>-46.2</td>
<td>-45</td>
<td>-44.4</td>
</tr>
<tr>
<td>100a</td>
<td>-38.6</td>
<td>-39.4</td>
<td>-42.4</td>
<td>-43</td>
<td>-42.4</td>
<td>-42.4</td>
<td>-40.8</td>
</tr>
<tr>
<td>100b</td>
<td>-37.4</td>
<td>-40.4</td>
<td>-42.6</td>
<td>-44.8</td>
<td>-42.8</td>
<td>-42.2</td>
<td>-42</td>
</tr>
</tbody>
</table>

corresponding results are displayed in Table 2, indicate that twin removal with \( r = 0.8 \), i.e., having 80% and above similarity being removed, has obtained the best performance. Introduction of the twin removal helps improved generically.

4.3 Intelligence in Chromosomal Encoding

The encoding used in the HP lattice models was mainly isomorphic, which add unwanted variations for the same solution (conformation). Non-isomorphic encoding scheme [76] further constrains the search space, aids convergence and similarity comparisons are made easier while applying a twin removal and removes implicit controlling of the crossover and mutation rates (see Fig. 12), thus provides superior results.

In the literature, four different encoding strategies have been reported [76]: i) Direct coordinate presentation, ii) Absolute encoding, iii) Relative encoding and iv) Non-isomorphic encoding. Rather than using a binary string, preference to use conformations themselves is known as direct coordinate presentation.

Fig. 9. Absolute moves (a) 2D square lattice based representation and (c) 3D cube lattice based representation. (b) Coordinate frame used for encoding.

Fig. 10. The relative moves in 3D, namely (a) Straight / Forward (S or F) (b) Left (L) (c) Right (R) (d) Up (U) and (e) Down (D). However, Backward (B) move does not need a self avoiding walk.
Absolute encoding [34, 42, 77–79] replaces the direct coordinate presentation with letters representing directions with respect to the lattice structure. The permitted moves for absolute encoding are: \( f \) (forward), \( l \) (left), \( r \) (right), \( b \) (back), \( u \) (up) and \( d \) (down) (see Fig. 9), while \( u \) and \( d \) indicate \( +z \) and \( -z \) direction respectively. A conformation \( c \) in 2D with \( n \) residues could be \( c \in \{f, l, r, b, u, d\}^{n-1} \) while in 3D it would be \( c \in \{f, l, r, b, u, d\}^{n-1} \). Alternatively, in relative encoding the move direction is defined relative to the direction of the previous move as shown in Fig. 10, rather than relative to the axis defined by the lattice. These moves are lattice automorphic [34], with the initial move always expressed by \( F \) (forward). A conformation \( c \) of \( n \) residues in 2D and 3D could then be \( c \in \{F, L, R\}^{n-2} \) and \( c \in \{F, L, R, U, D\}^{n-2} \), respectively. Relative encoding (Fig. 10) was developed with a view to improving presentation over absolute encoding with pivot mutation being represented as the single locus or character alteration of a chromosome as shown in Fig. 11.

![Fig. 11.](image1)

(a) Single mutation at residue number 6 (red colour) using absolute encoding using changes in genotype and in the corresponding phenotype is not a pivot rotation (b) Single mutation at residue 6 using relative coding results in true pivot rotation

![Fig. 12.](image2)

The cross-exchange indicated by the dotted contour in the identical conformations (a) and (b) result conformation in (c), which can also be constructed from (a) or (b) by applying mutation (i.e. pivot rotation) at residue number 5. Hence, the crossover can result equivalent to the mutation operation for identical parents.
It is clear that the coordinates of a rotating object change so direct coordinate presentation is inherently isomorphic. Moreover as shown in Fig. 13 and Fig. 14, absolute and relative encodings are also isomorphic. Thus, a non-isomorphic encoding algorithm is essentially proposed in [76] by assigning a fixed directions for a growing chain based upon the first occurrences of the move in a particular dimension. The direction from first residue towards the second is marked ‘1’ and the reverse is marked ‘2’, which defines the complete move in 1-dimension. The first occurrence of a direction perpendicular to the 1-dimension is marked as ‘3’ and the reverse as ‘4’, which completes the moves in 2-dimensions. The first occurrence of the move perpendicular to the plane formed by ‘1’ and ‘3’ moves is marked as ‘5’ and the reverse as ‘6’, which finally defines the moves in 3-dimensions.

