applied sciences

MDPI

*Concept Paper*

# DNA Computing: Concepts for Medical Applications

**Sebastian Sakowski [1,*], Jacek Waldmajer [2], Ireneusz Majsterek [3] and Tomasz Poplawski [4]**

[1] Faculty of Mathematics and Computer Science, University of Lodz, Banacha 22, 90-238 Lodz, Poland
[2] Institute of Computer Science, University of Opole, Oleska 48, 45-052 Opole, Poland; jwaldmajer@uni.opole.pl
[3] Department of Clinical Chemistry and Biochemistry, Medical University of Lodz, 90-419 Lodz, Poland; ireneusz.majsterek@umed.lodz.pl
[4] Department of Molecular Genetics, Faculty of Biology and Environmental Protection, University of Lodz, 90-236 Lodz, Poland; tomasz.poplawski@biol.uni.lodz.pl
* Correspondence: sebastian.sakowski@wmii.uni.lodz.pl

**Abstract:** The branch of informatics that deals with construction and operation of computers built of DNA, is one of the research directions which investigates issues related to the use of DNA as hardware and software. This concept assumes the use of DNA computers due to their biological origin mainly for intelligent, personalized and targeted diagnostics frequently related to therapy. Important elements of this concept are (1) the retrieval of unique DNA sequences using machine learning methods and, based on the results of this process, (2) the construction/design of smart diagnostic biochip projects. The authors of this paper propose a new concept of designing diagnostic biochips, the key elements of which are machine-learning methods and the concept of biomolecular queue automata. This approach enables the scheduling of computational tasks at the molecular level by sequential events of cutting and ligating DNA molecules. We also summarize current challenges and perspectives of biomolecular computer application and machine-learning approaches using DNA sequence data mining.

**Keywords:** machine learning; DNA computer; biochips; queue automata; type IIB endonucleases



**Citation:** Sakowski, S.; Waldmajer, J.; Majsterek, I.; Poplawski, T. DNA Computing: Concepts for Medical Applications. *Appl. Sci.* **2022**, *12*, 6928. https://doi.org/10.3390/app12146928

Academic Editor: Cezary Czaplewski

Received: 11 May 2022
Accepted: 6 July 2022
Published: 8 July 2022



## 1. Introduction

For the last several years, there has been a growing interest in the possibility of computing by means of DNA molecules (called "DNA computing" later in this paper). The different directions of studies in this area include construction of biomolecular computers hardware and software which are based on biochemical components (bioorganic chemical compounds). Such computers are nanodevices built exclusively of organic components. Biomolecular computers may have a number of practical uses in the future, owing to their various properties, such as parallelism of operation or the ability to store information. Importantly, the biomolecular computers may, in the predictable future, fill some gaps in the areas not yet accessible to conventional computers. Particularly interesting is the compatibility between biomolecular computers and the cellular environment via biochemical reactions taking place both in vitro and in vivo.

An important part of DNA computing is involved in the construction of intelligent biochips (meaning decision making in the choice of a diagnosis/treatment direction), as such technological solutions may simplify and automate molecular diagnostics. This paper presents the use of biomolecular computers for constructing diagnostic biochips based on DNA chain cutting and ligating reactions carried out by restriction enzymes. The study was inspired by the concept of the hypothetical enzymatic Turing machine that was built of biomolecules by Charles Bennett in 1982 [1]. It also indicates the feasibility of using only biochemical components for designing computers characterized by high energy efficiency—with low energy consumption for performing calculations scheduled by humans. It should be pointed out that Charles Bennett noticed a similarity between the biochemical processes

taking place in live organisms (specifically DNA polymerase) and the operation of the Turing machine—specifically, a model of a programmable universal Turing machine that enables data processing and which is very well known in computer science.

In 1995, Paul W.K. Rothemund published his concept of a Turing machine based on commercially available class IIS restriction enzymes [2]. This concept indicated a theoretical possibility (not requiring laboratory experiments) of encoding, in double-stranded DNA, the transition table of the Turing machine, the idea of which is based on alternate cutting and ligating the double-stranded DNA with the class IIS restriction enzymes and a ligase. Moreover, Paul W.K. Rothemund suggested a method for constructing symbols as well as input words, for instance: the input word $x$ = 000111 (built of the symbols 0 or 1), of Turing machines which were designed by recording them as double-stranded DNA. In that approach, encoding the information (the symbol encoded in the input word $x$ = 000111) as double-stranded DNA is feasible, as modern laboratories offer production of double-stranded DNA with a preset nucleotide sequence. Paul W.K. Rothemund also proposed a method for encoding the state of the biomolecular Turing machine which, in his approach, was interpreted as a sticking-out, single-stranded DNA (the so-called sticky end), obtained by cutting the double-stranded DNA with class IIS restriction enzymes.

Further studies, including experimental ones, were carried out by our and Ehud Shapiro's teams, and demonstrated the potential of restriction enzymes in developing practical programmable biomolecular nanodevices, functioning in actual laboratory conditions [3–6]. In 2001, Ehud Shapiro's group built biomolecular computers in which double-stranded DNA was employed for encoding processed input data (symbols encoded in the input word). They also used double-stranded DNA for developing molecular software to enable such a biomolecular computer to be programmed. The hardware of such a biomolecular computer consisted of *Fok*I restriction enzyme and ligase. They achieved a computational result by alternately cutting and ligating double-stranded DNA, placing in a test tube the double-stranded DNA encoding the input word, the software in the form of double-stranded DNA, and the hardware (*Fok*I, ligase). It is worth noting that the entire process was run autonomously in a reaction mixture comprising the appropriate reaction buffers until the final computational result was obtained. This approach showed that it is feasible to practically construct a biomolecular computer, working only in a test tube (without any electronic components), in which the computations are based exclusively on biochemical reactions.

The use of restriction enzymes in typical laboratory conditions requires optimization of reaction conditions [5] and an appropriate approach to encoding various components of biomolecular computers, especially when multiple restriction enzymes are used. A number of laboratory experiments were carried out using multiple restriction enzymes, operating alternately on DNA chains in a single reaction mixture [5,6]. After solving various practical problems, we developed an algorithmic method that enabled an ad hoc addition of more restriction enzymes acting alternately on the appropriately encoded DNA [6]. Understanding the successive properties of the biomolecular computers led us to the formulation of a new mathematical theory involving a base formal apparatus concerning performance of computation by means of a single restriction enzyme and a double-stranded DNA [7]. Other theoretical studies included the fundamentals of designing biomolecular computers with memory and discussed the potential use of type IIB restriction endonucleases for developing a biomolecular push-down automaton [8].

One concern with biochip design is targeting particular specific molecular goals. Typically, these targets are specific DNA or RNA sequences defined and determined by the biochip's application target (infectious agent or pathological protein/sequence). This issue can be solved with machine learning.

Machine learning is an important part of the new concept of designing biochips based on biomolecular computers that has been proposed in this paper. This approach (concept) requires knowledge of unique DNA fragments, which is obtained by using machine-learning methods, such as sequence pattern mining [9]. Knowledge about unique

DNA fragments, the presence of which we want to diagnose, makes it possible to use biomolecular computers to read these DNA fragments. We selected from the various approaches to the application of machine-learning methods in biology those that will be useful for finding unique DNA fragments—important from the point of view of designing biochips based on biomolecular computers.

## 2. Machine-Learning Approaches in Nucleotide Sequence Data Mining

In recent years, the use of algorithms and mathematical methods has become widespread in biological sciences [10,11]. This is due to a dynamic increase in the number of biological data sets, which prompts the use of various methods typical of exact sciences [12], including artificial intelligence and machine-learning algorithms, for example, recurrent neural networks [13] or convolutional neural networks [14]. This requires using sophisticated methods characteristic of exact sciences to deepen the biological knowledge (see Table 1). The current knowledge on living organisms demonstrates great complexity of processes occurring at the molecular level, e.g., the expression of genetic information is a complex and not fully understood process.

**Table 1.** Summary of the existing machine-learning approaches for DNA sequence mining.

| Paper | Method | Type of Input | Target | Dataset | Result |
|---|---|---|---|---|---|
| Luedi et al. (2007) [15] | Multiple classification algorithms | DNA sequence | Imprint status of human genes | Ensembl | 156 imprinted genes identified |
| Chen et al. (2016) [16] | Hierarchical neural networks | cDNA microarrays | Molecular signal transduction | PUMAdb | Novel model for evaluating the machinery regulating gene expression |
| Kelley et al. (2016) [17] | Convolutional neural networks | Genome sequence | To annotate and interpret the noncoding genome parts | DNaseI-seq peak BED format files for 125 cells | Noncoding genome parts annotated and interpreted |
| Amin et al. (2018) [18] | Long short-term memory | Genome sequence | Annotate genome sequences | NCBI genomedomek-database | DeepAnnotator algorithms and models |
| Zeng et al. (2018) [19] | Natural language processing | Gene sequence | Enhancer-promoter interactions | Various databases from TargetFinder | Framework EP2vec |
| Yuan and Bar-Joseph (2019) [20] | Convolutional neural network | RNA sequences | Gene–gene relationships | scRNA-seq and bulk RNA-seq | Framework CNNC |
| Fudenberg et al. (2020) [21] | Convolutional neural network | DNA sequences | Genome folding | Five Hi-C anddomekMicro-C datasets | Akita network |

A dynamic development of the next-generation sequencing (NGS), which became cheap and available, brought about an increase in the amount of data containing nucleotide sequences. NGS is a sequencing method that makes it possible to determine the order of nucleotides in a sample of nucleic acids and high-throughput whole-genome sequencing. The GenBank database, which collects research results from the sequencing of living organisms, is of particular interest here. Thus, it is possible to quickly find information about the nucleotide sequence in the form of files containing nucleotide sequences, e.g., for a selected group of organisms for which we want to check for differences in nucleotide sequences. This makes it possible to develop new approaches to the analysis of data derived from the sequencing of living organisms. In recent years, many different machine-learning approaches have been used in life sciences. They focus on different aspects related to sequencing data analysis, such as sequence alignment, classification, and pattern finding [22].

