Visualizing High Dimensional Datasets
Using Parallel Coordinates:
Application to Gene Prioritization

Thomas Boogaerts, Léon-Charles Tranchevent, Georgios A. Pavlopoulos, Jan Aerts, Joos Vandewalle
ESAT-SCD-SISTA / IBBT-K.U.Leuven Future Health Department, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, box 2446, 3001, Leuven, Belgium
Thomas.Boogaerts@student.kuleuven.be
{Leon-Charles.Tranchevent, Georgios.Pavlopoulos, Jan.Aerts, Joos.Vandewalle} @esat.kuleuven.be

Abstract—In this paper, we introduce a visualization tool for interactive and efficient exploration of high dimensional data using parallel coordinates. An algorithm is developed to find an optimal permutation of dimensions, which allows the data miner to immediately see the most important features or irregularities in the dataset. This is implemented as a genetic algorithm based on the travelling salesman problem using maximal correlation as fitness. Other features of the tool include selection operators to group the data such as selection by intersection or by angle, orthogonal and density plots complementing the parallel coordinates plot, manual arrangement of permutation order of the dimensions, possibility to show all plots necessary to see all dimensional relations and displaying a certain number of standard deviations for each dimension separately. The tool is applied to multiple gene prioritization cases in search of genes that are relevant to certain genetic disorders. The used datasets are obtained with the MerKator and Endeavour tools and include a Breast cancer, Cataract, Charcot-Marie-Tooth and Cardiomyopathy dataset, as well as a dataset relating 29 diseases with 22206 genes. Our tool, manual and data can be downloaded from http://www.toomas.be/parcoord/.

I. INTRODUCTION

Data mining is in essence the extraction of relevant information from large datasets. Usually, it is not known in advance which information one is trying to find. This is why visualization is so important; it can reveal small irregularities by transforming the data to a more intuitive and clear form. These irregularities are easy to spot visually but very hard to define mathematically. Due to the huge amount of data being generated by modern experimental methods in all areas, the need for efficient data visualization rapidly increases.

Traditional data visualization methods usually deliver poor results when applied to large datasets. Parallel Coordinates is a technique in which datapoints in an orthogonal coordinate system are projected onto parallel axes, which transforms these points to polygonal lines (see Section 2). This technique does not suffer from the curse of dimensionality as much as for example the scatterplot technique. Parallel Coordinates is a technique onto parallel axes, which transforms these points to polygonal lines when applied to large datasets. This technique does not suffer from the curse of dimensionality as much as for example the scatterplot technique.

The main parallel coordinates theory used to design the program is good when it enables interesting patterns to emerge. In this case a measure for what constitutes an interesting pattern is needed. 2) An ordering is good when it hides certain aspects of the data so that interesting irregularities become more prominent. The genetic algorithm is implemented and tested in the ParCoord program (see Section 4) and several gene prioritization datasets are analyzed with it (see Section 5).

II. BACKGROUND AND RELATED WORK

The main parallel coordinates theory used to design the program can be found in Inselberg’s book [3], which focuses mainly on the underlying geometry, however there is also a standalone chapter on data mining present. Several papers discuss the possibility of changing the permutation order: in [4], the clutter is defined as the proportion of outliers against the total number of data points, and this clutter is minimized by reordering the axes. In [5], the idea of Similarity-Oriented Dimension Ordering is explored, in which dimensions with similar patterns are placed adjacently. The problem of rearranging dimensions in a parallel coordinates plot is closely related to a travelling salesman problem (TSP). A genetic algorithms approach is thought to be able to find a very good (but not necessarily optimal) solution very fast for these kinds of problems. The main text used to program the Genetic Algorithms is ‘Genetic algorithms and genetic programming: modern concepts and practical applications’ [6]. A special crossover operator was used, namely a modified version of the Sequential Constructive Crossover (SCX), as defined in [7]. Thorough reviews to better understand gene prioritization and how the datasets were generated are presented in [8] and [9].
III. ParCoord PROGRAM

The main reason to develop the ParCoord program was that none of the previously existing parallel coordinates plotting tools seem to possess all of the features of the ParCoord program described below:

- **Loading datasets:** Most delimiter-separated values (DSV) datasets can be loaded. Dimension labels can be loaded separately or included as first line of the data file. Infinity values are plotted as 110% of the maximum value in each dimension. Datapoints containing missing values are plotted as an interrupted line. The values in discrete dimensions (finite number of possible values) are plotted equidistantly.

