Methodology to Explore Co-expression in Microarray Data

Bertrand De Meulder, Eric Bareke, Michael Pierre, Sophie Depiereux, Eric Depiereux
Bioinformatics & Biostatistics lab, URBM
University of Namur
Namur, Belgium
bertrand.demeulder@fundp.ac.be

Abstract - In the past several years, the amount of microarray data accessible on the Internet has grown dramatically, representing millions of Euros worth of underused information. We propose a method to use this data in a coexpression study. The method is simple in principle: the aim is to detect which genes react in the same way in certain circumstances (such as a disease, stress, medication,), potentially highlighting new interaction partners or even new pathways. We propose to study coexpression using a large amount of data, process it through an adequate algorithm and visualize the results with a dynamic graphical representation. We gather the microarray data using the PathEx database developed in our lab, which allows searching through more than 120,000 microarrays experiments on *Homo sapiens* using specific criteria such as the tissue sample, the biological background or any information contained in the metadata describing the experiment. Then, we process the data using the Minet R package, which allows for coexpression analysis using cutting-edge algorithms such as ARACNE or MRNET methods. This step computes the weighted relations between all the probesets in the microarrays and provides a GraphML representation of the relations. In order to explore the relations optimally, we channel the GraphML into a dynamic graphical program we developed called gViz. This program allows for data visualization but also for exploration and post-analysis. We can extract meaningful information from the network computed, compare this information with curated databases such as KEGG (Kyoto Encyclopedia of Genes and Genomes), highlight the discrepancies and hopefully discover new interactions or add new steps in canonic pathways. We present here a fast, free and user-friendly working methodology to analyze co-expression in microarray data

Keywords: Co-expression; microarray; methodology

I. INTRODUCTION

Co-expression analysis in gene networks in an infant domain in the teenager field of network biology. It is only a few years ago that the technology, knowledge, and data mass mandatory for these analysis has been made available to researchers. Co-expression analysis is the study of the similarities in changes of genes expressions in various circumstances (such as diseases). Using this technique, researchers aim to discover new relationships between known genes, new partners in known pathways or even entirely new pathways. This holds promises for new insights into complex biological states, such as cancer or degenerative diseases, as well as further comprehension of the cell fine machinery. Ultimately, this approach could provide a compendium of gene interactions maps in an ever-growing array of cell states.

Currently, there lies in databases millions of Euros worth of underused microarray data, since researchers conducting those experiments usually focus only on a small number of genes among the thousands available on the chip. We propose a way to make use of this data mass, by studying co-expression relations between all genes represented on the chip, using state of the art processing algorithms and an adapted graphical interface for exploration. With this we hope to predict new interactions, which we will then try to confirm using wet-lab analysis.

Our approach implies the possession of several elements: a large deposit of microarray data, if possible in database form; an algorithm to process this data efficiently; a graphical interface to browse the map of interactions and an external verification database for the validations.

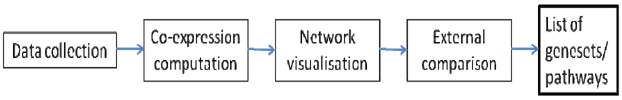

Figure 1. General layout of the co-expression analysis

Each step of this general layout will be developed hereunder. We first discuss about data collection then present the different steps of the co-expression computation process. Section IV is about the visualization solutions, while Sections V and VI present the validations and conclusions respectively.

II. DATA COLLECTION

The main microarray deposits are the well known Gene Expression Omnibus (GEO) [1] and ArrayExpress [2] databases. The data collection, although laborious and time-consuming, could be done directly from these websites. However the construction of the websites does not allow for advanced querying on the biological parameters of the experiment. This aspect is crucial: indeed, the relations we potentially will highlight in the end of the analysis are related to the biological state of the cells on which the microarray experiment was done. In other words, all the information we will extract from the analysis has to be already buried in the microarray experiment we select at this step.

To help the process of selection, we developed a database with the data included in both GEO and ArrayExpress and we included all the information in the description field manually into a database, thus making high-level queries possible. This database – PathEx [3] – allows us to query all the microarray experiments on criterions such as disease, cell lines, age of the patient, organ studied, etc., thus allowing for a much more precise choosing of data.

III. Co-expression computation

During this part of the methodology, gene expression data from the microarrays selected in the previous step are processed to generate a representation of the co-expressions between those genes. This processing can be done by several algorithms such as the R packages nem, qpgraph or GeneNet [4 , 5 , 6]. We chose to use the R package MINET to process our data based on its speed, ease of use, large choice of methods at each steps and possibility of parameterization. This package was developed by Patrick Meyer in the Machine Learning Group at the University of Bruxelles (ULB) [7]. MINET computes the weighted relations between every probeset in the microarray; it takes as input the preprocessed microarray data and outputs a GraphML representation of the interactions.