To implement the biomolecular computers in medicine, as proposed in this paper, it is crucial to find unique DNA fragments that can be read/identified by biomolecular

computers. It is important to find unique patterns for the DNA sequences tested so that genetic differentiation of the investigated organisms is possible. DNA fragments encode various information, e.g., they encode amino acids that make up proteins. Therefore, comparing genomes based on specific DNA fragments makes it possible to find similarities between the tested organisms or to differentiate them with respect to the occurrence of unique nucleotide sequence arrangements. The problem of finding patterns in large sets of biological data that contain genomic sequences [9,23] is a challenge both for computer scientists and mathematicians, but also for biologists. Figure 1 presents the main idea of finding patterns in various nucleotide sequences that are analyzed by applying machine-learning methods to files containing nucleotide sequences. In our approach to the use of biomolecular computers, we amplify the unique DNA fragments by PCR (fragments number 2 and number 4 to be exact); then, we read the amplified DNA fragments using a properly designed biomolecular computer. Thus, the step of finding unique patterns using machine-learning methods is a key stage in the application of biomolecular computers for molecular diagnostics.

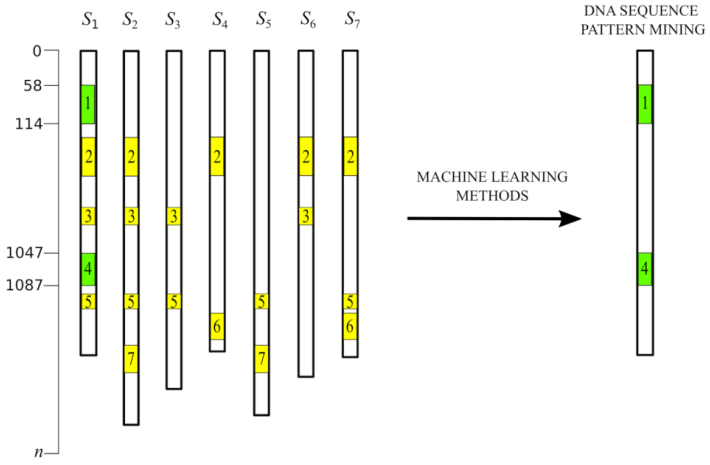


**Figure 1.** A diagram showing the use of machine-learning methods to find patterns and similarities in nucleotide sequences. By studying different nucleotide sequences with the use of machine-learning methods, unique DNA fragments can be found that are distinctive patterns of a DNA sequence. Two unique sequences (1 and 4) in the $S_1$ sequence tested are marked green. Abbreviations: $S_1, \ldots, S_7$—denote different genome sequences (different nucleotide sequences in genomes); the numbers 1, 2, 3, 4, 5, 6, 7—denote nucleotide sequences that occur in the sequences tested; the numbers: 58, 114, 1047, 1087—denote the sequence position in the genome; the unique DNA fragments are denoted by green color; the nonunique DNA fragments are denoted by yellow color.

Machine-learning algorithms are of particular interest, as they allow classifying DNA sequences, for example, the use of convolutional neural networks to analyze DNA sequences [14]. The classification of DNA sequences is very useful in understanding the relationship between DNA sequences encoding different proteins, as well as the relationships between proteins [13]. The main problem with these studies is that the functions of DNA fragments are not fully understood and the relationships between DNA fragments are still being discovered. There are many different approaches to finding homology in files containing nucleotide sequences [24], e.g., the use of basic local alignment search tool (BLAST) to find similarities in nucleotide sequences [25]. These problems are similar to those of fast search in text files encountered in computer sciences [26].

The key element of the proposed approach of employing the biomolecular computers is the use of machine-learning methods to find unique and characteristic DNA fragments of the diagnosed organism. These can be unique patterns of DNA fragments that are found by machine learning. Our proposed new approach is a combination of different research results in the field of machine learning and theoretical and practical work in the field of

biomolecular computers. This approach requires an interdisciplinary treatment of looking at the problem of molecular diagnostics, which needs to be solved by a joint action of computer science and molecular biology.

## 3. New Concept of Designing Diagnostic Biochips

Extraction of DNA sequence patterns with the use of machine-learning methods provides the background for the construction of diagnostic biochips based on biomolecular computers. In the proposed concept of designing diagnostic biochips based on biomolecular computers, unique DNA fragments play a special role, as they enable proper programming of a biomolecular computer in such a way that it reads unique DNA nucleotide sequences.

In the previous section, we discussed different machine-learning approaches to nucleotide sequence data mining. In our new concept of designing diagnostic biochips, it is possible to use artificial neural networks, for example, recurrent neural network (RNN) [18], at the stage of machine learning in the sequence data mining. RNNs can be used for data containing ordered strings, e.g., nucleotide sequences. From the point of view of the research methodology involving the RNN, it is important that the nucleotide sequences of the studied living organism genome are the input layer of such a network. RNNs can be used to generate output based on a nucleotide sequence of a given length. For example, RNNs can analyze nucleotide sequences that are characteristic of protein-coding genes and identify promoter sequences [10]. As part of the methodology of working with sequence extraction machine-learning methods, classic elements of machine learning, such as the process of training, validation, and testing, should be distinguished in individual steps. In the first step, it is necessary to well understand the set of input data and then to formulate an appropriate research question. In the next step, the data should be divided into training, testing, and validation sets. The next step is to choose the most suitable model for the research question. At this stage, it is especially important to check assumptions on the possibility of using the model. From the point of view of machine-learning methodology, it is also important to fine-tune the hyperparameters for the methods used. It is worth noting that in recent years, the model called transformers have attracted a lot of interest from researchers, as it allows better accuracy when studying character strings such as nucleotide sequences [27,28].

We propose a new approach to biochip design with biomolecular computing as the hardware and software (Figure 2) to enable DNA-level diagnostics. In this approach, the main mechanism is based on the use of biomolecular computers, built of appropriately encoded DNA chains as the software, and restriction enzymes and ligase as the hardware.

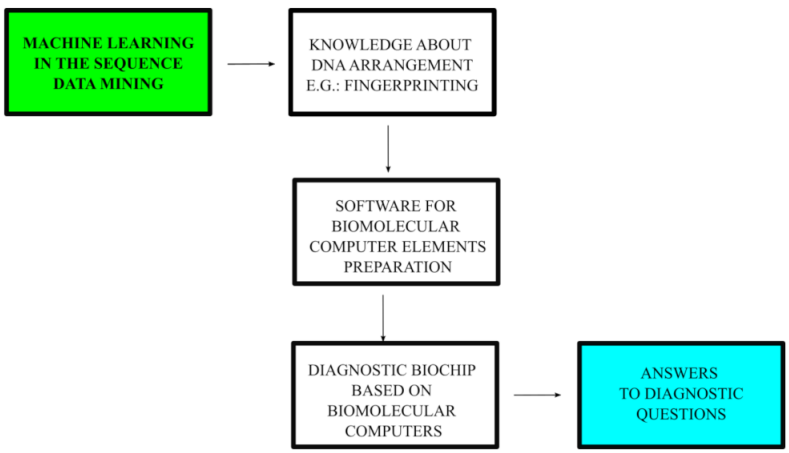


**Figure 2.** Schematic diagram of the new approach to using biomolecular computers as diagnostic biochips.

In this approach, the DNA fragments, for example, unique to a pathogenic virus, are determined by means of machine-learning methods in the sequence data mining.

Knowledge on unique DNA fragments, e.g., viral genomes, can come from analyzing an open access genetic sequence database, such as the GenBank database. Particularly interesting studies in this area include fingerprinting, which enables detection of the specific DNA fragments (obtained by PCR) characteristic of the investigated organisms. Recent years have also seen the development of methods based on artificial intelligence, which enable an automatic retrieval of information from genetic data. To this end, we propose the development of various machine-learning methods enabling detection of those DNA fragments that are unique with reference to the tested species. An important part of this approach is the software that will enable automatic encoding of the indispensable parts of diagnostic biochips based on biomolecular computers. This type of software will allow highly precise designing of indispensable components of the diagnostic biochips. At the molecular level, the mechanism of action of the diagnostic biochips consists of a sequential (alternate) reading of characteristic genetic features by the programmable biomolecular nanodevices built of a double-stranded DNA and the restriction enzymes. The devices complete their action as soon as they detect the presence of the desired DNA fragments and output signal, e.g., by means of fluorescence. It is worth noting that the so-designed diagnostic biochips can be manufactured commercially as laboratory kits or diagnostic devices.

