Introduction to Bioinformatics

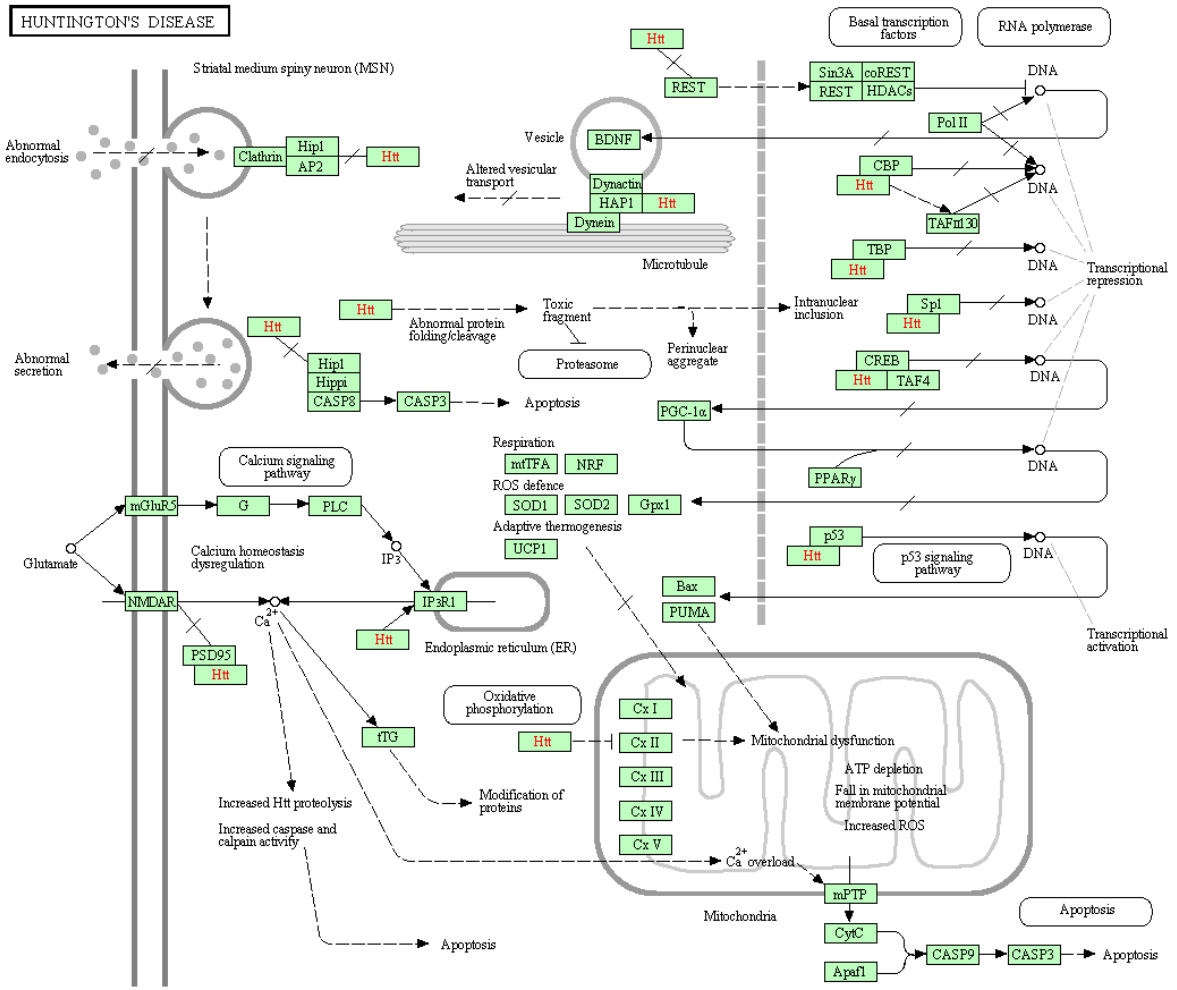
Network Enrichment

Lecturer: Jan Baumbach
Teaching assistant(s): Diogo Marinho

Overview

- Introduction
- Active Modules
- Detecting dysregulated pathways
  - *CUSP algorithm*
- Key-pathway mining
  - *Ant Colony Optimization*
- Example application: Huntington's Disease
- Summary

Pathway Databases

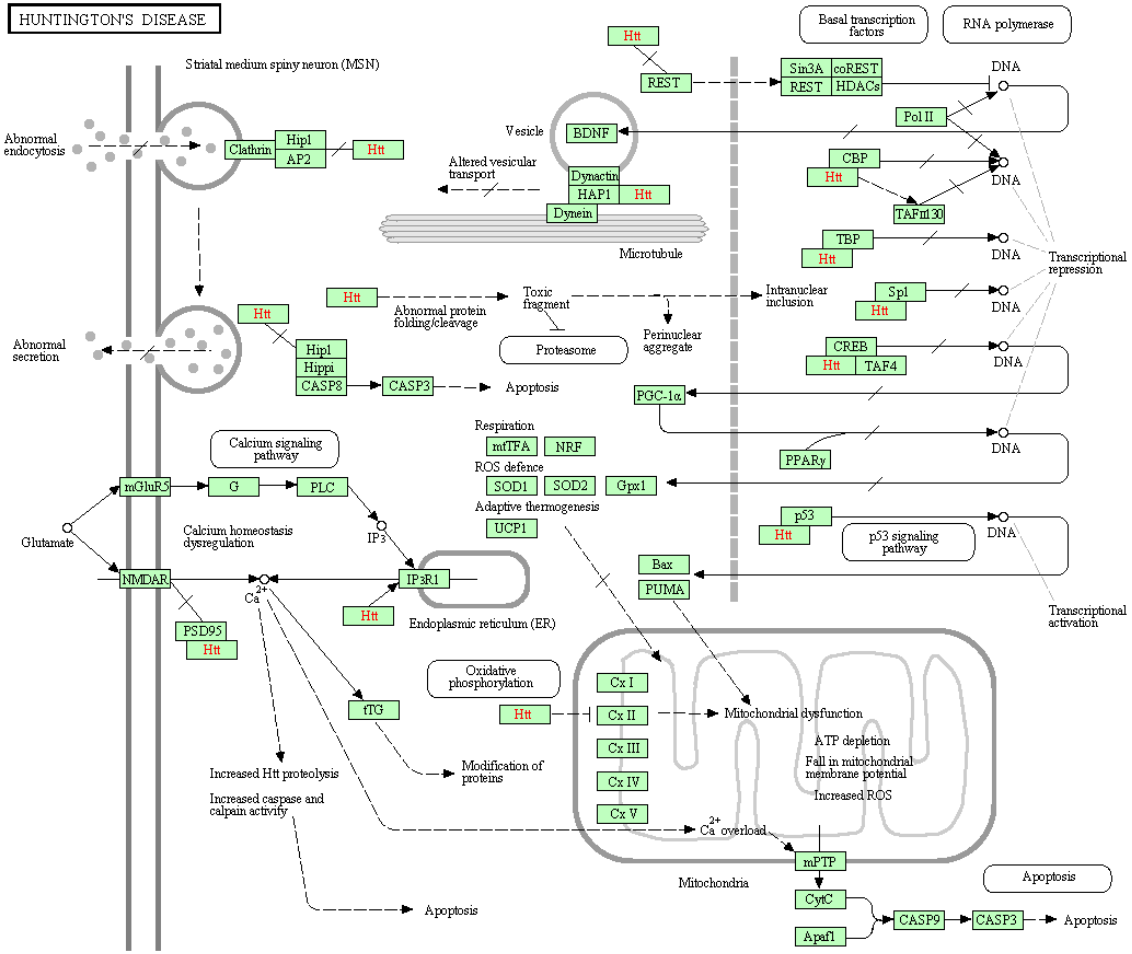


Huntington's Disease pathway – Kegg Database

Pathway Databases

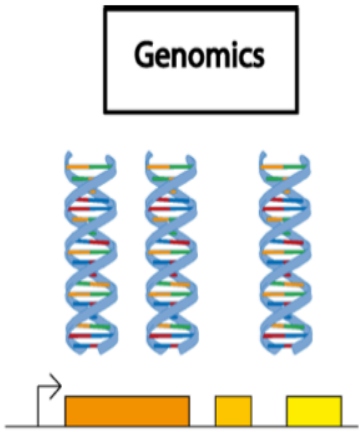
Main Problems:

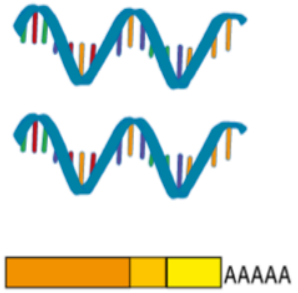
- Limited disease annotations
- Incomplete
- Noisy
- Static picture

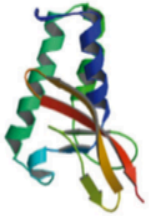


Huntington's Disease pathway – Kegg Database

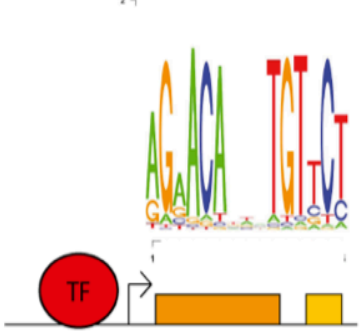
The OMICS revolution







> Replicase polyprotein 1ab, P19811
MATFSATGFG GSFVRDWSLD LPDACE-
HGAG LCCEVDGSTL CAECFRGCEG M
.....

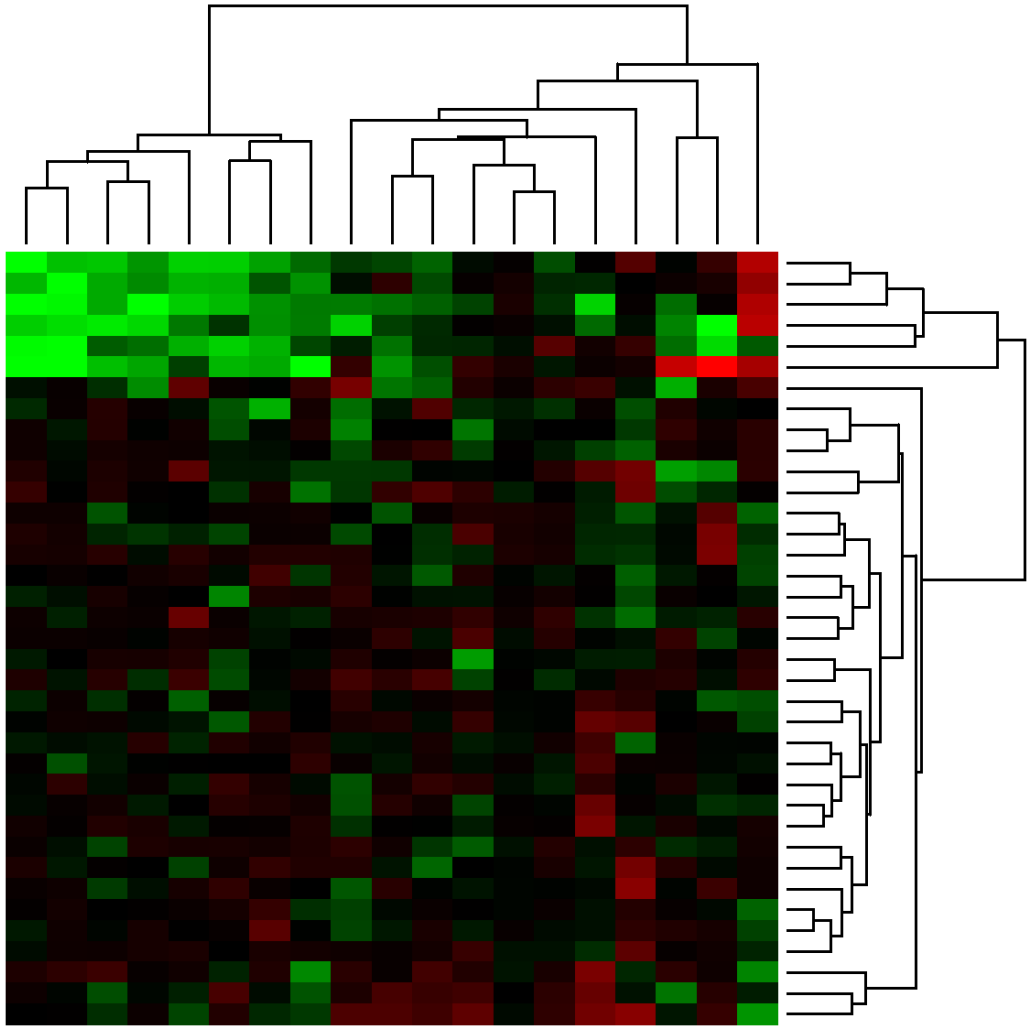


Genomics	Transriptomics	Proteomics	Epigenomics	Cistromics
- genomic DNA Sequencing - annotations	-microarray hybridization -RNA-seq	- liquid chromatography + - Mass Spectrometry	- ChIP-Seq - bisulfite seqencing	- ChIP-Seq
Genes sequences	mRNA sequences mRNA expression level	Protein sequences	Histones modifications DNA methylation	Genomic repertoire of the binding sequences of a transcriptome factor

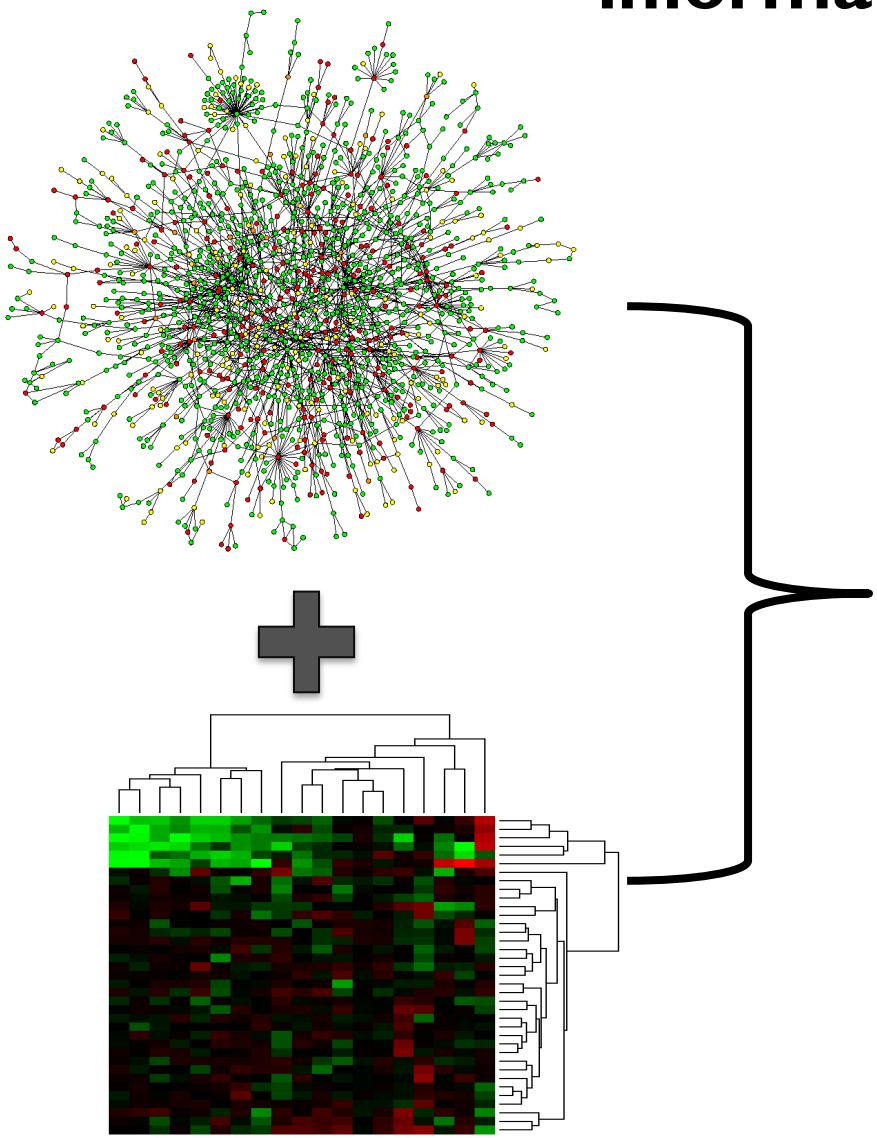
Measuring the activity of genes/proteins

Main Problems:

- Large number of genes/ proteins
- Small number of samples
- Noisy
- Standard analyses ignore effects of interactions



Introduction: Integrating OMICS and network information



Applications:

- Improve on shortcomings of single analysis
- Find novel potential biomarkers and drug targets
- Personalized medicine

BIOINFORMATICS

Vol. 18 Suppl. 1 2002
Pages S233–S240

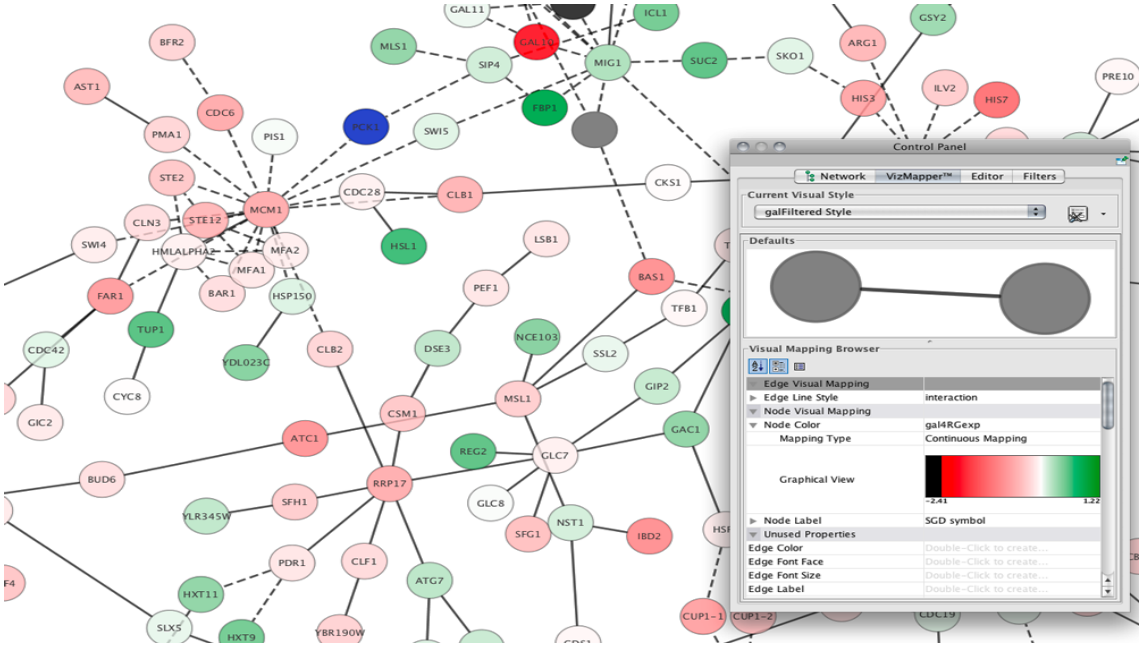
Discovering regulatory and signalling circuits in molecular interaction networks

Trey Ideker[1,*], Owen Ozier[1], Benno Schwikowski[2] and Andrew F. Siegel[2,3]

[1]Whitehead Institute for Biomedical Research, Cambridge, MA 02142, USA, [2]Institute for Systems Biology, Seattle, WA 98103, USA and [3]Departments of Management Science, Finance, Statistics, and Genome Sciences, University of Washington, Seattle, WA 98195, USA

Active Modules

- **Idea:** Given a biological network and a set of expression studies over one or multiple conditions, extract sub-networks that are differentially expressed ("active modules").