The Mutual Information networks are a subcategory of inference methods. In those networks, a link is set if it exhibits a high score based on pairwise mutual information. One of the advantages of these methods is the low computational complexity. This is due to the fact that $n(n-1)/2$ calls of mutual information, based on bivariate probability distributions, are required to compute the mutual information matrix. Since each estimation of a bivariate distribution can be done quickly and does not require a large number of samples, this method is ideal to analyze microarray data [8].

The MINET package consists of three successive steps: Discretization, Mutual Information computation and Network Inference.

A. Discretization

This step is mandatory for the next computing step. However, it is known that there is an inevitable loss of information when discretizing continuous data. To minimize this loss, two discretization algorithms are implemented: *equalWidth* and *equalFreq*. The principle of the first is to divide the interval [a, b] into [X_i] intervals of same size, while the principle of the latter is to divide [a, b] into [X_i] intervals, each having the same number of data points. The number of bins is also important, as it controls the ratio between the bias and the variance. Practically, if m is the number samples, it is considered that a number of bins equal to the square root of m is a fair trade-off between bias and variance [9].

B. Mutual Information Matrix (MIM) computation

This step consists in the computing, between all pairs of genes present in the dataset, of the mutual information, i.e. the similarity in the behaviors of the genes. This step is very tricky, often biased but is crucial for the good results of the whole procedure. It is not surprising that there exist many alternative algorithms for this step. Without going into too many details, here is an overview of the MIM computation algorithms available in the MINET package:

1) General formulation

The MIM computation requires the computation of a square matrix whose m_{ij} element is given by

$$mim_{ij} = I(X_i; X_j) \qquad (1)$$

where $I\ (X_i;\ X_j)$ is the Mutual Information between variable X_i and X_j.

The difference between the methods lies in the computing of this term $I(X_i;\ X_j)$ Mutual Information computation requires the determination of three entropy terms:

$$I(X_i; X_j) = H(X_i) + H(X_j) - H(X_i; X_j) \qquad (2)$$

where $H\ (X)$ is the entropy of the variable X.

Entropy has to be estimated and an effective and fast entropy estimator is essential. The reduction of the bias inherent to the entropy estimation has gained much interest over the last years and most approaches have focus on minimizing this bias. However, in the case of microarray analysis, the reduction of the bias should not be the only criterion, as computational complexity/speed should also be minimized. To save space, we only discuss the *Shrink* and the *Schurmann-Grassberger* estimators.

2) Shrink Estimator

The rationale behind this algorithm is to combine two different estimators: one with low bias and one with low variance, by use of a weighting factor *λ [0, 1]*. The general formulation is the following [7]:

$$\hat{p}_\lambda(x) = \lambda \frac{1}{|\chi|} + (1-\lambda)\frac{\#(x)}{m} \qquad (3)$$

where λ is the weighting factor [0, 1], |χ| is the number of non null bins, #(x) is the number of data points having the value x and m is the number of samples.

The entropy can then be estimated with:

$$\hat{H}^{shrink}(X) = -\sum_{x\in\chi} \hat{p}_{\lambda^*}(x)\log \hat{p}_{\lambda^*}(x) \qquad (4)$$

where H is the entropy and λ^* is the value of λ minimizing the mean square error [10] [8].

3) *The Schurmann-Grassberger estimator*

It is a Bayesian estimator which assumes the sample distribution follow a Dirichlet distribution. A Dirichlet distribution is the generalization of the Beta distribution [11]. The density of this distribution is described by [8]

$$p(X;\Theta) = \frac{\prod_{i\in\{1,2,\ldots|\chi|\}}\Gamma(\Theta_i)}{\Gamma(\sum_{i\in\{1,2,\ldots|\chi|\}}\Theta_i)} \prod_{i\in\{1,2,\ldots|\chi|\}} x_i^{\Theta_i - 1} \qquad (5)$$

where θ_i is the prior probability of an event x_i, x_i being the ith element of the set χ and Γ (.) is the gamma function.

The entropy can then be estimated by

$$\hat{H}^{dir}(X) = \frac{1}{m + |\chi|N} \sum_{x\in\chi} (\#(x) + N) \\ (\psi(m + |\chi|N + 1) - \psi(\#(x) + N + 1)) \qquad (6)$$

where $\psi(z) = \frac{d \ln \Gamma(z)}{dz}$ is the digamma function, N is a weighting factor. Various choices of parameters for this factor N have been proposed [12 , 13].