One of the requirements for the correct and accurate designing of biochips based on biomolecular computers is to develop theoretical fundamentals of the implementation of practical biomolecular computers. In the case of diagnostic biochips, we propose the use of a formal system called queue automata [29] which have a memory that works according to the first in, first out (FIFO) principle—queue memory [30] (Figure 3).

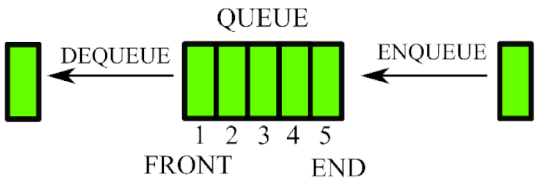


**Figure 3.** Schematic diagram of the queue memory operation (FIFO principle). A new element added to the queue is placed at the end of the queue, and the first element is removed from the front of the queue. The operation of adding a new element to the queue is called ENQUEUE, while the operation of removing an element from the queue is called DEQUEUE. Abbreviations: 1, 2, 3, 4, 5 denote the positions of the elements that are in the queue.

The queue automata consist of a head (finite control), a type with cells containing an input word created from symbols of a certain finite alphabet and a queue (Figure 3). A finite control reading of the symbols of the input word runs one after another and changes its state according to the transition rules followed. At every step of the queue automaton operation, the first symbol in the queue may be removed or retained, and another symbol may be added at the end of the queue. The transition depends on the current state of the queue automaton and on the symbol read out from the input word. A variant of queue automata is the deterministic input-driven queue automata [31]. In these automata, "input-driven" means that the automata are controlled by the input, i.e., that the input word controls the queue. The deterministic input-driven queue automata are a formal system $M = (Q, \Sigma, \Gamma, q_0, F, \perp, \delta_e, \delta_r, \delta_i)$, where $Q$ is the finite set of internal states; $\Sigma$ is the finite set of input symbols consisting of the disjoint union of sets $\Sigma_e$, $\Sigma_r$, $\Sigma_i$; $\Gamma$ is the finite set of queue symbols; $q_0 \in Q$ is the initial state; $F \subseteq Q$ is the set of accepting states; $\perp \notin \Gamma$ is the empty queue symbol; $\delta_e$ is the partial transition function mapping $Q \times \Sigma_e \times (\Gamma \cup \{\perp\})$ to $Q \times \Gamma$; $\delta_r$ is the partial transition function mapping $Q \times \Sigma_r \times (\Gamma \cup \{\perp\})$ to $Q$; and $\delta_i$ is the partial transition function mapping $Q \times \Sigma_i \times (\Gamma \cup \{\perp\})$ to $Q$.

The choice of that theoretical model (exactly queue automata) was dictated by the possibility of task scheduling with the use of queue automata, which is required for the controlled genome reading at the molecular level with the use of type IIB restriction

endonucleases. The biomolecular implementation of the queue automata (biomolecular queue automata) requires specific encoding of the respective components of the queue automata (Figure 4).

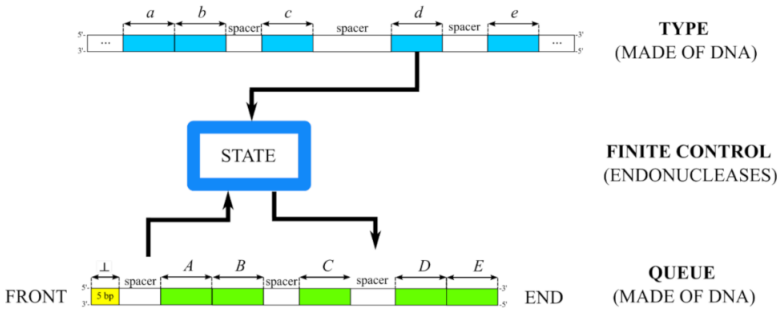


**Figure 4.** Schematic diagram of a biomolecular queue automaton. Abbreviations: *a*, *b*, *c*, *d*, *e* denote the symbols of the biomolecular queue automaton; *A*, *B*, *C*, *D*, *E*—the queue symbols of a biomolecular computer; the empty queue symbol denotes the beginning of the biomolecular queue; the spacer denotes the DNA fragment between the input symbols or the queue symbols of the biomolecular queue automaton; 5′, 3′—the DNA chain direction.

We propose that the symbols in the queue automaton should be separated by spacers, that is, the DNA fragments which do not encode the input symbols of a biomolecular computer—similarly to a computing machine made of biomolecules and presented by Ehud Shapiro's group in 2003 [4]. In addition, our proposal is that the spacers between the symbols should be of different length as this provides an opportunity to perceive a genome as a system of different symbols separated by spacers (Figure 5A). In our approach, the spacers can be used as technical DNA fragments encoding additional information at the biological level, but they do not play a significant role in queue automata. From the point of view of queue automata, the spacers are not very important for the calculations, but can be used as carriers of biological information encoded in DNA, e.g., a spacer can be used to store DNA sequences encoding proteins. It is suggested that the input symbols of the queue automata are encoded with same-length DNA chains, for instance: 10 base pairs, since this will enable appropriate encoding of the states of the queue automaton. In this approach, we are able to find the respective DNA fragments in the genome and then read the symbols using type IIB restriction endonucleases. The queue symbols of the biomolecular computer are encoded with DNA chains which have the same encoding lengths and which may contain spacers—just like the symbols of the queue automaton (Figure 5B). The states of the biomolecular queue automata are understood as the cut places of the symbol of a queue automaton within a fragment of the diagnosed genome [7].

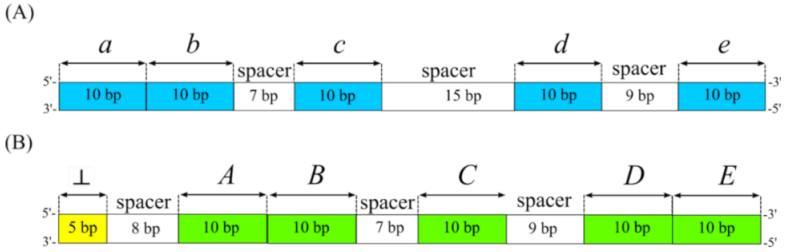


**Figure 5.** (**A**) An example of the input symbols of a biomolecular queue automaton. (**B**) An example of the queue symbols of a biomolecular queue automaton. Abbreviations: bp—the number of base pairs, encoding information in DNA; 10 bp—the length of a DNA with 10 base pairs (other base pair numbers such as 7 bp are to be understood accordingly); the input symbols of a biomolecular queue automaton are denoted by blue color; the queue symbols of a biomolecular queue automaton are denoted by green color, the empty queue symbol is denoted by yellow color.

The distinguished initial state is the first cut place of a type IIB restriction endonuclease in the appropriate place of the symbol of a genome. The set of accepting states is understood as cut places generated after the last cut with type IIB restriction endonuclease. The empty queue symbol is a separate DNA molecule, encoded with a unique DNA fragment with a sticky end. The queue symbols are encoded within the DNA fragment located in the transition molecule—to the left of the action site of the type IIB restriction endonuclease. The queue symbols are released only after the transition molecule relates to the symbol, encoded in the input word consisting of the symbols of the biomolecular queue automaton. The transitions of the biomolecular queue automaton are encoded with DNA chains which contain the restriction site for a type IIB restriction endonuclease in the middle of the transition. One queue symbol is encoded on the left of the restriction site and on the right, there is the sticky end complementary to the sticky end of the symbol of the biomolecular queue automaton.

Type IIB restriction endonucleases may constitute the hardware for biomolecular computers with queue memory, as they have the ability to simultaneously read and write information by cutting the double-stranded DNA to the right and to the left of the restriction site (Figure 6A). It is also acceptable to use multiple type IIB restriction endonucleases that act alternately in a single reaction mixture as well as to use restriction enzymes of other classes, for instance Class II, which cut the DNA chain only in one direction from the restriction site. This is of interest particularly in the aspect of earlier laboratory studies on the applicability of multiple restriction enzymes [5,6], as well as the concept of a theoretical design of a push-down automaton with the use of multiple restriction enzymes [32].

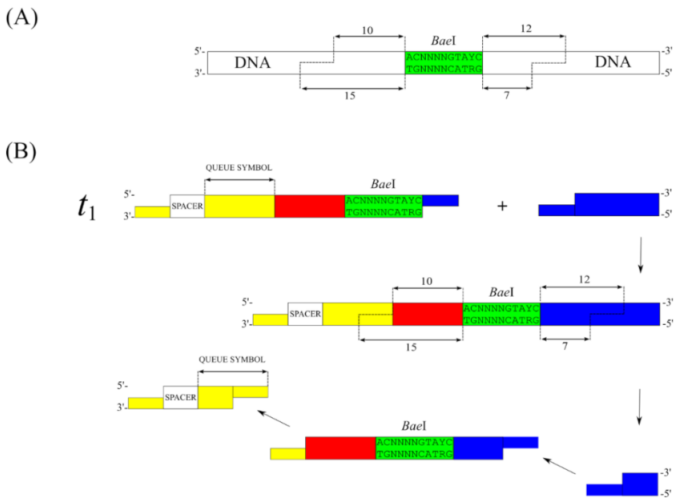


**Figure 6.** (**A**) The operation of the restriction enzyme *Bae*I. (**B**) The mechanism of writing the queue symbols using the restriction enzyme *Bae*I. Abbreviations: $t_1$—the transition molecule named $t_1$.

