Analysis of DNA Chips and Gene Networks **Fall Semester, 2010**

Lecture 14: January 21, 2010

Lecturer: Prof. Ron Shamir *Scribe: David Ze'evi*

Integrated Analysis of Gene Expression and Other Data

14.1 Introduction

Over the past weeks, we have seen a number of methods for gene expression profile analysis. From clustering *(see scribe 3-5)* via classification *(see scribe 7-9)* to biclustering *(see scribe 6, 10 and 11)*, all the methods we dicussed dealt solely with the analysis of expression patterns. We would now like to introduce methods that combine gene expression data with data from other sources such as protein-protein interactions networks, patient status and patient clinical parameters. This combination of different data types may help in mapping the underlying cause of a certain phenotype.

14.2 Analysis of expression profiles and a network

14.2.1 Goal

A large amount of data exists on protein-protein interactions (PPIs), usually presented as a large network. Since most biological processes are carried out by a set of interacting proteins, the superimposition of a protein-protein interactions network to a gene expression model can show which part of the network is active in a particular experiment. Our goal is to detect **active functional modules**: a connected subnetwork of genes that are co-expressed.

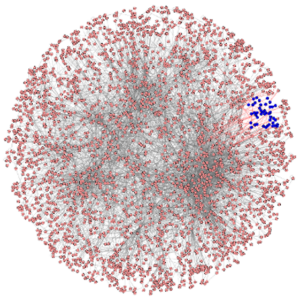

Figure 14.1: An active functional module (in blue) as a subset of a PPI network

14.2.2 MATISSE

MATISSE – Modular Analysis for Topology of Interactions and Similarity SEts [1] receives as input expression data and a PPI network and outputs a collection of modules: connected PPI subnetworks that have correlated expression profiles.

The probablistic model for expression similarities

For each module, we assume that the data is a mixture of two Gaussians: (1) mates – highly co-expressed genes and (2) non-mates – genes with low similarity values. We therefore compare two hypotheses: H_M (Module) assumes most of the genes are mates, and H_N (Null) assumes the number of mates is as expected at random. For a candidate group U, the likelihood ratio of originating from a module or from the background is:

$$W_U = log\frac{P(S_{U\times U}|H_M)}{P(S_{U\times U}|H_N)} = \sum_{(i,j)\in U\times U} log\frac{P(S_{ij}|H_M)}{P(S_{ij}|H_N)} = \sum_{(i,j)\in U\times U} w_{ij} \qquad (14.1)$$

The module's score is the gene group's likelihood ratio, which is the log likelihood sum over all gene pairs in U. See [1] for more details.

Front and back nodes

Not all genes may have similarity values. We call the genes that have significant similarity values front nodes and they may be connected by MATISSE using other genes (back nodes). Back nodes correspond to unmeasured transcripts, post-translationally regulated genes and partially regulated pathways (Fig. 14.2).

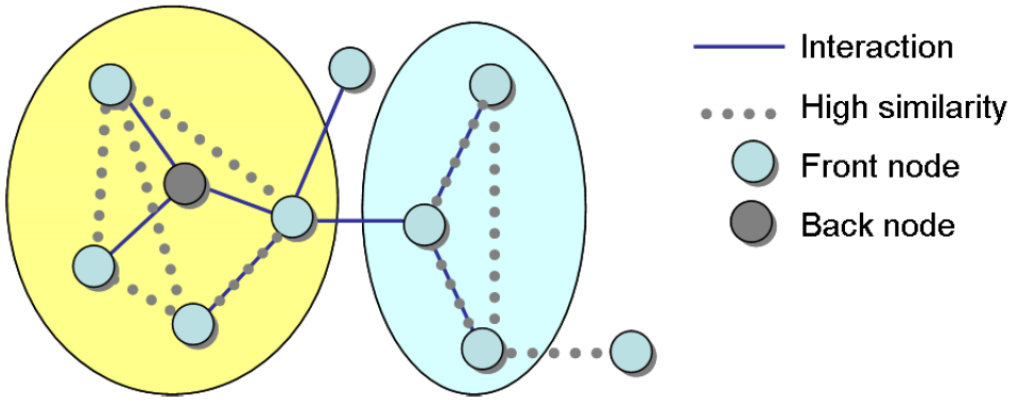

Figure 14.2: **Toy input example.** A toy example of an input problem with two distinct modules and with front and back nodes. Both modules (circled) are connected in the interaction network and heavy in the similarity graph. Note that the four front nodes in the left module form a connected subgraph only after the addition of the back node. *Image source: Ulitsky et al. 2007[1]*