**Fig. 13.** Absolute encoding is isomorphic. For six directions, namely +x, -x, +y, -y, +z and -z, 24 (= 6×4) different genotypes are possible for a given 3D conformation.

**Fig. 14.** Relative encoding is isomorphic. Four variations shown in 3D by rotating around axis formed by 1-2 connecting line, but no variation achieved by the change in direction (x or y or z).

### 4.4 Domain Knowledge Based Heuristic

Despite the aforementioned improvements, PSP remains an intractable problem because during the search, the solution becomes phenotypically more compact,
thereby increasing the number of collisions [3, 57]. To solve, alternate operators and move sets have also been applied [75]. An operator that is able to move the intended portion of the converging conformation with a predefined target, while concomitantly having minimal impact on the stable portion, exhibits considerable promise. One such operator, short pull move, or pull move was proposed by Lesh in the square 2D lattice model [75], which subsequently extended by Hoque et al. [3], with the introduction of the tilt move, which is applicable when other moves fail due to congestion. The tilt move however can disturb the stability more than the pull move. A selective implementation of the move sets based on current scenario could represent a powerful combination such as for instance, firstly attempting a diagonal move [3] and if this cannot be performed to reach a predefined goal then next applying a pull move and then a tilt move if the pull move perchance fails. Fig. 15 describes these moves in further detail.

Lesh’s experiment demonstrates the superior performance in achieving the minimum energy conformation for longer sequences using the pull move in moving

Fig. 15. Various move operators (a) if ‘D’ is free, then ‘B’ can be move to ‘D’ via a diagonal move. (b) Before and after applying pull move is displayed. In first case ‘B’ can be pulled to ‘B’ if ‘C’ is free or ‘C’ is already at the position of ‘C’ and the rest of the chain upto one end can be pulled until a valid conformation is reached. (c) Tilt move, ‘C’ and ‘D’ can be moved to ‘C’ and ‘D’ respectively and pull will propagate towards both ends.

Fig. 16. The subsequence -123- in (a) need to remap to sub-conformation corresponds to –HPPH-. If the position 2′ is free then 2 can be placed at 2′ and a pull (indicated in (a)) applied towards the higher indexed end. The pull moves 3 to 2, 4 to 3 and 5 to 4 and then finds a valid conformation without pulling further leaving (b). The |fitness| in (b) is increased by 1. In (b) assume, 4′ and 5′ are free positions and the segment 3 to 6 can be recognized as –PHHP-. To enforce a mapping to highly probable sub-conformation, 4 and 5 can be shifted to 4′ and 5′ respectively applying a pull move which results (c). In (c), 8 can pass through position 9, 10, 11 and results (d) and increases |fitness| by 1 further. The position of H-Core centre (HCC) (‘•’) is the arithmetic mean of the coordinates of all Hs.
Further, both (d) and (e) relate to -PHHP- (2S_H). Symbol ●, ○ and □, respectively indicate an H, a P and the approximate position of HCC.

phonotypically compact conformation, but it also provides lessons that random application of the move can consume significant computational resources. Hoque et al, has subsequently proven that incorporating domain specific knowledge [3, 80−82] with the move and their combinations afford considerable promise. As illustrated in Fig. 16, the pull move in both 2D and 3D FCC model helps to improve the fitness. Furthermore, as the parity problem is absent in the FCC model, the pull move does not need to be moved diagonally [81, 82] to start as in an ordinary pull because with more neighbours, the model is likely to get a valid conformation without the need to propagate the pull often upon the terminal residue.