- **Identifiers:** Each datapoint can be labeled by an identifier. This enables the user to select certain datapoints and immediately see a list of corresponding identifiers or vice versa.

- **Groups:** Each datapoint is part of a specific group, and each group has a colour. For example: when loading a new dataset, all datapoints are part of the “red” group and the polygonal lines representing these datapoints are coloured red. Each group’s opacity can be changed from 0 to 255 to make the plot clearer for huge datasets. This way, the density (number of lines) of each group can be estimated at each location. Lines can be made invisible, which is useful when certain groups have to be excluded from possibly being selected.

- **Switching Permutation Order:** The permutation order of the dimensions can be changed manually. The program can also calculate a number of permutations necessary to see all the relations between dimensions and switch between these permutations. For an N-dimensional dataset, the remaining \( \left\lfloor \frac{N}{2} \right\rfloor - 1 \) permutations are obtained from the first very easily by adding 1 to each element in the permutation (modulo N) successively to obtain each new permutation. Several genetic algorithms are included to find “optimal” permutation orders.

- **MinMax and SDs:** There are two possible modes in each dimension: MinMax plots the datapoints so the minimum value appears at the bottom of the axis and the maximum value appears at the top of the axis. SDs plots the data so the mean of the data is exactly in the middle of the axis, and a certain number of standard deviations is displayed. In this mode, there are two variables that can be changed for each axis separately: the number of standard deviations to be plotted and where the mean should be located on the axis. The SDs mode is especially useful when outliers are present. In most of the traditional parallel coordinate plotting programs, something similar to the MinMax mode is used, which obscures the plot a great deal when there are outliers present; all the other data is squashed together. The SDs mode can also be used for zooming purposes to carefully scrutinize the plot.

- **Values on axes:** When hovering over the axes, the values are displayed. If the axis is in MinMax mode, the original values are shown. If the axis is in SDs mode, then the number of standard deviations from the mean (normalized value) is displayed, as well as the original value. If the axis is discrete, then the closest category is displayed.

- **Density plot:** There is a density plot present: it shows a histogram for each dimension separately. This density plot also takes groups into account, so when there is for example a red group of datapoints and a blue group of datapoints, then the density plot also shows red histograms and blue histograms for these groups separately. The density plot changes in real time when selecting groups.

- **Orthogonal coordinates plot:** For each two adjacent dimensions, it is possible to show an orthogonal plot of the plane formed by these dimensions.

- **SelectLines:** It is possible to select specific datapoints very easily by clicking-and-dragging the mouse: all the polygonal lines intersecting the line formed by the dragging of the mouse will be selected and coloured according to the active group.

- **SelectAngles:** It is also possible to select datapoints by the angle formed by the line representing this datapoint in between two dimensions and a horizontal line. So, for example, in a parallel coordinates plot with dimensions X, Y and Z, you can select all the lines that have an angle of 45 degrees (or a slope of 1) between dimensions X and Y. There is also the option of selecting a certain range for the angle e.g., all the lines with an angle between 35 and 55 degrees.

- **SelectOrthogonal:** In the orthogonal coordinates plot, datapoints are selected by drawing a line in the plot, separating the datapoints into two groups.

- **Complex selections:** What makes these selection tools very useful is the possibility of combining selections: performing one selection after the other, the specific set of datapoints needed can be formed or “cut out” of the dataset. Groups can be hidden, so that selections do not influence them.

- **Synchronized Plots:** When making changes in the parallel, orthogonal or densities plot, the other two plots are updated automatically.

IV. PERMUTATION ORDER GENETIC ALGORITHM

To see all relations between dimensions, \( \left\lfloor \frac{N}{2} \right\rfloor \) different permutation orders have to be generated. It would be useful to generate a permutation order automatically in which all interesting relations are present. A possible approach is maximizing correlation, which is possibly not only important in itself (in the sense that correlated variables are interesting features of the data), but also to be able to see other features or irregularities better e.g., positive correlation minimizes crossing lines in the parallel coordinates plot which makes it more clear. Two correlation measures are used, namely Pearson’s (linear) correlation and Spearman’s (rank-based, monotonic) correlation.

