Internet Mathematics Vol. 7, No. 4: 299–313

# KeyPathwayMiner: Detecting Case-Specific Biological Pathways Using Expression Data

Nicolas Alcaraz, Hande Kücük, Jochen Weile, Anil Wipat, and Jan Baumbach

**Abstract.** Recent advances in systems biology have provided us with massive amounts of pathway data that describe the interplay of genes and their products. The resulting biological networks can be modeled as graphs. By means of "omics" technologies, such as microarrays, the activity of genes and proteins can be measured. Here, data from microarray experiments is integrated with the network data to gain deeper insights into gene expression. We introduce KeyPathwayMiner, a method that enables the extraction and visualization of interesting subpathways given the results of a series of gene expression studies. We aim to detect highly connected subnetworks in which most genes or proteins show similar patterns of expression. Specifically, given network and gene expression data, KeyPathwayMiner identifies those maximal subgraphs where all but $k$ nodes of the subnetwork are expressed similarly in all but $l$ cases in the gene expression data. Since identifying these subgraphs is computationally intensive, we developed a heuristic algorithm based on Ant Colony Optimization. We implemented KeyPathwayMiner as a plug-in for Cytoscape. Our computational model is related to a strategy presented by Ulitsky et al. in 2008. Consequently, we used the same data sets for evaluation. KeyPathwayMiner is available online at http://keypathwayminer.mpi-inf.mpg.de.


ISSN: 1542-7951 print

## 1. Introduction

Currently, there are genome sequences for several thousand organisms available in the National Center for Biotechnology Information (NCBI) databases [Sayers et al. 10]. However, these data provide only the first step in understanding how complex organisms evolve and how cells modulate their behavior when exposed to changing environmental conditions. Using different experimental conditions we can unravel systems of molecular interactions that control the expression and activity of genes and proteins. The results of such experiments allow the construction of systems biology models. This approach has the potential to revolutionize the investigation of complex diseases for which a deeper understanding of the interplay of many genes and proteins is crucial; cancer and neurodegenerative diseases may serve as examples here.

To date, networks and gene expression are typically studied in isolation. There are many approaches that aim to identify typical statistical network features, such as scale-free distributions [Balaji et al. 06], network centralities, or hub nodes [Assenov et al. 08]. Other approaches address overrepresented network patterns [Hartsperger et al. 10]. On the other hand, microarray experiments, for example, can be performed on a set of patients suffering a certain disease. Clustering approaches may be applied to the resulting data to find sets of genes with similar expression behavior across all patients. In a similar fashion, we may also cluster the patients to unravel cases in which most genes correlate in their expression, e.g., to identify subcategories of cancer [Wittkop et al. 10].

One of the major problems is the huge amount of available data, which is growing continuously. To date (October 2011), there are over half a million samples in the Gene Expression Omnibus database GEO [Edgar et al. 02]. Furthermore, there are 50 million interactions in the STRING database [Jensen et al. 09], 155,578 protein–protein interactions in IntAct [Aranda et al. 10], and about 33,420 reactions in the Reactome database [Croft et al. 10].

In previous work, Ulitsky et al. presented a method for finding a "dysregulated pathway, which is a minimal connected subnetwork with at least $k$ nodes differentially expressed in all but $l$ analyzed samples" [Ulitsky et al. 08]. In other words, they combine biological network data with gene expression data. The aim is the identification of subnetworks that show a similar behavior over many expression samples. However, finding maximal connected subnetworks that maximize a given scoring function based on all nodes of the reported subnetworks is NP-hard [Ideker et al. 02]. Therefore, Ulitsky et al. simplified the problem by searching for minimal connected subgraphs with at least $k$ nodes. Furthermore, they preprocessed each node individually: To be considered, an underlying gene has to be differentially expressed in all but $l$ cases. However, this causes a

practical problem: A suitable value for $k$ has to be determined, but $k$ is generally unknown a priori. If there is no subgraph of, for example, ten nodes or more, the algorithm needs to be restarted with a smaller value of the parameter $k$.

