BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 8 2005, pages 1592–1595
doi:10.1093/bioinformatics/bti169

Genetics and population analysis

Generalized Venn diagrams: a new method of visualizing complex genetic set relations

Hans A. Kestler[1,2,*,†], André Müller[2,†], Thomas M. Gress[2,‡] and Malte Buchholz[2,‡]

[1]Neuroinformatics, University of Ulm, 89069 Ulm, Germany and [2]Internal Medicine I, University Hospital Ulm, Robert-Koch-Strasse 8, 89081 Ulm, Germany

Received on August 18, 2004; revised on October 15, 2004; accepted on November 18, 2004
Advance Access publication November 30, 2004

ABSTRACT

Motivation: Microarray experiments generate vast amounts of data. The unknown or only partially known functional context of differentially expressed genes may be assessed by querying the Gene Ontology database via GOMiner. Resulting tree representations are difficult to interpret and are not suited for visualization of this type of data. Methods are needed to effectively visualize these complex set relationships.

Results: We present a visualization approach for set relationships based on Venn diagrams. The proposed extension enhances the usual notion of Venn diagrams by incorporating set size information. The cardinality of the sets and intersection sets is represented by their corresponding circle (polygon) sizes. To avoid local minima, solutions to this problem are sought by evolutionary optimization. This generalized Venn diagram approach has been implemented as an interactive Java application (VennMaster) specifically designed for use with GOMiner in the context of the Gene Ontology database.

Availability: VennMaster is platform-independent (Java 1.4.2) and has been tested on Windows (XP, 2000), Mac OS X, and Linux. Supplementary information and the software (free for non-commercial use) are available at http://www.informatik.uni-ulm.de/ni/mitarbeiter/HKestler/vennm together with a user documentation.

Contact: hans.kestler@medizin.uni-ulm.de

1 INTRODUCTION

Microarray technologies are increasingly being used in biological and medical sciences for high throughput analyses of genetic information on the genome, transcriptome and proteome levels. These types of analysis generate vast amounts of data, often in the form of large lists of genes differentially expressed between different sample sets, leaving the researcher with the task of identifying the functional relevance of the observed expression changes. Comprehensive functional annotation of gene products as provided by the Gene Ontology (GO) database (http://www.geneontology.org) is an invaluable resource for performing this task.

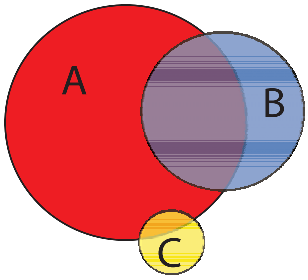

Fig. 1. A generalized Venn diagram with three sets *A*, *B* and *C* and their intersections. From this representation, the different set sizes are easily observed. Furthermore, if individual elements (genes) are contained in more than one set (functional category), the intersection sizes give a direct view on how many genes are involved in possibly related functions. During optimization, the localization of the circles is altered to satisfy the possibly contradictory constraints of circle size and intersection size.

Gene lists can be queried for associated functional categories (GO terms) which are significantly over-represented among the differentially expressed genes using query tools such as GOMiner (Zeeberg *et al*., 2003). However, due to the association of genes with multiple GO terms and the resulting complex interdependencies of categories sharing differentially expressed genes, the results of such an analysis remain hard to interpret and are not easily visualized. Standard tree representations, as e.g. provided with the GOMiner tool, are in many cases an improper choice for this task, especially for representing intersections. Venn diagrams can provide much more information to the researcher. Full containment of one set in another, partial intersections and disjunctness can be seen at a glance with Venn diagrams (Fig. 1). Simple Venn diagrams are already being used in microarray data analysis software packages such as GeneSpring® and SilicoCyte® to visualize intersections of up to three different lists of genes. In the present paper, we propose to extend the use of Venn diagrams to the faithful visualization of the results of GO queries as performed e.g. with the GOMiner tool. We present a method to represent GO terms which have been identified as significantly over-represented among the differentially expressed genes in the form of polygons with areas directly proportional to the true cardinalities of the sets (i.e. the numbers of differentially expressed genes in the categories) and intersections proportional to the numbers of genes shared by two or more categories.

