# Integrating Gene Regulatory Networks to identify cancer-specific genes

**Valeria Bo[1] and Allan Tucker[1]**

**[1]Department of Computer Science, Brunel University, London, UK**

**Abstract**

*Consensus approaches have been widely used to identify Gene Regulatory Networks (GRNs) that are common to multiple studies. However, in this research we develop an application that semi-automatically identifies key mechanisms that are **specific** to a particular set of conditions. We analyse four different types of cancer to identify gene pathways **unique** to each of them. To support the results reliability we calculate the prediction accuracy of each gene for the specified conditions and compare to predictions on other conditions. The most predictive are validated using the GeneCards encyclopaedia[1] coupled with a statistical test for validating clusters. Finally, we implement an interface that allows the user to identify unique subnetworks of any selected combination of studies using AND & NOT logic operators. Results show that unique genes and sub-networks can be reliably identified and that they reflect key mechanisms that are fundamental to the cancer types under study.*

**Introduction**

When an organism is subjected to a different condition either internal or external to it (environmental changes, stress, cancer, etc.) its underlying mechanisms undergo some changes. To build robust and reliable Gene Regulatory Networks (GRNs) from microarrays, it is necessary to integrate multiple data collected from different studies[2,3,4,5]. To identify links in common among a set of independent studies, researchers apply consensus networks analysis. Swift et al.[6] apply a clustering technique coupled with a statistically based gene functional analysis for the identification of novel genes. While Segal et al.[7] group genes that perform similar functions into *'modules'* and then build networks of these modules to identify mechanisms at a more general (higher) level. More recently, a similar approach[8] was applied to a large number of cancer datasets where case and control are compared. For each dataset, the pairwise correlation of gene expression profile is computed and a frequency table is built. Then the values in the table are used to build a weighted gene co-expression frequency network. After this they identify sub-networks with similar members and iteratively merge them together to generate the final network for both cancer and healthy tissue.
In[9], we expand on this work but rather than focusing on consensus networks, we develop a method to 'home in' on both the similarities and *differences* of GRNs generated from different independent studies by using a combination of partial correlation network building and graph theory. The method goes beyond the simple pairwise correlations between genes, as in[8], by building independent networks for each study using *glasso* which identifies the inverse covariance matrix using the lasso penalty. Rather than identifying consensus studies, we detect the edges that are $unique/specific$ for each study and build Bayesian Networks to identify the most predictive group of genes and further refine our networks.
In this work we extend the work presented in[9] by exploring the performances of the pipeline using four different cancer datasets and identifying, through the GeneCards encyclopaedia[1], the list of genes known to be involved in each type of cancer. We apply a statistical test to measure the significance of detecting these genes in our unique networks. In addition, we develop an interface that allows the user to select combinations of studies using AND and NOT logic operators and to identify the related unique sub-networks and genes.

## Materials and Methods

In this paper we adapt the Unique Network Identification Pipeline (UNIP) developed in[9]. Each step of the pipeline applied for the specific case of this paper are explained in the following sections.

**Dataset Description.** Four different cancer datasets are downloaded from the NCBI Gene Expression Omnibus (GEO) website[10]. To avoid platform bias the datasets selected are all generated using Affymetrix HU133 Plus 2.0 Genechip. Given the raw series of data, the *rma* (Robust Multi-Array Average) expression measure (available in the R package 'affy'[11]) is applied as a pre-processing step. Each study identification code, description and samples number are summarized in Table 1.

**i) Selection of Informative Genes.** The high discrepancy between the number of genes (order of thousands) and the samples (tens or hundreds) measured simultaneously in microarray data leads to the necessity of reducing the number of variables (genes) involved in the analysis. R statistics provides the 'pvac' package[12] which applies the PCA (Principal Component Analysis)[13] and returns a subset of the original variables: the closest to the principal components identified. To further refine the variable reduction and to select the most active genes, the standard deviation of each gene across all the samples in each separate study is calculated and only genes with $sd \geq 1.5$ in at least one of the 4 studies are selected. The reduced datasets are used as input to the following steps of the analysis.