In this paper, we present a similar approach but model the underlying biological question slightly differently. We introduce KeyPathwayMiner, a method to detect all maximal connected subnetworks in which all but $k$ nodes are differentially expressed in all but $l$ analyzed samples. Since we formulate a maximization problem and allow for noise in the network as well as the gene expression data (the *exceptions* $k$ and $l$), the user does not need to know the minimal number of nodes. We simply report all subgraphs in which all nodes but $k$ are dysregulated in all but $l$ samples. Furthermore, we offer different scoring functions for the reported pathways. Since the maximization problem is computationally hard, we developed and implemented an adapted Ant Colony Optimization (ACO) procedure. In the following sections, we provide formal definitions, introduce the ACO approach, and finally test KeyPathwayMiner on the same data sets used in [Ulitsky et al. 08].

## 2. Methods and Data

### 2.1. Definitions

Let $G = (V, E)$ be a graph representing a biological network in which all $v \in V$ represent genes or gene products (e.g., proteins, metabolites) and edges $(v, u) \in E$ stand for known physical, regulatory, or genetic interactions between two nodes $v$ and $u$. We define the matrix $C_{p\times q}$ as follows:

$$C_{ij} = \begin{cases} 1 & \text{if gene } i \text{ is differentially overexpressed in case } j, \\ -1 & \text{if gene } i \text{ is differentially underexpressed in case } j, \\ 0 & \text{otherwise.} \end{cases}$$

Let $Z : V \to \{-1, 0, 1\}^q$ be a mapping of vertex $v$ to its corresponding $q$-dimensional vector row in $C$. We compute the number of differentially expressed cases of vertex $v$ as

$$R(v) = \sum_{j=1}^{q} |Z_j(v)|.$$

Furthermore, we define a set of vertices $D(k, l) \subset V$ that satisfies the following conditions:

1. $D(k, l)$ induces a connected component in $G$.

2. With the exception of at most $k$ vertices, all other vertices $v \in D(k,l)$ have $l \leq R(v) \leq q$.

Given a graph $G(V,E)$, a matrix $C_{p\times q}$, and parameters $k,l$, our goal is to find maximal sets $D(k,l)$ such that $D(k,l) \subseteq D'(k,l) \Longrightarrow D(k,l) = D'(k,l)$. In other words, we are searching for maximal connected components of differentially expressed genes.

## 2.2. Running Time

We can find maximal $D(k,l)$ sets exactly by first labeling each node in the graph either as a differentially expressed node if $R(v) \geq q - l$ or otherwise as an exception node.

Hence, we may construct a graph $G'$ by first removing all exception nodes from $G$. Afterward, we compute all connected components. For each connected component $V_{\text{cc}} \subseteq V$ left in $G$ we create a new node $v' \in V(G')$ with weight $w(v') = |V_{\text{cc}}|$. Finally, we insert all exception nodes in $G'$ and create an edge between each exception node $u \in G'$ and weighted node $v'$ if $\exists v \in V_{\text{cc}}$ such that $\exists (u,v) \in E(G)$.

All exception nodes are now a vertex cover in $G'$, and no edges exist between weighted nodes; if there were, we would have merged them in the construction of $G'$. Hence, all edges in $G'$ are incident to at least one exception node. See Figure 1 for an example.

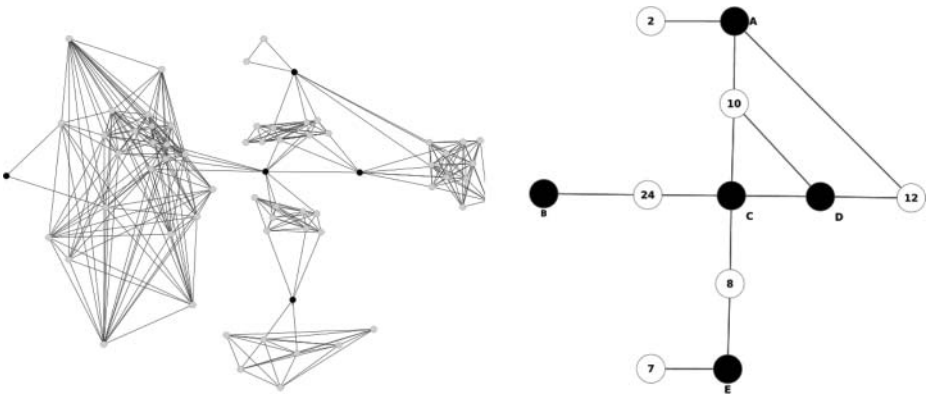


**Figure 1.** A new compressed graph (right) is constructed from the original (left). In this process, all connected differentially expressed nodes are merged into a single weighted node. Also, if at least one node in the original uncompressed connected component is connected to an exception (black) node, then an edge from its corresponding weighted differentially expressed node to the black node exists (color figure available online).