Active Modules

Step 1: Define an activity score for each individual gene:

1. Compute p-values p_i for each expression study.
2. Convert p-values to z-scores: $z_i = \Phi^{-1}(1 - p_i)$
   where Φ^{-1} is the inverse normal CDF

Step 2: Given a set A of k genes in a **single condition,** the aggregate score for the whole set is defined as:

$$z_A = \frac{1}{\sqrt{k}} \sum_{i \in A} z_i$$

Active Modules

Step 3: Given a set *A* of *k* genes in ***m* conditions**:

1. Compute z-scores for each condition ***m*** in gene set A: and sort them in descending order: $z_{A(1)} \ldots\ldots \geq z_{A(j)} \geq \ldots\ldots z_{A(m)}$
2. Probability that any single condition has z-score above $z_{A(j)}$:

$$P_z = 1 - \Phi(z_{A(j)})$$

3. Probability that at least *j* of the *m* conditions have scores above $z_{A(j)}$:

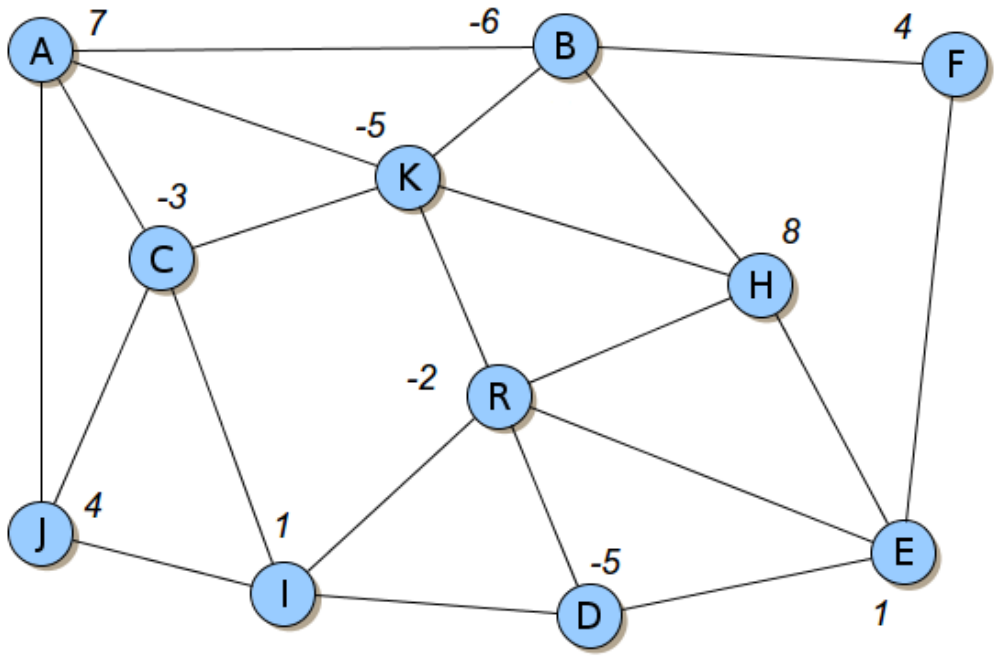
$$p_{A(j)} = \sum_{h=j}^{m} \binom{m}{h} (P_z)^h (1 - P_z)^{m-h}$$

4. Convert back to z-scores: $r_{A(j)} = 1 - \Phi^{-1}(p_{A(j)})$
5. Final aggregate score for the gene set:

$$r_A = max_j(r_{A(j)})$$

Active Modules

Goal: Find the set of genes *A* with maximal score r_A that induces a connected component in the network:



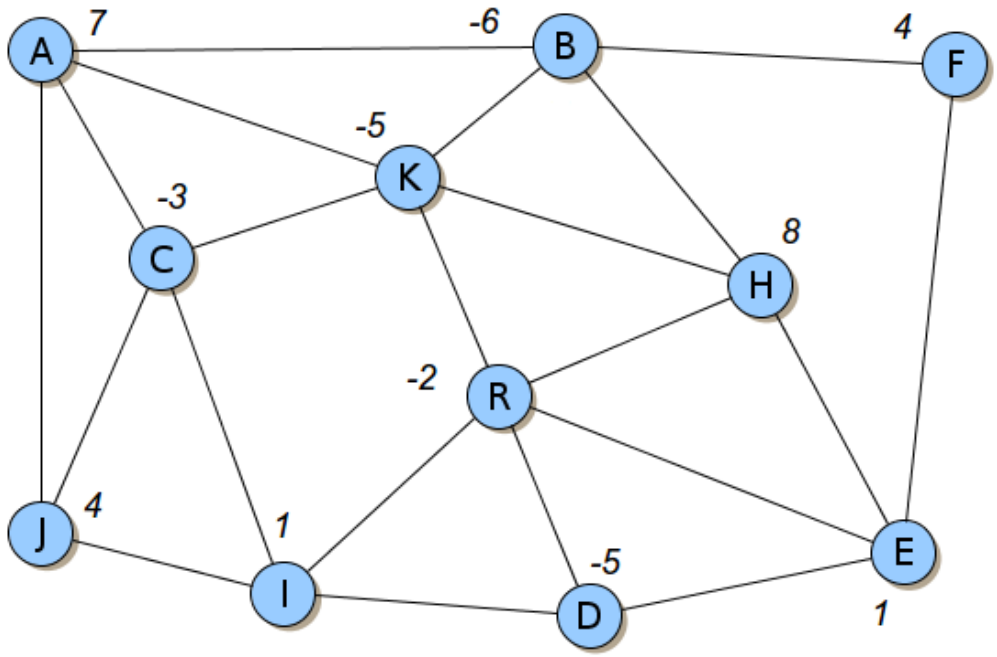
- Instance of the Maximum Weight Connected Subgraph problem: NP HARD !
- Solved with heuristics (Simulated Annealing)

Active Modules

Main problems:

- Relevant genes may not always show differential expression

  even if we find the optimal solution, this may not be the most biologically meaningful.
- Relies on statistics where we can assume data is normally distributed (ok for microarray data but not RNA-seq)



Detecting Disease-Specific Dysregulated Pathways Via Analysis of Clinical Expression Profiles

Igor Ulitsky[1], Richard M. Karp[2], and Ron Shamir[1]

[1] School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel
{ulitskyi,rshamir}@post.tau.ac.il.
[2] International Computer Science Institute, 1947 Center St., Berkeley, CA 94704
karp@icsi.berkeley.edu

Detecting dysregulated pathways

Idea:
Extract connected subnetworks but controlling the number of outliers allowed in each pathway (genes that are not differentially expressed in certain cases).

Given:

$G(V, E) \leftarrow$ graph representing the biological network

$C_{r \times n} \leftarrow$ matrix with expression studies for *r* genes and *n* samples/cases

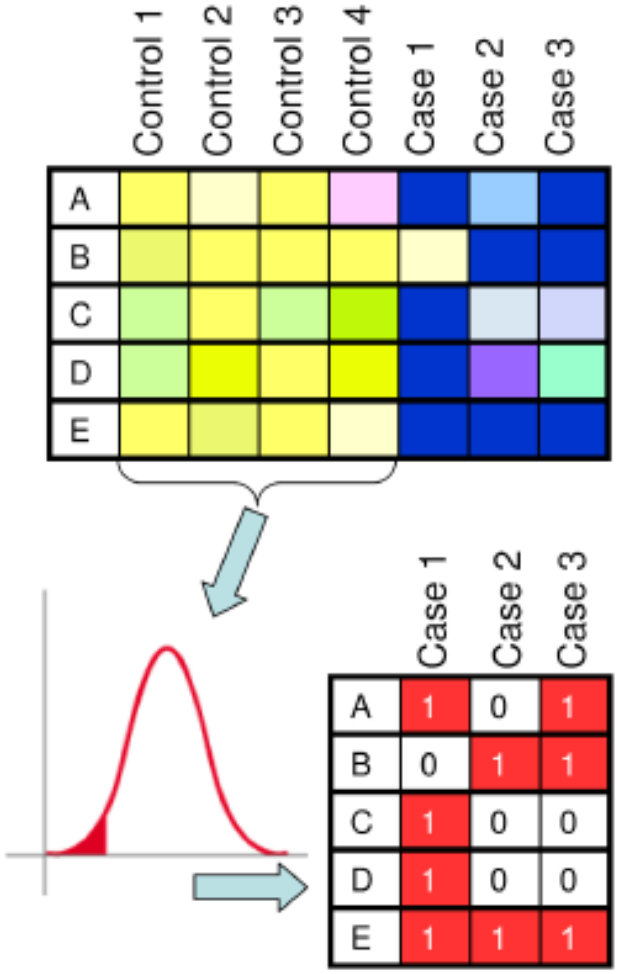
$k, l \leftarrow$ parameters to control the number of allowed outliers in the solution

Detecting dysregulated pathways

Data pre-processing

Convert expression matrix to indicator matrix:

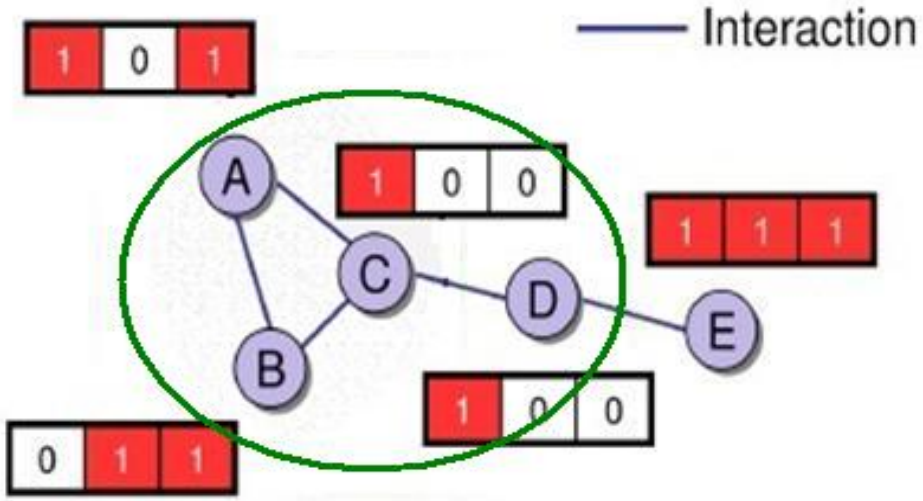
- If we have control cases, compute differential expression with preferred statistics (t-test, normal test, etc)
- Compute p-values, then select threshold for significance (usually 0.05 or 0.01).



Connected Cover

Definition:

A subset of genes/nodes $CC(k, l) \subseteq V$ is a (k, l)-connected cover if: **a)** it induces a connected component in G, and **b)** for each case but l there are at least k genes differentially expressed.

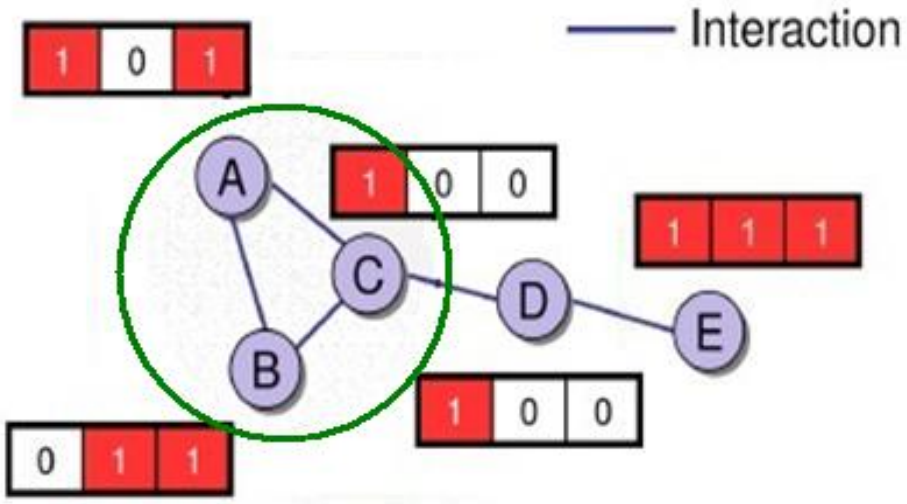


$\{A, B, C, D\}$ is a $CC(2, 1)$

Connected Cover

Goal:

Extract a **minimal** connected cover $MCC(k, l)$ (minimal w.r.t to the number of nodes):



$\{A, B, C\}$ is a $MCC(2, 1)$

Connected Cover

Computational complexity:

- If G is a complete graph then, it is equivalent to the Minimum Set Cover Problem: Given a set of elements $E = \{1, ...m\}$ and a list of sets $L = \{S_1, ..., S_n\}$ where each set $S \subset E$. Find a list $L' \subset L$ such that $\cup_{i \in L'} S_i = E$.
- Otherwise it is equivalent to the Minimum Connected Set Cover Problem, where each set S must induce a connected component in G.
- Both problems are NP-Hard

Covering Using Shortest Paths

Let $d(v, w)$ be the distance between node *v* and *w*. Given a set of root nodes $r_1, \ldots, r_s \in V$:

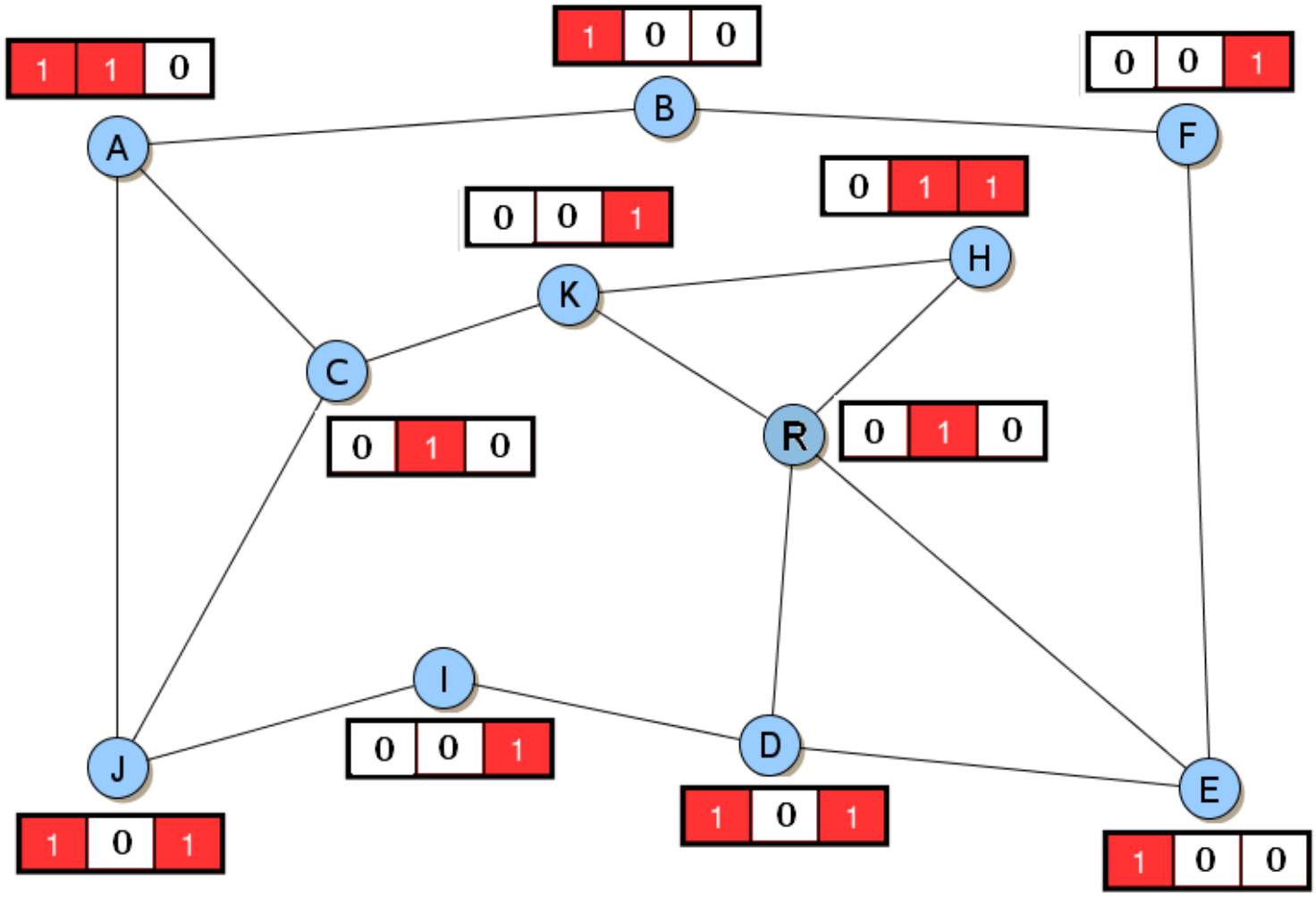
1. For each root node *r* , and for each case *u* compute:
   - Pointers $P[r, u]_1, \ldots., P[r, u]_k$
   - Distances $M[r, u]_1, \ldots., M[r, u]_k$

   to the *k* closest nodes to *r* that cover case *u*
2. Constuct X_r as the union of paths between root node *r* and the *k* closest nodes that cover *n* – *l* cases for $max_q\{M[r, u]_q, 1 \leq q \leq k\} \cdot \{P[r, u]_i\}$ are the smallest.
3. Compute the final solution as $X = argmin_v|X_v|$.

