Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

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, Charcoth-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/*.**

***Index Terms*—data visualization, parallel coordinates, genetic algorithm, gene prioritization**

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. However, it is thought that interactivity plays a large role in the effective use of parallel coordinates.

One of the emerging fields in which visualization is very relevant is gene prioritization, which is the ordering of a list of candidate genes according to relevancy with respect to certain biological processes (usually hereditary diseases) in order to obtain the most promising genes (see Section 2). Several gene prioritization tools which try to automate this process by using supervised machine learning techniques and statistical tools are available. The result of these tools is an ordered list or selection of genes that have the highest probability of influencing the disorder. Usually, it is very hard to intuitively understand these results without visual aid.

A program (hereafter called *ParCoord*) is developed using Java, the Processing library [1] and the Watchmaker framework [2] to visualize high dimensional datasets (see Section 3).

To find a good permutation order of the dimensions (to be able to immediately see the most interesting features of the data), a method was developed based on the solution to the Travelling Salesman Problem using a genetic algorithm. As to what constitutes a good permutation order, two routes are explored: 1) An ordering 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].

Funding: The authors would like to acknowledge support from: Research Council KUL:ProMeta, GOA Ambiorics, GOA MaNet, CoE EF/05/007 SymBioSys en KUL PFV/10/016 SymBioSys , START 1, several PhD/postdoc & fellow grants. Flemish Government: FWO: PhD/postdoc grants, projects G.0318.05 (subfunctionalization), G.0553.06 (VitamineD), G.0302.07 (SVM/Kernel), research communities (ICCoS, ANMMM, MLDM); G.0733.09 (3UTR); G.082409 (EGFR) IWT: PhD Grants, Silicos; SBO-BioFrame, SBO-MoKa, TBM-IOTA3 FOD:Cancer plans, IBBT. Belgian Federal Science Policy Office: IUAP P6/25 (BioMaGNet, Bioinformatics and Modeling: from Genomes to Networks, 2007- 2011); EU-RTD: ERNSI: European Research Network on System Identification; FP7-HEALTH; CHeartED

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

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 $\lceil \frac{N}{2} \rceil - 1$ 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 squished 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 inbetween 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, $\lceil \frac{N}{2} \rceil$ 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) [10]. 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 can be used for very high dimensional data to extract a lot of information without having to display a large number of parallel coordinates plots. Niching (*e.g.*, crowding, fitness sharing) can potentially be used to obtain several highly interesting but maximally dissimilar permutations (*i.e.*, without much overlap of dimensional pairs).

More information and experiments regarding the implementation of the algorithm can be found at *http://www.toomas.be/parcoord/*.

V. Application to Gene Prioritization

Gene prioritization consists in predicting which candidate genes are promising with respect to a disease under study. More precisely, candidate genes that are highly similar to the known disease genes are considered promising, and therefore should be investigated first. Several gene prioritization tools have been developed in the last decade [8] and they rely on many genomic databases [8]. There are two main classes of databases: 1) prior knowledge about different disease-gene links (*e.g.*, genes that are already known to play an important role in the development of the genetic disorder) and 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 candidate genes. 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.

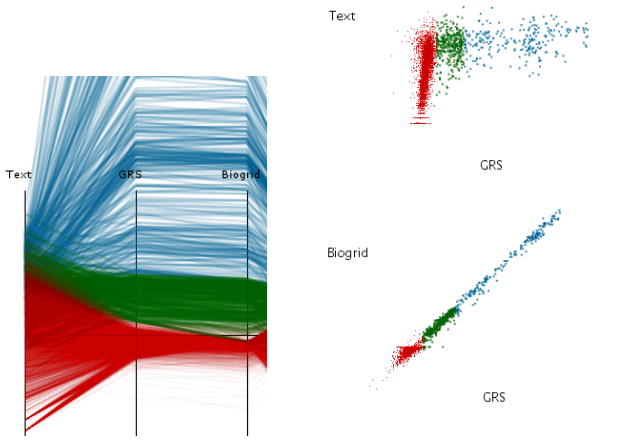
To illustrate the features of the *ParCoord* program and to demonstrate its usefulness when dealing with gene prioritization, three case studies are performed. The data used in all three cases was obtained using either MerKator [11] or Endeavour [12], two gene prioritization tools. These two methods combine several data sources in order to derive a global ranking. Briefly, the following data sources were used (see [11] and [12] for a more thorough description):