Further, Hoque et al. [3, 80−82] conceptualised the folded protein as a three-layered kernel (Fig. 17). The inner kernel, called the H-Core, is assumed compact and mainly formed of Hs while the outer kernel consists mostly of Ps. The H-Core [83] Centre is named HCC. The composite thin layer between the two kernels consists of those Hs that are covalent-bonded with Ps and is referred to as the HP-mixed-layer. To integrate domain knowledge, Hoque et al, showed that the optimal core for a square 2D [3], cube 3D [80], 2D FCC [81] and 3D FCC [82] lattice are square, cube and regular hexagon respectively, which concludes the optimal core that maximizes the fitness can be predicted based upon the properties and dimension of the model. To form the cavity of H-Core Hoque et al. further introduced, motifs or sub-conformations based approach which are highly
probable to a sub-sequence (defined in Fig. 13 for 2D FCC) are forced to re-map. The rationale is to form immediate TN and place P as far away as possible from HCC while concomitantly placing H as near as possible to the HCC. For the mapping, two broad categories of sub-sequences are defined; \( gS_H \) and \( gS_P \), where \( g \in N \), where N is a natural number. These two categories completely cover the HP-mixed-layer including outer kernel. Let \( S_H \) and \( S_P \) represent segments of H and P respectively. A segment refers to a contiguous string of length \( g \), e.g. \( 2S_H \) means -PHHP-, so \( g = 2 \) with the two boundary residues being of the opposite type. \( g \) is divided into even \( g_e \) and odd \( g_o \) numbers. For \( 1S_P, 1S_H, 2S_P \) and \( 2S_H \), there are only a few possible sub-conformations, so only highly potential sub-conformations (Fig. 18) are chosen, based on embedded TN and core formation [83, 84] concepts. Collectively they are referred to as 

**H-Core Boundary Builder Segments (HBBS)** [3] and are mapped to potential sub-conformations which are known as **H-Core Boundary Builder sub-Conformation (HBBBC)**. HBBBC forms part of a corner (especially when \( g = 1 \) and through the composition with other group having \( g = 2 \)) and an edge (especially when \( g = 2 \) and with the composition of the former group) of the H-Core boundary. The selection for mapping HBBBC into HBBS is probabilistically applied while searching.

**Formulation of Multi-Objectivity**

Combining the moves with domain knowledge, Hoque et al., formulated the prediction into multi-objective optimization [3, 80–82] by combining an additional probabilistic constrained fitness (PCF) measure along with the original fitness. When searching for an optimum conformation, if any member of a HBBBC corresponds to the related sub-sequence exists PCF rewards otherwise penalizes the search.

**Implementation of the Heuristics in a way to Enable Backtracking Capacity**

Here aforementioned heuristics are combine strategically as: The conformational search process is divided into two alternative phases namely, *Phase 1* (see (4)) in which F dominates PCF and starts building the core. In the alternate *Phase 2* (see (4)), PCF dominates which covers the formation of an HP-mixed-layer, i.e. the Core boundary. The enforcement of HBBBC is also performed in *Phase 2*, since PCF helps to sustain and stabilize any applied change. The HBBBC mapping is performed only if they are not found according to the likely sub-conformations for the corresponding sub-sequences. This may reduce the achieved fitness F, but it is expected that it will help reformulate a proper cavity that will maximize the \( H \) bonding inside the core, while shifting to the favorable *Phase 1* will maximize \( |F| \). As the phases alternate during the search process (using (3)), the impact becomes such that F and PCF come up with common goal that is more likely to be optimal. The total or combined fitness is defined as:
where \( t \) is \( t^{th} \) generation while search is carried out by the GA. To adjust the weights \( \alpha \) and \( \beta \) to dominate \( F \) and \( PCF \) over each other, the oscillatory function \( \delta(t) \) shown in Fig. 19, is introduced. The setup maintains a variation in the amplitude (A).

\[
Total\ Fitness = \alpha(t) \times F + \beta(t) \times PCF
\]  

(2)

Fig. 19. Plot of \( \delta(t) \) function

\[
\delta(t) = A(1 + \cos \omega_m t) \cos \omega_0 t
\]  

(3)

where \( \omega_m << \omega_0 \) and \( t = \) number of generations. The assignment of \( \alpha \) and \( \beta \) are as:

**Phase 1:** \( \alpha(t) = \delta(t), \beta(t) = 1, \) when \( \delta(t) > 0 \)  

(4)