[*]To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

[‡]M. Buchholz and T.M. Gress made equal contributions to this study and should both be considered senior authors.

2 METHODS

Unfortunately, this visualization problem is not easily solvable. Sometimes no perfect solution exists, as the constraints of set size (polygon or circle size) and intersection set size are in some cases contrary. As a consequence and to avoid local minima, the visualization approach was implemented using an evolutionary strategy for optimization. The goal is to find the best solution possible and mark unavoidable non-intersections in gray.

Efficient algorithms are known for finding intersecting polygons and computing their area: The intersection of two convex polygons with L and M edges can be computed efficiently within $O(L+M)$ steps (O'Rourke, 2000) and the area of a polygon with L edges can be determined in $O(L)$ steps.

2.1 Area calculation

The polygon areas are computed by applying the Gaussian integration theorem in the plane (Harris and Stocker, 1998)

$$\int_B \underbrace{\left(\frac{\partial P}{\partial x} - \frac{\partial Q}{\partial y}\right)}_{:=F} \mathrm{d}(x, y) = \oint_{\partial B} P\,\mathrm{d}y + Q\,\mathrm{d}x$$

It states that the value of an area integral of the above form (the left side integrates over a scalar field) on a closed domain $B \subset \mathbb{R}^2$ can be expressed by a curve integral along the boundary ∂B (right side). Let $(x, y)_{k=1}^{L} \in \mathbb{R}^2$ be a polygon. After some conversions the area computes to

$$A = \sum_{k=1}^{L} x_k(y_{k+1} - y_k), \quad y_{L+1} := y_1$$

2.2 Evolutionary optimization

The problem is partitioned into independently solvable subproblems. It is therefore assumed that all sets have at least one intersecting partner.

Let $A_1, \dots, A_m \subseteq U$ be a sequence of intersecting subsets of the element set U and $G_1^t, \dots, G_m^t \subseteq \mathbb{R}^2$ the corresponding polygons at optimization step $t = 1, 2, \dots$. For each k-subset $I \subseteq \{1, \dots, m\}(|I| = k \geq 2)$ we observe the cardinality c of the intersection set $\bigcap_{i \in I} A_i$ and the area of the corresponding polygonal intersection $A = \eta$ area $\left(\bigcap_{i \in I} G_i^\tau\right)$. The factor $\eta > 0$ is a predefined constant describing the correspondence between area and cardinality, so for all polygons $|A_i| = \eta$ area(G_i^t) holds (with these assumptions all costs for $|I| = 1$ are equal to 0). We define the partial cost of an observed intersection of order $k = |I| \geq 2$, cardinality c and area A:

$$f(k, c, A) = \begin{cases} \alpha A^2/(k-1) & \text{if } c = 0 \\ \beta c^2/(k-1) & \text{if } c > 0, A = 0 \\ (A - c)^2/(k-1) & \text{otherwise.} \end{cases}$$

The two parameters $\alpha, \beta \geq 0$ allow the weighting of the different cases. In the current version they are set to $\alpha = 10$ and $\beta = 20$. So the unwanted (first case) and missing overlaps (second cases) are weighted stronger than a small area deviation (last case).

The overall cost is defined as the sum over all partial costs:

$$E^t = \sum_{\substack{I \subseteq \{1, \dots, m\} \\ 2 \leq |I| \leq K}} f\left(|I|, \left|\bigcap_{i \in I} A_i\right|, \eta \text{ area}\left(\bigcap_{i \in I} G_i^t\right)\right)$$

In the optimum, all areas multiplied by η should be equivalent to the cardinality of the corresponding intersection sets. The number of considered set combinations sometimes needs to be restricted via an upper bound $2 \leq K \leq m$, which may be necessary for large m due to memory and time limitations. The problem grows exponentially with the number of sets m: A single cost function evaluation requires $O(Lm2^{m-1})$ steps for all set combinations; in the restricted case this reduces to $O\left(L \sum_{k=2}^{K} k\binom{m}{k}\right)$ steps, with L being the number of polygon edges.