X_r is a tree of shortest paths where *n* – *l* cases are covered at least *k* times by the respective $\{P[r, u]_i\}$, thus it is a $CC(k, l)$

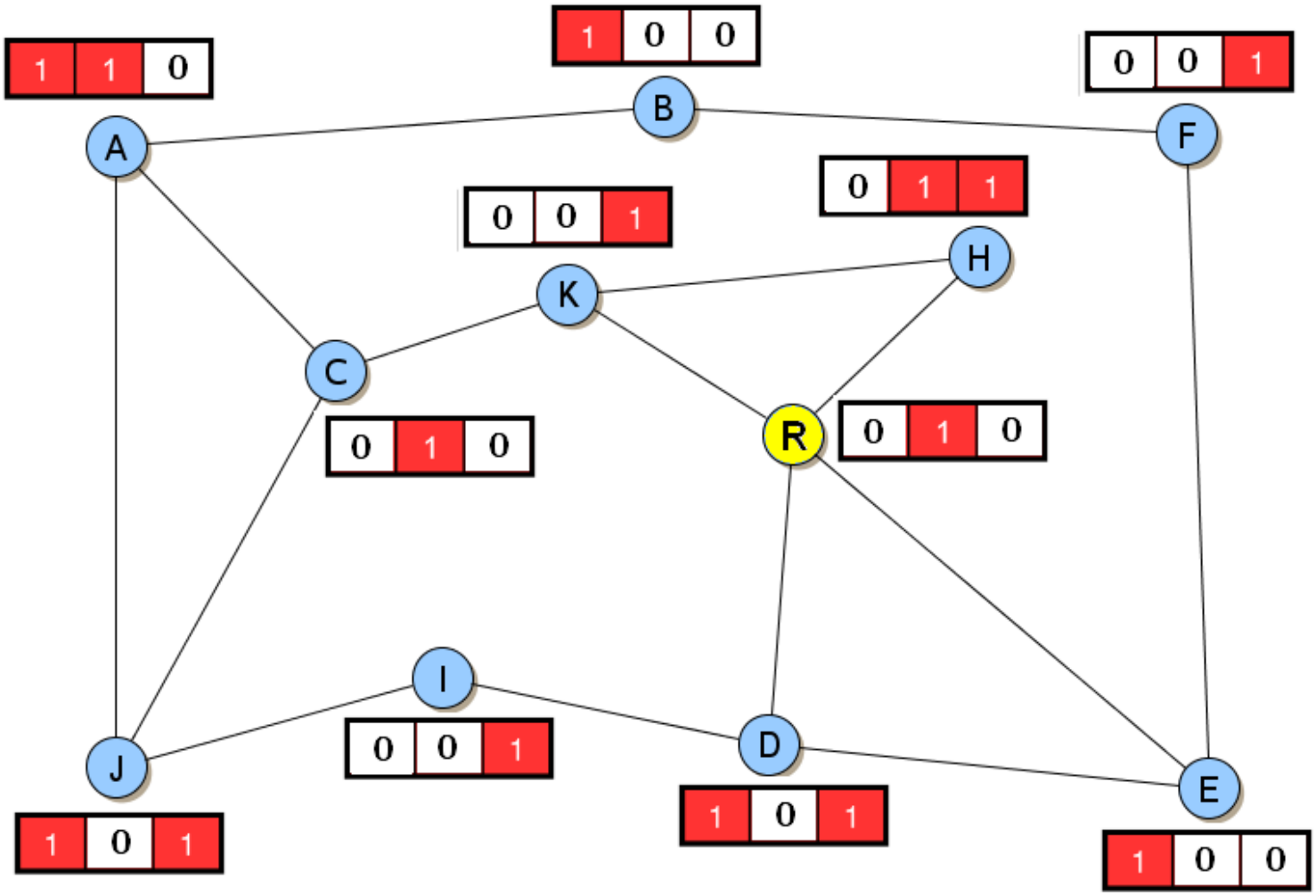
Covering Using Shortest Paths

Example for $k = 3, l = 1$



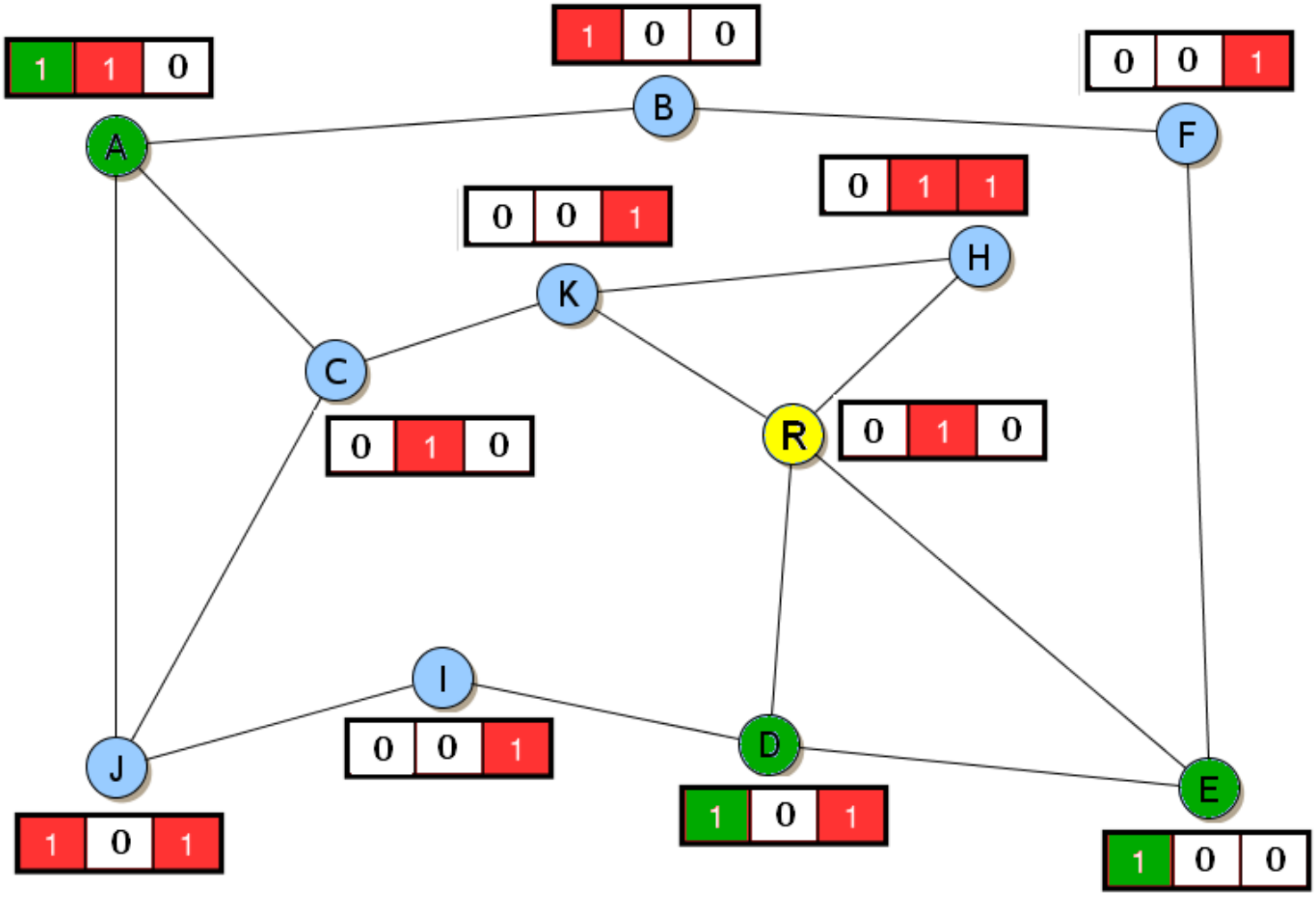
Covering Using Shortest Paths

Root node = *R*



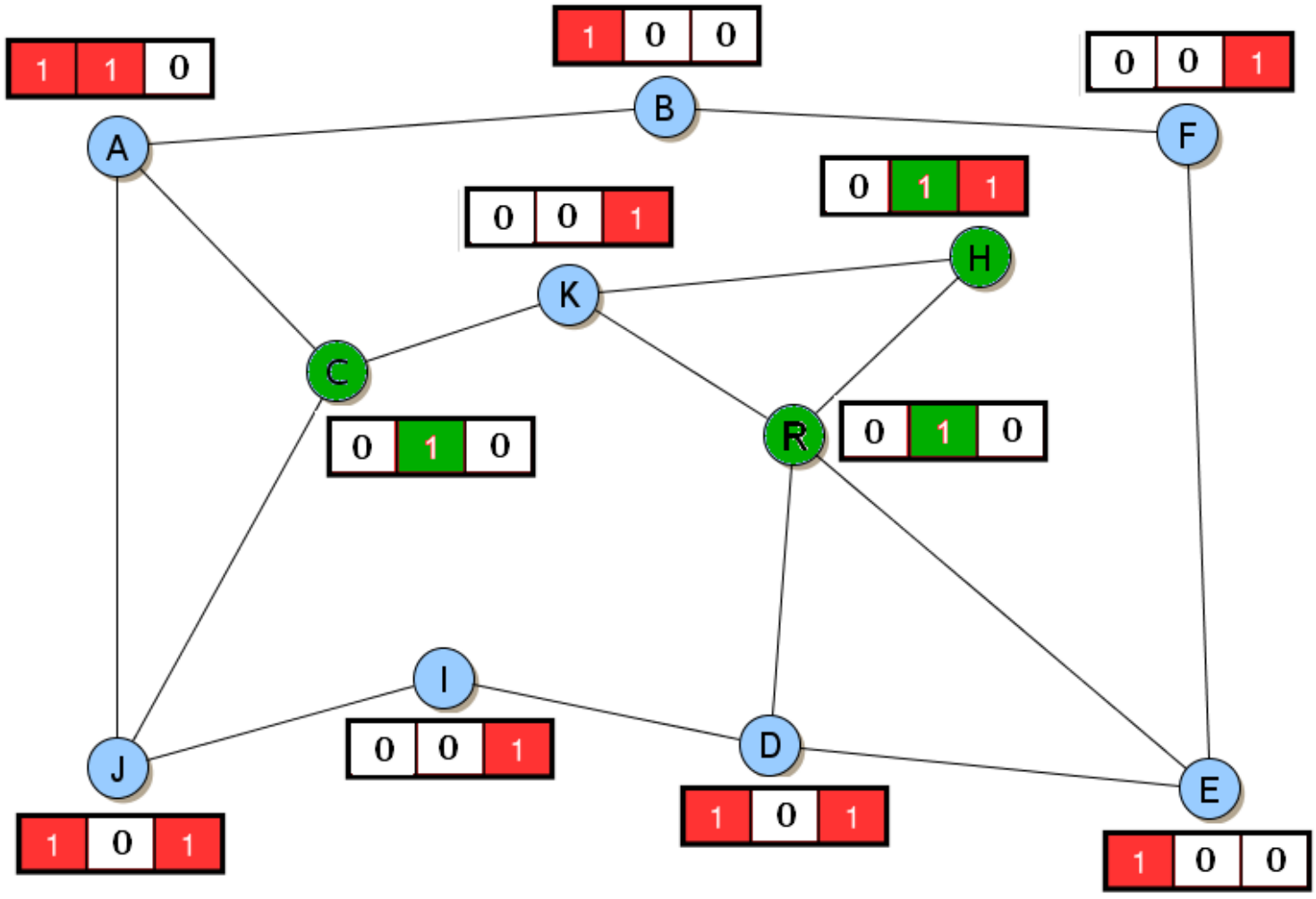
Covering Using Shortest Paths

Compute pointers to the *k* = *3* closest nodes that cover case *1*



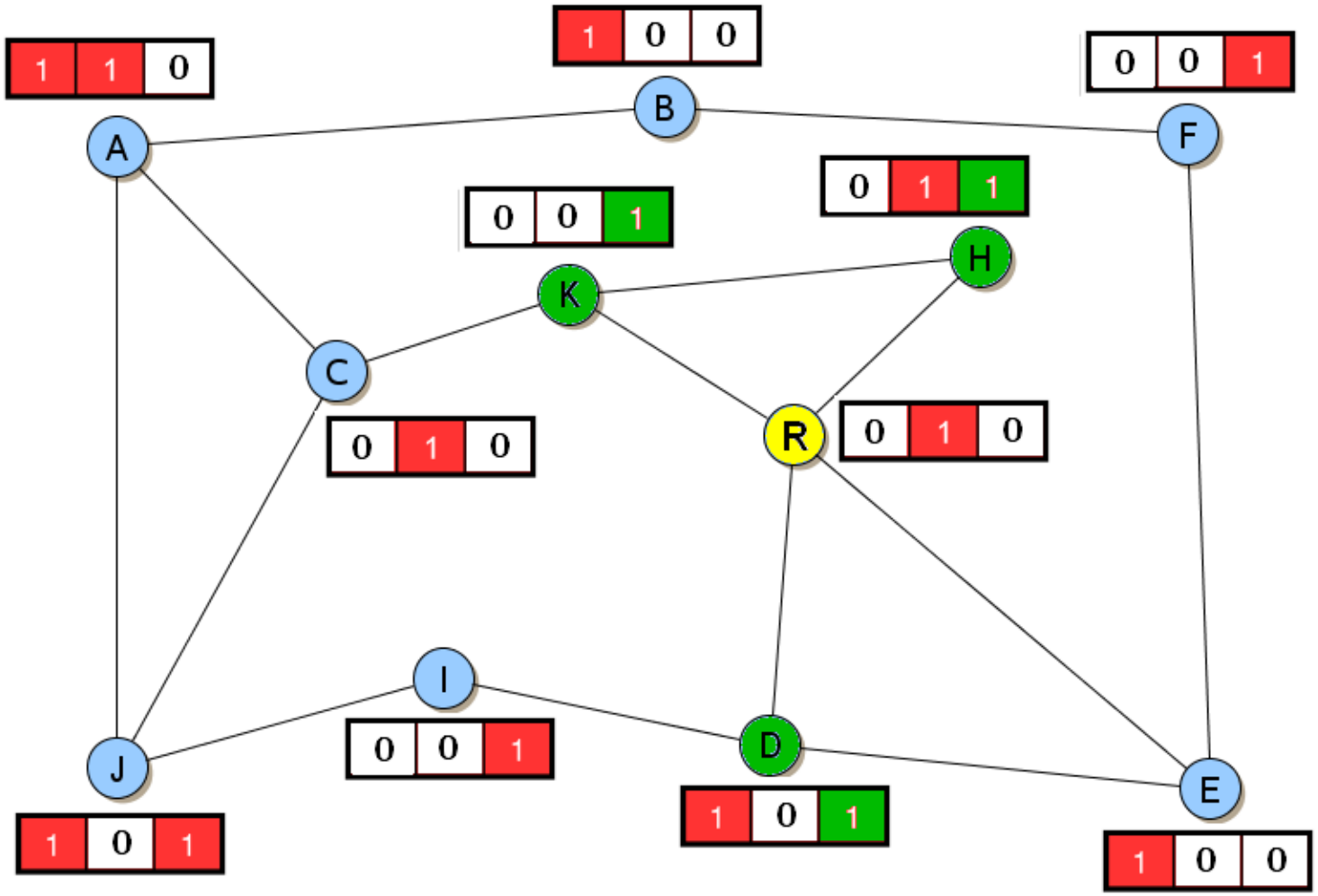
Covering Using Shortest Paths

Compute pointers to the *k* = *3* closest nodes that cover case *2*



Covering Using Shortest Paths

Compute pointers to the *k* = *3* closest nodes that cover case *3*



Covering Using Shortest Paths

Pointers and distances to *k* closest nodes to *R* that cover case u:

u \ k	1	2	3
1	D, d(R,D) = 1	E, d(R,E) = 1	A, d(R,A) = 3
2	R, d(R,R) = 0	H, d(R,H) = 1	C, d(R,C) = 2
3	D, d(R,D) = 1	H, d(R,H) = 1	K, d(R,K) = 1

Covering Using Shortest Paths

For each row, mark the entry with the longest distance:

u \ k	1	2	3
1	D, d(R,D) = 1	E, d(R,E) = 1	A, d(R,A) = 3
2	R, d(R,R) = 0	H, d(R,H) = 1	C, d(R,C) = 2
3	D, d(R,D) = 1	H, d(R,H) = 1	K, d(R,K) = 1