C. *Network inference*

Once the MIM computation is done, the network inference step can take place. This step is essentially a translation of the Mutual Information links computed at the previous step into a graph of the probable relations between the variables. In the case of microarray data the nodes in the graph represent probesets and the arcs represent the regulator/regulated relations between them. Various network inference methods are available in the MINET package: *Relevance network*, *CLR*, *ARACNE* and *MRNET* algorithms [14-16]. We will only discuss the *MRNET* method, to save space.

MRNET is based on the Maximum Relevance / Minimum Redundancy (MRMR) rationale [7]. Simply put, if we consider a set of genes (*X*) and a target gene *Y*, the algorithm will first select the gene X_i with the highest mutual score to variable *Y*. Then, the next selected, X_j, will be the one with a high $I(X_j; Y)$ score (maximum relevance), and at the same time a low $I(X_i; X_j)$ (minimum redundancy). At each step, the algorithm is thus expected to select the variables with and efficient trade-off between relevance and redundancy, for every gene X_i in the set of X genes. A selection based on a score above a certain threshold $I(X_i; X_j) < \theta$ is performed in both directions: for two genes X_i and X_j, there will be an edge if X_i is a well predictor of X_j ($s_i > \theta$) or if X_j is a well predictor of X_i ($s_j > \theta$). The complexity of this methods lies between O(n^2) and O(n^3).

D. *Methods selection*

Following MINET author's recommendation based on validations on external datasets, we use the following methods to analyze our data: *equalFreq* discretization, the *Shrink* estimator if the number of replicates is low and the *Schurmann-Grassberger* estimator otherwise and the *MRNET* algorithm [7]. Our figure 1 then becomes:

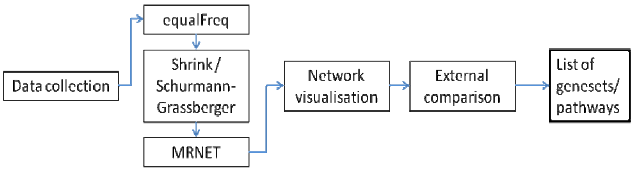

Figure 2. Layout of the analysis with choice of methods in MINET

IV. NETWORK VISUALIZATION

The easiest way to explore the results found at the previous step is to use a graphical representation of the network. Although there are several softwares available for the task [17-19], we were not satisfied either with the functionalities proposed or with the interactions possibilities of said softwares. We decided to develop our visualization software, which we called gViz. An application note describing it has been submitted to Bioinformatics on January 22nd [20].

gViz takes as input the network computed by MINET in GraphML format. It can then translate the probeset IDs into a wide range of mainstream identifiers: Entrez gene, Ensembl, UniGene or KEGG IDs. The advantage of gViz over the other network softwares is that it can be used to visualize specific parts of the network. The user can select one of several identifiers in the left panel (see fig 3) and display the network containing only the relations it wants to focus on. The network displayed in gViz is dynamic and interactive. The user can choose to display the whole network (although it can be resources consuming) or specific parts of it. When clicking on a node, the user can highlight as well the neighbors of the node, with an adjustable deepness. gViz also has a feature capable of filtering the entire network based on the MRMR score given by MINET or by nodes degree (i.e. the number of neighbors) or also based on annotation criteria (involved in same biological process). The user can at any time adjust the value of the exclusion threshold.

Several layout algorithms are available. Other 'visual' features allow to display the thickness of the nodes (representing the degree of said node) and of the edges (representing the MINET score for said edge).

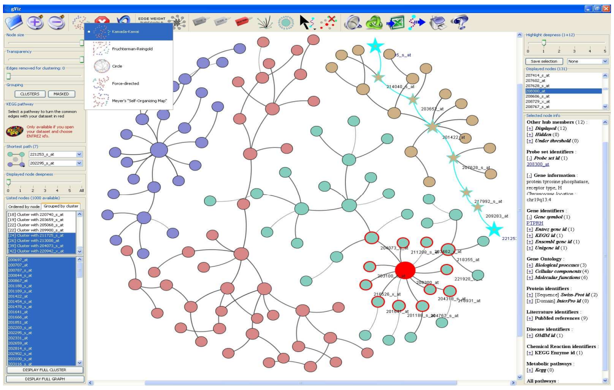

Figure 3. General layout of the gViz interface

V. Validations

This part is still in progress. However, we have a clear picture of the three validations that should be done. First we will perform a consistency validation, to ensure that our method can retrieve known interactions, and a coverage validation to estimates the method's ability to retrieve all data present in the original dataset. Once those steps are done, we will be able to make predictions by comparing our results with an external database, such as KEGG [21], highlight discrepancies and test the corresponding genes in wet-lab analysis, therefore biologically validating our approach. These experiments will be performed in the course of 2011.