Computational aspects

When the PPI network is a clique (a fully connected graph), our problem is to search for a single module without connectivity constraints. This problem then is to find the heaviest subgraph with positive and neative edge weights, which is known to be NP-Hard. We shall describe three heuristics for the problem. All heuristic methods follow three steps: (1) detection of small high-scoring seeds, (2) greedy optimization and (3) significance based filtering [1].

Seed generation. Three methods were tested for seed generation:

- Best-Neighbors: High scoring seeds of size k are created by initially ranking the graph's nodes, according to the edge weight of their neighbors, iterating over the highest scoring nodes, and heuristically selecting their $k-1$ best scoring neighbors to the potential seed.
- All-Neighbors: This method is similar to Best-Neighbors, but instead of selecting $k-1$ neighbors for a potential seed, in this version, all the neighbors of v with a non-negative edge score (including neighboring back nodes with zero score) enter the seed.
- Heaviest-Subset: This heuristic is inspired by the *Maximum Density Subgraph* 2-approximation algorithm[2]. For each connected component in the constraint graph, nodes are removed from the graph one at a time until none remain. After each node removal, the overall score of the remaining graph is recorded. After all nodes are removed, the highest scoring (possibly size-constrained) subgraph that was encountered is selected as the seed. That subgraph is then removed from the graph and the next seed is sought. This method is computationally more costly than the other two.

Greedy optimization. MATISSE simultaneously optimizes all the seeds, while not allowing modules to overlap and keeping an upper bound on module size. The optimization is done greedily, and therefore might reach local maxima. The following steps (Fig. 14.3) are considered:

- Node addition: addition of an unassigned node to an existing module.
- Node removal: removal of an interrupting node from a module.
- Assignment change: exchange of a node between modules.
- Module merge: forming a new module by taking the union of two existing ones.

Significance filtering. After optimizing each seed, the modules created are filtered in a two-step process. First, the significance of each module is tested by randomly sampling gene groups of the same size, and comparing module scores (Eq. 14.1). In a second step, to avoid possible bias in the score, module significance is tested using only expression similarity scores. The same sampling procedure is performed using the raw expression pairwise similarity values, and modules whose average similarity is not sufficiently high compared to the sampled sets of the same size are removed.

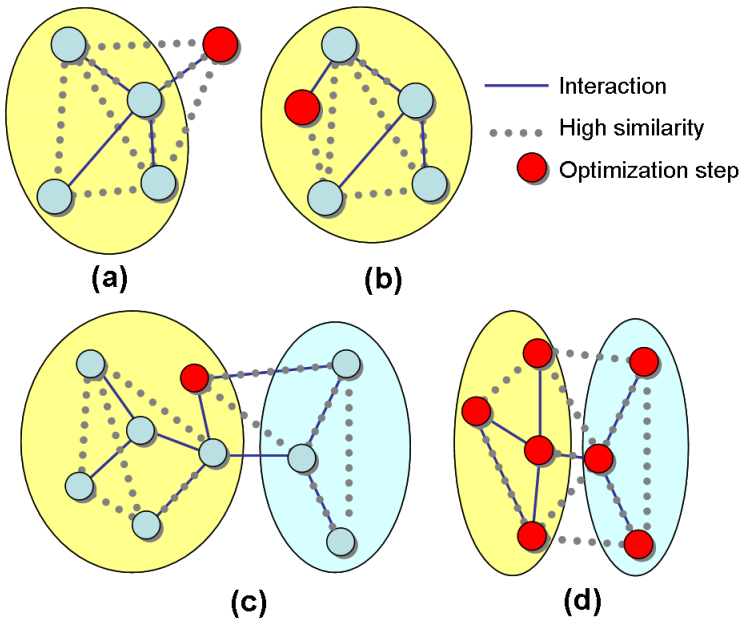

Figure 14.3: **Examples of the moves performed by the optimization algorithm.** (a) Node addition; (b) Node removal; (c) Assignment change; (d) Module merge. *Image source: Ulitsky et al. '07* [1]