The problem of finding an optimal permutation order according to a specific measure strongly resembles a Travelling Salesman Problem (TSP). Each node in the graph is a dimension and each edge in the graph is a possible pair of dimensions. Each edge has an associated value with it according to a specific measure e.g., Pearson’s correlation measure between the two dimensions. In a regular TSP, edges are typically associated with (Euclidean) distance values and the path to be found is a closed loop (Hamiltonian cycle). In the dimension ordering problem, the path to be found is an open “route” (Hamiltonian path in a graph) because the last and first dimension in a parallel coordinates plot are not connected.

A genetic algorithm typically consists of a genotype representation of the individual, a mutation operator, a crossover operator, selection operators, a fitness function and an initial population. Each new generation is formed by the selection of individuals from the previous generation according to fitness (how well the specific individual solves the problem). The selected individuals are crossed over and mutated with a certain probability.

The dimension ordering genetic algorithm is implemented in Java using the Watchmaker framework [2]. We use a path representation as genotype, with random permutations as initial population. The mutation operator is a simple reciprocal exchange mutation: two dimensions in the route are simply swapped. The used crossover
operator is Sequential Constructive Crossover (SCX) [7]. The selection operator is rank-based Stochastic Universal Sampling (SUS). The fitness function is the sum of absolute values of a certain correlation measure for each pair of dimensions present in the permutation. The population exists of 160 individuals and remains constant. The algorithm is run for 800 generations. The five percent fittest individuals in each generation is automatically inserted in the next generation (elitism).

The genetic algorithm is applied to a highly nonlinear dataset, namely the MAGIC gamma telescope dataset [10] using Spearman’s correlation to search for the highest positively correlated dimension pairs (Fig. 1 and 2 (a)) and highest negatively correlated dimension pairs (Fig. 1 and 2 (b)). This dataset consists of 19020 entries in 11 dimensions (of which one class variable).

The genetic algorithm can be used for very high dimensional data, e.g., 2) gene-gene links and individual gene information. The gene-disease links and the gene-gene links can be combined to compare the genes that are known to play a role in the development of the disorder with the input genes (candidate search space). This way the candidate genes can be prioritized or a selection of the most promising genes can be made using statistical or machine learning techniques.

To illustrate the features of the ParCoord program and to demonstrate its usefulness when dealing with gene prioritization, three case studies are performed. The data used in all three cases was collected from public repositories (both disease-gene links and gene-gene links) and aggregate scores for the different diseases for each gene were calculated with the MerKator [11] and Endeavour [12] gene prioritization tools. The following data sources were used:

- **Text Mining Data:** Represents the strength of the associations between genes and the results of a text mining approach performed on the whole corpus of scientific papers for specific biological processes [13] [14].
- **Annotation Data:** One binary vector per gene, containing the association between this gene and specific functional terms. A lot of this data is not yet experimentally verified. Used Annotation databases: Interpro (Terms are active protein domains [15]), Kegg (Terms are different biological pathways [16]) and Ensembl (Terms are tissues [17]).
- **Expression Data:** Full matrix, genes are rows and columns are tissues tested (e.g., brain, liver, blood, ...). Each value represents the level of activity of the gene in the tissue. Used Expression databases: Son (168 human tissues [18]) and Su (168 mouse and rat tissues, mapped to human [19]).
- **Protein-Protein Interactions:** Proteins that physically interact with each other usually do so to exert a common function, which means genes that code for physically interacting proteins are more likely to have disease dependencies. Used databases are: Biogrid [20] and Biogrid [21].
- **Regulatory information:** ‘Controller’ proteins can bind in front of ‘controlled’ genes, which change their activity. The Motif data tries to capture this by analyzing which motifs are present in front of every gene [22].
- **Blast:** Sequence similarities in general (not focusing on active sites like in Interpro) between all proteins pairwise [23].

### A. Gene-Score Data

The first dataset (Case 1) contains 10603 genes in 11 dimensions, each dimension representing a different measure of how much these genes are involved in Breast cancer (three other datasets are available, for Cardiomyopathy, Charcot-Marie-Tooth and Cataract). One of the 11 dimensions is the global (aggregate) score, calculated by machine learning algorithms from the other 10 dimensions. Note that there are a lot of null values present in this data.