It is worth mentioning that, in the area of DNA computing, earlier practical solutions based on the use of restriction enzymes, only offered the possibility of reading the information encoded in the DNA, but not that of writing it [3,5]. The type IIB restriction endonucleases enable cutting the DNA chains in two directions and, in addition, they leave relatively long sticky ends, e.g., the length of the sticky ends left by *Bae*I is five nucleotides (Figure 6A). This effect of the type IIB restriction endonucleases enables the biomolecular computer to be programmed so that, after cutting the DNA chain, the enzyme writes information on the read-out DNA fragment (Figure 6B)—this is similar to the case of the biomolecular push-down automaton [8]. It is worth mentioning that a representative of the type IIB restriction endonucleases, *Bae*I, was used in practical experiments aimed at the implementation of the biomolecular computer involving multiple restriction enzymes [6]. This provided an experimental ground in the area of DNA computing for constructing various practical solutions based on type IIB restriction endonucleases.

## 4. A Concept for PCR Automation by Means of Biomolecular Computers

Biomolecular computers, specifically those with queue memory, which use type IIB restriction endonucleases may be used for automating the PCR method. The proposed new approach (called Queue-PCR) to automating the PCR method consists of the use of biomolecular computers as the hardware and the software for a wide range of PCR solutions. In comparison with the conventional PCR, this approach has the advantage of reading numerous replicated DNA fragments by the appropriately programmed biomolecular computer (Figure 7). The first step involves the replication of selected DNA fragments using the starters and the polymerase—as in the conventional PCR protocol. In the next step, the appropriately programmed biomolecular software in the form of transition molecules (see $t_1$, $t_2$, $t_3$, Figure 8) enables cutting the respective DNA fragments. The transition molecules enable both reading the DNA fragments and writing the read-out DNA fragment at the same time (Figure 8). The key elements in this approach are the appropriately programmed transition molecules that are unique for each transition executed by the biomolecular computer—they have the unique sticky ends and the encoded queue symbols.

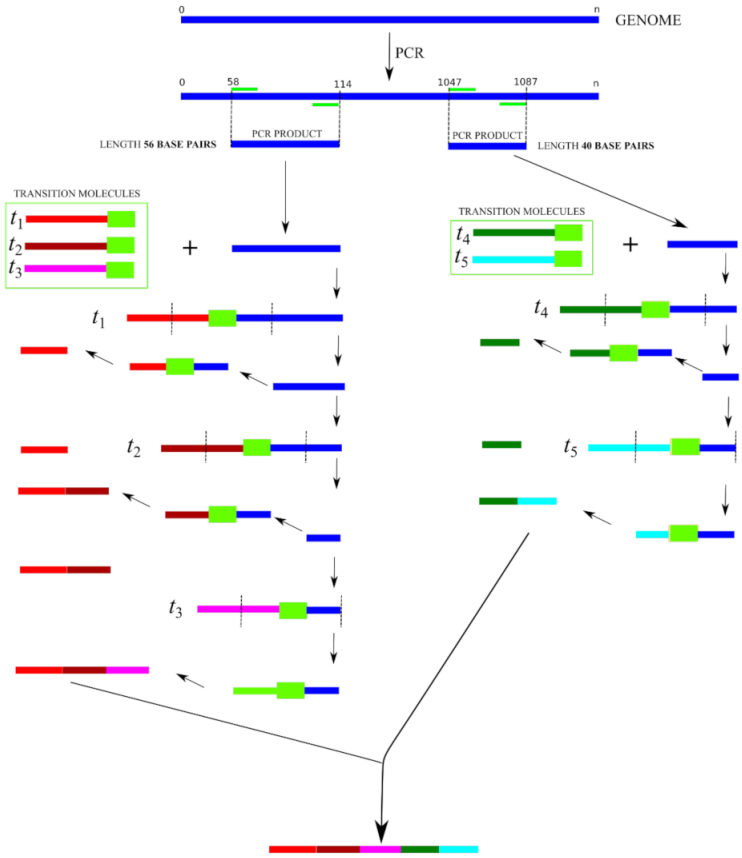


**Figure 7.** Diagram Queue-PCR of a new PCR automation concept of the use of biomolecular computers with type queue memory based on type IIB restriction endonuclease. The first step of Queue-PCR concept is replication of selected DNA fragments and the next step is sequential cutting and ligating DNA molecules by means of molecular software and hardware. Abbreviations: $t_1$, $t_2$, $t_3$, $t_4$, $t_5$—respective transition molecules (molecular software); PCR—conventional PCR method; the numbers: 58, 114, 1047, 1087 denote positions on the genome. The blue line denotes the genome fragment multiplicated by the PCR. The green rectangle denotes the sequence recognized by type IIB restriction endonuclease (molecular hardware). The DNA fragments amplify by PCR are denoted by blue color. The restriction sites are denoted by light green color. The remaining colors schematically illustrate the queue symbols.

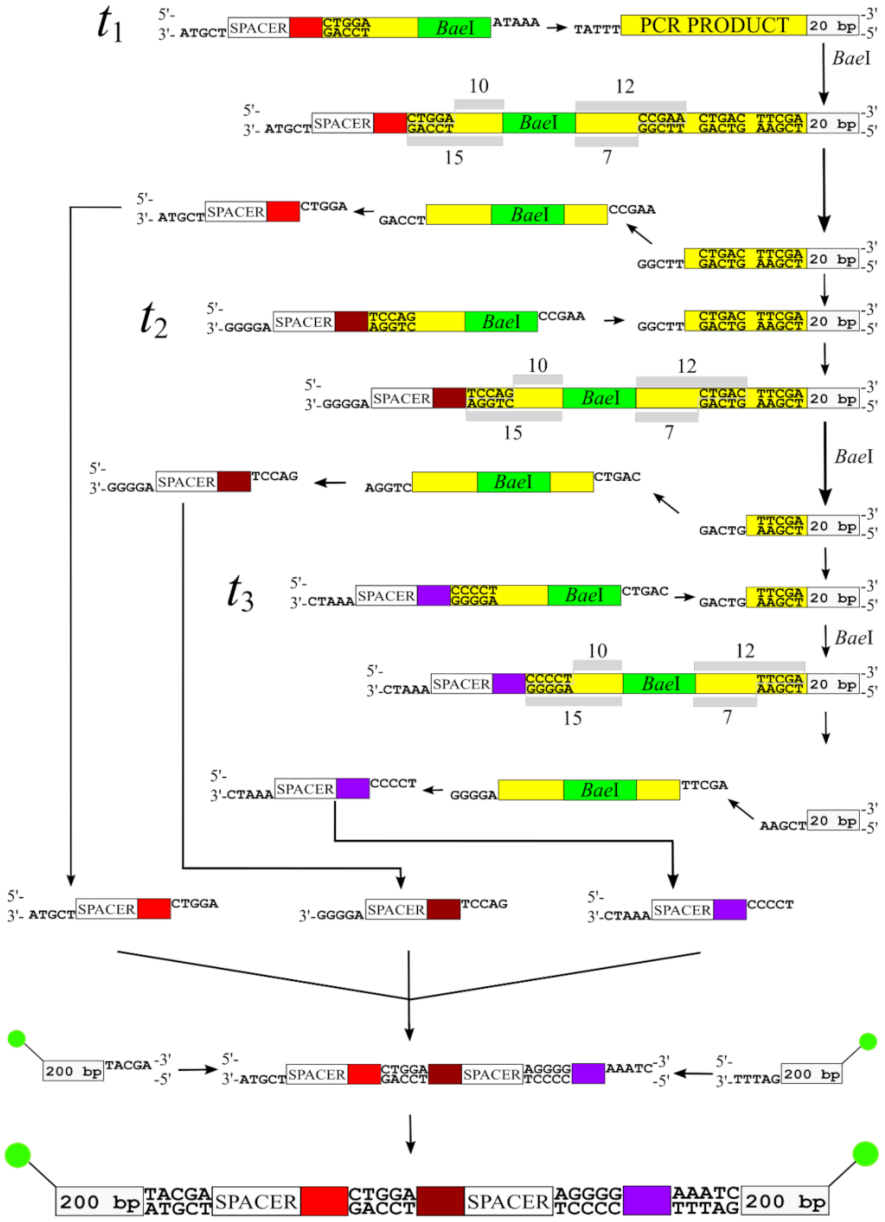


**Figure 8.** Operation of the restriction enzyme *Bae*I in a system which reads the genome fragment—reading and writing information into the queue. The molecular software ($t_1$, $t_2$, $t_3$) allows alternating and autonomous cleavage of DNA molecules which are genome fragments obtained by the PCR method. The biomolecular computer produces the final DNA fragment, as a standard output, after the cyclic reactions of cutting and ligating the genome fragment obtained by the PCR method. The green circle denotes the factor of fluorescence. The DNA fragments that are cleaved by restriction enzyme (*Bae*I) are denoted in yellow. The restriction sites are denoted by light green color. The other colors schematically illustrate the queue symbols that are placed in the queue.