**ii) Glasso.** At this stage we need to build a GRN for each condition/study in the dataset.
As we want to identify networks that go beyond simple pairwise relationships, our procedure uses *glasso*[14,15,16], which calculates the inverse covariance matrix using the lasso penalty to make it as sparse as possible.
In this paper, we apply *glasso* with the penalization parameter $\rho = 0.05$, to build a GRN for each study dataset. In addition, to further improve the sparsity and reduce the nodes involved, we maintain only the connections with an inverse covariance value greater or equal to 0.8.

**iii) Unique Bayesian Networks and Prediction.** In this paper we are exploring four different studies, each of which we want to explore the unique mechanisms, we consider each of the four studies as a study-cluster of one element and the related glasso-network (built earlier) as the consensus network for that study-cluster. Although consensus approaches are popular, here we are interested in exploring the study-specific mechanisms through that we call *unique-networks*. Given a generic graph $G = (V, E)$. We have $m$ fixed graphs $G_i$ such that $G_i = (V, E_i)$, where $V = 1, ..., n$ is the set of vertices(nodes) of the graph and $E_i = \{e_i\} = \{(u_{i1}, v_{i1}), ..., (u_{ik_i}, v_{ik_i})\}$, $k_i = \left|E_i\right|$ and $k_i \leq n(n-1)/2$. We define the unique function as $\Phi : G \mapsto G$, where, given $\hat{E}_i = \bigcup_{j=1, j\neq i}^{m} E_j$.

**Definition 1:** We define a function $\Phi(G_i)$ such that $\Phi(G_i) : (V, \{e_i : e_j \in E_i \text{ and } e_j \notin \hat{E}_i\})$
In other words, a *unique-network* contains only those edges present in no other condition-specific network. We choose to measure the reliability of the unique-networks through prediction using Bayesian Networks (BNs)[17,18] which naturally perform this using inference, given the graphical structure obtained using the genes involved in the unique-networks provided by *glasso*. Given the unique edges in the *glasso*-derived networks we first build one BN for each of the study-clusters using the R package *bnlearn*[19,20] and then identify the most predictive (how well it predicts other expression level values) and predictable (how well its expression level values are predicted) genes within (intra) and outside (inter) the study using the package *gRain*[21] and the leave one out cross validation technique. Given the $m$ samples and $n$ genes within each study we use $m$-1 samples as a training set and the remaining one as test set. Then, given the $n$-1 genes, we predict the expression value of the one left out. We compare the predicted with the real value, return 1 if they correspond and zero otherwise. We do this within all the studies and for all possible combinations of training and test sets of studies and genes. Finally, we average the amount of correctly-predicted values among the total predictions to obtain the correct-prediction for each gene. The idea is that genes that are predictive or predicted better within the selected study than on other studies are more likely to be relevant to the unique-network.

**iv) Gene cards.** As we detect study-specific sub-networks we also want to verify that our method captures study-specific genes. We query GeneCards encyclopaedia[1] to obtain the list of genes that are known to be involved in each cancer. We compare the list for each study to the others and select the genes that appear *only in the study under consideration*. To compare the unique-gene list for each type of cancer with the genes found in the corresponding unique-network, we apply a probability score developed in[6] used to test the significance of observing multiple genes with known function in a given cluster against the null hypothesis of this happening by chance. This score is based on the hypothesis that, if a given cluster, $i$ of

size $s_i$, contains $x$ genes from a defined functional group of size $k_j$, then the chance of this occurring by chance follows a binomial distribution and is defined by: $\Pr(\text{Observing} x \text{from group} j) = \binom{k_j}{x} p^x q^{k_j - x}$ where $p = \frac{s_j}{n}, q = 1 - p$ and $n$ is the number of genes in the dataset. As in this paper, when $k_j$ and $x$ are very large Pr cannot be evaluated. Therefore we use the normal approximation of the binomial distribution where: $z = \frac{x-\mu}{\sigma}$, $\mu = k_j p$ and $\sigma = \sqrt{k_j pq}$. Values of $z$ above zero mean that the probability of observing $x$ elements from functional group $j$ in cluster $i$ by chance is very small (values of $z \geq 2.326$ correspond to a probability less than 1%). The test performed is the one tailed test.