Advantages of MATISSE

MATISSE has several advantages over previous methods. First, there is no need for confidence estimation on individual measurements. Second, *MATISSE* works even when only a fraction of the genes' expression patterns are informative, by employing back nodes. The method can handle similarity data, and there is no need to pre-specify the number of modules.

14.2.3 Performance comparison

MATISSE, Co-clustering [3], CLICK [4], a method producing random connected modules and a method producing random modules of the same sizes were tested on data for osmotic shock response in *S. cerevisae* [1]. The network consisted of 6,246 genes, with 65,990 protein-protein and protein-DNA interactions. The expression of the genes was measured in 133 different conditions – response of perturbed strains to osmotic shock. In the MATISSE solution, 2,000 genes were selected as front nodes, based on their variation, and the size of the modules was limited to 120 proteins. It is visible (Table 14.1) that MATISSE obtained the highest edge density of all methods (together with the random connected graph) and very good homogeneity. CLICK had a better homogeneity score, as it disregards the network structure. CLICK and Co-Clustering tend to produce highly fragmented modules.

Solution	Expression Homogeneity	Edge Density	Connected components
MATISSE	0.361	**0.035**	1.00
Co-Clustering	0.354	0.010	89.67
CLICK	**0.438**	0.011	77.61
Random connected	0.063	**0.036**	1.00
Random	0.033	0.003	89.78

Table 14.1: **Methods comparison**. The right column is the average number of connected components per module.

Next, we check the enrichment of the modules with biological annotations (GO, MIPS and KEGG) for each solution (Fig. 14.4) and the portion of biological annotations that are enriched in the modules. MATISSE has a significant advantage over other methods.

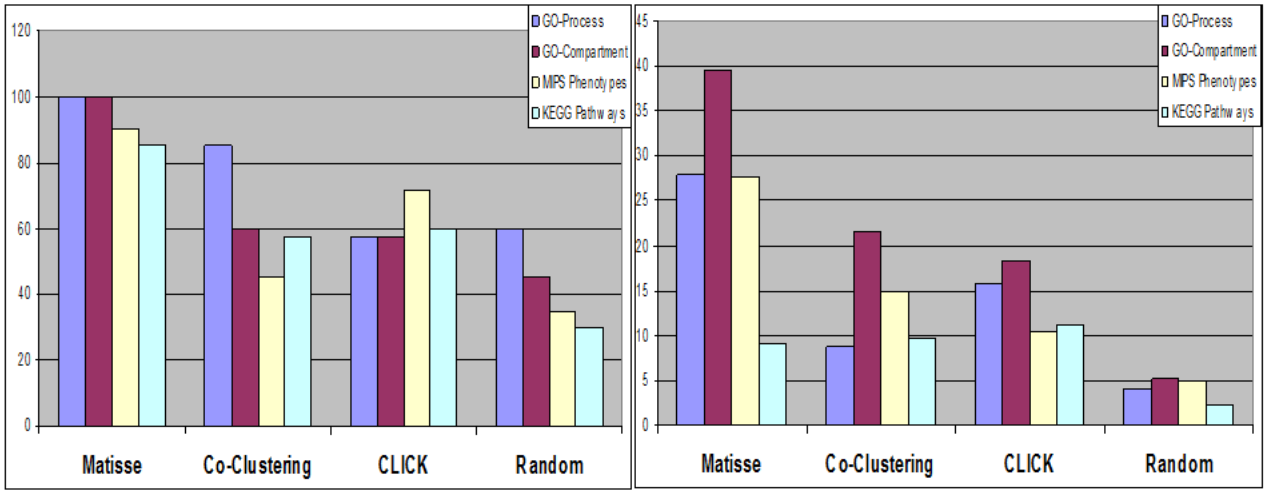

Figure 14.4: **Performance comparison.** The left chart depicts the percent of modules with category enrichment at a p-value $< 10^{-3}$. The right chart depicts the percentage of annotations with enichment at a p-value $< 10^{-3}$ in modules.