After reading the DNA fragments (see Appendix A), the appropriately programmed and designed biomolecular computer will return, as the standard output, information on the result of the operation, e.g., in the form of fluorescence (see Appendix B). An important element in the new approach to designing diagnostic biochips is the use of a restriction enzyme cutting DNA in two directions (as described in the previous subsection). This enables designing transition molecules which also comprise writing the information (queue symbols) and reading multiple DNA fragments replicated by the PCR method. In the final phase of DNA reading, a chain is formed that changes the color of the solution due to

fluorescence. This approach may additionally be supplemented with DNA sequencing using the restriction enzymes [33,34].

## 5. Discussion

Searching for unique nucleotide sequences using machine-learning methods is crucial for the proposed applications of biomolecular computing. An important direction of research in this area has been shown to be the discovery of unique patterns in nucleotide sequences [9], as this knowledge enables biomolecular computer programming to read unique patterns in nucleotide sequences.

Current and potential future research directions and an example of the use of biomolecular computers as the hardware and software in technological solutions are presented. Among the vast array of available restriction enzymes, type IIB restriction endonucleases were selected as having a great potential in applications related to biomolecular computers. In addition, the idea of biomolecular queue automata is proposed along with queuing systems and queuing networks based on the use of type IIB restriction endonucleases and appropriate encoding of double-stranded DNA. This approach shows that it is possible to design biomolecular computers with queue memory as well as queuing networks using type IIB restriction endonucleases. The earlier technological solutions included practical laboratory experiments testing biomolecular finite state automata that were computers without memory [3,4]. Moreover, laboratory tests were already run on the functioning of type IIB restriction endonuclease (*Bae*I) in a system of biomolecular computers [6]. This provides practical foundations for creating different technological solutions linked with broadly understood biomolecular computers involving type IIB restriction endonucleases, specifically for creating diagnostic biochips.

The proposed approach to designing biochips with biomolecular computers can incorporate one or more than one type IIB restriction enzyme. The choice of multiple restriction enzymes (type IIB) depends on the possibility of using an appropriate type IIB restriction enzyme to read a genome fragment. If a given DNA fragment cannot be read by a given type IIB restriction enzyme (in use with designed transition molecules), it should be checked whether it is possible to read it using two or more restriction enzymes. When selecting the number of type IIB restriction enzymes, one should be guided by the principle of minimalism and choose the minimum number of restriction enzymes that can read a given DNA fragment. In this regard, attention should be paid to the main advantage of such an approach, namely that the use of more types of IIB restriction enzymes makes it possible to read DNA fragments of greater lengths. Thus, the use of multiple restriction enzymes in the diagnostic biochip (while maintaining the principle of minimalism) allows building more complex biochips.

Additionally, in the case of diagnostic biochips, we propose the use of a formal apparatus available in the theory of queue automata [29,35]. Particularly interesting are the deterministic input-driven queue automata [31]; nevertheless, this approach does not take into account encoding the input data with DNA chains and the action of restriction enzymes. In addition, we used the idea of queue automata—a concept mentioned as early as 1943 by Emil Leon Post [36]. The queue automata were the subject of studies in various areas [31,35,37–39]. From the PCR perspective, especially interesting are the queue automata in the context of scheduling problems [40], as in this method scheduling of the resulting DNA chains is particularly useful. We propose to advance the idea of the deterministic input-driven queue automata [31]. The other theories we used in our paper include the queuing theory [41] and queuing networks [42]. The former was used for modeling biological processes [43].

The use of type IIB restriction endonucleases also makes it possible to implement other models, based on the type of queue memory. Particularly promising seem the queuing systems based on the Erlang queuing theory [44,45]. It is worth noting that queuing systems can be combined to form queuing networks [46], which enable complex systems based on queues to be designed. Biomolecular implementation consists of the appropriate encoding

of all elements of the system, where the queue is the hardest element to implement. It should be noted that the idea of biomolecular implementation of a queue is provided in the present paper. For instance, in every queuing system, one can use one type IIB restriction endonuclease and combine it with others so that they form a queuing network.

Theoretical studies on biomolecular computers may provide practical solutions. Such new methods and models may have numerous medical applications, complementing conventional therapies and laboratory diagnostics. Early research on biomolecular computers focused on human-operated, laboratory-scale computers for solving complex computational problems. More recently, simple, autonomous and programmable molecular computers have been presented in which the input and output information can be given in a molecular form. Such computers, using biological molecules as input and biologically active molecules as output, could create a system for the "logical" control of biological processes. An autonomous biomolecular computer is proposed, which, at least in vitro, logically analyzes mRNA levels and responds by producing a molecule capable of affecting gene expression levels. This approach can be applied in vivo for biochemical sensing, genetic engineering, and even medical diagnosis and treatment.

A good example of the use of biomolecular computers is the concept of cancer diagnostics and cancer treatment based on silencing a cancer gene expression [47]. With this approach in mind, Ehud Shapiro's team developed a method for the activation of biomolecular software for cases where the reaction mixture contains typical neoplastic mRNA. For that purpose, they used a simple biomolecular computer, developed in a similar way as in their previous experiments [3,4]: it was built of software (in the form of double-stranded DNA) and hardware (in the form of the *Fok*I restriction enzyme), which could only assume one of two states—"yes" or "no". If "cancer mRNA" was present, then the biomolecular computer switched to the "yes" state; otherwise, it was in the "no" state. When all the sought for cancer features in the form of mRNA were present in the reaction mixture, then the biomolecular computer completed its operation and yielded the diagnosis of "yes", confirming the presence of cancer genes. Then, it released a drug capable of silencing the cancer gene expression. A part of that approach was also the proposed new concept of designing the input word, which enabled the release of the drug after the final cutting of the double-stranded DNA encoding the symbols of the biomolecular computer.

Another interesting application of DNA computing is the general concept of designing reconfigurable DNA nanovaults capable of controlling interactions between the enzymes [48]. Of particular interest seem to be issues involving DNA sequencing, where Sanger's sequencing remains the basic method [49]. Interestingly, one of the methods of studying sequencing is based on the use of class IIS restriction enzymes [33,50], and another, similar approach to sequencing is used as well [34,51]. The proposed concept, of which DNA sequencing is the main aim, is based on reading DNA by the restriction enzymes. It is worth noting that the idea of sequencing with the use of the restriction enzymes was shown before the first practical experiment, i.e., implementing biomolecular computers with a single restriction enzyme [3]. It is particularly interesting to study the feasibility of sequencing in the aspect of programming and using biomolecular computers, because the two approaches complement each other. It seems that comprehensive studies on sequencing with the use of biomolecular computers will yield numerous practical solutions in the field of sequencing. Another interesting application of restriction enzymes, regarding type IIB, is fingerprinting with the use of a class of such enzymes [52]. The researchers noted that type IIB restriction endonucleases can be used for studies on the genome of live organisms. Details of the biochemical cutting of DNA by means of type IIB restriction endonucleases were studied in a number of papers, including ours [6,53]. Type IIB enzymes were used in DNA-computing studies for developing a theoretical biomolecular push-down automaton model, that is, for a stack-memory computer. The main idea was based on using a circular double-stranded DNA and a single type IIB restriction endonuclease, *Psr*I [8], which enables cutting of the double-stranded DNA to the left and to the right of the restriction site. This approach is similar to the idea presented by Paul W.K. Rothemund in 1995, in which

the symbols are encoded by means of the double-stranded DNA, and input data processing is affected by alternately cutting and ligating the double-stranded DNA. It should also be noted that our practical studies on the use of multiple restriction enzymes in a single reaction mixture also showed the applicability of type IIB restriction endonuclease for designing biomolecular computers [6].

The action of a typical type IIB restriction endonuclease (more specifically, *Bae*I) was studied in earlier works in a system of biomolecular computers using multiple restriction enzymes [6]. Those studies demonstrated how to design synthetic nucleotides with LITMUS 38i plasmids, and how to obtain the respective components of a biomolecular computer working in a system of multiple alternately functioning restriction enzymes. The biochemical reactions were optimized for two restriction enzymes acting in a single reaction mixture [5]. In those studies, we also investigated in detail the optimization of the reaction medium in which a biomolecular computer operates and identified problems that may occur in preparing the respective components of biomolecular computers. Moreover, we explained how to solve any problems that may occur in the practical functioning of the biomolecular computer comprising multiple restriction enzymes within a single reaction mixture.

It is worth mentioning that earlier experiments demonstrated the action of a single restriction enzyme in a biomolecular computer system [3,4,54]. The experimental conditions of the work of various restriction enzymes were also investigated as potentially useful in the construction of biomolecular computers [55].

Numerous research papers describe the development of biosensors, or small devices enabling detection of organic compounds, for instance DNA, by means of miniature physical detectors. Another concept is to design biochips, or devices that enable a number of biochemical reactions to take place in a single device, usually integrated with electronic components that constitute the hardware [56,57]. A compelling approach to designing diagnostic devices is the lab-on-a-chip (LOC), with a number of strategies to design actual devices that are based on the lab-on-a-chip idea [58] and have practical applications involving operations on DNA [59]. An interesting illustration of how the lab-on-a-chip idea may be coupled with DNA computing is the concept of a molecular inference system [60–62]. One more example of an innovative approach to using biomolecular computing is the idea of creating biosensors [63,64].