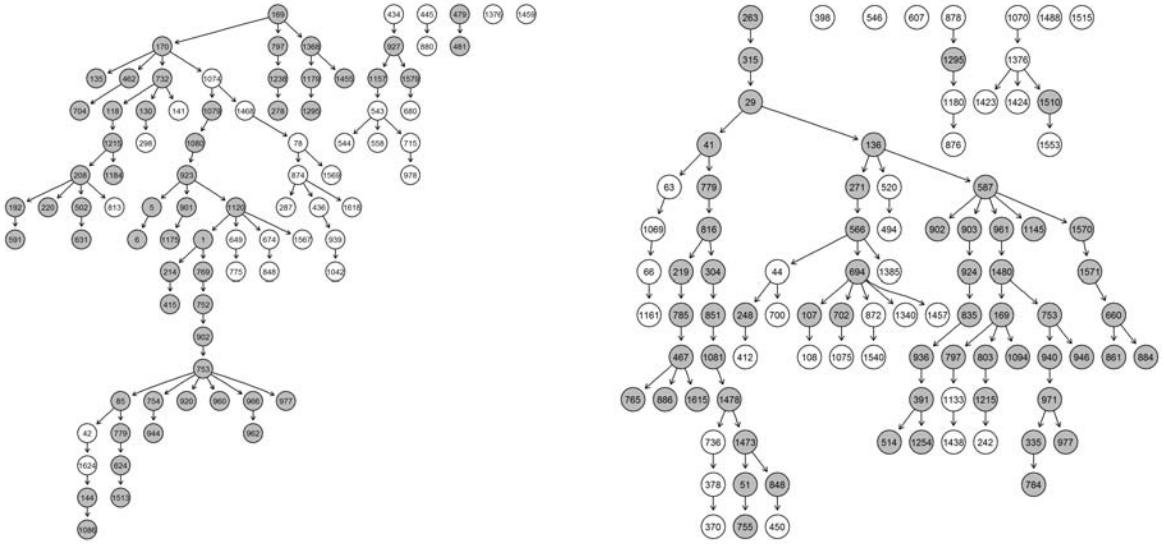
**v) Logic and GUI.** Finally a user interface has been developed using the R package *shiny* [22]. This interface allows the user to input the networks obtained with glasso and let the user choose which combination of unique networks to identify, using the logic operators AND and NOT. For example setting **1 AND 2 - NOT 3** will identify the sub-networks that study 1 and 2 have in common but do not appear in study 3. The unique sub-networks for that rule/pattern are identified and plotted on the interface together with the list of genes involved. Finally, the user has the possibility to save the network in a tiff file and the list of genes involved in csv format.