14.3 Analysis of expression profiles, network and patient status

14.3.1 Goal

The most popular human expression studies compare two cohorts of individuals, for example, sick and healthy ones. Alternatively, two disease subtypes are compared. Hundreds of comparison studies were published over the last decade and were used for patient classifi-

cation and for selection of biomarkers. In order to improve our understanding of disease mechanisms, we can integrate case-control profiles with network information. It is possible to extract dysregulated pahways specific for the *cases*, i.e., ones that are altered in diseased individuals compared to *controls*. A meaningful pathway should be a connected subgraph of the original graph and a good solution should account for heterogeneity among *cases*.
Note: for simplification, we will refer to two groups as *case* and *control*, even though the methodology hereinafter is applicable for any other pairwise comparison.

14.3.2 Methods

The following section was derived from a study on the detection of disease-specific dysregulated pathways (DPs) from the analysis of clinical expression profiles[5], published in 2008. In the study, the expression data are preprocessed, a mathematical problem is formulated and solved using approximation algorithms and heuristics.

Preprocessing

The initial input is the gene expression matrix in which the columns correspond to samples taken from case and control patients and the rows correspond to genes. For each of the genes, we use the distribution of its values among the controls to decide if the gene is dysregulated in each of the cases (Fig. 14.5 B). This way, we obtain a binary matrix of cases vs. genes. An additional input is the protein-protein interactions network, with nodes corresponding to proteins and edges corresponding to interactions. Each gene in this pattern now has a dysregulation pattern, which is simply its row in the binary matrix (Fig. 14.5 C).

Problem formulation

The known gene network is presented as an undirected graph, where each node (gene) has a corresponding set of elements (samples) in which it is differentially expressed (Fig 14.5 C). Our goal is to detect a minimal connected subnetwork with at least k nodes differentially expressed in all but l analyzed samples (l thus denotes of the number of allowed 'outliers'). We call such submetwork a *dysregulated pathway* (DP).

Formalization follows. We are given an undirected graph $G = (V, E)$ and a collection of sets $\{S_v\}_{v \in V}$ over the universe of elements U (the gene rows from the binary matrix), with $|U| = n$. For ease of representation, we will use, in addition to G, a bipartite graph $B = (V, U, E^B)$ where $(v, u) \in E^B, v \in V, u \in U$ if and only if $u \in S_v$ (Fig 14.5 D). A set $C \subseteq V$ is a *connected* (k, l)*-cover* (denoted $CC(k, l)$) if C induces a connected component in G and a subset $U' \subseteq U$ exists such that $|U'| = n - l$ and for all $u' \in U'$, $|N(u') \cap C| \geq k$, i.e., in the induced subgraph (C, U') the minimal degree of nodes in U' is at least k ($N(x)$ is the set of neighbors of x in B). We are interested in finding a $CC(k, l)$ of the smallest cardinality. We denote this minimization problem by $MCC(k, l)$.

Hardness Without connectivity constraints, i.e., when G is a clique, $MCC(k = 1, l = 0)$ is *Set Cover*, $MCC(k > 1, l = 0)$ is *Set k-Cover* and $MCC(k = 1, l > 0)$ is *Partial Set Cover*. All these problems were shown to be NP-Hard. For a general G, $MCC(1, 0)$ is the *Connected*