**Phase 2:** \( \alpha(t) = 1, \beta(t) = -\delta(t) , \) when \( \delta(t) < 0 \)  

(5)

**Transient Phase:** \( \alpha(t):=1, \beta(t):=1, \) when \( \delta(t) = 0 \)  

(6)

Typical parameter values for the \( \delta(t) \) function (see plot in Fig. 19) were set as follows: \( A = 30, \omega_m = 0.004 \) and \( \omega_0 = 0.05. \) The choice of \( A \) came from \( 2A \geq \max \left( |F_i|, |PCF_i| \right) \) where \( F_i \) and \( PCF_i \) respectively imply the upper bounds of \( F \) and \( PCF, \) which is predictable from the chosen model. The lower bound of \( F \) can be defined by (7) for 2D square and 3D cube HP lattice model and (8) for 2D FCC and 3D FCC model.

\[
F_i = -2*\dim^*(\min\{E[Seq], O[Seq]\}) + n_{TH}
\]  

(7)

\[
F_i = -(\dim \times n_H + n_{TH})
\]  

(8)

where in (7), \( E[Seq] \) and \( O[Seq] \) indicate the number of even and odd indexed H residues in the sequence and \( n_{TH} \) indicates number of terminal H residue, where
0 \leq n_{T_H} \leq 2$. The value of $\dim$ in (7) for 2D square and 3D cube HP lattice model is 1 and 2 respectively (and in (8) for 2D FCC and 3D FCC the values are 2 and 5 respectively). The ‘min’ implies ‘minimum of’. The $n_H$ in (8) indicates the total number of hydrophobic residues. Note, the minimum value of both $|\alpha(t)|$ and $|\beta(t)|$ is 1 and so never becomes zero in (4), (5) and (6), thereby preserving the sub-conformation or schema possessing good features, that may have been created in the alternate phase. The search uses a simple GA (SGA) paradigm which is hybridized (see Algorithm 1) with the aforementioned move sets, PCF etc with a population size of 200 for all sequences. The elite rate = 0.1, crossover rate = 0.85, mutation rate = 0.5 and single point mutation by pivot rotation was applied. The implementation of both crossover and mutation operations were as in [2], but without any special treatment such as cooling. The roulette wheel selection procedure was used.

The experiment results were very impressive (see Table 3) and outperformed in all available low resolution models including square 2D [3], cube 3D [80], 2D FCC lattice [81] and 3D FCC lattice model [82]. This general concept is referred to as guided GA [3, 80] or hybrid GA [12, 81], and it importantly provides a intelligent backtracking capability if any local minimum is assumed. Combining HGA with twin removal (as mentioned in Section 4.1) having $r = 0.8$, it was shown in [82] to obtain best performance over the form of $i$) SGA, $ii$) SGA + $r = 0.8$ and $iii$) HGA - for the 3D FCC lattice model.

As there could be a number of possible lattice structure or orientations [34], we next justify the preferred on for PSP problem (in Section 5) and modify the two bead HP model further to improve the prediction in Section 6.
Table 3. Performance comparison of nondeterministic search approaches [12] using 2D HP square lattice model

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20 / (HP)2PH(HP)2</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td>24 / H2P2HP2(HPP)4H2</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td>25 / P2HP2H2P4H2P4H2</td>
<td>-8</td>
<td>-8</td>
<td>-8</td>
<td>-8</td>
<td>-8</td>
<td>-8</td>
</tr>
<tr>
<td>36 / P3(H2P2)2P2H7P2H2P4H2P2HP2</td>
<td>-14</td>
<td>-14</td>
<td>-14</td>
<td>-12</td>
<td>-13</td>
<td>-14</td>
</tr>
<tr>
<td>48 / (P2H)2(HP2)2P4H10P6(H2P2)2HP2H5</td>
<td>-23</td>
<td>-23</td>
<td>-23</td>
<td>-22</td>
<td>-20</td>
<td>-23</td>
</tr>
<tr>
<td>50 / H2(PH)3PH4PH(P3H)2P4H(P3H)2PH4P(HP)3H2</td>
<td>-21</td>
<td>-21</td>
<td>-21</td>
<td>-21</td>
<td>-21</td>
<td>-21</td>
</tr>
<tr>
<td>60 / P2H3PH8P3H10PHP3H12P4H6PH2PHP</td>
<td>-36</td>
<td>-35</td>
<td>-35</td>
<td>-34</td>
<td>-33</td>
<td>-35</td>
</tr>
<tr>
<td>64 / 12(PH)2(P2H2)2P2HP2H2PHP2H2P2(H2P2)2 (HP)</td>
<td>-42</td>
<td>-39</td>
<td>-39</td>
<td>-37</td>
<td>-35</td>
<td>-40</td>
</tr>
</tbody>
</table>