The aforementioned error function is optimized over the positions of the polygon centers in the 2D plane. The shape and orientation of the polygons remain fixed (the number of edges can be chosen in advance). An evolutionary strategy (Bäck, 1996) was used to minimize E (inverse of fitness). In this strategy, only mutation was used to modify the population. Following Bäck, we used the self-adaptation of the mutation variances to achieve a better convergence and to eliminate the need for specifying mutation variances.

A generation contains N individuals (this parameter defaults to 100) each consisting of a parameter vector $\mathbf{v}_1^t, \dots, \mathbf{v}_m^t \in \mathbb{R}^2$ representing the polygon centers and a mutation vector $\sigma^t \in \mathbb{R}_+^m$ describing the mutation rate for each parameter. For the first population, all centers $\mathbf{v}_i^t$ are set to random values so that the polygons are contained within a bounding box, and the mutation parameters are uniformly drawn from an pre-specified interval $[\tau_{\text{lower}}, \tau_{\text{upper}}]$. In the mutation step the mutation parameters itself are mutated:

$$\sigma_i^{(t+1)} = \sigma_i^t e^{N(0,\tau)}, \quad i = 1, \dots, m$$

and restricted to the interval $[\tau_{\text{lower}}, \tau_{\text{upper}}](\tau$ defaults to 0.5). Then the locations of the polygons are updated as follows:

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^t + \begin{bmatrix} N(0, \sigma_i^{(t+1)}) \\ N(0, \sigma_i^{(t+1)}) \end{bmatrix}, \quad i = 1, \dots, m$$

where $N(0, \sigma)$ is a normal distributed variate with mean 0 and variance σ. The result of each partial mutation operation (mutation of a single polygon) is restricted to match the following two conditions:

(1) In the case that a polygon is not in contact with at least one other polygon, its position is reset in the direction of the nearest polygon whose corresponding set has a non-empty intersection with the observed set so that the distance of the centers will be the sum of both radii.

(2) The polygons are restricted to stay in the bounding box $[0, 1]^2$.

Evolutionary selection and offspring generation is performed by assigning each individual a rank $r = 1, \dots, N$ according to its fitness determined by the value of its cost functional (the best individual with the lowest cost has $r = 1$). Each individual is then duplicated reciprocal to its rank value. So each individual with rank r will have at most $\lceil qN/r \rceil (0 < q < 1)$ offsprings. Starting with the individual with the highest rank $r = 1$ the new population will be filled up until it has size N. The fittest individual is always included in the new population. All but the fittest individual are mutated. The displayed polygon arrangement always shows the fittest individual and will only change if there is a better solution found.

The optimization process stops when a configurable upper number of steps is exceeded or the cost functional has not changed over a certain number of steps.

3 RESULTS AND DISCUSSION

The visualization approach described was implemented in a small and easy-to-use, platform-independent Java application (VennMaster). It allows an interactive exploration of the data sets and was tested on Windows XP, Linux and Mac OS X using the Java Runtime Environment 1.4.2 (http://www.java.sun.com). VennMaster supports the interactive exploration of sets and intersection sets. When touching the polygons with the cursor, the region will be highlighted and the involved group names and the cardinality of the intersection set will be shown (Fig. 2). The edge number of the polygons is user-configurable. Furthermore, a gene list of the selected intersection set(s) is shown in an information field. Unresolved intersections (for which no corresponding polygon intersection exists) are listed in the field 'Inconsistencies'. For each set or intersection set, a text label can be attached. Labels and polygons can be moved by drag-and-drop (the cost function will be updated immediately). So the user can interactively modify the configuration and may restart the evolutionary optimization process on the changed arrangement. Set positions can be locked so that they will not be moved by the optimizer. The optimization process can be controlled via a parameter