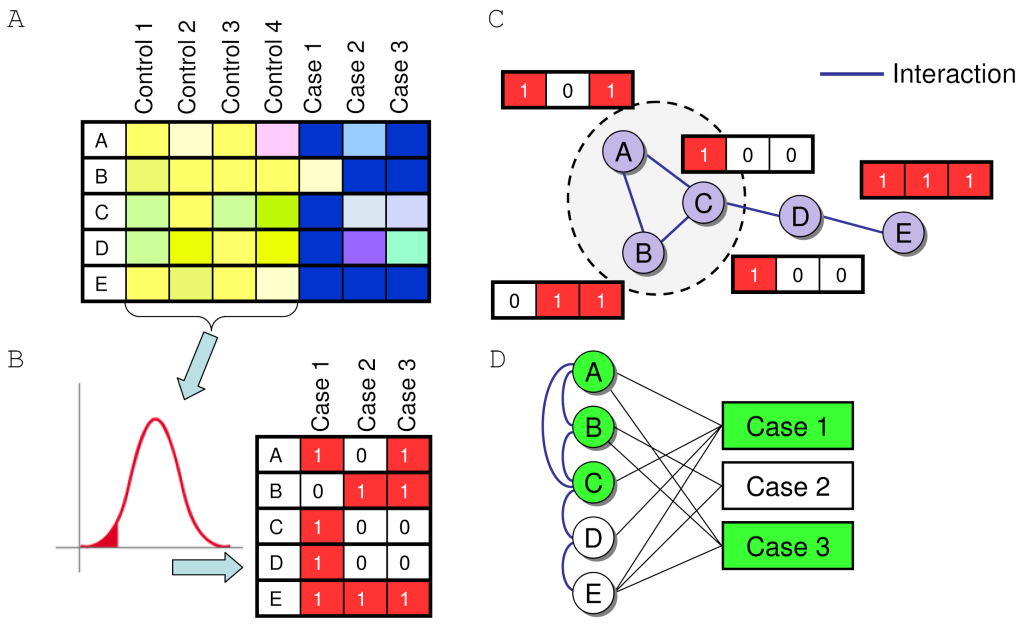

Figure 14.5: **From case-control profiles to dysregulated pathways.** (A) The initial input: a gene expression matrix. (B) Data is preprocessed and the dysregulation pattern is determined. (C) A second input is a protein interaction network. The row next to each gene is its dysregulation pattern. The goal is to find a smallest possible subnetwork in which, in all but l cases, at least k genes are differentially expressed. In this example, the circled subnetwork satisfies the condition with $k = 2$, $l = 1$: (i) A and C are dysregulated in case 1; (ii) A and B are dysregulated in case 3. (D) The bipartite graph representation of the data. Genes (left) are connected to the cases (right) in which they are differentially expressed. Edges between genes constitute the protein interaction network. The genes of the minimal cover and the samples covered by them are in green. *Image source: Utilsky et al. '08* [5]

Set Cover problem, which was recently studied in the context of wavelength assignment of broadcast connections in optical networks [6]. It was shown to be NP-Hard even if at most one vertex of G has degree greater than two.

Greedy algorithms for $MCC(k, l)$

Two variants of the classical greedy approximation for *Set Cover* were tested for the approximation of this problem. For simplicity we will describe them for $MCC(1, 0)$.

Expanding Greedy The first algorithm, *ExpandingGreedy* works as follows: Given a partial cover $W \subseteq V$ and the set of corresponding covered elements $X \subseteq U$, the algorithm picks a node $v \in V$ that is adjacent to W and that covers the largest number of elements of $U \setminus X$, adds v to the cover and adds $N(v) \setminus U$ to X. Initially $W = \emptyset$, $X = \emptyset$ and the first node is picked without connectivity constraints. Unfortunately, *ExpandingGreedy* can be shown to give a solution that is $\theta(|V|)$ times the optimal solution. Specifically, it runs into difficulties in cases where all the nodes in the immediate neighborhood of the current solution have equal benefit, and the next addition to the cover is difficult to pick.

Connecting Greedy The second algorithm, *ConnectingGreedy*, first uses the simple greedy algorithm [7] to find a set cover C that ignores the connectivity constraints and then augments it with additional nodes in order to obtain a proper cover. The *diameter* of a graph is the maximum length of a shortest path between a pair of nodes in V. *ConnectingGreedy* guarantees an approximation ratio of $O(Dlogn)$ for $MCC(1,0)$, where D is the diameter of G, since we can connect C using $|C|-1$ paths of length $\leq D$ each.