- **Text Mining Data:** the associations between genes and ontological terms resulting from a text mining approach of the MEDLINE corpus [13] [14].
- **Functional Annotation Data:** association between genes and specific functional terms. A lot of this data is not yet experimentally verified. This category includes Interpro (terms are active protein domains [15]), Kegg (terms are biological pathways [16]) and EnsemblEst (terms are tissues [17]).
- **Expression Data:** this measures the level of activity of the genes in several tissues. Data derived from Son et al. [18] (168 human tissues including replicates) and Su et al. [19] (168 mouse and rat tissues including replicates, mapped to human). In this case, the raw expression values are extracted from the microarray experiments (different from EnsemblEST, for which binary values are computed - either a gene is active or inactive in a tissue).
- **Protein-Protein Interactions:** Proteins that 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. Examples of used databases are Bind [20] and Biogrid [21].
- **Regulatory information:** 'controller' proteins can bind in front of 'controlled' genes, in order to change their activity. The Motif data tries to capture this by analyzing which binding motifs are present in the upstream sequences of the genes [22].
- **Blast:** protein sequence similarities between all proteins pairwise computed using NCBI Blast (Sig filtering, max e-value set to 1000, other parameters set to default) [23].

A. Visualizing Single Disease Prioritizations

The first dataset (*Case 1*) contains prioritizations for four diseases (breast cancer, cardiomyopathy, Charcoth-Marie-Tooth and cataract) obtained using MerKator. For each disease, there are in total 10603 genes and 11 dimensions, 10 of which represent the gene scores computed using the databases mentioned above. The last dimension is the global (aggregate) score, calculated by Merkator from the 10 individual scores. Note that there are a lot of missing values in this data. In our system, a missing value means that there is no value for one of the dimensions, the other values are kept and the corresponding lines are drawn.

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. 1 (a), it seems that datapoints with high global scores are likely to also have a high Text score, while those with a low global 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 of not having 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. 1 (b) confirms these findings.



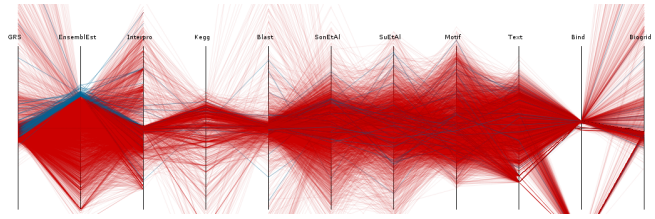
(a) *Parallel coordinates plots of Text, global score (GRS) and Biogrid*

(b) *Orthogonal coordinates plots of Text-GRS plane and Biogrid-GRS plane*

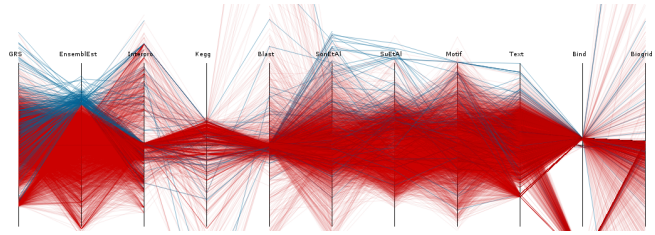
Fig. 1: *Breast Cancer data*

In the Breast Cancer dataset, the datapoints with the highest EnsemblEst scores dont 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. 2).



(a) *Breast cancer dataset*



(b) *Cardiomyopathy dataset*

Fig. 2: *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. 3, 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. 4 (a), genes with a high value in the Charcoth-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].

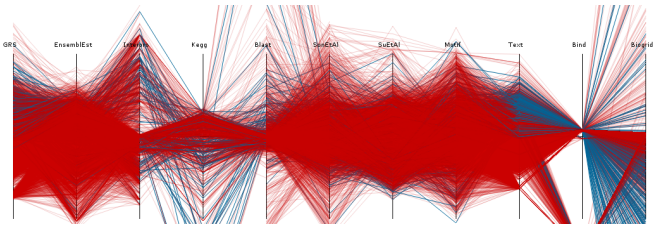


Fig. 3: *Parallel coordinates plot of Cardiomyopathy dataset with genes that have high global score for Breast cancer in blue*

B. Visualizing Multiple Disease Prioritizations

We then extend our analysis to study several diseases at once. Using Endeavour, prioritizations are run for 29 selected diseases [12]. Two extra datasets with 29 dimensions are then created (Cases 2 and 3), for which each dimension contains the global (aggregate) scores for a specific genetic disorder. The number of genes however differ. The 538 genes that are known to be involved in any of the 29 diseases are in Case 3. The remaining 22206 genes are in *Case 2*. Notice that for *Case 2*, only the first 13 dimensions are analyzed.