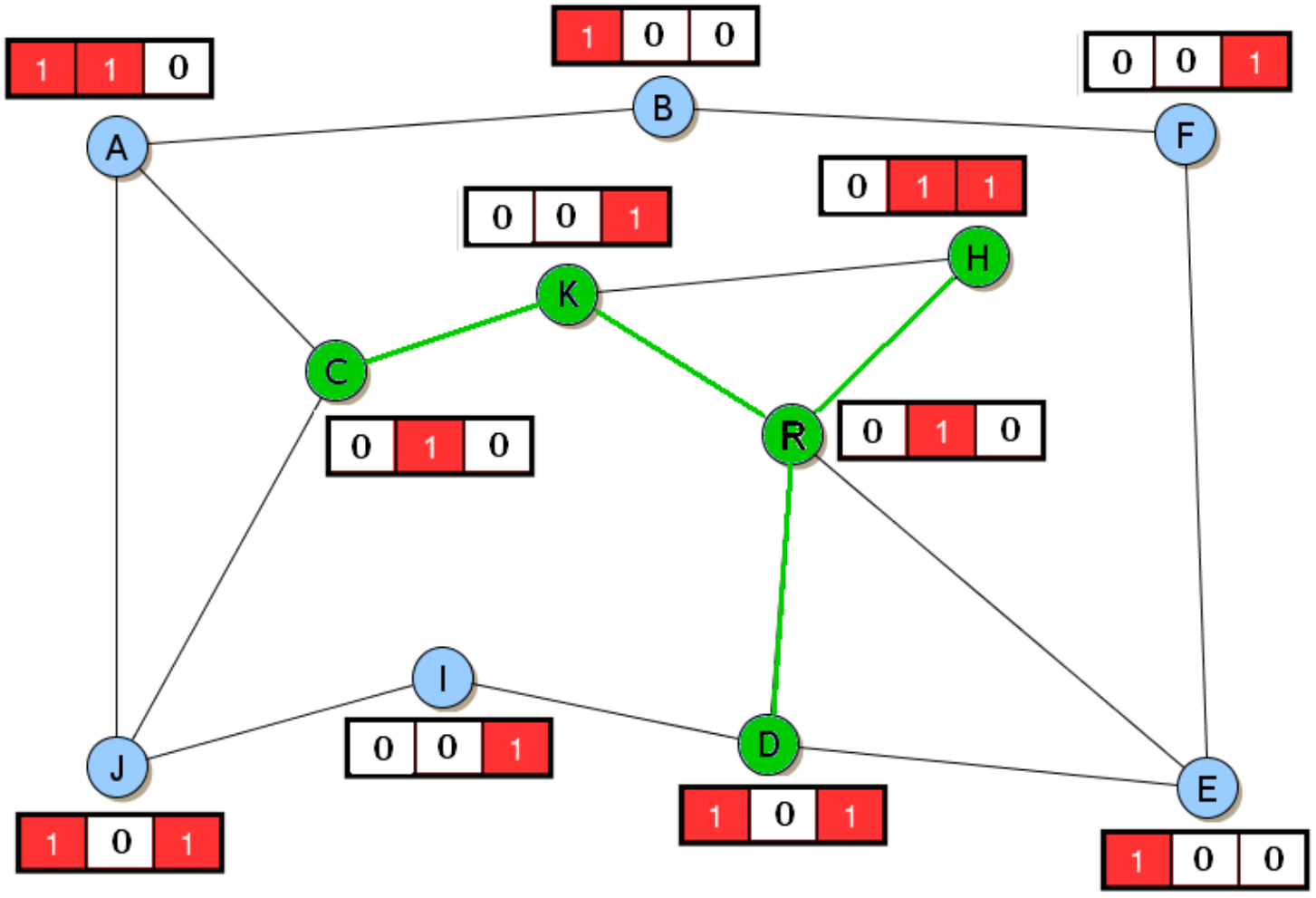
Covering Using Shortest Paths

Remove the $l = 1$ rows with the largest longest distances and construct the solution X_R with the remaining pointers:

u \ k	1	2	3
1	D, d(R,D) = 1	E, d(R,E) = 1	A, d(R,A) = 3
2	R, d(R,R) = 0	H, d(R,H) = 1	C, d(R,C) = 2
3	D, d(R,D) = 1	H, d(R,H) = 1	K, d(R,K) = 1

Covering Using Shortest Paths

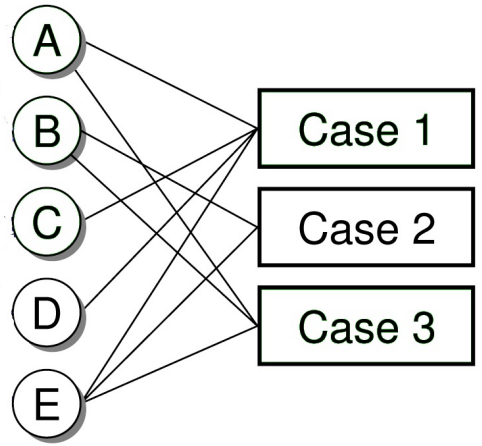
Solution for root node *R*: $X_R = \{R, K, H, C, D\}$



Covering Using Shortest Paths

Repeat for all root nodes v and report the **minimal** X_v as the final solution.

- Has two biologically meaningful parameters k, l
- Gives a provable $k(n-l)$-approximation to the optimal
- Runtime: $O(|V|(|V|+|E|+|E_B|))$ where B is a bipartite graph such that an edge $(v,u) \in E_B$ exists if gene v is differentially expressed / active in case u



Covering Using Shortest Paths

Main drawbacks:

- A solution for given parameters *k, l* may not necessarily exist (no way to know this beforehand)
- Assumption that affected biological pathways are compact (i.e. have low diameter).

KeyPathwayMiner: Detecting Case-Specific Biological Pathways Using Expression Data

Nicolas Alcaraz [a] , Hande Kücük [a] , Jochen Weile [b] , Anil Wipat [b] & Jan Baumbach [a]

[a] Saarland University, Cluster of Excellence for Multimodal Computing and Interaction, Max Planck Institute for Informatics , Campus E2.1, 66123, Saarbrücken, Germany

[b] Newcastle University, School of Computing Science , Newcastle upon Tyne, NE1 7RU, United Kingdom

Published online: 30 Nov 2011.

KeyPathwayMiner: Individual Nodes Exceptions (INEs) model

Given: $G(V, E)$, $C_{r \times n}$ and parameters *k, l*

Definition: Let $R(v)$ be the number of differentially expressed cases in node *v*. A set of vertices $D(k, l) \subseteq V$ is one such that:

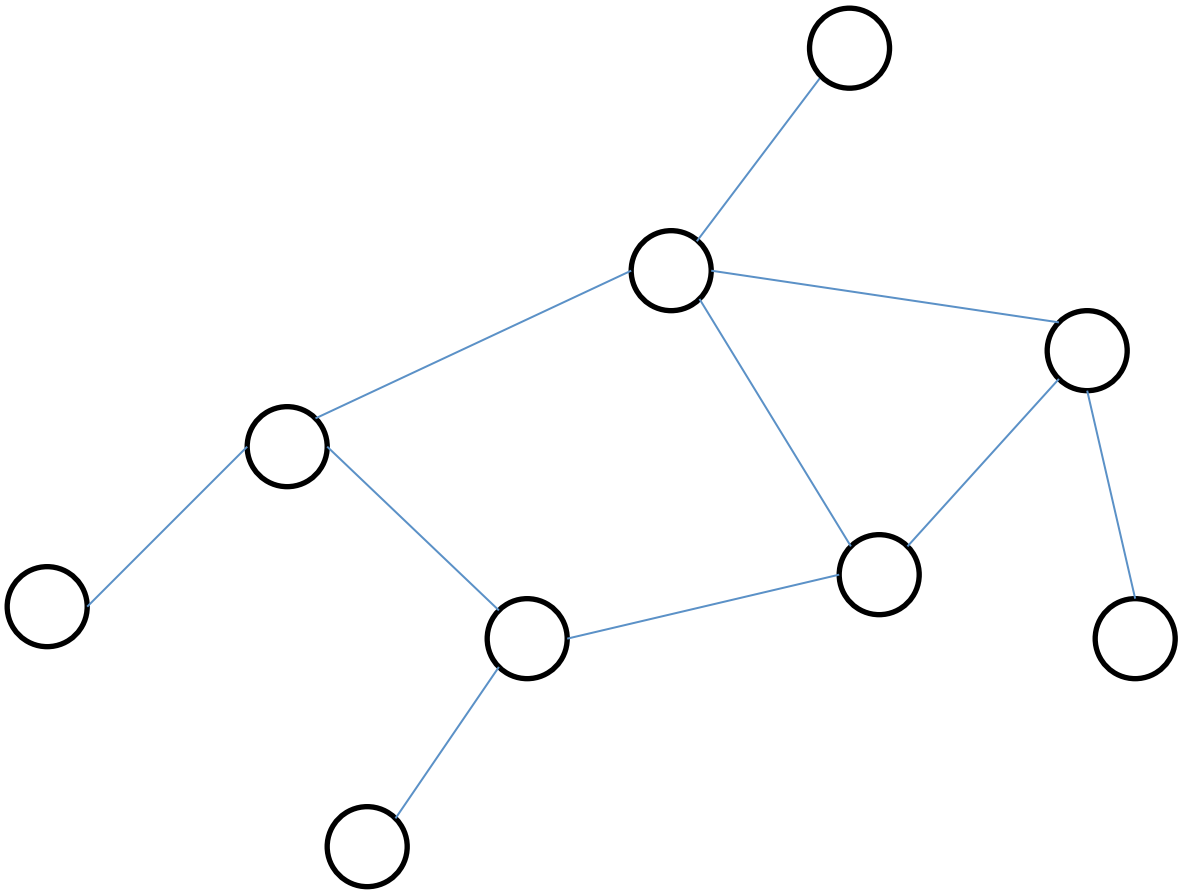
- Induces a **connected component** in G
- Satisfies: $|\{v \in D(k, l) | R(v) < n - l\}| \leq k$

  (**at most** *k* genes in D are **not** differentially expressed in more than *l* cases)

Goal: Extract all (or up to a user defined limit) maximal sets $D(k, l)$

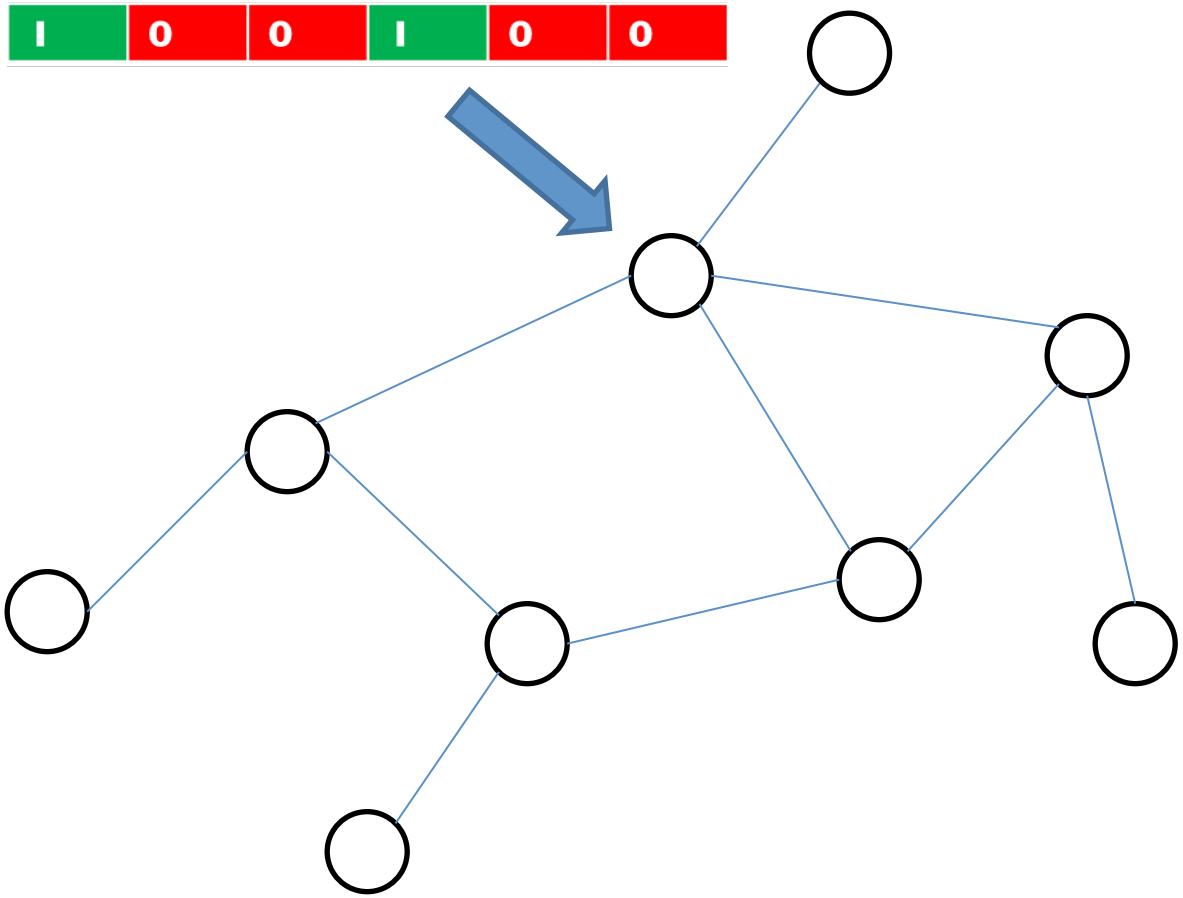
KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



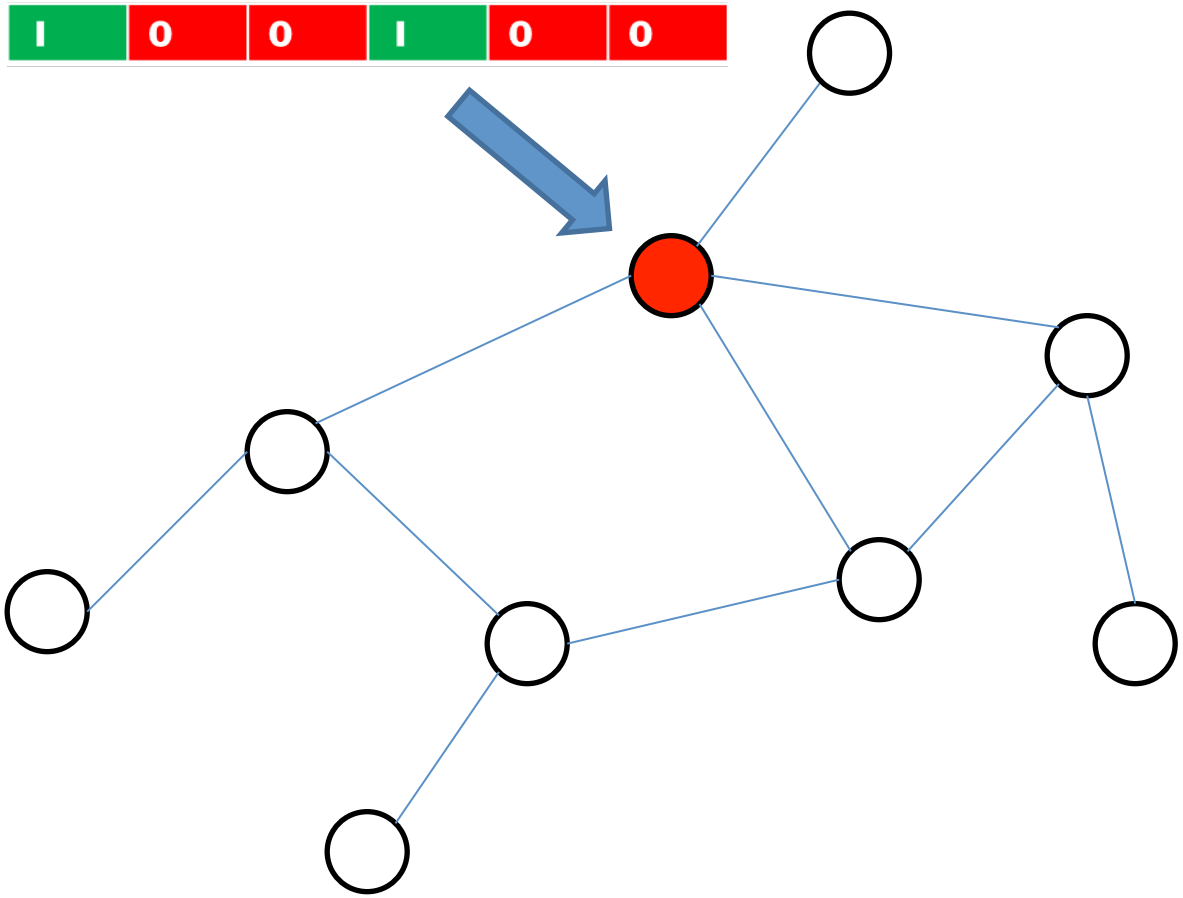
KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



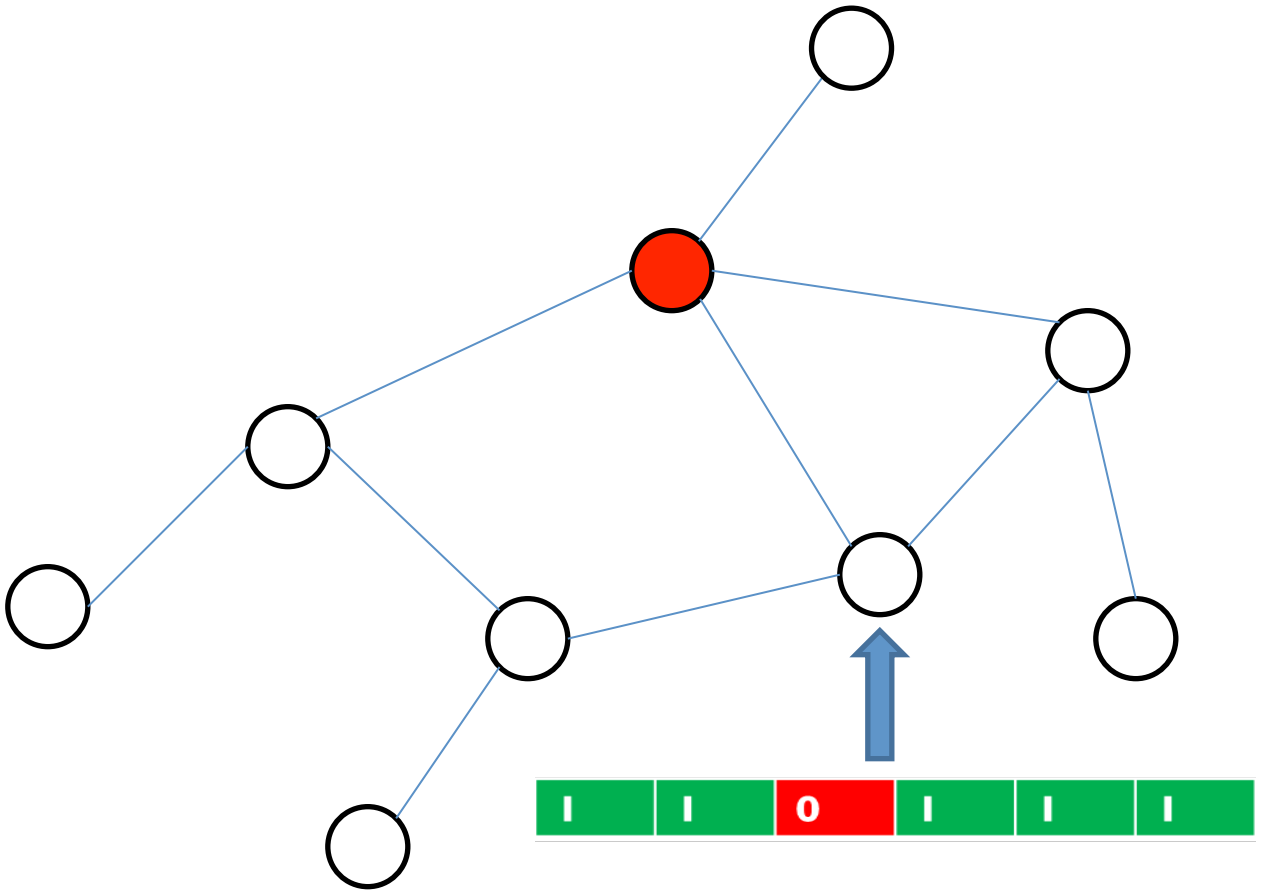
KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