The CUSP Algorithm

We next describe an algorithm called *Covering Using Shortest Paths* (*CUSP*). Let $d(v,w)$ be the distance in edges between v and w in G. For each root node r and for each element $u \in U$ the algorithm computes distances $(M[r,u]_1,\ldots,M[r,u]_k)$ and pointers $(P[r,u]_1,\ldots,P[r,u]_k)$ to the k nodes closest to r that cover u. This can be done by computing the distances from r to all the nodes in V that cover u, and then retrieving the k closest nodes, which is an instance of the selection problem and can be solved in expected linear time [8]. Now take X_r, the union of the paths to the nodes covering the $n-l$ elements for which $max_q\{d(r,P[r,u]_q), 1 \leq q \leq k\}$ is the smallest. The final solution is $X = argmin_v|X_v|$.

Claim 14.1 *X_r is a proper $CC(k,l)$*

Proof: (a) X_r is a subtree of T and thus induces a connected component in G; (b) $n-l$ elements of U are covered k times by the corresponding $\{P[r,u]_i\}$ ■

Claim 14.2 *CUSP yields an n-approximation for $MCC(1,0)$.*

Proof: Let C_{OPT} be the optimal solution and let C_{CUSP} be the *CUSP* solution. Choose some $v_{OPT} \in C_{OPT}$. Since C_{OPT} is a cover, every vertex $u \in U$ has a neighbor w_u in C_{OPT}. Let w_u^* be the vertex for which $d(v_{OPT}, w_u^*)$ is maximal. C_{OPT} must contain all the vertices along some path from v_{OPT} to w_u^*.

$$|C_{OPT}| \geq d(v_{OPT},w_u^*)+1 = max_{u\in U}\{d(v_{OPT},w_u)|(w_u,u)\in E^B\}+1 = max_{u\in U}\{M[v_{OPT},u]\}+1 \tag{14.2}$$

CUSP minimizes over all the $|X_r|$s and therefore $|X_{v_{OPT}}| \geq |C_{CUSP}|$. $X_{v_{OPT}}$ is a union of shortest paths from vertices for which $M[v_{OPT},u]$ is minimal. The number of vertices in each of these paths $\leq max_{u\in U}\{M[v_{OPT},u]\}+1$ and for n paths:

$|C_{CUSP}| \leq n(max_{u\in U}\{M[v_{OPT},u]\}+1) \leq n|C_{OPT}|$ ■

Note that n is the number of cases and is often much smaller than $|V|$ – in human data, usually $|V| > 10,000$ and $n < 100$. In the general case, this algorithm can be proved to give a $k(n-l)$-approximation for $MCC(k,l)$. In terms of computational complexity, the total amount of work for each choice of r is $O(|V|+|E|+|E^B|)$, by simply using DFS, and the overall complexity is $O(|V|(|V|+|E|+|E^B|))$. See [5] for some biomedical applications of the method.

14.4 Analysis of expression profiles, network and clinical parameters

14.4.1 Goal

We would now like to expand the model by adding clinical parameters for each patient. Clinical parameters can be either logical – for example, the gender of the patient, the mutation status of a certain gene – or numerical – for example, the age of the patient, the size of the tumor or the patient's metastasis free survival period. We report here on a study that aims to analyze both types of parameters [9].

14.4.2 Handling clinical parameters

For each parameter, we compute a profile P_i across all patients. For a numerical parameter, the profile will be its normalized expression pattern (Fig. 14.6 A). For a logical parameter, we construct a seperate pattern for each value (Fig. 14.6 B).