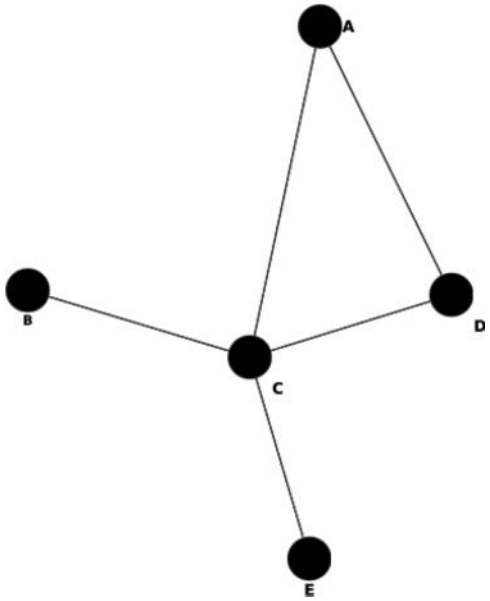


**Figure 2.** Exception graph for $k = 2$ (color figure available online).

We finally construct a third graph $G''$ consisting of all the black nodes in $G'$ by adding an edge between two black nodes in $G''$ if there is a path containing at most $k-2$ other black nodes in $G'$ (see Figures 2 and 3 for examples). Finding $D(k,l)$ sets in $G$ is then equivalent to computing all paths of length $k-1$ in $G''$.

Constructing $G'$ takes $O(|E| * |V| + |E|)$ in the worst case, constructing $G''$ can take up to $O(|V|^3)$, and computing all paths of length $k-1$ can be accomplished in $O(|V|^k)$. Hence, the overall running time is bounded by $O(|V|^k)$.

### 2.3. Finding Maximal Sets with Ant Colony Optimization

Finding maximal sets can become infeasible for large networks and large values of $k$. In order to find maximal sets $D(k,l)$ in a reasonable amount of time, we applied the widely known Ant Colony Optimization (ACO) strategy to our algorithm. ACO is a bio-inspired probabilistic algorithm used mainly for solving hard computational problems that can be formulated using graph theory. An extensive explanation of ACO and its variants has been given in [Dorigo and Stuetzle 04]. Here, we give a brief summary of the algorithm's basic elements that were used in our approach.

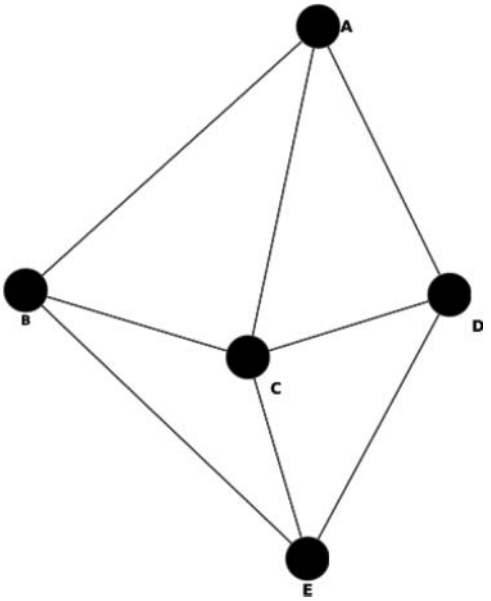


**Figure 3.** Exception graph for $k = 3$ (color figure available online).

A set of ants is initially placed on the vertices of the graph. The start vertex for each ant can be chosen randomly or arbitrarily (e.g., based on some heuristic). In the next step, each ant chooses an edge incident to the vertex $v$ where it is currently located by utilizing the following probability function:

$$p_{uv} = \frac{\tau_{uv}^{\alpha} \eta_{uv}^{\beta}}{\sum_{v \in N(u)} \tau_{uv}^{\alpha} \eta_{uv}^{\beta}}.$$

Here, $\tau$ is the amount of pheromone placed on edge $u, v \in E$, $\eta$ is the desirability of the edge, and $\alpha, \beta$ are parameters that control the importance of the pheromone and the desirability of the edge, respectively. In order to give higher preference to edges connected to highly differentially expressed nodes, we set the desirability of an edge $\eta_{uv} = R(v)$ to the number of differentially expressed cases in the opposite vertex.