What is frequently required in the practical implementations of biomolecular computers is an advanced mathematical formal apparatus, the purpose of which is to organize state-of-the-art knowledge of various approaches to constructing biomolecular computers. A precursor formal apparatus, which enables computations performed with the use of restriction enzymes and a double-stranded DNA to be grasped, was described as the theory of tailor automata [7]. A uniform formal system helps to precisely define a formal model used for implementing biomolecular computers. This approach to DNA computing enables problems connected with the practical designing of biomolecular computers to be solved effectively. For instance, from the practical point of view, it is an important thing to increase the number of states of a biomolecular computer [6,54,65], and to correctly encode the symbols of the biomolecular computer, simply because their precise design is required for the optimum operation of a biomolecular computer in the conditions of a reaction medium [66,67].

In addition to that, a number of other concepts of building biomolecular computers as well as their applications are currently in the phase of theoretical considerations. As a proof of the principle, a computer can be used to identify and analyze mRNA of disease-related genes from in vivo cancer models. Theoretical studies on biomolecular computers include, as a leading topic, studies on the splicing system [68–71]. Another interesting approach to DNA computing is the reaction system [72,73]—the subject of numerous studies recently [74–76].

In this paper, we proposed an idea of biochips that can be constructed using DNA. Therefore, the natural target processed easily by our biochips is cell-free DNA (cfDNA)

present in a wide variety of pathological conditions ranging from cancer to autoimmune diseases [77]. It is associated not only with disease occurrence, but also with the severity of symptoms and even treatment options. It is worth mentioning that cfDNA can easily be isolated using noninvasive methods and further processed. To date, cfDNA analysis has only been quantitative; however, it has recently been shown that cfDNA can be used to detect the sign of genomic instability. In other words, cfDNA profiling, defined as a common set of disease-specific mutations, can be a useful tool for screening healthy individuals for early symptoms of cancer and other diseases with the genomic instability background. Such a screening is time- and cost-consuming, as it requires sequencing followed by library preparation. Our conceptual biochips could be used instead of DNA sequencing. It is possible to configure our biochips in such a way that individual mutations would be scheduling by queue biomolecular automata and the signal is only emitted when the exact mutation is detected.

## 6. Conclusions

This paper presents a novel concept for building biochips of which the hardware and the software are based on biomolecular computers. Due to their unique properties, the biomolecular computers are of particular interest in building biochips. Among the various known theoretical models of computation, we selected the one that is exceptionally interesting regarding the aspect of PCR automation, namely, the deterministic input-driven queue automaton. We also indicated a group of restriction enzymes (type IIB restriction endonucleases), especially important for biomolecular computers. The use of such enzymes provides a foundation for designing biomolecular computers with memory, e.g., queue automata. It would also be interesting to pursue studies on the applicability of type IIB restriction endonucleases for designing various data structures known in computer science, e.g., tree data structure.

We also propose a new approach—Queue-PCR—for automation of the PCR method using the biomolecular computers. The Queue-PCR concept is important as a new idea of the use of biomolecular computers. It shows their applications and indicates the potential directions of study on new functionalities of programmable biomolecular nanodevices. This approach to PCR automation may be the beginning of new directions of study, focused on the automation of molecular genetics methods using biomolecular computers as the hardware. This provides fundamentals for creating biomolecular software for PCR solutions in their broad sense, thus enabling them to be programmed at the molecular level.

In the proposed approach, it is important to use machine-learning methods at the stage of designing biochips based on biomolecular computers, because the key element of the proposed solutions is knowledge on unique nucleotide sequences.

**Author Contributions:** Conceptualization, S.S.; investigation, S.S.; methodology, S.S., J.W. and T.P.; formal analysis, S.S. and J.W.; writing—original draft preparation, S.S., J.W., T.P. and I.M.; writing—review and editing, S.S., J.W., T.P. and I.M.; supervision, S.S.; project administration, S.S.; funding acquisition, I.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

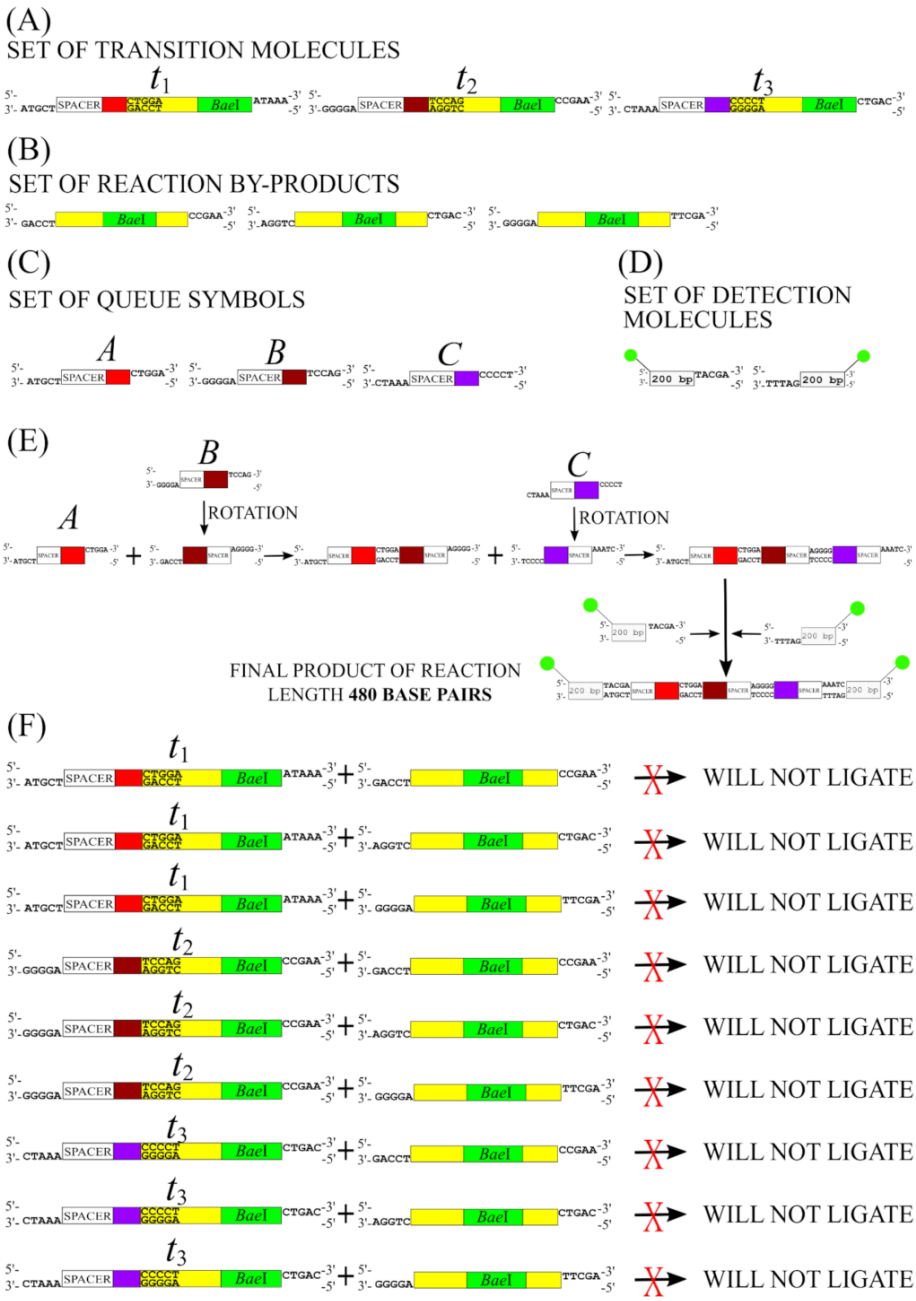


**Figure A1.** The list of DNA chains that are formed during the action of a biochip—an example: (**A**)—the set of all transition molecules; (**B**)—the set of all reaction by-products formed during the alternating action of type IIB restriction endonuclease (more specifically, *Bae*I); (**C**)—the set of all queue symbols. (**D**)—the set of all detection molecules; (**E**)—the process of biomolecular queue symbols ligation; (**F**)—the set of all reactions in which DNA molecules are not ligated together. Abbreviations: *A*, *B*, *C* (in italic text)—the queue symbols of a biomolecular computer; $t_1$, $t_2$, $t_3$, $t_4$, $t_5$—molecular software. The color markings are described in Figure 8.