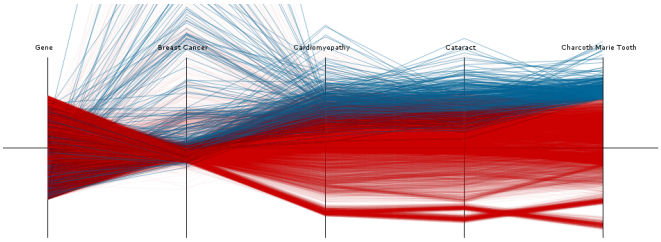
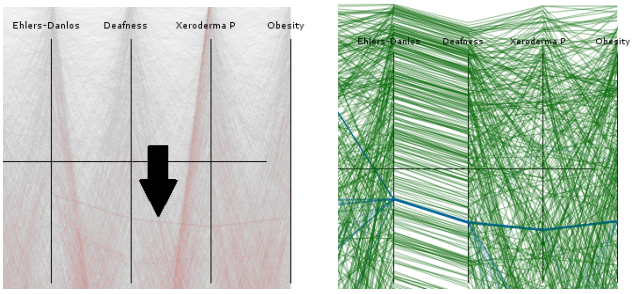


Fig. 4: *Parallel coordinates plot of global ranking scores for all four diseases with top global scores for Charcot-Marie Tooth in blue*

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. 5).



(a) *Zooming in on irregularities between Ehlers-Danlos and deafness dimensions (black arrow indicates irregularity)*

(b) *Selection by* SelectAngles *tool (in green) and subsequently* SelectLines *tool (in blue)*

Fig. 5: *Highlighting irregularity between Ehlers-Danlos and deafness dimensions*

Fig. 6 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.

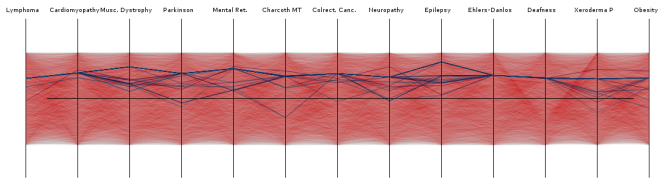
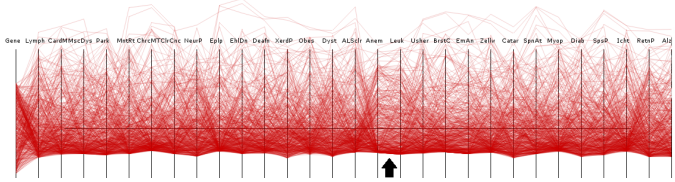


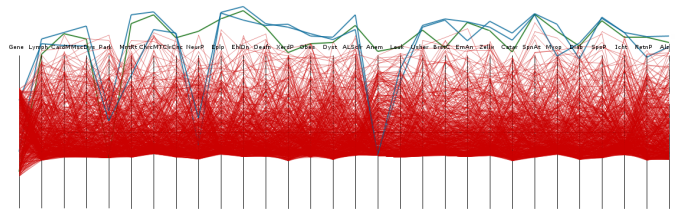
Fig. 6: *Irregularities between Ehlers-Danlos and deafness dimensions in the* Case 2 *parallel coordinates plot (in blue)*

The *Case 3* data contains only the disease-genes (genes that are known to influence at least one of the 29 diseases). In figure 7 (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. 7 (a)).



(a) *Anemia-Leukemia indicated with arrow*



(b) *Outliers in green and blue*

Fig. 7: *Parallel coordinates plot of all 29 dimensions of the* Case 3 *data*

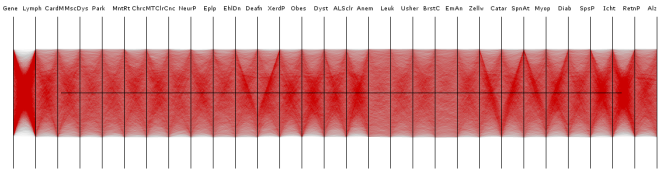
C. Permutation Order Genetic Algorithm

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. 8 (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. 8 (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.

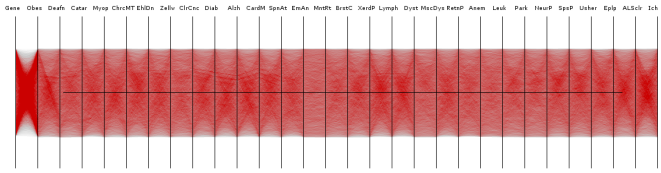
The majority of highly correlated dimension pairs can be found without having to plot the 15 permutations necessary to see all relations between dimensions. Note that the goal in this case is not finding the most correlated pairs, it is finding a permutation order in which as many highly correlated pairs 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. 8 (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.