5 Preferred Lattice Structure for PSP

A number of lattice models are used for studying the PSP problem. However, towards preferring a lattice structure or orientation in 3D for effectively mapping the real folded protein, we advocate the preference of the 3D face-centred-cube (FCC) orientation for the following reasons:

i) Based on the full proof of Kepler Conjecture [86], a 3D FCC is proven to be the densest sphere packing orientation. It can provide densest protein core [87] while predicting a protein structure (though the protein core may not necessarily need to be in the most compact form [88]).

ii) In 3D FCC orientation, a residue can have 12 neighbours in a 3D space [82]. Such orientation allows maximum excluded volume for offering densest compactness [3, 80, 81]. Therefore logically inferring, for a region with fixed volume, an FCC model has more option for placing a residue in suitable neighbouring position with respect to another residue than any other lattice models. A rudimentary example is, the FCC model is parity [88] problem free, whereas the square or the cube lattice is not.

iii) Therefore, within the lattice constraints, the FCC lattice can provide maximum degree of freedom and FCC can provide closest resemblance to the real or high resolution folding [3, 75, 80, 81].

In the FCC orientation, if its 12 neighbours are assumed to be covered with a thin outer layer, the overall structure resembles to a cuboctahedron [3, 80, 82] (see the shape of the inner kernel in Fig. 17 (b)), where a cuboctahedron has 14 faces, 6 of them are square and 8 of them are equilateral triangle and it has 12 corners of vertices.
6 hHPNX – An Extension of the HP Model

For an effective and faster exploration of the PSP landscape, the lattice models are indispensable. However, this crucial HP model (i.e., for interaction potential, see Fig. 20 (a)) having two beads, produces relatively large number of degeneracy [89] (i.e., the chance of different possible conformations but having same ground state energy), consequently which can result in useful conformations being lost in the multitude. Second, the positions of polar segments (i.e., P) are not optimized [90], can result in deformed structures, especially if the segment is too long or located at the end of the sequences. Thus necessarily a modification and an extension to the HP model, keeping simplicity as much as possible, lead to proposing the HPNX model (for interaction potential, see Fig. 20 (b)), where a logical extension of the HP model being proposed [79, 89]. In the HPNX model, the splitting of P (polar) monomer of HP model is actually based on the variations of electric charge, namely positive (P), negative (N) and neutral (X) among amino acids.

\[
\begin{array}{cccc}
H & P & N & X \\
H & -4 & 0 & 0 & 0 \\
P & 0 & 1 & -1 & 0 \\
N & 0 & -1 & 1 & 0 \\
X & 0 & 0 & 0 & 0 \\
\end{array}
\]

(a)          \[\begin{array}{cccc}
h & H & P & N & X \\
h & -2 & -4 & 0 & 0 \\
P & 0 & 0 & 1 & -1 \\
N & 0 & 0 & -1 & 1 \\
X & 0 & 0 & 0 & 0 \\
\end{array}\]
(b)          \[\begin{array}{cccc}
h & H & P & N & X \\
h & -4 & -3 & 0 & 0 \\
P & 0 & 0 & 1 & -1 \\
N & 0 & 0 & -1 & 1 \\
X & 0 & 0 & 0 & 0 \\
\end{array}\]
(c)

Fig. 20. Interaction potential matrixes of (a) HP (b) HPNX [89] and (c) hHPNX model. Negative entry indicates reward for being topological neighbors (TN) in the lattice model, whereas interaction for TN with positive value represents a penalty, ‘0’ indicates neutral (i.e., no) interaction.