## Appendix B

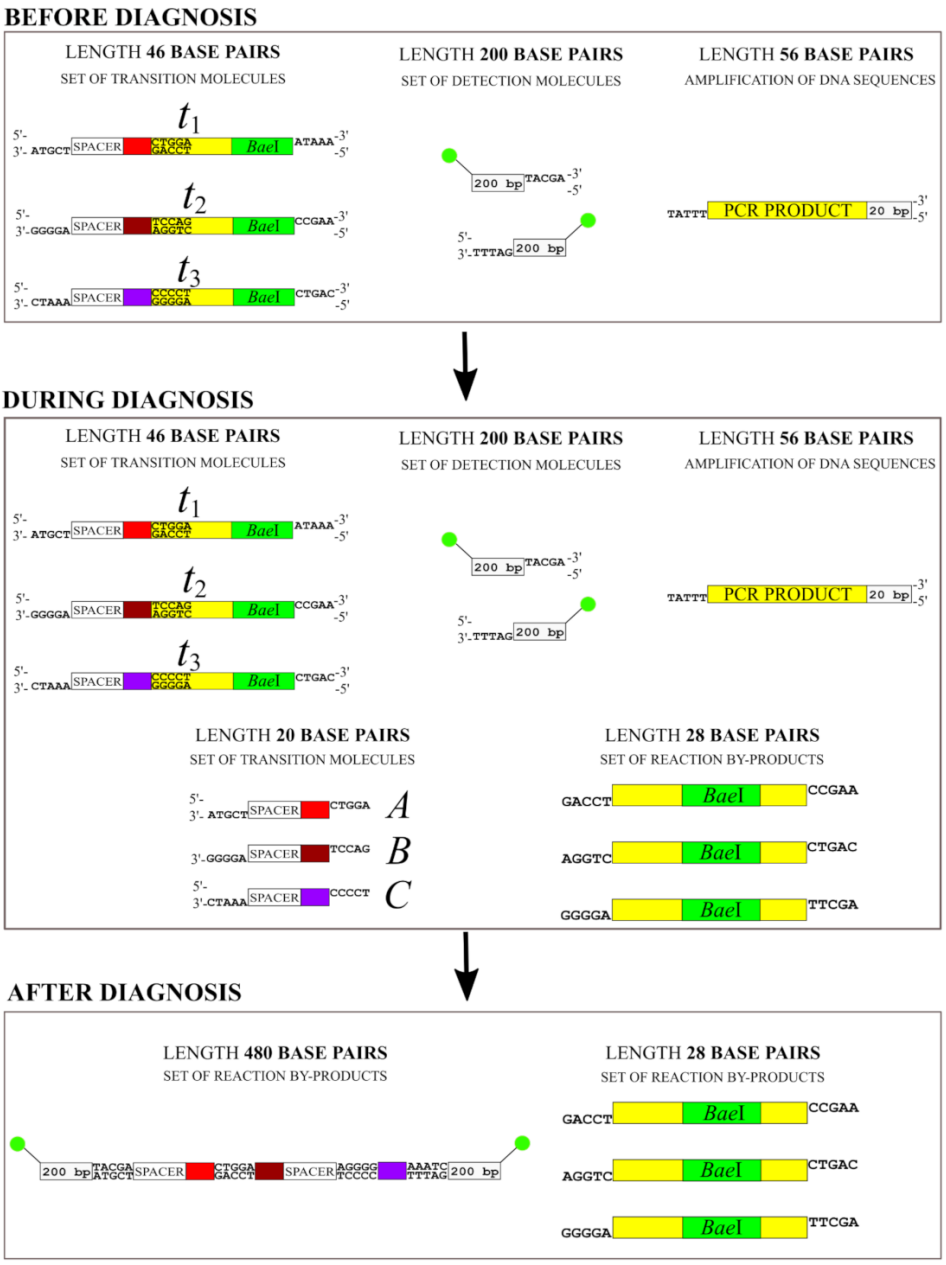


**Figure A2.** The list of DNA chain lengths before, during and after molecular diagnosis by means of a biochip based on a biomolecular computer. The length of DNA strands before the diagnosis is: 200 bp, 56 bp, and 46 bp, while after the molecular diagnosis it is: 480 bp, and 28 bp. The fluorescence mechanism is used, in which fluorescence occurs when a DNA chain of 480 appears in the solution. The color markings are described in Figure 8.

## References

1. Bennett, C.H. The thermodynamics of computation—A review. *Int. J. Theoret. Phys.* **1982**, *21*, 905–940. [CrossRef]
2. Rothemund, P.W.K. A DNA and restriction enzyme implementation of Turing machines. In *DNA Based Computers. DIMACS Series in Discrete Mathematics and Theoretical Computer Science*; Lipton, R.J., Baum, E.B., Eds.; American Mathematical Society: Providence, RI, USA, 1995; pp. 75–120.
3. Benenson, Y.; Paz-Elizur, T.; Adar, R.; Keinan, E.; Livneh, Z.; Shapiro, E. Programmable and autonomous computing machine made of biomolecules. *Nature* **2001**, *414*, 430–434. [CrossRef] [PubMed]
4. Benenson, Y.; Adar, R.; Paz-Elizur, T.; Livneh, Z.; Shapiro, E. DNA molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 2191–2196. [CrossRef] [PubMed]

5. Sakowski, S.; Krasiński, T.; Sarnik, J.; Blasiak, J.; Waldmajer, J.; Poplawski, T. A detailed experimental study of a DNA computer with two endonucleases. *Z. Naturforsch. C* **2017**, *72*, 303–313. [CrossRef]
6. Sakowski, S.; Krasiński, T.; Waldmajer, J.; Sarnik, J.; Blasiak, J.; Poplawski, T. Biomolecular computers with multiple restriction enzymes. *Genet. Mol. Biol.* **2017**, *40*, 860–870. [CrossRef]
7. Waldmajer, J.; Bonikowski, Z.; Sakowski, S. Theory of tailor automata. *Theor. Comput. Sci.* **2019**, *785*, 60–82. [CrossRef]
8. Cavaliere, M.; Jonoska, N.; Yogev, S.; Piran, R.; Keinan, E.; Seeman, N. Biomolecular implementation of computing devices with unbounded memory. *Lect. Notes Comput. Sci.* **2005**, *3384*, 35–49.
9. Agrawal, R.; Srikant, R. Mining sequential patterns. In Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, 6–10 March 1995; pp. 3–14.
10. Greener, J.G.; Kandathil, S.M.; Moffat, L.; Jones, D.T. A guide to machine learning for biologists. *Nat. Rev. Mol. Cell Biol.* **2022**, *23*, 40–55. [CrossRef]
11. Li, R.; Li, L.; Xu, Y.; Yang, J. Machine learning meets omics: Applications and perspectives. *Brief. Bioinform.* **2022**, *23*, bbab460. [CrossRef]
12. Eraslan, G.; Avsec, Ž.; Gagneur, J.; Theis, F.J. Deep learning: New computational modelling techniques for genomics. *Nat. Rev. Genet.* **2019**, *20*, 389–403. [CrossRef]
13. Bosco, G.L.; Di Gangi, M.A. Deep learning architectures for DNA sequence classification. In *International Workshop on Fuzzy Logic and Applications 2016*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 162–171.
14. Nguyen, N.G.; Tran, V.A.; Phan, D.; Lumbanraja, F.R.; Faisal, M.R.; Abapihi, B.; Kubo, M.; Satou, K. DNA sequence classification by convolutional neural network. *J. Biomed. Sci. Eng.* **2016**, *9*, 280–286. [CrossRef]
15. Luedi, P.; Dietrich, F.; Weidman, J.; Bosko, J.; Jirtle, R.; Hartemink, A. Computational and experimental identification of novel human imprinted genes. *Genome Res.* **2007**, *17*, 1723–1730. [CrossRef] [PubMed]
16. Chen, L.; Cai, C.; Chen, V.; Lu, X. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinform.* **2016**, *17*, S9. [CrossRef] [PubMed]
17. Kelley, D.R.; Snoek, J.; Rinn, J.L. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **2016**, *26*, 990–999. [CrossRef] [PubMed]
18. Amin, M.; Yurovsky, A.; Tian, Y.; Skiena, S. Deepannotator: Genome annotation with deep learning. In Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, Online, 15 August 2018; pp. 254–259.
19. Zeng, W.; Wu, M.; Jiang, R. Prediction of enhancer-promoter interactions via natural language processing. *BMC Genom.* **2018**, *19*, 84. [CrossRef] [PubMed]
20. Yuan, Y.; Bar-Joseph, Z. Deep learning for inferring gene relationships from single-cell expression data. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 27151–27158. [CrossRef]
21. Fudenberg, G.; Kelley, D.R.; Pollard, K.S. Predicting 3D genome folding from DNA sequence with Akita. *Nat. Methods* **2020**, *17*, 1111–1117. [CrossRef]
22. Yang, A.; Zhang, W.; Wang, J.; Yang, K.; Han, Y.; Zhang, L. Review on the application of machine learning algorithms in the sequence data mining of DNA. *Front. Bioeng. Biotechnol.* **2020**, *8*, 1032. [CrossRef]
23. Srikant, R.; Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. In Proceedings of the International Conference on Extending Database Technology, Avignon, France, 25–29 March 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 1–17.
24. Pearson, W.R. An introduction to sequence similarity ("homology") searching. *Curr. Protoc. Bioinform.* **2013**, *42*, 3.1.1–3.1.8. [CrossRef]
25. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403–410. [CrossRef]
26. Wu, S.; Manber, U. Fast text searching: Allowing errors. *Commun. ACM* **1992**, *35*, 83–91. [CrossRef]
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2017; p. 30.
28. Clauwaert, J.; Menschaert, G.; Waegeman, W. Explainability in transformer models for functional genomics. *Brief. Bioinform.* **2021**, *22*, bbab060. [CrossRef] [PubMed]
29. Kutrib, M.; Malcher, A.; Wendlandt, M. Queue Automata: Foundations and Developments. In *Reversibility and Universality. Emergence, Complexity and Computation*; Adamatzky, A., Ed.; Springer: Cham, Switzerland, 2018; Volume 30, pp. 385–431.
30. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; MIT Press: Cambridge, MA, USA, 2009.
31. Kutrib, M.; Malcher, A.; Mereghetti, C.; Palano, B.; Wendlandt, M. Deterministic input-driven queue automata: Finite turns, decidability, and closure properties. *Theor. Comput. Sci.* **2015**, *578*, 58–71. [CrossRef]
32. Krasiński, T.; Sakowski, S.; Popławski, T. Autonomous push-down automaton built on DNA. *Informatica* **2012**, *36*, 263–276.
33. Brenner, S. DNA Sequencing by Stepwise Ligation and Cleavage. U.S. Patent Application No. 5599675, 4 February 1997.
34. Jones, D. Iterative and Regenerative DNA Sequencing Method. U.S. Patent Application No. 5858671, 12 January 1999.
35. Brandenburg, F.J. Multiple equality sets and post machines. *J. Comput. Syst. Sci.* **1980**, *21*, 292–316. [CrossRef]
36. Post, E.L. Formal reductions of the classical combinatorial decision problem. *Am. J. Math.* **1943**, *65*, 197–215. [CrossRef]
37. Brandenburg, F.J. On the intersection of stacks and queues. *Theor. Comput. Sci.* **1988**, *58*, 69–80. [CrossRef]