Once an edge is chosen, the ant will move to the corresponding vertex at the other end, remembering the edge it has visited, and again will try to move to a new edge using the same probability function. If a dead end is reached, the ant is allowed to jump back to previously visited nodes with still unvisited edges incident to them. Also, ants will be allowed to visit only up to $k$ nodes that are not differentially expressed in at least $q - l$ cases, thus ensuring that all ants

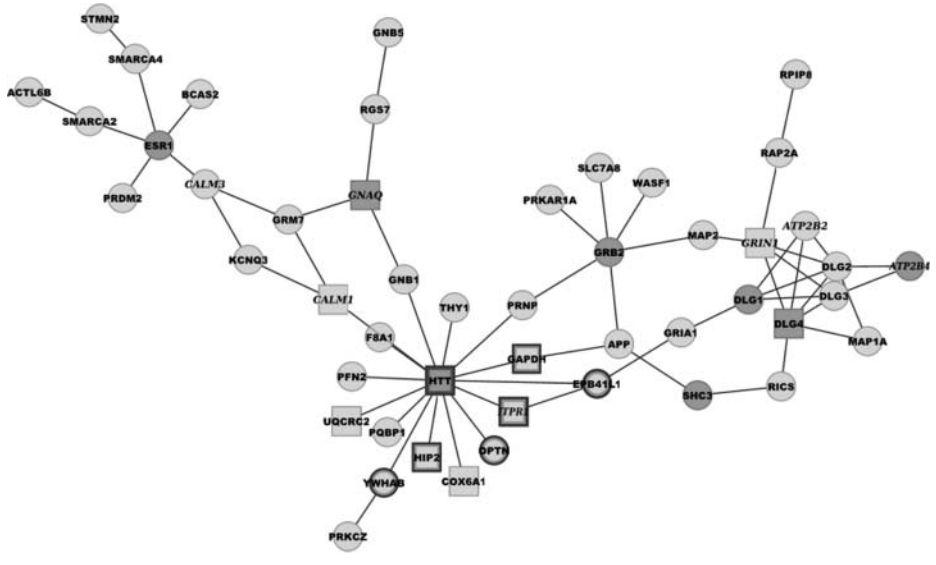


**Figure 4.** Largest subnetwork found for $k = 8$. Dark grey nodes represent exception nodes, squared nodes are genes also reported as part of the Huntington's disease KEGG pathway, nodes with dark borders are HD modifiers, and nodes with italic font are also part of the calcium signaling pathway.

report a valid differentially expressed pathway. Once all ants have found their respective solutions, they will drop pheromone on all the edges visited based on the quality of the solution found. The general pheromone update function for each ant $s$ is

$$\tau_{uv}^{s} = (1-\rho)\tau_{uv}^{s} + \triangle\tau_{uv}^{s},$$

where $\rho \in [0,1]$ is a parameter that controls the pheromone decay rate and $\triangle\tau_{uv}^{s}$ is the pheromone drop function. Since we are looking for maximal connected components, we implemented each ant $s$ to drop pheromone in an amount proportional to the number of nodes reported in the pathway ($\triangle\tau_{uv}^{s} = |V(D_s)|$). See Algorithm 1 for a formal description.

The algorithm ensures that no ant will visit more nodes or edges than are contained in a valid differentially expressed pathway. Let $D_{\max}$ be the largest differentially expressed pathway contained in $G$. Then in the worst case, $D_{\max} = |V|$, and every ant will visit $|E|$ edges of this pathway, resulting in a worst-case running time of $O(|S| * |E|)$ for each generation $S$ of ants.

## 3. Results and Discussion

### 3.1. Analysis of Huntington's Disease Expression Profiles

Huntington's disease (HD) is a degenerative neurological disorder caused by a genetic defect on chromosome number four that encodes a mutated version of the huntingtin (HTT) protein. The pathology of HD has been extensively described. However, the behavior of mutated HTT protein and its effect at the molecular level, especially in the human brain, are not completely understood. Recent studies have shown that mutant huntingtin interferes with the function of widely expressed transcription factors, suggesting that gene expression may be altered in a variety of tissues in HD.