## Results

In this study four cancer datasets are explored: breast, ovarian, medullary breast (a subtype of breast cancer) and lung, in human patients. Each dataset contains a different number of samples (see Table 1). The variable selection approach reduces the number of variables/genes to analyse from 54675 to 1629. Variable reduction is followed by the implementation of *glasso* with the parameter $\rho = 0.05$. Given the glasso networks for each study we consider only the edges that are present in the network under consideration but not in the others. Once the unique-edges are detected, the genes involved are used to build a BN for each study called unique-networks (U-Ns). An example of these networks is shown in Figure 1. The structure of the glasso U-Ns differ from the structure of the Bayesian U-Ns. In the Figures 1a and 1b the nodes with a grey background indicate genes with a predicted accuracy for the gene greater than 0.6 (based on our findings in [9]). Because of the study description in Table 1, we would expect breast cancer to be very similar (involving almost the same genes) to medullary breast cancer and slightly less similar to ovarian, but very different from lung cancer. This implies that the average internal prediction for each study will not differ much from the external prediction. The internal vs external prediction for each study shown in Figure 2 reveals, as expected a very clear difference only in Network 3 and 4, medullary-breast and lung cancer respectively, with a small difference in 1 and 3. This deduction is supported by the p-values obtained from the applied t-test as shown in Table 1. We now evaluate the significance of detecting the identified unique-genes by calculating the probability score using the normal approximation. For this paper $s_i$ is the size of each unique network, $k_j$ the number of genes in the unique gene-list obtained for each cancer type comparing the geneCards gene lists, $x$ the number of genes that are present on both the unique network and the corresponding unique gene-list and $n$ is the number of genes in the original unprocessed dataset. The results in Table 2 show the $z$-score and the corresponding $p$-value indicating that the probability of observing $x$ elements from functional group $j$ in cluster $i$ by chance is in all four cases very small. This implies that the unique genes identified by our pipeline are highly significant in all studies.
Finally, Figure 3 shows the Logic Application interface. The example allows the user to visualize the unique sub-networks and the list of related genes that study 1 AND 4 have in common but do not appear in study 2.

Table 1: Cancer datasets description and t-test p-value

| Study ID | Study title | Samples | t-test p-value |
|---|---|---|---|
| GSE18864 | Triple Negative Breast Cancer | 84 | 0.55 |
| GSE9891 | Ovarian Tumour | 285 | 0.00 |
| GSE21653 | Medullary Breast Cancer | 266 | 0.02 |
| GSE10445 | Adenocarcinoma and large cell Lung Carcinoma | 72 | 0.00 |



(a) Bayesian U-N for medullary-breast cancer.

(b) Bayesian U-N for lung cancer.

Figure 1: Nodes with grey background indicate a prediction accuracy for the nodes greater than 0.6. Isolated nodes do not have connections due to the structure differences between glasso U-Ns and Bayesian U-Ns. Nodes are labelled with numbers (directly corresponding to the gene ID) for visualization purposes.

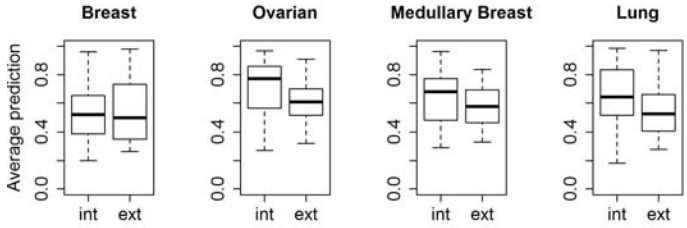



Figure 2: Internal vs External prediction accuracy for each study averaged among all genes involved in the related unique-network.

Table 2: Parameters values, z-score and p-value for each study.

| Parameters values for each study | | | | | | |
|---|---|---|---|---|---|---|
| Study ID | $s_i$ | $k_j$ | $x$ | $n$ | $z$-score | p-value |
| GSE18864 | 117 | 2982 | 11 | 54675 | 1.83 | $\leq 3.4\%$ |
| GSE9891 | 61 | 692 | 4 | 54675 | 3.68 | $\leq 1\%$ |
| GSE21653 | 89 | 0 | 0 | 54675 | NaN | $\leq 1\%$ |
| GSE10445 | 80 | 240 | 3 | 54675 | 4.47 | $\leq 1\%$ |

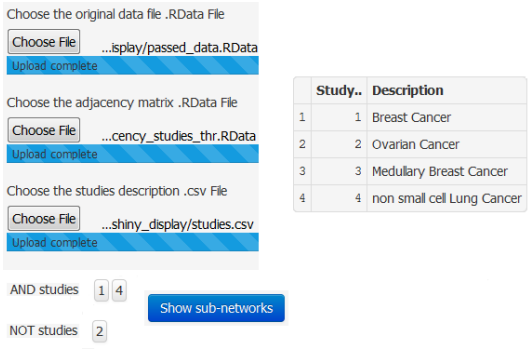



Figure 3: Logic Application interface.