38. Allevi, E.; Cherubini, A.; Reghizzi, S.C. Breadth-first phrase structure grammars and queue automata. In *International Symposium on Mathematical Foundations of Computer Science*; Springer: Berlin/Heidelberg, Germany, 1988; pp. 162–170.
39. Cherubini, A.; Citrini, C.; Reghizzi, S.C.; Mandrioli, D. QRT FIFO automata, breadth-first grammars and their relations. *Theor. Comput. Sci.* **1991**, *85*, 171–203. [CrossRef]
40. Breveglieri, L.; Cherubini, A.; Crespi-Reghizzi, S. Real-time scheduling by queue automata. In *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 131–147.
41. Thomopoulos, N.T. *Fundamentals of Queuing Systems: Statistical Methods for Analyzing Queuing Models*; Springer Science & Business Media: New York, NY, USA, 2012.
42. Garrido, J.M. Queuing Networks. In *Performance Modeling of Operating Systems Using Object-Oriented Simulation: A Practical Introduction*; Springer: New York, NY, USA, 2000; pp. 61–83.
43. Arazi, A.; Ben-Jacob, E.; Yechiali, U. Bridging genetic networks and queueing theory. *Phys. A Stat. Mech. Its Appl.* **2004**, *332*, 585–616. [CrossRef]
44. Erlang, A.K. The theory of probabilities and telephone conversations. *Nyt. Tidsskr. Mat.* **1909**, *20*, 33–39.
45. Kleinrock, L. *Queueing Systems*; Wiley: New York, NY, USA, 1975.
46. Zimmermann, A. Queuing Models. In *Stochastic Discrete Event Systems: Modeling, Evaluation, Applications*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 65–78.
47. Benenson, Y.; Gil, B.; Ben-Dor, U.; Adar, R.; Shapiro, E. An autonomous molecular computer for logical control of gene expression. *Nature* **2004**, *429*, 423–429. [CrossRef] [PubMed]
48. Grossi, G.; Jepsen, M.D.E.; Kjems, J.; Andersen, E.S. Control of enzyme reactions by a reconfigurable DNA nanovault. *Nat. Commun.* **2017**, *8*, 992. [CrossRef] [PubMed]
49. Sanger, F.; Nicklen, S.; Coulson, A.R. DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA* **1977**, *74*, 5463–5467. [CrossRef] [PubMed]
50. Brenner, S.; Livak, K.J. DNA fingerprinting by sampled sequencing. *Proc. Natl. Acad. Sci. USA* **1989**, *86*, 8902–8906. [CrossRef]
51. Jones, D.H. An iterative and regenerative method for DNA sequencing. *Biotechniques* **1997**, *22*, 938–946. [CrossRef]
52. Tengs, T.; LaFramboise, T.; Den, R.B.; Hayes, D.N.; Zhang, J.; DebRoy, S.; Gentleman, R.C.; O'Neill, K.; Birren, B.; Meyerson, M. Genomic representations using concatenates of Type IIB restriction endonuclease digestion fragments. *Nucleic Acids Res.* **2004**, *32*, e121. [CrossRef]
53. Marshall, J.J.; Gowers, D.M.; Halford, S.E. Restriction endonucleases that bridge and excise two recognition sites from DNA. *J. Mol. Biol.* **2007**, *367*, 419–431. [CrossRef]
54. Soreni, M.; Yogev, S.; Kossoy, E.; Shoham, Y.; Keinan, E. Parallel biomolecular computation on surfaces with advanced finite automata. *J. Am. Chem. Soc.* **2005**, *127*, 3935–3943. [CrossRef]
55. Chen, P.; Li, J.; Zhao, J.; He, L.; Zhang, Z. Differential dependence on DNA ligase of type II restriction enzymes: A practical way toward ligase-free DNA automaton. *Biochem. Biophys. Res. Commun.* **2007**, *353*, 733–737. [CrossRef]
56. Vo-Dinh, T.; Cullum, B. Biosensors and biochips: Advances in biological and medical diagnostics. *Fresenius J. Anal. Chem.* **2000**, *366*, 540–551. [CrossRef]
57. Vo-Dinh, T.; Cullum, B.; Stokes, D. Nanosensors and biochips: Frontiers in biomolecular diagnostics. *Sens. Actuators B Chem.* **2001**, *74*, 2–11. [CrossRef]
58. Temiz, Y.; Lovchik, R.D.; Kaigala, G.V.; Delamarche, E. Lab-on-a-chip devices: How to close and plug the lab? *Microelectron. Eng.* **2015**, *132*, 156–175. [CrossRef]
59. Lehmann, U.; Vandevyver, C.; Parashar, V.K.; Gijs, M.A. Droplet-based DNA purification in a magnetic lab-on-a-chip. *Angew. Chem. Int. Ed.* **2006**, *45*, 3062–3067. [CrossRef] [PubMed]
60. Wąsiewicz, P.; Mulawka, J. Lab-on-a-chip molecular inference system. In *Prace Naukowe Politechniki Warszawskiej. Elektronika*; Oficyna Wydawnicza Politechniki Warszawskiej: Warsaw, Poland, 2007; Volume 160, pp. 293–300.
61. Wąsiewicz, P.; Janczak, T.; Mulawka, J.J.; Płucienniczak, A. The inference based on molecular computing. *Cybernet. Syst.* **2000**, *31*, 283–315.
62. Wąsiewicz, P.; Malinowski, A.; Nowak, R.; Mulawka, J.J.; Borsuk, P.; Węgleński, P.; Płucienniczak, A. DNA computing: Implementation of data flow logical operations. *Future Gener. Comput. Syst.* **2001**, *17*, 361–378. [CrossRef]
63. de Murieta, I.S.; Rodríguez-Patón, A. DNA biosensors that reason. *Biosystems* **2012**, *109*, 91–104. [CrossRef]
64. Aiassa, S.; Terracciano, R.; Carrara, S.; Demarchi, D. Biosensors for Biomolecular Computing: A Review and Future Perspectives. *BioNanoScience* **2020**, *10*, 554–563. [CrossRef]
65. Unold, O.; Troć, M.; Dobosz, T.; Trusiewicz, A. Extended molecular computing model. *WSEAS Trans. Biol. Biomed.* **2004**, *1*, 15–19.
66. Krasiński, T.; Sakowski, S.; Waldmajer, J.; Popławski, T. Arithmetical analysis of biomolecular finite automaton. *Fundam. Inform.* **2013**, *128*, 463–474. [CrossRef]
67. Waldmajer, J.; Sakowski, S. A solution to the problem of the maximal number of symbols for biomolecular computer. *Informatica* **2019**, *43*, 485–494. [CrossRef]
68. Head, T. Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behavior. *Bull. Math. Biol.* **1987**, *75*, 737–759. [CrossRef]
69. Paun, G. On the splicing operations. *Discret. Appl. Math.* **1996**, *70*, 57–79. [CrossRef]

70. Kari, L.; Kopecki, S. Deciding whether a regular language is generated by a splicing system. *J. Comput. Syst. Sci.* **2017**, *84*, 263–287. [CrossRef]
71. Kari, L.; Ng, T. Descriptional Complexity of Semi-Simple Splicing Systems. *Int. J. Found. Comput. Sci.* **2021**, *32*, 685–711. [CrossRef]
72. Ehrenfeucht, A.; Rozenberg, G. Basic Notions of Reaction Systems. *Lect. Notes Comput. Sci.* **2005**, *3340*, 27–29.
73. Ehrenfeucht, A.; Rozenberg, G. Reaction systems. *Fundam. Inform.* **2007**, *75*, 263–280.
74. Corolli, L.; Maj, C.; Marini, F.; Besozzi, D.; Mauri, G. An excursion in reaction systems: From computer science to biology. *Theor. Comput. Sci.* **2012**, *454*, 95–108. [CrossRef]
75. Męski, A.; Penczek, W.; Rozenberg, G. Model checking temporal properties of reaction systems. *Inf. Sci.* **2015**, *313*, 22–42. [CrossRef]
76. Bottoni, P.; Labella, A. Transactions and contracts based on reaction systems. *Theor. Comput. Sci.* **2021**, *881*, 25–61. [CrossRef]
77. Volik, S.; Alcaide, M.; Morin, R.D.; Collins, C. Cell-free DNA (cfDNA): Clinical significance and utility in cancer shaped by emerging technologies. *Mol. Cancer Res.* **2016**, *14*, 898–908. [CrossRef]