We tested our method with the same human protein interaction network and expression data sets as utilized in [Ulitsky et al. 08] for Huntington's disease. The expression data sets, which were obtained using oligonucleotide arrays [Hodges et al. 06], consist of 32 unaffected control samples and 38 affected samples taken from the caudate nucleus region of the brain. The protein interaction network consists of 7384 nodes corresponding to Entrez gene identifiers and 23,462 interactions based mostly on small-scale experiments and obtained from several interaction databases. The network and sources information can be obtained from the website http://acgt.cs.tau.ac.il/clean.

**Algorithm 1:** Finding maximal sets with ACO.

**input** : A graph $G$ with corresponding expression mapping $Z$, a set $S$ of ants, number of allowed gene exceptions $k$, number of allowed case exceptions $l$.
**output**: A list $L$ of subgraphs $D_1, D_2, \ldots, D_n$ representing differentially expressed pathways

```
forall the ants s in S do
    v_current ← v_start;
    exceptions ← 0;
    visitedVertices ← ∅;
    visitedEdges ← ∅;
    stack ← ∅;
    stack.push(v_current);
    candidateEdges ← ∅;
    L ← ∅;
    while stack ≠ ∅ do
        candidateEdges ← {edges incident to v_current} \ visitedEdges;
        if exceptions == k then
            candidateEdges ← candidateEdges \ {(v_current, u) ∈ E|R(u) < q − l}
        end
        if candidatedEdges == ∅ then
            v_current ← stack.pop()
        end
        else
            chosenEdge ← choose edge from candidateEdges with probability p;
            if |candidatedEdges| >= 2 then
                stack.push(v_current)
            end
            v_current ← vertex on opposite end of chosenEdge;
            if R(v_current) < q − l then
                exceptions ← exceptions + 1
            end
            visitedNodes ← visitedNodes ∪ v_current;
            visitedEdges ← visitedEdges ∪ chosenEdge
        end
    end
    D_s ← {visitedNodes, visitedEdges};
    L ← L ∪ D_s;
end
return L
```

We used the same p-values and threshold as Ulitsky et al. for determining differential expression, and we also allowed for up to $l = 8$ nondifferentially expressed cases for each node for the down-regulated data sets. Table 1 summarizes our results.

In these expression data sets, the huntingtin (HTT) protein is not differentially expressed in more than eight cases, which makes it an exception node in the network. Since Huntington's disease is caused by a mutation in the huntingtin