## Conclusions

We have developed a tool that aims to identify unique sub-networks and genes based upon a number of microarray studies. We explore networks and genes that are robust and unique to a pre-selected number of studies. We support our results using prediction accuracy and a score to test the significance of identifying a subset of unique genes. Furthermore, we created an application interface which allows the user to combine different studies through AND and OR logic operators. Based on the findings we conclude that our pipeline is a robust and reliable method to analyse large sets of transcriptomic data. It detects relationships between transcriptional expression of genes that are specific to different conditions and also highlights structures and nodes that could be potential targets for further research.

## References


1. Safran M, Dalah I, Alexander J, Rosen N, Stein TI, Shmoish M, et al. GeneCards Version 3: the human gene integrator. Database. 2010;2010:baq020.
2. Choi JK, Yu U, Kim S, Yoo OJ. Combining multiple microarray studies and modeling interstudy variation. Bioinformatics. 2003;19(suppl 1):i84–i90.
3. Kirk P, Griffin JE, Savage RS, Ghahramani Z, Wild DL. Bayesian correlated clustering to integrate multiple datasets. Bioinformatics. 2012;28(24):3290–3297.
4. Steele E, Tucker A. Consensus and Meta-analysis regulatory networks for combining multiple microarray gene expression datasets. Journal of biomedical informatics. 2008;41(6):914–926.
5. Anvar SY, Tucker A, et al. The identification of informative genes from multiple datasets with increasing complexity. BMC bioinformatics. 2010;11(1):32.
6. Swift S, Tucker A, Vinciotti V, Martin N, Orengo C, Liu X, et al. Consensus clustering and functional interpretation of gene-expression data. Genome biology. 2004;5(11):R94.
7. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nature genetics. 2003;34(2):166–176.
8. Zhang J, Lu K, Xiang Y, Islam M, Kotian S, Kais Z, et al. Weighted Frequent Gene Co-expression Network Mining to Identify Genes Involved in Genome Stability. PLoS Computational Biology. 2012;8(8):e1002656.
9. Bo V, Curtis T, Lysenko A, Saqi M, Swift S, Tucker A. Discovering Study-Specific Gene Regulatory Networks. PLoS ONE. 2014;9(9):e106524.
10. Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic acids research. 2002;30(1):207–210.
11. Gautier L, Cope L, Bolstad BM, Irizarry RA. affy—analysis of Affymetrix GeneChip data at the probe level. Bioinformatics. 2004;20(3):307–315.
12. Lu J, Bushel PR. pvac: PCA-based gene filtering for Affymetrix arrays; 2010. R package version 1.12.0.
13. Pearson K. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901;2(11):559–572.
14. Friedman J, Hastie T, Tibshirani R. glasso: Graphical lasso- estimation of Gaussian graphical models; 2014. R package version 1.8.
15. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics. 2008;9(3):432–441.
16. Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. The Annals of Statistics. 2006;34(3):1436–1462.
17. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning. 1995;20(3):197–243.
18. Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. Journal of computational biology. 2000;7(3-4):601–620.
19. Scutari M. Learning Bayesian networks with the bnlearn R package. arXiv preprint arXiv:09083817. 2009;.
20. Scutari M, Scutari MM. Package bnlearn. 2012;.
21. Højsgaard S. Graphical Independence Networks with the gRain package for R. Journal; 2012.
22. RStudio, Inc . shiny: Web Application Framework for R; 2014. R package version 0.9.1.