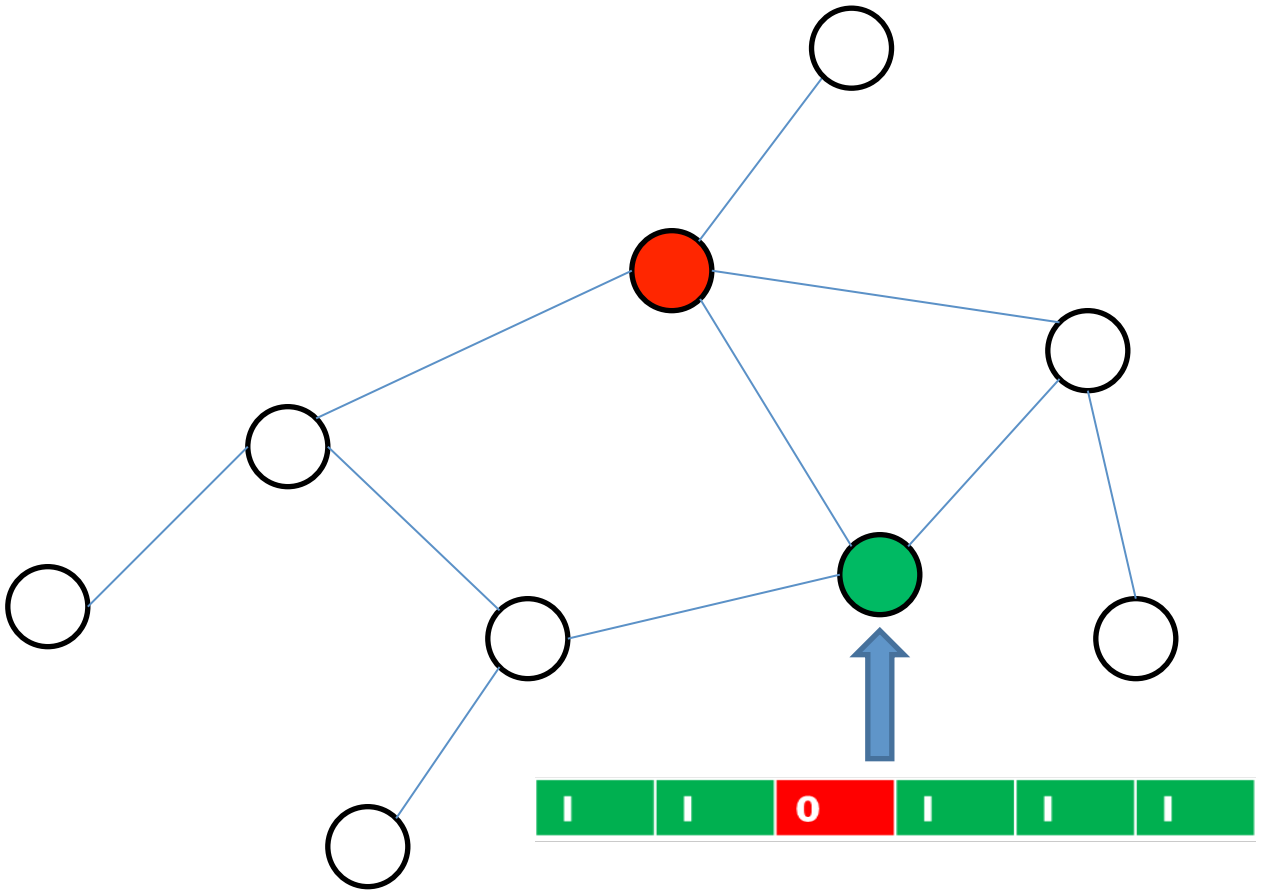
KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



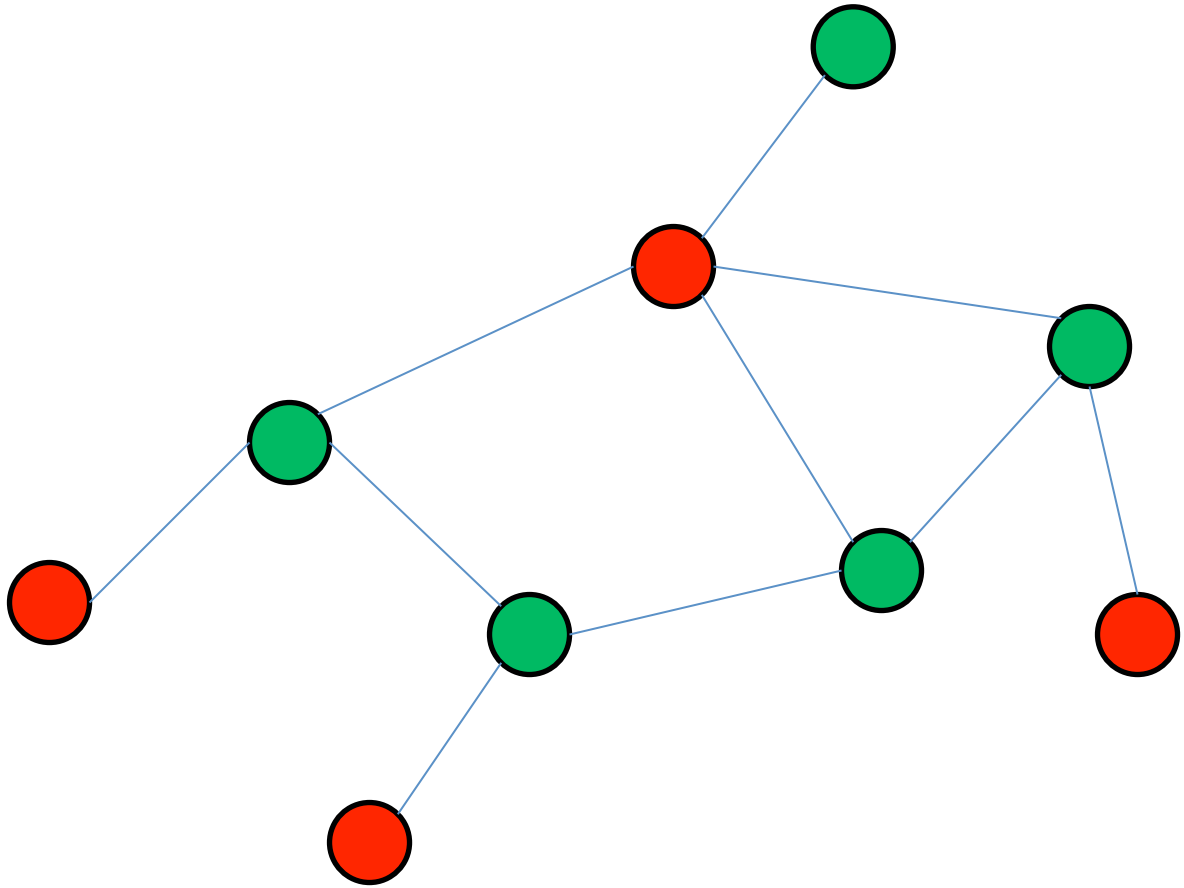
KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



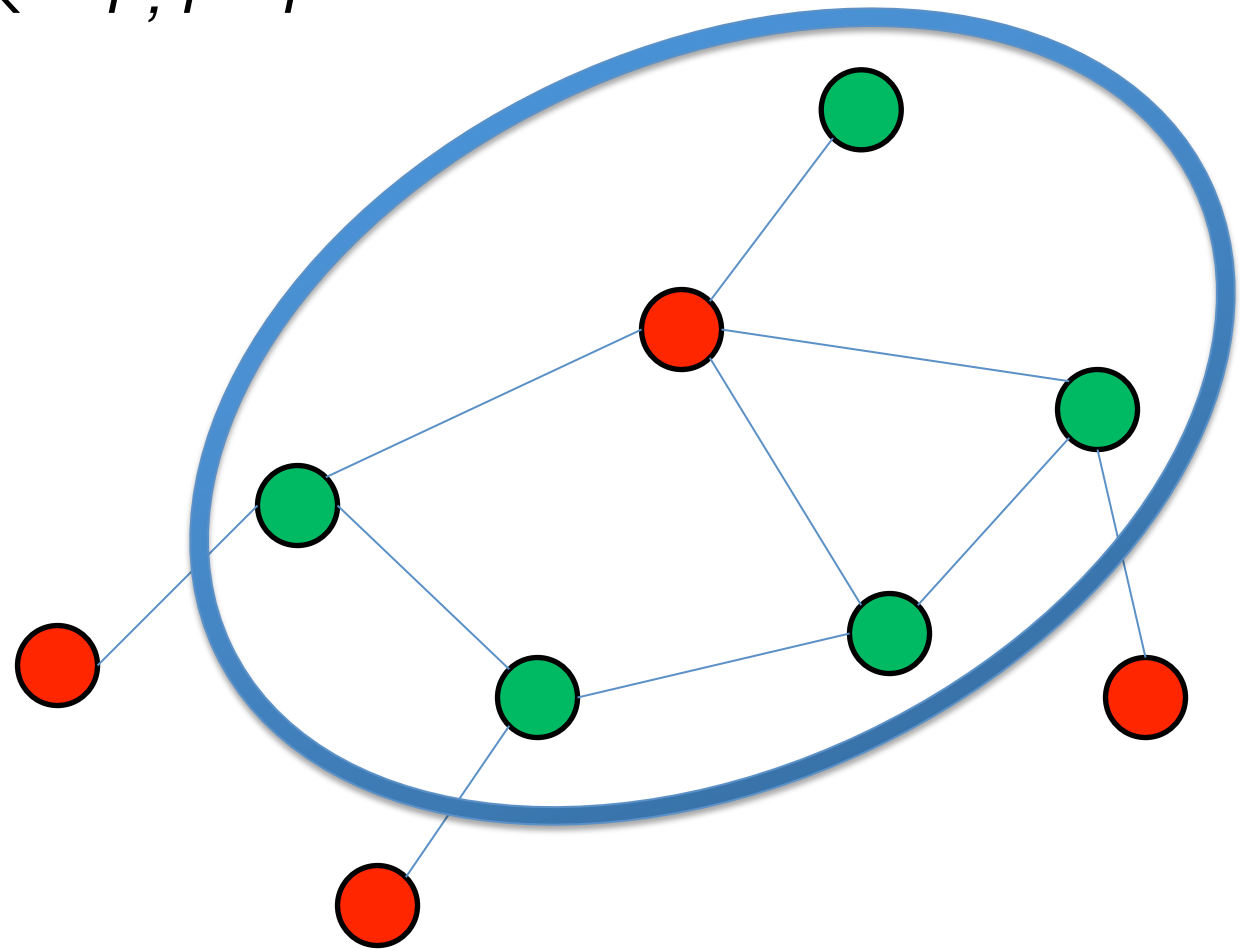
KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



KeyPathwayMiner: INEs model

Example: *k = 1 , l = 1*



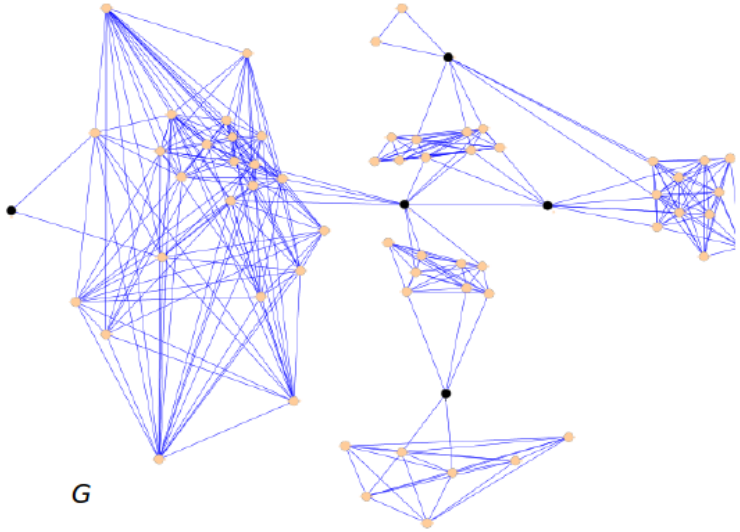
KeyPathwayMiner: INEs model

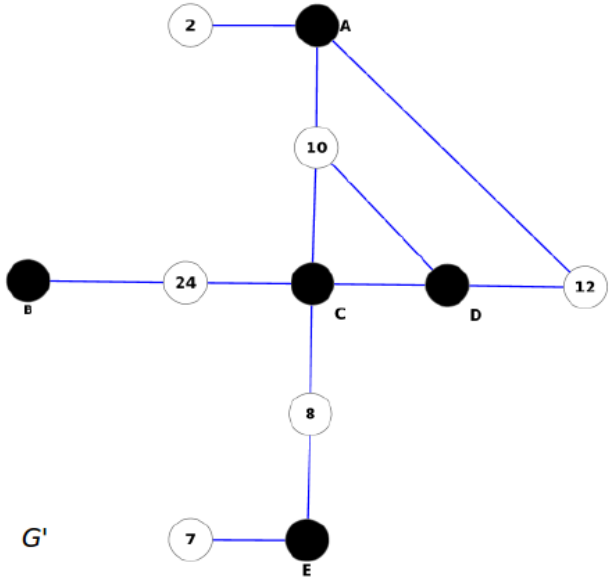
- Also two parameters *(k, l)*, but easier to interpret than CUSP
- As long as there is at least one node *v* satisfying $n - l \leq R(v)$ (easy to compute beforehand), a solution will always exist.
- Can be solved exactly for small values of *k*

KeyPathwayMiner: INEs model

Complexity: Extracting maximal $D(k, l)$ sets has complexity $O(|V|^k)$

- Construct a graph *G'* by adding all exception nodes $\{v \in V | R(v) < n - l\}$ and all edges connecting them.
- Collapse all non-exception nodes that are connected in *G* into a single node v'. Connect v' to every exception node in *G'* that is connected at least one of the non-exception nodes of v' in *G.*
- All exception nodes are a vertex cover in *G'*

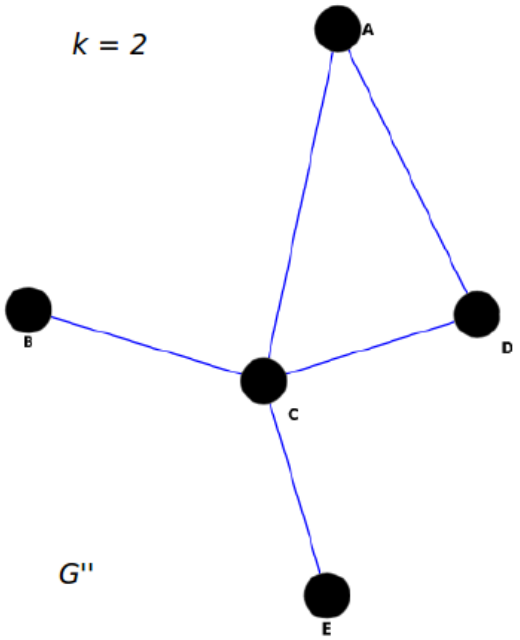


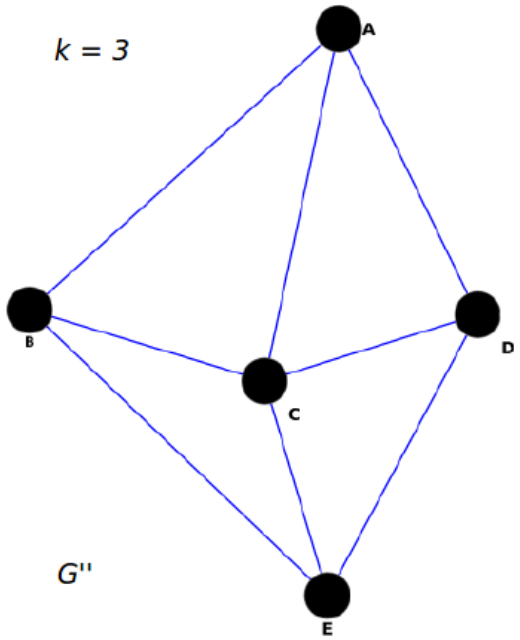


KeyPathwayMiner: INEs model

Complexity: Extracting maximal $D(k, l)$ sets has complexity $O(|V|^k)$

- Construct a graph *G"* of all exception nodes and add and edge between two exception nodes if there exists a path containing at most *k* – 2 connecting them.
- Extracting all maximal $D(k, l)$ sets in *G* is equivalent to extracting all paths (no cycles) of length *k* in *G"*