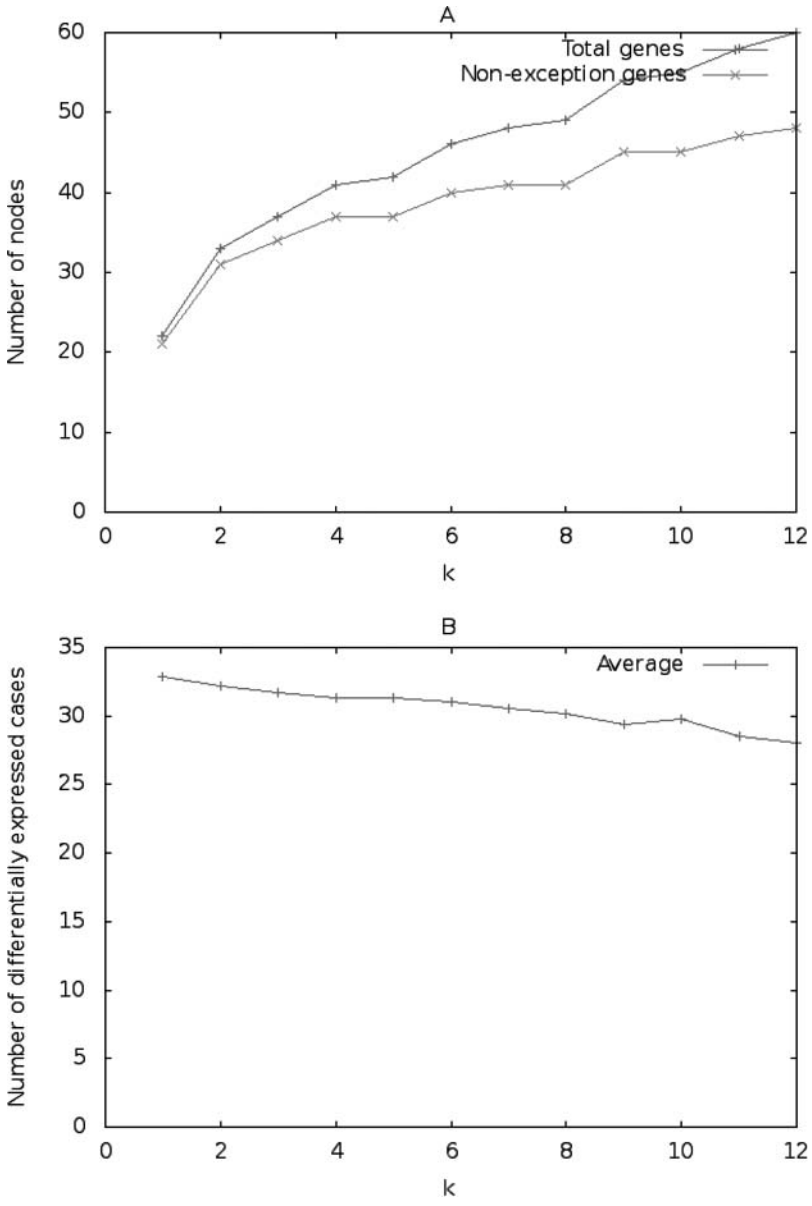


**Figure 5.** (A) The number of nodes (total and nonexception) of the largest subnetwork found for an increasing number of allowed exceptions $k$. Allowing for more exceptions increases the number of differentially expressed genes included in solutions. (B) The average number of differentially expressed cases per node in the largest subnetworks for an increasing number of allowed exceptions $k$. Allowing for more node exceptions slightly decreases the average differential expression for the largest subnetworks (color figure available online).

| | KPM ($k = 2$) | KPM ($k = 8$) | CUSP | GiGA | jActive-Modules | t-test top |
|---|---|---|---|---|---|---|
| Number of genes | 33 | 49 | 34 | 34 | 282 | 34 |
| Contains HTT ? | Yes | Yes | Yes | No | No | No |
| HD modifiers | 7 | 7 | 7* | 3 | 12 | 2 |
| KEGG HD pathway | 8 | 10 | 4 | 0 | 4 | 0 |
| Calcium pathway | 5 | 7 | 6 | 5 | 10 | 3 |

* This entry was updated from 6 to 7 to account for gene OPTN, which is included in Ulitsky's reported solution but not labeled as HD modifier, though it is reported in [Kaltenbach and Romero 07].

**Table 1.** Gene sets identified as down-regulated in HD caudate nucleus with KeyPathwayMiner (KPM) and compared to the results of [Ulitsky et al. 08] produced with their own CUSP algorithm, GiGA [Breitling et al. 04], jActiveModules [Ideker et al. 02], and the top 34 down-regulated genes with the most significant t-scores.

gene, it may not always lead to a change in expression patterns. Nevertheless, for $k = 1$, the HTT gene is still included in the largest subnetwork, which highlights the ability of our method to include important genes in the reported network even if they do not satisfy the user-defined expression threshold $l$.

For $k = 2$, the largest subnetwork found contains 33 nodes (see Figure 6), one fewer than the number in the largest subnetwork reported by Ulitsky et al. However, our network contains only two exception nodes, seven genes that have been reported as HD modifiers in [Kaltenbach and Romero 07], eight genes that are included in the KEGG HD pathway (twice as many as in Ulitsky's solution), and also five genes from the calcium signaling pathway, which is known to have an important role in the development of HD [Rockabrand et al. 07].

As $k$ increases, we allow more exception nodes to be reported, and the size of the highest-scoring subnetworks significantly increases, highlighting the ability of our method to find "bridge" exception nodes. Such nodes connect two or more highly differentially expressed regions that can play an important role in HD. These bridge nodes would have been disregarded if the expression profiles had been analyzed in isolation. For example, for $k = 8$, the size of the largest subnetwork found (see Figure 4) rises to 49 nodes, and also two additional genes are included from both the KEGG and the calcium signaling pathways. Three of these new genes (GNAQ, ATP2B4, DLG4) are exception genes, which again reinforces the importance of allowing nondifferentially expressed nodes to be reported in the solutions.