VI. Conclusion

We presented a method to analyze co-expression in microarray data, with the help of a large data mass, cutting-edge algorithms and suitable visualization solution. In a short time we will finish the first validations. We hope to spot new interactions which we will explore further in wet-lab analysis. We will then have produced a reliable, fast, cheap and user-friendly way for researchers to analyze their microarray data prior to the wet-lab. In the near future, we will apply this methodology to try and discover new genes or interactions involved in the metastatic transformation of primary cancer cell and eventually provide new targets for cancer treatments.

Acknowledgement

B. D.M. thanks Patrick Meyer and Raphaël Helaers for help and technical ideas. B. D.M. benefits from a FRS-FNRS Télévie Grant n° FC 81726.

References

1. T. Barrett et al., *NCBI GEO: mining tens of millions of expression profiles--database and tools update.* Nucleic Acids Res, 2007. **35**(Database issue): pp. D760-765.
2. H. Parkinson et al., *ArrayExpress--a public database of microarray experiments and gene expression profiles.* Nucleic Acids Res, 2007. **35**(Database issue): pp. D747-750.
3. E. Bareke et al., *PathEx: a novel multi factors based datasets selector web tool.* BMC Bioinformatics, 2010. **11**: pp. 528-537.
4. H. Frohlich, M. Fellmann, H. Sultmann, A. Poustka, and T. Beissbarth, *Estimating large-scale signaling networks through nested effect models with intervention effects from microarray data.* Bioinformatics, 2008. **24**(22): pp. 2650-2656.
5. R. Castelo and A. Roverato, *Reverse engineering molecular regulatory networks from microarray data with qp-graphs.* J Comput Biol, 2009. **16**(2): pp. 213-227.
6. R. Opgen-Rhein and K. Strimmer, *From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data.* BMC Syst Biol, 2007. **1**: p. 37. Last access date: 22 March 2011.
7. P.E. Meyer, F. Lafitte, and G. Bontempi, *minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information.* BMC Bioinformatics, 2008. **9**: p. 461. Last access date: 22 March 2011.
8. P.E. Meyer, *PhD Thesis.* 2008. Available from http://www.ulb.ac.be/di/map/pmeyer. Last access date: 22 March 2011.
9. Y. Yang and G. Webb, *Discretization for naive-bayes learning: managing discretization bias and variance*, in *Technical report*, S.o.C.S.a.S. Engineering, Editor. 2003, Monash University.
10. J. Hausser, *Improving entropy estimation and inferring genetic regulatory networks.* , in *National Institute of Applied Sciences*. 2006: Lyon. Available from http://jean.hausser.org/site/64. Last access date: 22 March 2011.
11. T. Schurmann and P. Grassberger, *Entropy estimation of symbol sequences.* Chaos, 1996. **6**(3): pp. 414-427.
12. N. Beerenwinkel et al., *Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype.* Proc Natl Acad Sci U S A, 2002. **99**(12): pp. 8271-8276.
13. L. Wu, P. Neskovic, E. Reyes, E. Festa, and W. Heindel, *Classifying nback eeg data using entropy and mutual information features.* in *European symposium on Artificial Neural Networks*. 2007.
14. A.J. Butte and I.S. Kohane, *Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements.* Pac Symp Biocomput, 2000: pp. 418-429.
15. J.J. Faith et al., *Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles.* PLoS Biol, 2007. **5**(1): p. e8. Last access date: 22 March 2011.
16. A.A Margolin et al., *ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.* BMC Bioinformatics, 2006. **7 Suppl 1**: p. S7. Last access date: 22 March 2011.

17. *yEd- Graph Editor*. 2010 Available from: http://www.yworks.com/en/products_yed_about.html. Last access date: 22 March 2011.
18. N. Salomonis et al., *GenMAPP 2: new features and resources for pathway analysis.* BMC Bioinformatics, 2007. **8**: p. 217. Last access date: 22 March 2011.
19. P. Shannon et al., *Cytoscape: a software environment for integrated models of biomolecular interaction networks.* Genome Res, 2003. **13**(11): pp. 2498-2504.
20. R. Helaers et al., *gViz - A novel co-expression network visualization tool*., unpublished.
21. H. Ogata, S. Goto, K. Sato, W. Fujibichi, H. Bono, and M. Kanehisa, *KEGG: Kyoto Encyclopedia of Genes and Genomes*. Nucleic Acids Res, 1999. **27**(1): pp. 29-34.