Looping through the six permutations necessary to see the relations between all dimensions (also using the opacity feature and grouping the high, medium and low global scores together with the SelectLines tool), it is obvious that the global score (“GRS”) is highly linearly (positively) correlated with the Biogrid score. Text is also correlated with the global score but in a different way. From Fig. 3 (a), it seems that datapoints with high global scores are likely to also have a high Text score, while those with a low local score do not seem favoured in any way concerning the Text ranking. The relation between global and Text scores seems to be a logical implication. This seems intuitive: in the corpus of scientific literature about Breast cancer, the genes that are never talked about have a high likelihood to not have any influence on Breast cancer, while the genes that
are known to have a large influence on Breast cancer are likely to be frequently mentioned in the literature. However, genes that are frequently mentioned in the Breast cancer literature do not necessarily have a large influence on Breast cancer. Fig. 3 (b) confirms these findings.

![Parallel coordinates plots of Text, global score (GRS) and Biogrid](image)

![Orthogonal coordinates plots of Text-GRS plane and Biogrid-GRS plane](image)

**Fig. 3: Breast Cancer data**

In the Breast Cancer dataset, the datapoints with the highest EnsemblEst scores don’t seem to have high scores in other dimensions. This in contrast to the data of another genetic disease namely Cardiomyopathy, where the datapoints with the highest EnsemblEst scores have a high probability of having high SonEtAl, SuEtAl, Interpro, Text and Bind scores. This means that for the Breast cancer case, the genes which are expressed in the same tissues as the Breast cancer genes (so the datapoints with high EnsemblEst scores) are not more likely to have other high scores: somehow Breast cancer genes are not expressed exclusively in the same tissues. This in contrast to Cardiomyopathy genes, which seem to have a common expression pattern (see Fig. 4).

![Parallel Coordinates plot of Breast Cancer data with highest EnsemblEst scores in blue](image)

![Cardiomyopathy dataset](image)

**Fig. 4: Parallel Coordinates plot of Breast Cancer data with highest EnsemblEst scores in blue**

To illustrate the use of identifiers, the top global score genes in the Breast cancer dataset are selected and grouped together in the Cardiomyopathy set. It seems the high score Breast cancer genes also score very well globally for Cardiomyopathy (see blue lines in Fig. 5, note that Text and global score are highly correlated). To see if this is true in general (for each disease), a parallel coordinates plot is created using the global scores of each disorder. In Fig. 6 (a), genes with a high value in the Charcot-Marie-Tooth dimension are highlighted in blue. It is clear that these also score well for the other diseases. This observation is found to be true for each of the four diseases, which is in agreement with the work of Gillis and Pavlidis who report that guilt-by-association methods tend to introduce a systematic bias towards multifunctional genes [24] [25].

![Parallel coordinates plot of Cardiomyopathy dataset with genes that have high global score for Breast cancer in blue](image)

**Fig. 5: Parallel coordinates plot of Cardiomyopathy dataset with genes that have high global score for Breast cancer in blue**

![Parallel coordinates plot of global ranking scores for all four diseases with top global scores for Charcot-Marie Tooth in blue](image)

**Fig. 6: Parallel coordinates plot of global ranking scores for all four diseases with top global scores for Charcot-Marie Tooth in blue**

### B. Gene-Disease Data

Two datasets are analyzed in which the dimensions represent global (aggregate) scores for specific genetic disorders. Both datasets consist of 29 dimensions, but the first contains 22206 genes. The second dataset contains only 538 genes, which are thought to be the disease-genes (each gene is associated with at least one of the disorders in the dataset). We will refer to these datasets as Case 2 and Case 3. For Case 2, only the first 13 dimensions are analyzed.

After changing the permutation order and using the opacity and zooming tools, some interesting irregularities are found. The irregular genes can be highlighted by combining the SelectAngles and SelectLines tools to create complex selections (Fig. 7).

![Irregular genes highlighted in blue](image)

**Fig. 7: Irregular genes highlighted in blue**

Fig. 8 shows the irregular genes in blue. These genes seem to have a very common pattern over all diseases. Checking the dataset, it is indeed clear that these genes have extremely similar values, but they are not identical. Now that the irregular genes are separated from the rest, their names can be extracted using identifiers.