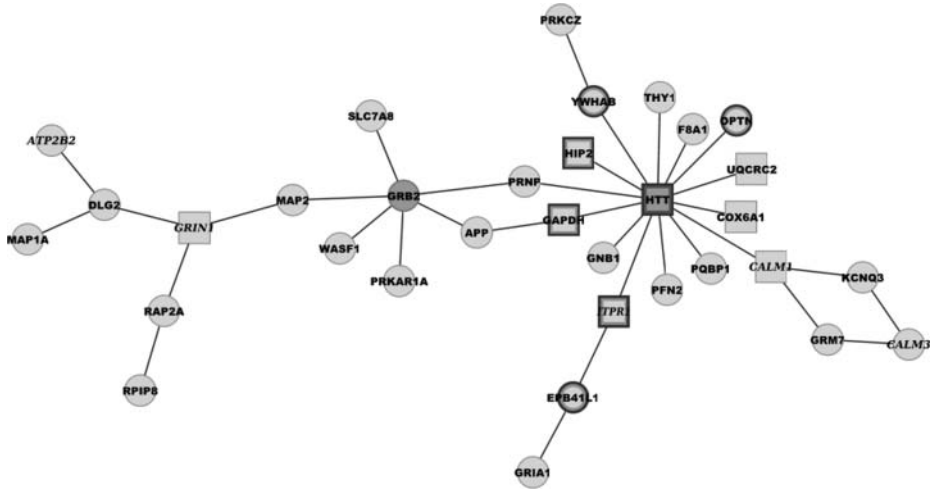


**Figure 6.** Largest subnetwork found for $k = 2$. Dark grey nodes represent exception nodes, squared nodes are genes also reported as part of the Huntington's disease KEGG pathway, nodes with dark borders are HD modifiers, and nodes with italic font are also part of the calcium signaling pathway.

It should be noted that allowing more exception nodes decreases the average number of differentially expressed cases for the whole subnetwork. However, this decrease is not large compared to the number of differentially expressed nodes that are included in potentially better solutions (see Figure 5). This difference is due to the fact that our method aims to include nodes that have a high number of differentially expressed cases with higher probability and thus reports results that have a higher biological relevance.

## 4. Conclusion

We have presented KeyPathwayMiner, a tool for the identification of highly connected subnetworks that show similar expression behavior in a given set of gene expression studies. KeyPathwayMiner is available as a Cytoscape plug-in and as a Java library. In contrast to [Ulitsky et al. 08], we model the biological question as a maximization problem. We report all maximal subnetworks in which all but $k$ nodes are differentially expressed in all but $l$ cases. In the future, we will integrate KeyPathwayMiner with the Ondex data integration framework [Köhler et al. 06]. Furthermore, we will investigate the application of the method to proteomics and metabolomics data (mass spectrometry and ion mobility spectrometry).

**Acknowledgments.** HK and JB are grateful for financial support from the Cluster of Excellence for Multimodal Computing and Interaction (MMCI), Germany. NA would also like to acknowledge the Max Planck Institute for Informatics (MPII) and the International Max Planck Research School (IMPRS) for their financial support. JW and AW would like to acknowledge funding from the Biotechnology and Biological Sciences Research Council (BBSRC) Systems Approaches to Biological Research (SABR) initiative (Grant number BB/F006039/1). Furthermore, we wish to thank Igor Ulitsky (Whitehead Institute) for providing us with the data sets used in this study.

## References

[Aranda et al. 10] B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, et al. "The Intact Molecular Interaction Database in 2010." *Nucleic Acids Res.* 38 (2010), Database issue, D525–31.

[Assenov et al. 08] Yassen Assenov, Fidel Ramírez, Sven-Eric Schelhorn, Thomas Lengauer, and Mario Albrecht. "Computing Topological Parameters of Biological Networks." *Bioinformatics* 24:2 (2008), 282–284.

[Balaji et al. 06] S Balaji, M Iyer Lakshminarayan, L Aravind, and M Madan Babu. "Uncovering a Hidden Distributed Architecture behind Scale-Free Transcriptional Regulatory Networks." *J. Mol. Biol.* 360:1 (2006), 204–212.

[Breitling et al. 04] Rainer Breitling, Anna Amtmann, and Pawel Herzyk. "Graph-Based Iterative Group Analysis Enhances Microarray Interpretation." *BMC Bioinformatics* 5:1 (2004), 100+.

[Croft et al. 10] David Croft, Gavin O'Kelly, Guanming Wu, Robin Haw, Marc Gillespie, et al. "Reactome: A Database of Reactions, Pathways and Biological Processes." *Nucleic Acids Res.* 39 (2010), D691–D697.