KeyPathwayMiner: Global Node Exceptions (GloNE) model

Given: $G(V, E)$, $C_{r \times n}$ and parameter L

Definition: Let $I(v)$ be the number of differentially expressed cases in node *v*. A set of vertices $D(L) \subseteq V$ is one such that:

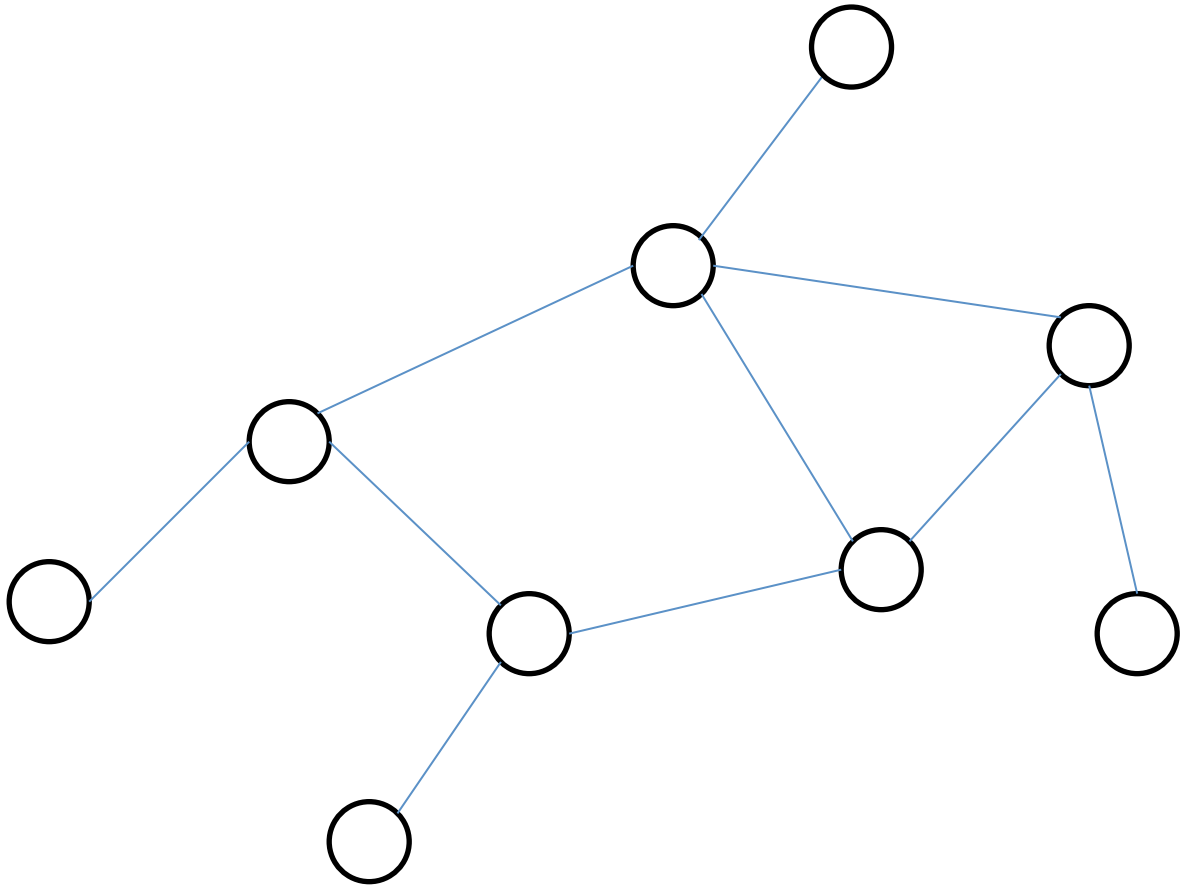
- Induces a **connected component** in G
- Satisfies: $\sum_{v \in D} I(v) \leq L$

(the sum of all **non differentially** expressed cases is **at most** *L*)