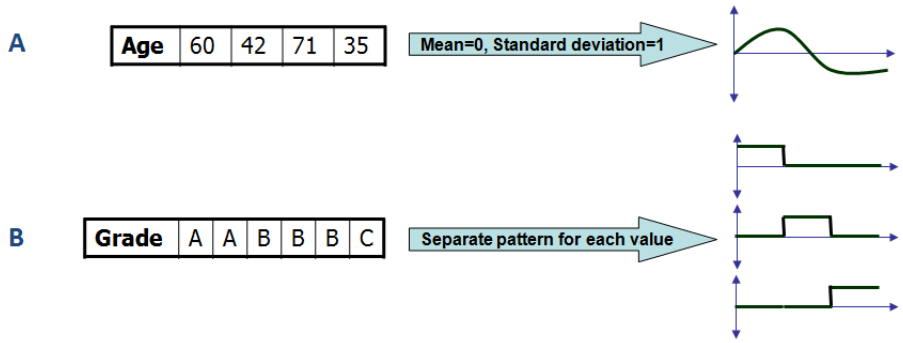

Figure 14.6: **Handling clinical parameters.** (A) Handling numerical parameters. (B) Handling logical parameters.

For genes i, j, and a clinical parameter k with profile P_k, we compute pairwise similarities:

$$S(i,j) = \frac{S_{diff}(i,j) + \lambda S_{corr}(i,j)}{1+\lambda} \tag{14.3}$$

where $S_{diff}(i,j) = min\{Corr(Pattern(i), P_k), Corr(Pattern(j), P_k)\}$
and $S_{corr}(i,j) = PartialCorrelation(Pattern(i), Pattern(j)|P_k)$
Hence, S_{diff} is the lower of the two correlations of the gene's pattern with the profile (we take the minimum as it would be meaningful only if both correlations are high) and S_{corr} is the correlations of the two gene patterns after correcting for their individual correlations with P_k. λ is a weighting factor.

14.4.3 Study outline

After transforming the clinical parameters to profiles, they are used together with the gene expression patterns (Equation 14.3) to form a new gene similarity matrix. This matrix, along with the pairwise protein interaction network are input for *MATISSE* [1] *(see above)*. The modules extracted are then filtered for redundancies (Fig. 14.7).

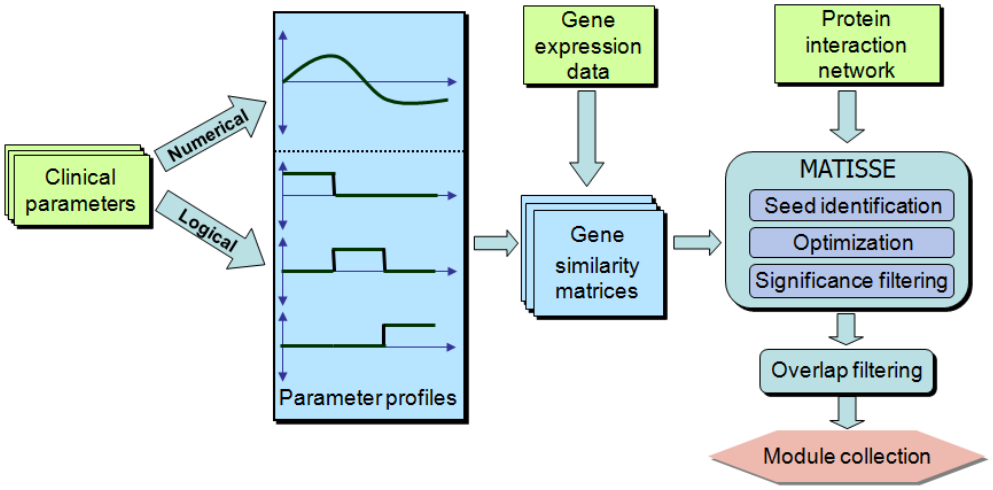

Figure 14.7: **Clinical parameters study outline.**

14.4.4 Breast cancer study results

Expression profiles of 99 breast cancer tumor samples were analyzed [10]. Ten clinical parameters such as the age at diagnosis, tumor size, mutation status of certain genes, etc. were used to compile a similarity matrix. A human protein-protein interaction network of 10,033 proteins and 41,633 interactions was used.