[Dorigo and Stuetzle 04] M. Dorigo and T. Stuetzle. *Ant Colony Optimization*. MA: The MIT Press, 2004.

[Edgar et al. 02] Ron Edgar, Michael Domrachev, and Alex E Lash. "Gene Expression Omnibus: NCBI Gene Expression and Hybridization Array Data Repository." *Nucleic Acids Res.* 30:1 (2002), 207–210.

[Hartsperger et al. 10] Mara L. Hartsperger, Robert Strache, and Volker Stümpflen. "HiNO: An Approach for Inferring Hierarchical Organization from Regulatory Networks." *PLoS One* 5:11 (2010), e13698.

[Hodges et al. 06] A. Hodges, A. D. Strand, A. K. Aragaki, A. Kuhn, T. Sengstag, et al. "Regional and Cellular Gene Expression Changes in Human Huntington's Disease Brain." *Hum. Mol. Genet.* 15:6 (2006), 965–977.

[Ideker et al. 02] Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F Siegel. "Discovering Regulatory and Signalling Circuits in Molecular Interaction Networks." *Bioinformatics* 18 (Suppl. 1) (2002), S233–240.

[Jensen et al. 09] Lars J. Jensen, Michael Kuhn, Manuel Stark, Samuel Chaffron, Chris Creevey, et al. "String 8—A Global View on Proteins and Their Functional Interactions in 630 Organisms." *Nucleic Acids Res.* 37 (Database Issue) (2009), D412–416.

[Kaltenbach and Romero 07] L. Kaltenbach and E. Romero. "Huntingtin Interacting Proteins Are Genetic Modifiers of Neurodegeneration." *PLoS Genet.* 3 (2007), e82.

[Köhler et al. 06] Jacob Köhler, Jan Baumbach, Jan Taubert, Michael Specht, Andre Skusa, et al. "Graph-Based Analysis and Visualization of Experimental Results with ONDEX." *Bioinformatics* 22:11 (2006), 1383–1390.

[Rockabrand et al. 07] E. Rockabrand, N. Slepko, A. Pantalone, V. Nukala, A. Kazanstev, et al. "The First 17 Amino Acids of Huntingtin Modulate Its subcellular Localization, Aggregation and Effects on Calcium Homeostasis." *Human Molecular Genetics* 16 (2007), 61–77.

[Sayers et al. 10] Eric W Sayers, Tanya Barrett, Dennis A Benson, Evan Bolton, Stephen H Bryant, et al. "Database Resources of the National Center for Biotechnology Information." *Nucleic Acids Res.* 38 (2010), D5–D16.

[Ulitsky et al. 08] Igor Ulitsky, Richard Karp, and Ron Shamir. "Detecting Disease-Specific Disregulated Pathways via Analysis of Clinical Expression Profiles."

*Proceedings of RECOMB, Research in Computational Molecular Biology* 4955 (2008), 347–359.

[Wittkop et al. 10] Tobias Wittkop, Dorothea Emig, Sita Lange, Sven Rahmann, Mario Albrecht, et al. "Partitioning Biological Data with Transitivity Clustering." *Nat. Methods* 7:6 (2010), 419–420.

Nicolas Alcaraz, Saarland University, Cluster of Excellence for Multimodal Computing and Interaction, Max Planck Institute for Informatics, Campus E2.1, 66123 Saarbrücken, Germany (nalcaraz@mpi-inf.mpg.de)

Hande Kücük*, Saarland University, Cluster of Excellence for Multimodal Computing and Interaction, Max Planck Institute for Informatics, Campus E2.1, 66123 Saarbrücken, Germany (hkucuk@mpi-inf.mpg.de)

Jochen Weile, Newcastle University, School of Computing Science, Newcastle upon Tyne NE1 7RU, United Kingdom (j.weile@ncl.ac.uk)

Anil Wipat, Newcastle University, School of Computing Science, Newcastle upon Tyne NE1 7RU, United Kingdom (anil.wipat@newcastle.ac.uk)

Jan Baumbach, Saarland University, Cluster of Excellence for Multimodal Computing and Interaction, Max Planck Institute for Informatics, Campus E2.1, 66123 Saarbrücken, Germany (jbaumbac@mpi-inf.mpg.de)

*Nicholas Alcaraz and Hande Kücük both contributed to this paper equally.

Received December 25, 2010; accepted April 14, 2011.