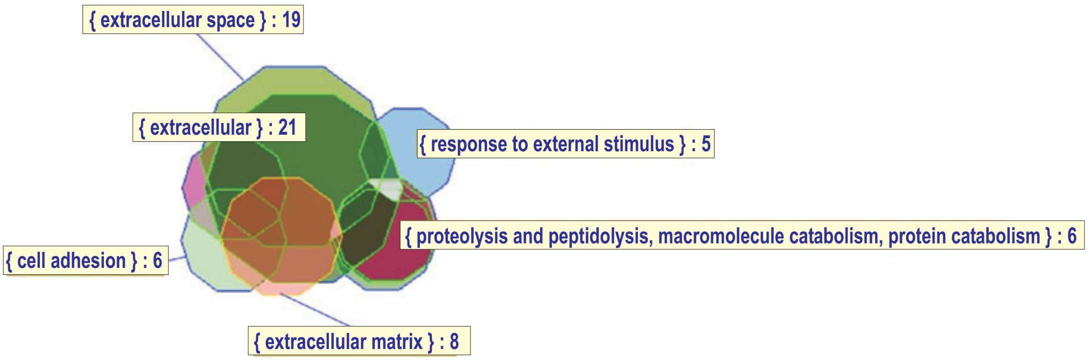
{ extracellular space } : 19
{ extracellular } : 21
{ response to external stimulus } : 5
{ proteolysis and peptidolysis, macromolecule catabolism, protein catabolism } : 6
{ cell adhesion } : 6
{ extracellular matrix } : 8

Fig. 2. Result of a visualization (with polygons) by importing files from GOMiner. Here, differentially expressed gene sets between a specialized mesenchymal cell type (stellate cells) and normal skin fibroblasts (minimum total: 100; max p-value: 0.05) are shown. Although a total of nine GO categories were reported to be significantly over-represented among the changed genes, the analysis revealed that these categories strongly overlap and form a single large cluster of cell surface/extracellular matrix related categories.

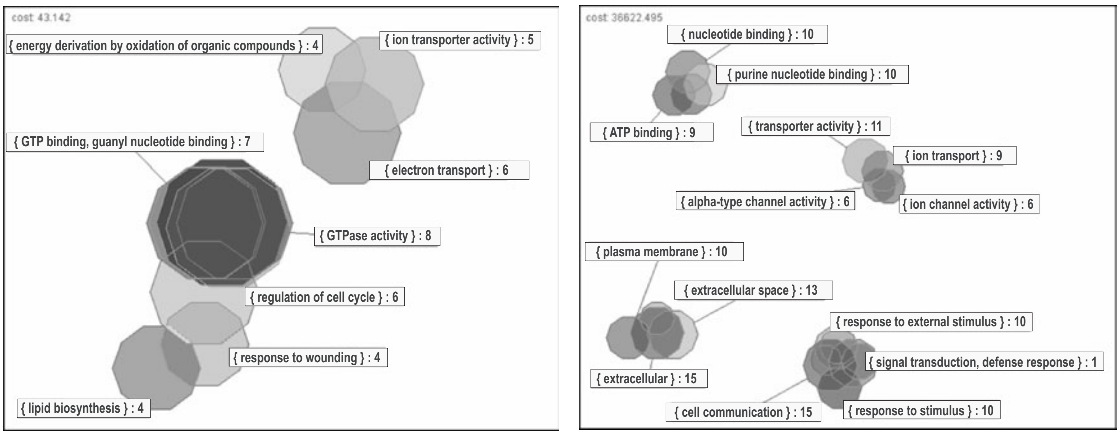
cost 43.142
{ energy derivation by oxidation of organic compounds } : 4
{ ion transporter activity } : 5
{ GTP binding, guanyl nucleotide binding } : 7
{ electron transport } : 6
{ GTPase activity } : 8
{ regulation of cell cycle } : 6
{ response to wounding } : 4
{ lipid biosynthesis } : 4
cost 36622.495
{ nucleotide binding } : 10
{ purine nucleotide binding } : 10
{ ATP binding } : 9
{ transporter activity } : 11
{ ion transport } : 9
{ alpha-type channel activity } : 6
{ ion channel activity } : 6
{ plasma membrane } : 10
{ extracellular space } : 13
{ response to external stimulus } : 10
{ signal transduction, defense response } : 1
{ extracellular } : 15
{ cell communication } : 15
{ response to stimulus } : 10