However, based on many structural observation of a protein data sets Crippen proposed [91] a new potential interaction matrix as shown in Fig. 21 (a), where the amino acids where divided into four different groups. Crippen emphasized the small set of particular group for the certainty of their distinguishing properties, namely Alanine (Ala or A) and Valine (Val or V). It has been detected [92] that this particular element of the matrix highlighted in Fig. 21 (a) was converted with few wrong entries by Bornberg [79] as shown in the matrix of Fig. 21 (b). and named YhHX matrix.

The emphasised [91] small group \{A, V\} has highest frequency among proteins on an average compared to the occurrence frequencies of all the amino acids [93], and hence it is important to amend the incorrect conversion of the element (2, 2) of matrix in Fig. 21 (a) to element (2, 2) of matrix in Fig. 21 (b). The element depicts the ‘hh’ interaction of the YhHX matrix of Fig. 21 (b). Note that h \equiv \{A, V\},
Fig. 21. (a) Crippen’s matrix [91]; classifies amino acid contacts, presented using single letter: 1 = \{GYHSRNE\}, 2 = \{AV\}, 3 = \{LICMF\} and 4 = \{PWTKDQ\}. (b) YhHX matrix as converted by Bornberg in [79] from Crippen’s matrix. Here, $fq.$ implies the percentage of occurrence frequencies of amino acid for each of the four groups. (c) Corrected YhHX as it should have been considered in [79]. Blacked and shared entries in (a), (b) and (c) are the problem area.

should have been recorded as ‘2’ instead of this highlighted element being incorrectly shown as ‘-2’ in Fig. 21 (b) which can be easily observed comparing rest of the entries of the original matrix in Fig. 21 (a) with entries of the matrix in Fig. 21 (b). Further, the frequencies, indicated by ‘$fq.$’ and the shaded elements shown in Fig. 21 (b), also need to be swapped. Moreover, the “10%” mentioned in the YhHX matrix needs to be corrected as 20%. The corrected matrix, incorporating all necessary changes, is shown in Fig. 21 (c). To incorporated further the essence of the HPNX model with the aforementioned correction, an hHPNX model has been proposed (see interaction potential, Fig. 20 (c)) [92]. In this hHPNX model basically the H of HP or HPNX model has been split into two by indicating $h \equiv \{A, V\}$, leaving the rest of the members of the H group as it is.

To compare, HP, HPNX and hHPNX model, developed HGA (reported in Section 4.3) was applied on sequences taken arbitrarily from Protein Databank (PDB) [94], measuring the models’ output using ‘alpha-carbon ($C_\alpha$) root-mean-square-deviation’ (cRMSD) [34]. As expected, hHPNX performed the best [92].

7 Conclusions

The *ab initio* protein structure prediction (PSP) is an important yet extremely challenging problem. It urges to involve a considerable amount of computational intelligence. Low resolution or simplified lattice models are very helpful in this regard to explore the search landscape of astronomical size in a feasible time scale. Due to the nature of the complex PSP problem, nondeterministic approaches such as genetic algorithm (GA), especially for its potential operators found to be relatively promising for conformational search. However, even GA often fails to provide reasonable outcome especially for longer sequences and also without the effectiveness in the conformational search in low resolution, the full-fledged prediction, which encompasses low to high resolution modelling in a hierarchical system, would suffer later on. Therefore, a way to improve the nondeterministic search
(such as GA) for PSP, has been reviewed in the context of a twin removal within population, intelligent encoding for problem presentation, so on, which become indispensable for providing further effectiveness. Domain knowledge based heuristics are shown very useful. Moreover, in the modelling point of view, simplified model can be made further effective by preferring a lattice orientation, beads and contact potential that can map real folded protein closely possible.

Acknowledgments. Support from Australian Research Council (grant no DP0557303) is thankfully acknowledged.