The Case 3 data contains only the disease-genes (genes that are known to influence at least one of the 29 diseases). In figure 9 (b), two genes that score extremely high for all diseases except for Parkinson’s, Neuropathy and Anemia are indicated in blue (Ensembl Gene IDs: ENSG00000176124 and ENSG00000116652). Another exceptional gene is indicated in green: it scores extremely
well for all diseases except for Parkinson’s (Ensembl Gene ID: ENSG00000183566). Most genes score high/medium for a few diseases and don’t score well at all for the rest of the diseases. Another feature of the data that is immediately obvious is the positive correlation between Anemia and Leukemia (Fig. 9 (a)).

C. Permutation Order Genetic Algorithm

The genetic algorithm is applied to the Case 2 dataset. The resulting plot clearly has less “X” patterns (which are caused by negative correlation patterns), as can be seen in Fig. 10 (b). In Case 2, all dimensions are positively correlated to a certain extent, but only the most correlated dimensions are shown which results in less visible “X” patterns (negative correlation patterns are less likely to be present in highly positively correlated dimensions). Fig. 10 (c) is obtained by applying the genetic algorithm using minimum correlation as a measure instead of maximum correlation. The “X” patterns are clearly more pronounced in this case.

The majority of highly correlated dimension pairs can be found without having to plot the 15 permutations necessary to see all relations between dimensions. Note that the goal in this case is not finding the most correlated pairs, it is finding a permutation order in which as many highly correlated pairs as possible are present.

The highly correlated disease pairs seem to be mental retardation ↔ Hemolytic Anemia, Leukemia ↔ Anemia, Muscular Dystrophy ↔ Dystonia and to a lesser extent Colorectal cancer ↔ Zellweger syndrome, Anemia ↔ Retinitis Pigmentosa and Breast cancer ↔ Mental retardation. This can be seen in Fig. 10 (b) by the absence of “X” patterns. It is interesting to see that some of these disease pairs are expressed in the same tissues. Lowly correlated disease pairs are Ichthyosis ↔ Anemia and Retinitis Pigmentosa ↔ Anemia. Zooming in on both plots (before and after genetic algorithm), it seems easier to spot patterns in the second plot. A possible explanation is the fact that the “X” pattern is gone and there’s an almost flat red space visible, in which it is much easier to spot patterns (so the reordering can hide certain aspects of the data in order for interesting irregularities to become more prominent). A second explanation is that particular patterns only become visible between highly correlated dimensions. It is now clear that sometimes maximizing positive correlation is not only important in itself (in the sense that correlated variables are interesting features of the data), but also to be able to see other features or irregularities better. In Fig. 11, the obvious patterns are indicated with blue ellipses. These patterns are clearly visible in both plots. Some more obscure patterns are only visible in the second plot (black ellipses). In Fig. 12, the Pearson’s correlation for each pair of dimensions is shown.
more should be tested. The additional benefit of using a correlation tried (Pearson’s correlation and Spearman’s correlation), but many in the gene prioritization field. A genetic algorithm is designed to find coordinates plot for the "visible after genetic algorithm (black ellipses) patterns visible in both plots (blue ellipses), less obvious patterns only Fig. 11: (b) After applying genetic algorithm (Pearson’s correlation version), zoomed (a) As is (before applying genetic algorithm), zoomed (a) As is (before genetic algorithm) (b) Using genetic algo- rithm favouring highly correlated pairs (c) Using genetic algo- rithm favouring loosely correlated pairs Fig. 12: Pearson’s correlation for each pair of dimensions in the parallel coordinates plot for the Case 2 dataset after applying genetic algorithms (Pearson’s correlation versions), pairs are in same order of display as in parallel coordinates plots.

VI. CONCLUSION
The ParCoord program can be used to efficiently analyze high dimensional datasets of any kind, and is also very useful for the extraction of possibly relevant genes regarding specific genetic disorders in the gene prioritization field. A genetic algorithm is designed to find a permutation order which reveals the most useful information about the data. Only two measures of how interesting a permutation is were tried (Pearson’s correlation and Spearman’s correlation), but many more should be tested. The additional benefit of using a correlation measure is the decluttering of the plot.

REFERENCES