Results Significant modules were identified for 9 of the 10 clinical parameters. After filtering, 10 modules were obtained, ranging from 84 to 118 genes each. Six of the modules were enriched with more than one biological process, and seven of the modules had enrichment for more than one breast cancer related gene set. Here are some observations on interesting modules:

- Module no. 1 - Age related:
  - Positively correlated with age at diagnosis.
  - Enriched with genes upregulated in aged Rhesus.
  - Contains genes from the PKC pathway that activates NF-κB, a transcription factor that was implicated in aging [11].

- Module no. 3 - Estrogen receptor:
  - Negatively correlated with tumor size.
  - Enriched with estrogen receptor (ER) targets ($p = 1.13 \times 10^{-4}$).
  - The module itself contains two estrogen receptors (ESR1 and ESR2).
  - Hypothesis: increased ER transcription factor activity could result in smaller tumors.
- Module no. 7 - Ribosomal proteins:
  - Expression correlated with longer metastases-free survival.
  - Enriched with ribosomal proteins (RPs).
  - High expression of RPs is indicative of milder ovarian tumors.
  - Supports finding that RP expression is correlated with longer survival.

14.5 Summary

We have just described several methods for use in the field of systems biology. Since the methods use protein-protein interactions graphs, we encounter many graph problems. As these tend to be NP-Hard, we also encounter approximation methods and heuristics. On a more theoretical level, seeing that additional information improves the analysis, an important challenge here is the integration of even more diverse data sources. The latter should be an important component in computational systems biology tools developed in the future.

Note that the methods we covered are generic, for example, when dealing with clinical parameters, we simply integrated them with expression profiles to fit as *MATISSE* input. To summarize, the above methods - as well as others - may provide shortcuts in search for good hypotheses, but we have to keep in mind that the decisive proof of the utility of the methods will eventually be done via experimental validation.

Bibliography

[1] Ulitsky I. and Shamir R.: *Identification of functional modules using network topology and high-throughput data.* BMC Systems Biology, Vol. 1, No. 8 (2007)

[2] Charikar M.: *Greedy Approximation Algorithms for Finding Dense Components in a Graph.* Lecture Notes in Computer Science (2000), Vol. 1913, pp. 84-95.

[3] Hanisch D., Zien A., Zimmer R., Lengauer T.: *Co-clustering of biological networks and gene expression data.* Bioinformatics 18 (2002) Suppl 1:S145-54.

[4] Sharan R. and Shamir R.: *CLICK: A clustering algorithm with applications to gene expression analysis.* In Proceedings of the 8th Annual International Conference on Intelligent Systems for Molecular Biology, (ISMB '00), pp. 307316. AAAI Press, 2000.

[5] Ulitsky I., Karp R.M. and Shamir R.: *Detecting Disease-Specific Dysregulated Pathways Via Analysis of Clinical Expression Profiles* Proc. RECOMB 2008, pp. 347-359, LNBI 4955, Springer, Berlin, (2008).

[6] Shuai, T., Hu, X.: *Connected set cover problem and its applications.* In Cheng S, Poon C, eds.: AAIM. Volume 4041 of Lecture Notes in Computer Science., Springer (2006) pp. 243254.

[7] Hochbaum, D.S.: *Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems.* In Hochbaum, D.S., ed.: Approximation algorithms for NP-hard problems. PWS, Boston (1997) pp. 94143.

[8] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: *Introduction to Algorithms.* MIT Press, Cambridge, MA (1990).

[9] Ulitsky I. and Shamir R.: *Detecting pathways transcriptionally correlated with clinical parameters* Proc. 7th Annual International Conference on Computational Systems Bioinformatics (CSB 08) pp. 249–258, Imperial College Press, London, UK (2008)

[10] Minn A.J., Gupta G.P., Siegel P.M., et al: *Genes that mediate breast cancer metastasis to lung.* Nature 436 (2005) pp. 518-524

[11] Adler A.S., Kawahara T.L., Segal E., Chang H.Y. *Reversal of aging by NFkappaB blockade* Cell Cycle 7(5) pp. 556-9 (2008)