Goal: Extract all (or up to a user defined limit) maximal sets $D(L)$

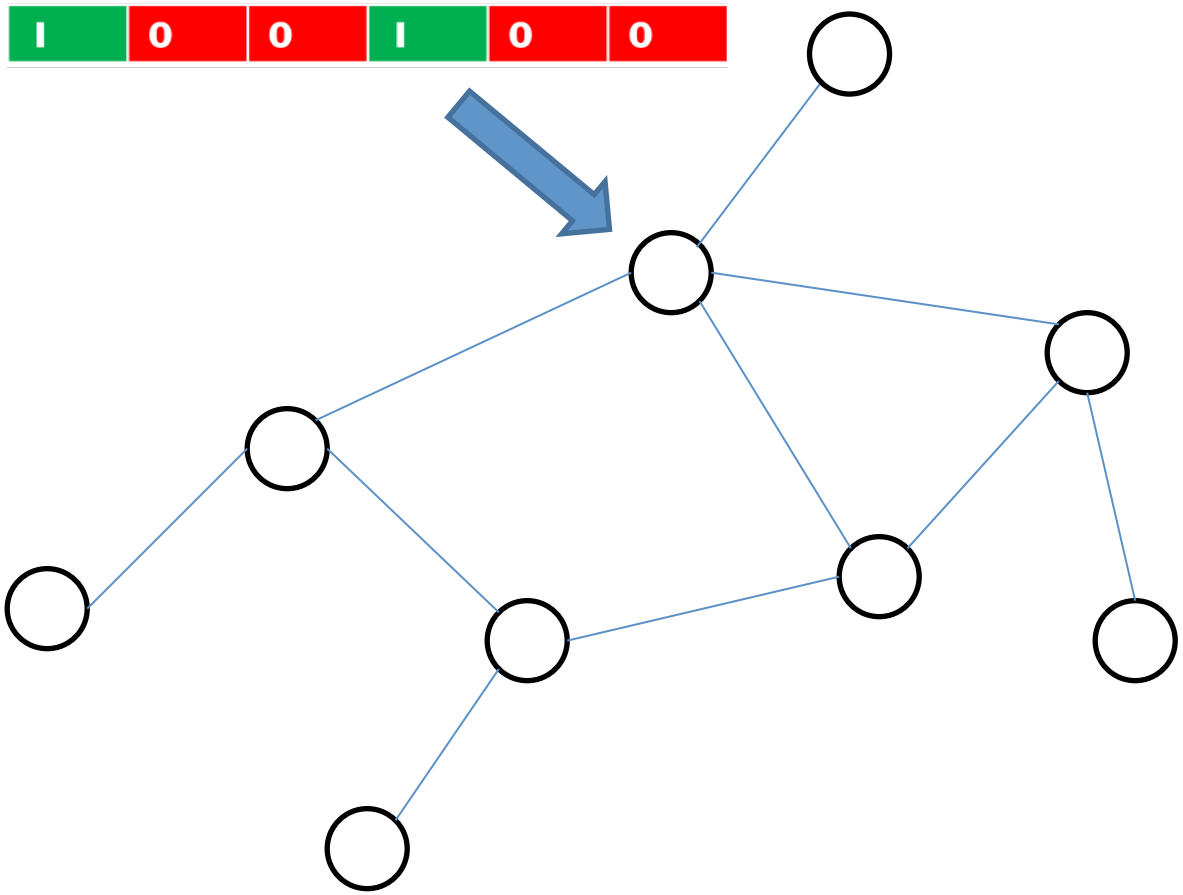
KeyPathwayMiner: GloNE model

Example: *L = 12*



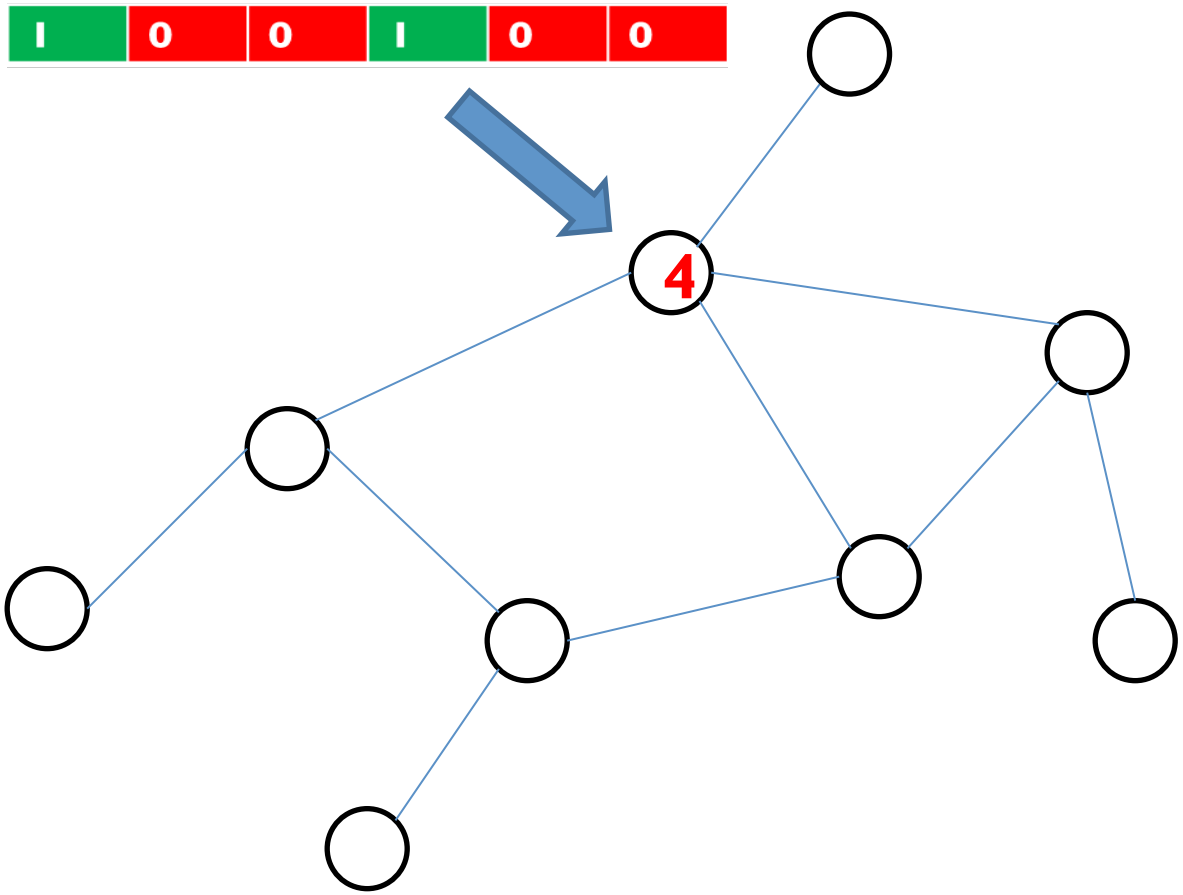
KeyPathwayMiner: GloNE model

Example: *L = 12*



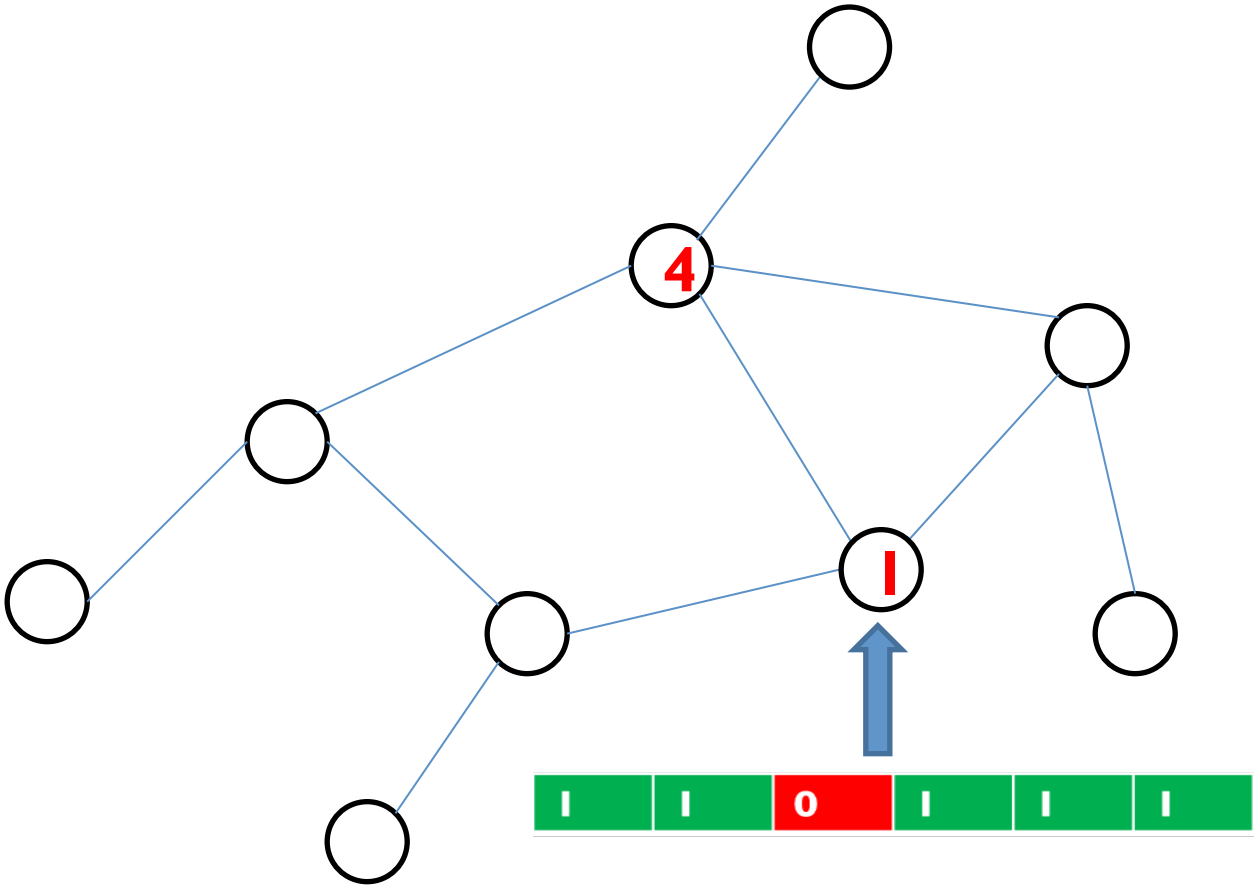
KeyPathwayMiner: GloNE model

Example: *L = 12*



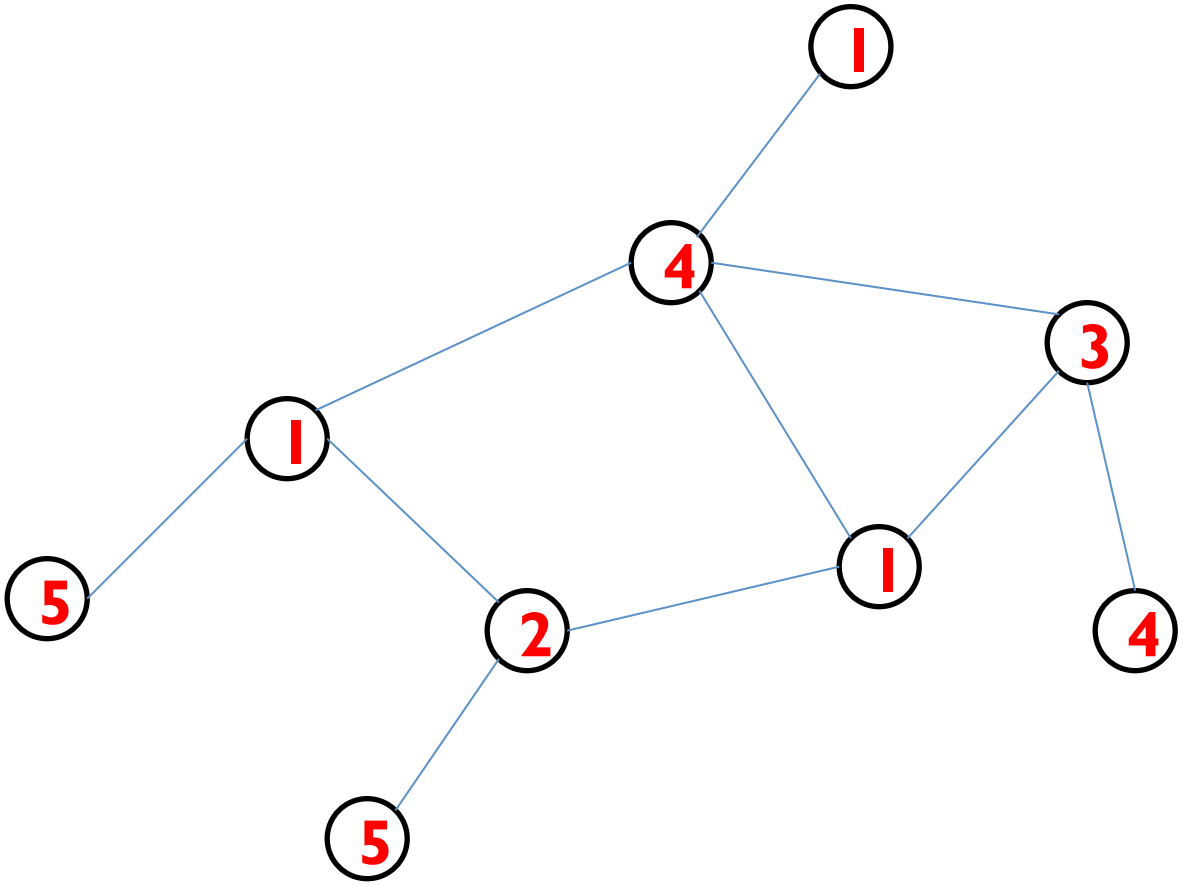
KeyPathwayMiner: GloNE model

Example: *L = 12*



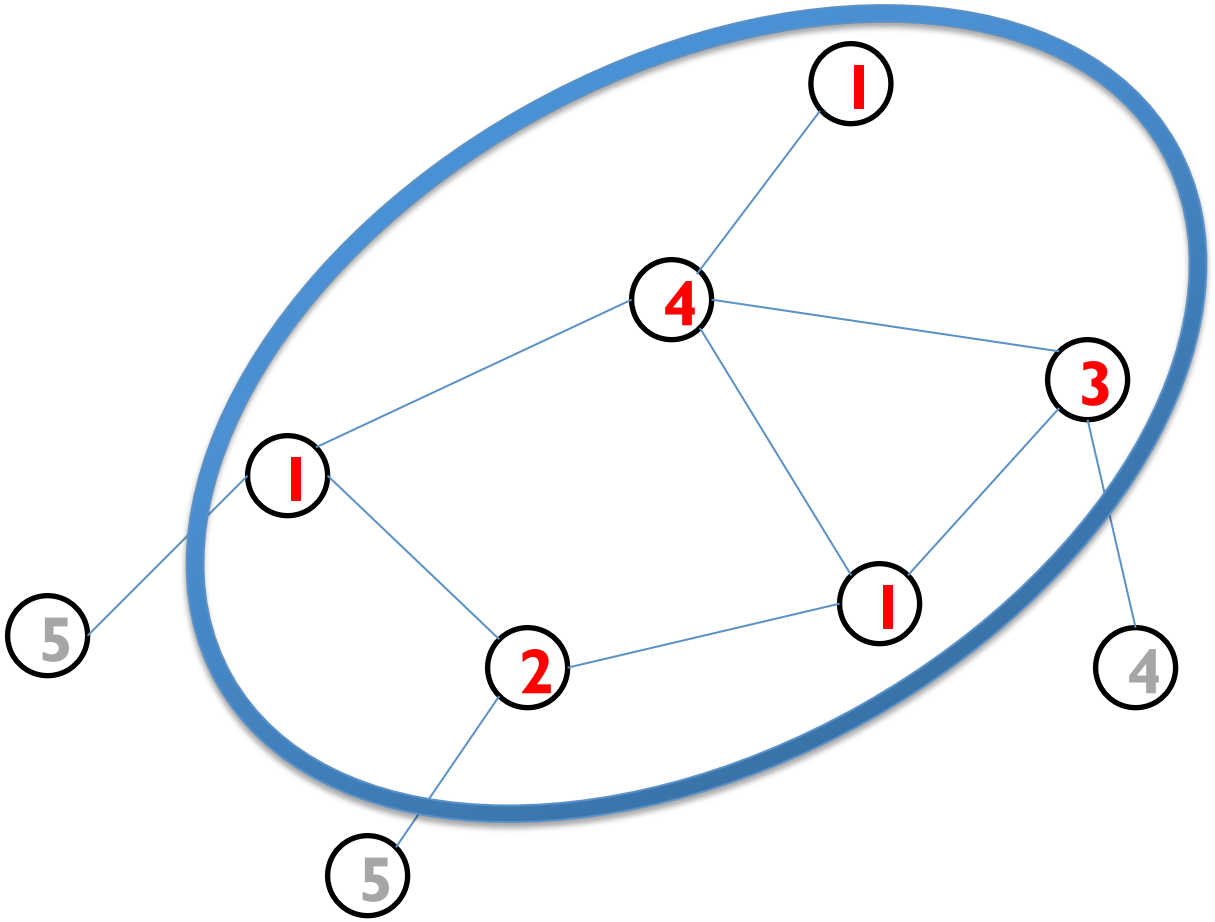
KeyPathwayMiner: GloNE model

Example: *L = 12*



KeyPathwayMiner: GloNE model

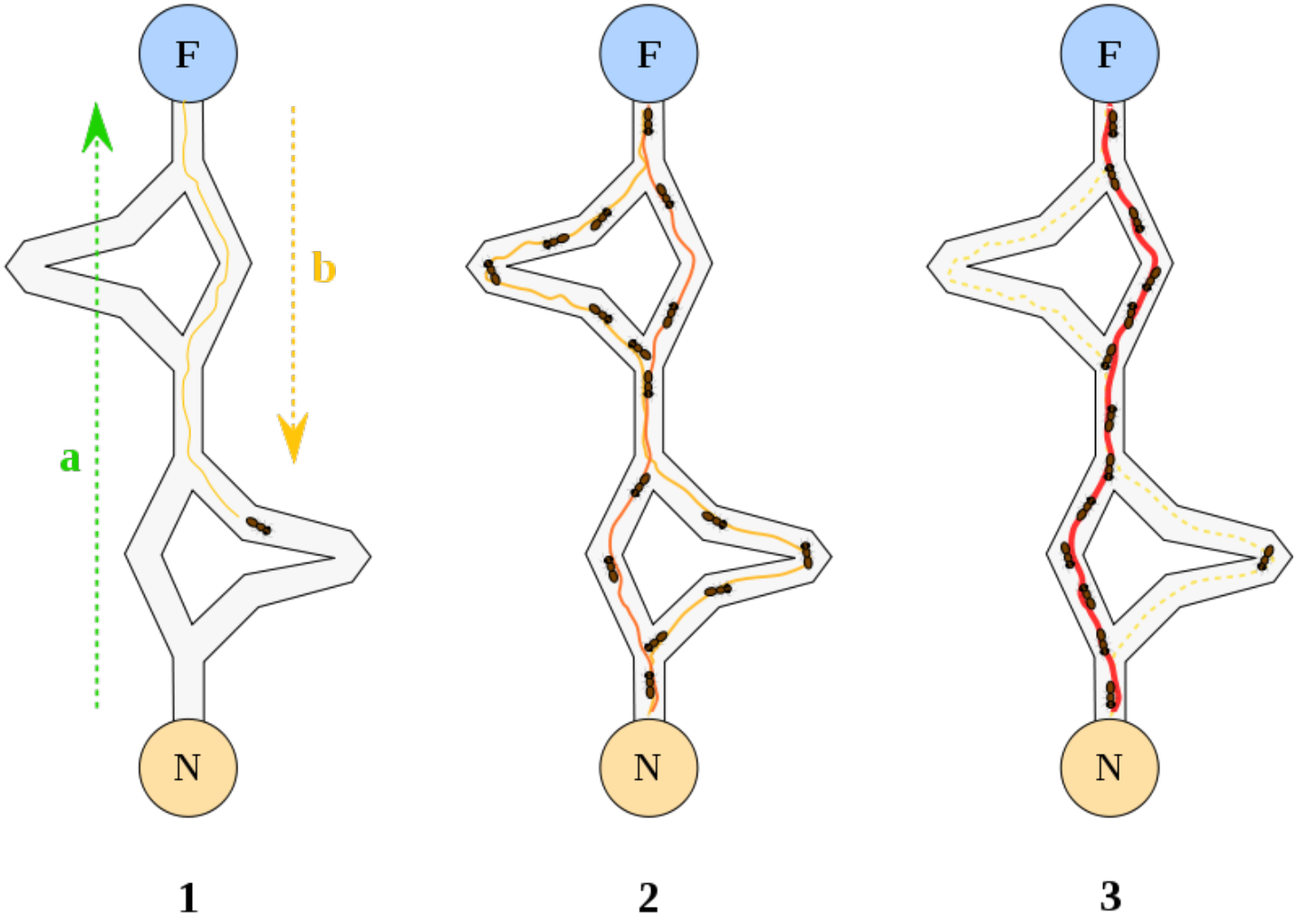
Solution for *L = 12*



KeyPathwayMiner: GloNE

- Complexity is NP-Hard (equivalent to the Maximum Weight Connected Subgraph Problem)
- Has less bias towards "hub" nodes compared to INEs
- Only one parameter *L*
- Can be solved exactly for small values of *L*

Ant Colony Optimization



ACO in a Nutshell

- A set *S* of ants are randomly placed on the working space, in this case a graph *G*
- Each ant tries to construct a valid solution to the problem by performing random walks
- The probability that an ant standing on vertex *u* moves to vertex *v* is:

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

where:

η_{uv} $\leftarrow$ heuristic value of edge *(u,v)*

τ_{uv} $\leftarrow$ current pheromone value of edge *(u,v)*

α, β $\leftarrow$ parameters to control the influence of η and τ

ACO in a Nutshell

- Once an ant s has successfully constructed a valid solution to the problem, it can update the pheromone value of the edges it visited as following:

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

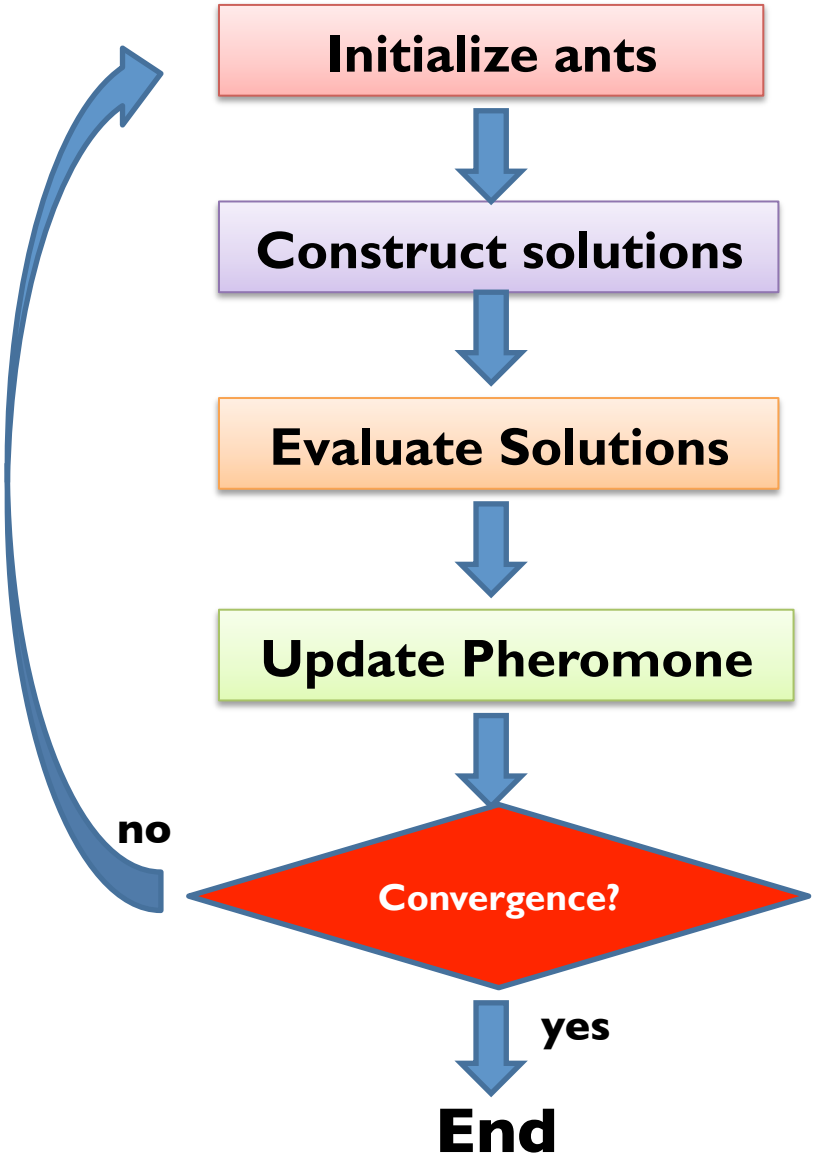
where:

ρ $\leftarrow$ pheromone evaporation rate (used to avoid fast convergence to suboptimal solutions).

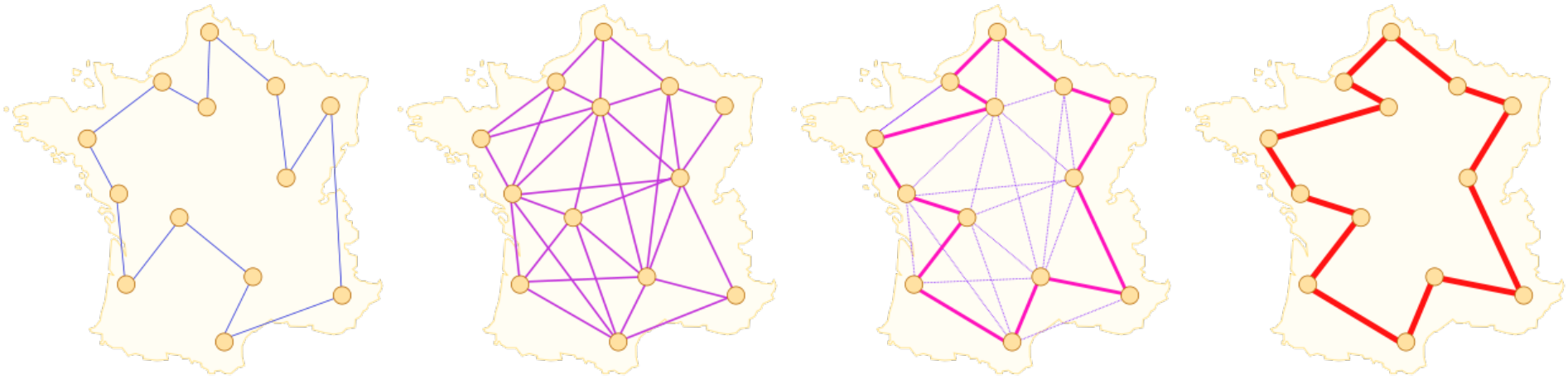
τ_{uv}^{s} $\leftarrow$ pheromone update function (should be proportional to the quality of the solution constructed by ant *s*).

ACO in a Nutshell

- Ant are initialized: randomly or in potentially "good" starting points.
- Each iteration ends until all ants have reported a solution
- Pheromone update can be done by all ants or only the best one (different ACO strategies)
- Repeat until the best solution has not improved after a certain amount of iterations



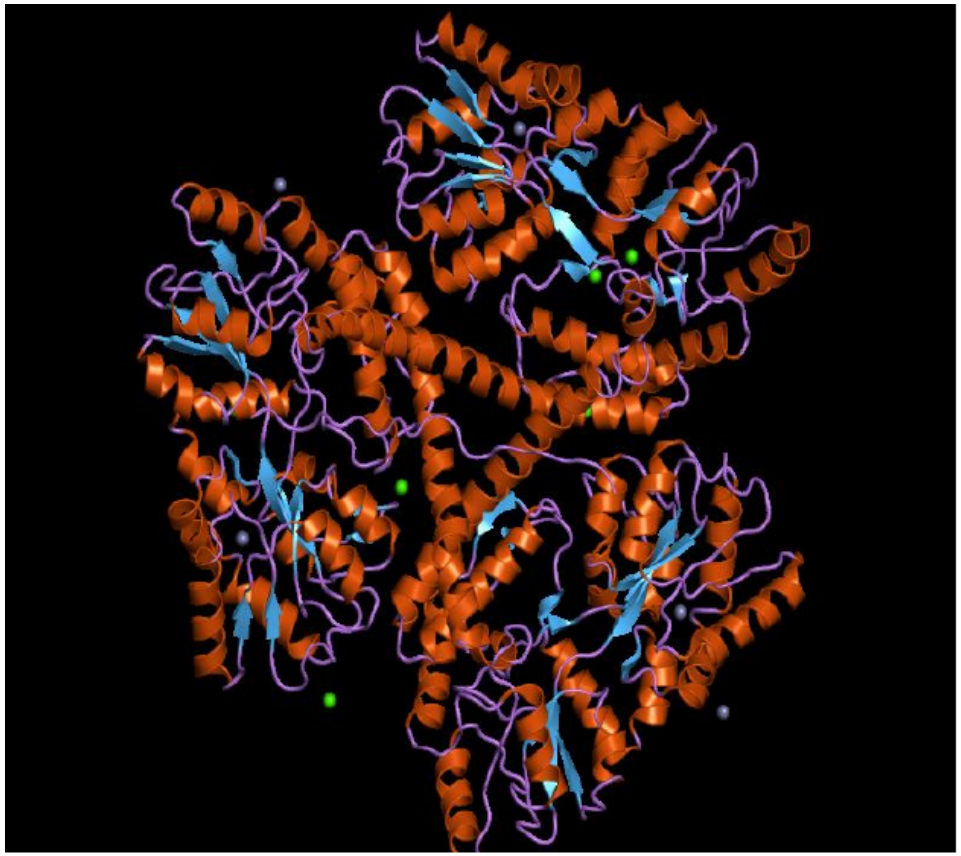
ACO in a Nutshell



Widely used strategy for NP-hard problems such as the travelling salesman problem.

Example application: Huntington's Disease

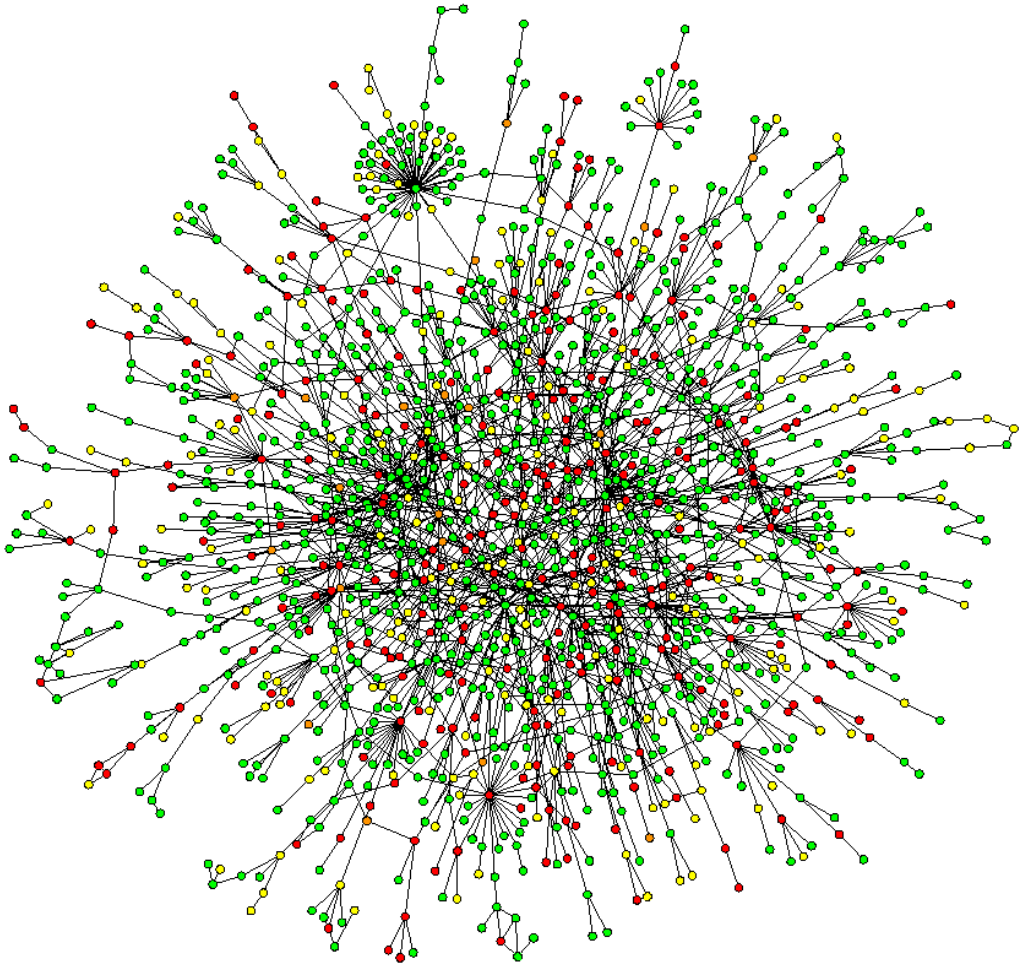
- Inheritable neurological disorder
- Caused by a mutation in the Huntingtin gene which causes a structural change in the Huntingtin protein (HTT).