Fig. 3. Visualization showing different subprocesses (two for left panel and four for right panel): Genes over-expressed (left) or under-expressed (right) in pancreatic ductal carcinoma as compared to normal pancreatic duct cells (minimum total: 50; max p-value: 0.05): the analysis identifies distinct clusters of biologically relevant GO categories over-represented among the over-expressed and under-expressed genes, respectively.

dialog (see Supplementary information). The Venn diagrams may be saved as JPEG files. To analyze functional categories of differentially expressed genes, we included the ability to import files from GOMiner (Zeeberg *et al*., 2003) in the program, in addition to a simple tab-delimited file format with an element/group pair in each line. For the GOMiner files, a pre-filtering of the genes/categories is included.

To validate our approach, an experiment using microarrays with 23 000 features was processed by VennMaster, the tool that supports the approach presented here. Differences between two types of cells (stellate cells and fibroblasts) involved in diseases associated with extensive fibrosis such as chronic pancreatitis or liver fibrosis were investigated. The GOMiner tool was used to classify the differential genes into functional categories. Due to the fact that one gene may belong to multiple functional categories, this analysis revealed a complex pattern of 29 GO terms. To identify the major functional categories differentiating the two cell types, the VennMaster program was applied to the GOMiner list of 74 genes. The list of differentially expressed genes was compared to the list of all genes exceeding a minimal expression threshold (normalized expression value >0.5 in at least one of the sample sets) to identify GO terms significantly over-represented among the differentially expressed genes. Since the tree format provided by GOMiner to visualize the results of the analysis is not suited to display the overlap of genes in different GO categories resulting from the association of genes with multiple GO terms, we applied the described Venn diagram approach which facilitates visualization of associations between GO categories based on the evaluation of genes mutually represented in different categories (Fig. 2). The Venn diagram representation revealed that the principal GO terms which were significantly over-represented among this set of genes all fell within a single cluster of interconnected categories relating to extracellular and cell surface genes, such as extracellular matrix genes, secreted proteins and cell adhesion genes and their associated functions. In a second experiment, expression profiles of pancreatic ductal carcinoma and normal pancreatic duct cells were compared (Fig. 3). Both analyses gave an instant overview of the involved GO terms.

To evaluate the performance of the self-adapting evolution strategy, a series of simulations were made with and without self-adaptation of the mutation variances. For non-adaptive mutation rates, the mutation parameter is critical for the convergence of the optimization process (see Supplementary Figures I and II). The self-adapting procedure gave superior results in all but the lowest mutation variances. For 17 different data sets, the overall maximal fitness gained was evaluated on 100 simulations for each configuration, i.e. a total of 3400. For every data set, the self-adapting algorithm resulted in smaller error values than the non-adaptive algorithm. Each of the *p*-values of the one-sided Wilcoxon rank sum test was below 6.28e-5 (Bonferroni correction requires a *p*-value <0.0015 to be considered significant; see Supplementary Figure V).

Clearly, the diagrams we used for visualization purposes are not true Venn diagrams according to Grünbaum (1992) and others (see http://www.combinatorics.org/Surveys/ds5/VennEJC.html for an excellent survey) as they allow empty and not connected intersection sets. Apart from this more theoretical aspect, the proposed generalized Venn diagrams proved nevertheless to be of great value for practical purposes requiring the visualization of complex set relationships.

ACKNOWLEDGEMENT

This research was funded in part by a 'Forschungsdozent' grant through the Stifterverband für die Deutsche Wissenschaft to H.A.K.

REFERENCES

Bäck,T. (1996) *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, Oxford.

Grünbaum,B. (1992) Venn diagrams I. *Geombinatorics*, **1**, 5–12.

Harris,J.W. and Stocker,H. (1998) *Handbook of Mathematics and Computational Science*. Springer Verlag, New York.

O'Rourke,J. (2000) *Computational Geometry in C*, 2nd edn. Cambridge University Press, Cambridge.

Zeeberg,B.R., Feng,W., Wang,G., Wang,M.D., Fojo,A.T., Sunshine,M., Narasimhan,S., Kane,D.W., Reinhold,W.C., Lababidi,S. *et al*. (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol.*, **4**, R28.