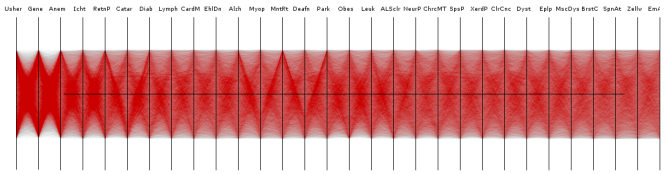
Zooming in on both plots (before and after genetic algorithm), it seems easier to spot patterns in the second plot. A possible explanation is the fact that the "X" pattern is gone and there's an almost flat red space visible, in which it is much easier to spot patterns (so the reordering can hide certain 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



(a) *As is (before applying genetic algorithm)*



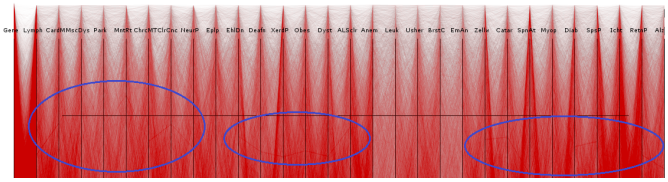
(b) *After applying genetic algorithm (Pearson's correlation version), maximum correlation*



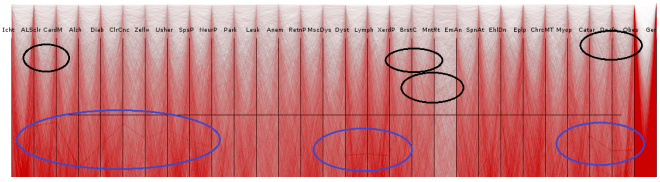
(c) *After applying genetic algorithm (Pearson's correlation version), minimum correlation*

Fig. 8: Case 2 *dataset Parallel Coordinates plots, opacity: 2*

Fig. 9, 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.



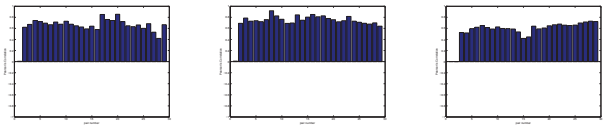
(a) *As is (before applying genetic algorithm), zoomed*



(b) *After applying genetic algorithm (Pearson's correlation version), zoomed*

Fig. 9: Case 2 *dataset parallel coordinates plots, opacity: 2, obvious patterns visible in both plots (blue ellipses), less obvious patterns only visible after genetic algorithm (black ellipses)*

To quantify the results of the genetic algorithm, an experiment is performed on the *case 3* dataset. The algorithm's fitness function is set to maximize the sum of all Pearson's correlations between adjacent dimensions (negative correlation meaning weak fitness). The algorithm is run on 100 random initial permutations, and for each of these 100 runs, the mean of Pearson's correlations between adjacent dimensions is calculated, and the same is done for the 100



(a) *As is (before genetic algorithm)* **(b)** *Using genetic algorithm favouring highly correlated pairs* **(c)** *Using genetic algorithm favouring lowly correlated pairs*

Fig. 10: *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.*

permutations resulting from the application of the genetic algorithm. This is followed by the calculation of the mean and standard deviation of these mean correlations before and after the application of the genetic algorithm. The results are as follows: a mean correlation of 0.55 with a 1.65e-2 standard deviation before and a mean correlation of 0.72 and a standard deviation of 1.12e-3 after the application of the genetic algorithm. It is clear that the mean correlation is higher after the application of the genetic algorithm while the standard deviation is divided by more than a factor of ten.

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

[1] B. Fry and C. Reas, "Processing," http://processing.org/.

[2] D. W. Dyer, "Watchmaker," http://watchmaker.uncommons.org/.

[3] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry And Its Applications*. Springer, 2009.

[4] W. Peng, M. Ward, and E. Rundensteiner, "Clutter reduction in multi-dimensional data visualization using dimension reordering," in *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, 0-0 2004, pp. 89 –96.

[5] J. Wang, W. Peng, M. Ward, and E. Rundensteiner, "Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets," in *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, oct. 2003, pp. 105 –112.