Example application: Huntington's Disease

Human protein-protein interaction network:

- 8,291 nodes (proteins)
- 23,462 edges (physical interactions)
- Collected from different sources

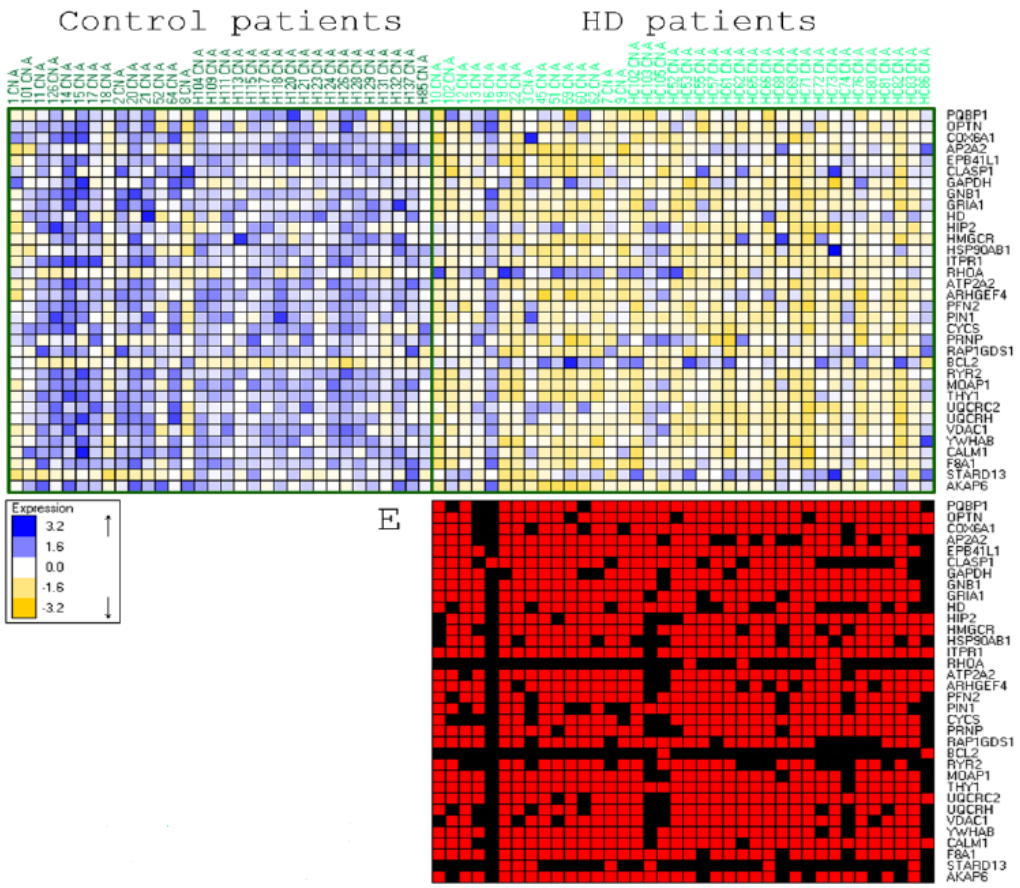


Ulitsky I et al. (2008) Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles, Proceedings of RECOMB, Research in Computational Molecular Biology 4955 (2008) 347–359.

Example application: Huntington's Disease

Microarray expression studies:

- 38 HD-affected samples
- 32 unaffected control samples
- The Huntingtin gene is differentially expressed in only ~50% of cases



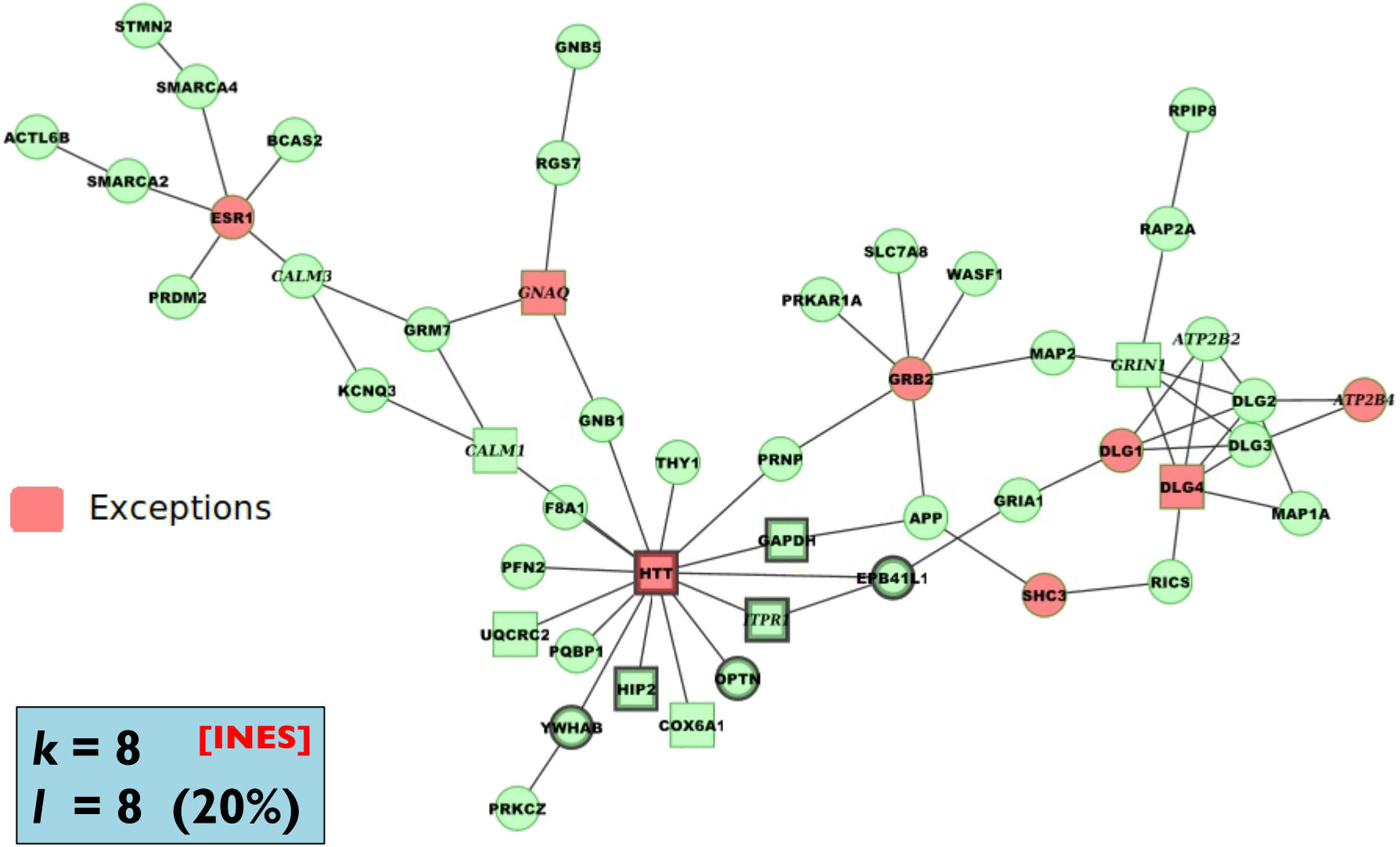
Hodges A et al. (2006) Regional and cellular gene expression changes in human Huntington's disease brain, Hum Mol Genet. 2006 Mar 15;15(6):965-77.

KeyPathwayMiner: Huntington's Disease

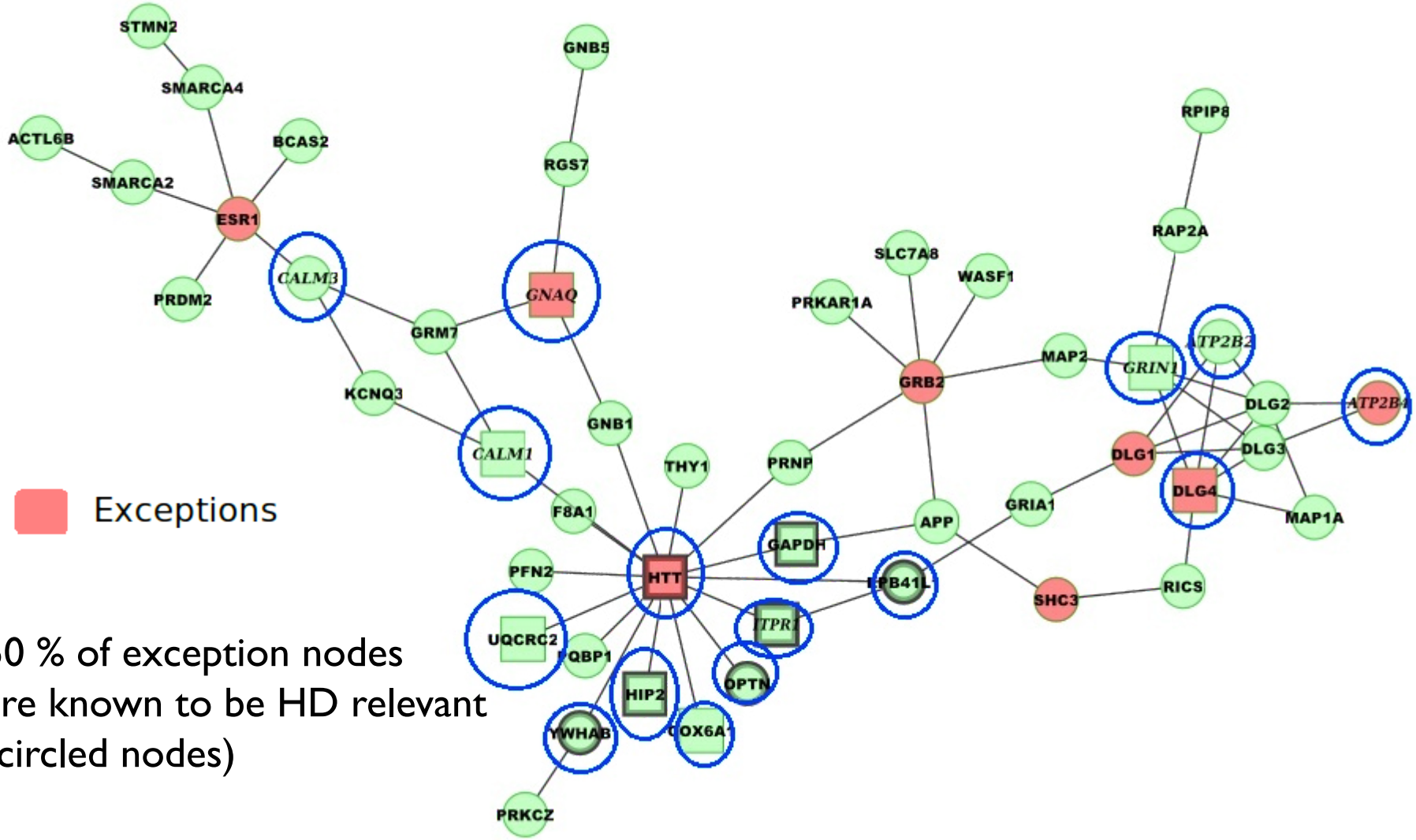
***k* = 8** **[INES]**

***l* = 8 (20%)**

Huntington's Disease Pathway



Huntington's Disease Pathway

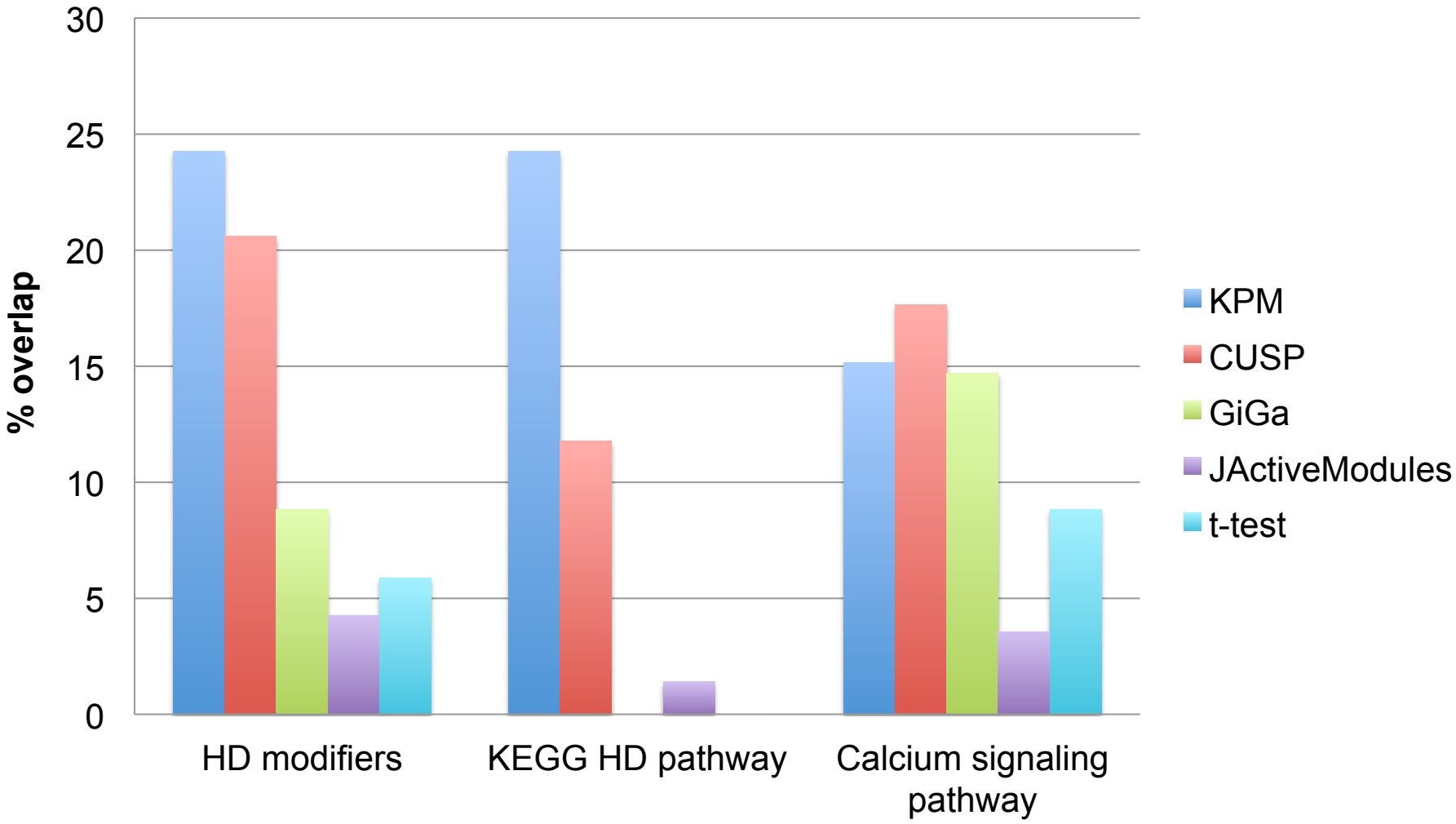


50 % of exception nodes
are known to be HD relevant
(circled nodes)

Evaluation

	INES	GLONE	CUSP	GiGA	jActive-Modules	t-test
Number of genes	37	38	34	34	282	34
Contains Htt	YES	YES	YES	NO	NO	NO
HD modifiers	8	7	7	3	12	2
KEGG HD pathway	8	10	4	0	4	0
Calcium Pathway	5	7	6	5	10	3

Evaluation


30
25
20
15
10
5
0
% overlap
HD modifiers
KEGG HD pathway
Calcium signaling pathway
KPM
CUSP
GiGa
JActiveModules
t-test

Summary

- Combining OMICs datasets with networks can improve on independent analysis of each
- Extracting differentially expressed pathways leads to hard computational problems
- For small instances, the problem can usually be solved exactly, otherwise approximation algorithms or more heuristic approaches (e.g. ACO) are needed.
- Active module approaches may overlook relevant genes in complex diseases
- Approaches like CUSP and KeyPathwayMiner improve on this by allowing a certain number of outliers in the solutions.

References

- T. Ideker et al., Discovering regulatory and signaling circuits in molecular interaction networks. Bioinformatics *Suppl I, 2002*

- I. Ulitsky et al., Detecting disease-specic dysregulated pathways via analysis of clinical expression proles, *Recomb, 2008*

- N. Alcaraz et al., KeyPathwayMiner: Detecting case-specic biological pathways using expression data, *Int Math. 2011, 7:4, 299-313*

- Baumbach J, et al., Efficient algorithms for extracting biological key pathways with global constraints. *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2012*

- M. Dorigo et al., Ant colony optimization, MIT Press, 2004

Thank you!