References

24. Hirst, J.D., Vieth, M., Skolnick, J., Brook, C.L.: Predicting leucine zipper structures from sequence. Protein Engineering 9, 657–662 (1996)
94. PDB, Protein Data Base (April 2007), http://www.rcsb.org/pdb/
<table>
<thead>
<tr>
<th>Author</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apiletti, Daniele</td>
<td>169</td>
</tr>
<tr>
<td>Baralis, Elena</td>
<td>169</td>
</tr>
<tr>
<td>Bellgard, Matthew</td>
<td>1, 55</td>
</tr>
<tr>
<td>Berlingerio, Michele</td>
<td>211</td>
</tr>
<tr>
<td>Bonchi, Francesco</td>
<td>211</td>
</tr>
<tr>
<td>Brezovan, Marius</td>
<td>71</td>
</tr>
<tr>
<td>Brouwer, Rick</td>
<td>281</td>
</tr>
<tr>
<td>Bruno, Giulia</td>
<td>169</td>
</tr>
<tr>
<td>Cannata, Nicola</td>
<td>13</td>
</tr>
<tr>
<td>Chen, Jake Y.</td>
<td>265</td>
</tr>
<tr>
<td>Chen, Meifania</td>
<td>281</td>
</tr>
<tr>
<td>Chetty, Madhu</td>
<td>317</td>
</tr>
<tr>
<td>Cook, Diane J.</td>
<td>237</td>
</tr>
<tr>
<td>Corradini, Flavio</td>
<td>13</td>
</tr>
<tr>
<td>Curcio, Michele</td>
<td>211</td>
</tr>
<tr>
<td>Dan Burdescu, Dumitru</td>
<td>71</td>
</tr>
<tr>
<td>De Francesco, Erika</td>
<td>37</td>
</tr>
<tr>
<td>Dillon, Tharam S.</td>
<td>1</td>
</tr>
<tr>
<td>Di Santo, Giuliana</td>
<td>37</td>
</tr>
<tr>
<td>Fernandez, Margarita</td>
<td>295</td>
</tr>
<tr>
<td>Ficarra, Elisa</td>
<td>169</td>
</tr>
<tr>
<td>Giannotti, Fosca</td>
<td>211</td>
</tr>
<tr>
<td>Godínez, Israel Román</td>
<td>187</td>
</tr>
<tr>
<td>Haarslev, Volker</td>
<td>143</td>
</tr>
<tr>
<td>Hadzic, Maja</td>
<td>281</td>
</tr>
<tr>
<td>Holder, Lawrence B.</td>
<td>237</td>
</tr>
<tr>
<td>Hoque, Md. Tamjidul</td>
<td>317</td>
</tr>
<tr>
<td>Lloyd, Frank</td>
<td>265</td>
</tr>
<tr>
<td>López-Yáñez, Itzamá</td>
<td>187</td>
</tr>
<tr>
<td>Merelli, Emanuela</td>
<td>13</td>
</tr>
<tr>
<td>Palopoli, Luigi</td>
<td>37</td>
</tr>
<tr>
<td>Piersigilli, Francesca</td>
<td>13</td>
</tr>
<tr>
<td>Rombo, Simona E.</td>
<td>37</td>
</tr>
<tr>
<td>Sattar, Abdul</td>
<td>317</td>
</tr>
<tr>
<td>Shaban-Nejad, Arash</td>
<td>143</td>
</tr>
<tr>
<td>Sidhu, Amandeep S.</td>
<td>1, 55</td>
</tr>
<tr>
<td>Stanescu, Liana</td>
<td>71</td>
</tr>
<tr>
<td>Streja, Dan</td>
<td>295</td>
</tr>
<tr>
<td>Taduri, Shailaja</td>
<td>265</td>
</tr>
<tr>
<td>Turini, Franco</td>
<td>211</td>
</tr>
<tr>
<td>Villasana, Minaya</td>
<td>295</td>
</tr>
<tr>
<td>Vito, Leonardo</td>
<td>13</td>
</tr>
<tr>
<td>Yáñez-Márquez, Cornelio</td>
<td>187</td>
</tr>
<tr>
<td>You, Chang hun</td>
<td>237</td>
</tr>
</tbody>
</table>