[6] M. Affenzeller and S. Winkler, *Genetic algorithms and genetic programming: modern concepts and practical applications*, ser. Numerical insights. CRC Press, 2009. [Online]. Available: http://books.google.be/books?id=EkELtZAXViEC

[7] Z. H. Ahmed, "Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator," *International Journal of Biometrics & Bioinformatics*, vol. 3, pp. 96 – 105, 2010.

[8] L.-C. Tranchevent, F. B. Capdevila, D. Nitsch, B. D. Moor, P. D. Causmaecker, and Y. Moreau, "A guide to web tools to prioritize candidate genes," *Briefings in Bioinformatics*, pp. 22–32, 2011.

[9] R. M. Piro and F. Di Cunto, "Computational approaches to disease-gene prediction: rationale, classification and successes," *FEBS Journal*, vol. 279, no. 5, pp. 678–696, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1742-4658.2012.08471.x

[10] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987, pp. 14–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=42512.42515

[11] T. De Bie, L. Tranchevent, L. M. M. van Oeffelen *et al.*, "Kernel-based data fusion for gene prioritization," *Bioinformatics (Oxford, England)*, vol. 23, no. 13, pp. i125–132, Jul. 2007, PMID: 17646288. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17646288

[12] S. Aerts, D. Lambrechts, S. Maity *et al.*, "Gene prioritization through genomic data fusion," *Nature Biotechnology*, vol. 24, no. 5, pp. 537–544, May 2006. [Online]. Available: http://www.nature.com/nbt/journal/v24/n5/full/nbt1203.html

[13] P. Glenisson, B. Coessens, S. Van Vooren *et al.*, "TXTGate: profiling gene groups with text-based information," *Genome biology*, vol. 5, no. 6, p. R43, 2004, PMID: 15186494. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15186494

[14] S. Yu, L. Tranchevent, B. De Moor *et al.*, "Gene prioritization and clustering by multi-view text mining," *BMC bioinformatics*, vol. 11, p. 28, 2010, PMID: 20074336. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20074336

[15] S. Hunter, P. Jones, A. Mitchell *et al.*, "InterPro in 2011: new developments in the family and domain prediction database," *Nucleic acids research*, vol. 40, no. Database issue, pp. D306–312, Jan. 2012, PMID: 22096229. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22096229

[16] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, Jan. 2000, PMID: 10592173. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10592173

[17] P. Flicek, M. R. Amode, D. Barrell *et al.*, "Ensembl 2012," *Nucleic acids research*, vol. 40, no. Database issue, pp. D84–90, Jan. 2012, PMID: 22086963. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22086963

[18] C. G. Son, S. Bilke, S. Davis *et al.*, "Database of mRNA gene expression profiles of multiple human organs," *Genome research*, vol. 15, no. 3, pp. 443–450, Mar. 2005, PMID: 15741514. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15741514

[19] A. I. Su, M. P. Cooke, K. A. Ching *et al.*, "Large-scale analysis of the human and mouse transcriptomes," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 7, pp. 4465–4470, Apr. 2002, PMID: 11904358. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11904358

[20] G. D. Bader, I. Donaldson, C. Wolting *et al.*, "BIND–The biomolecular interaction network database," *Nucleic acids research*, vol. 29, no. 1, pp. 242–245, Jan. 2001, PMID: 11125103. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11125103

[21] C. Stark, B. Breitkreutz, T. Reguly *et al.*, "BioGRID: a general repository for interaction datasets," *Nucleic acids research*, vol. 34, no. Database issue, pp. D535–539, Jan. 2006, PMID: 16381927. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16381927

[22] S. Aerts, P. Van Loo, G. Thijs *et al.*, "TOUCAN 2: the all-inclusive open source workbench for regulatory sequence analysis," *Nucleic acids research*, vol. 33, no. Web Server issue, pp. W393–396, Jul. 2005, PMID: 15980497. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15980497

[23] J. Ye, S. McGinnis, and T. L. Madden, "BLAST: improvements for better sequence analysis," *Nucleic acids research*, vol. 34, no. Web Server issue, pp. W6–9, Jul. 2006, PMID: 16845079. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16845079

[24] J. Gillis and P. Pavlidis, "The impact of multifunctional genes on "guilt by association" analysis," *PloS one*, vol. 6, no. 2, p. e17258, 2011, PMID: 21364756. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/21364756

[25] ——, ""Guilt by association" is the exception rather than the rule in gene networks," *PLoS computational biology*, vol. 8, no. 3, p. e1002444, Mar. 2012, PMID: 22479173. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22479173