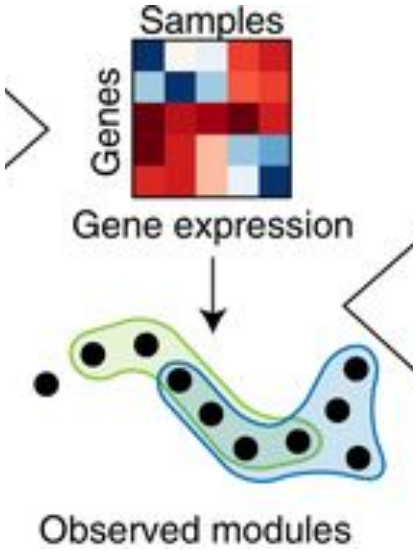
V14 – Gene Regulation

Tue, June 5, 2018

- Co-expression modules
- Motifs in GRNs
- Master Regulatory Genes in GRNs

Module detection methods



Module detection is a cornerstone in the biological interpretation of large gene expression compendia.

Such modules are groups of genes with similar expression profiles, which also tend to be functionally related and co-regulated.

Approaches:

(a) clustering

(b) decomposition methods

(c) biclustering – local co-expression (also (b))

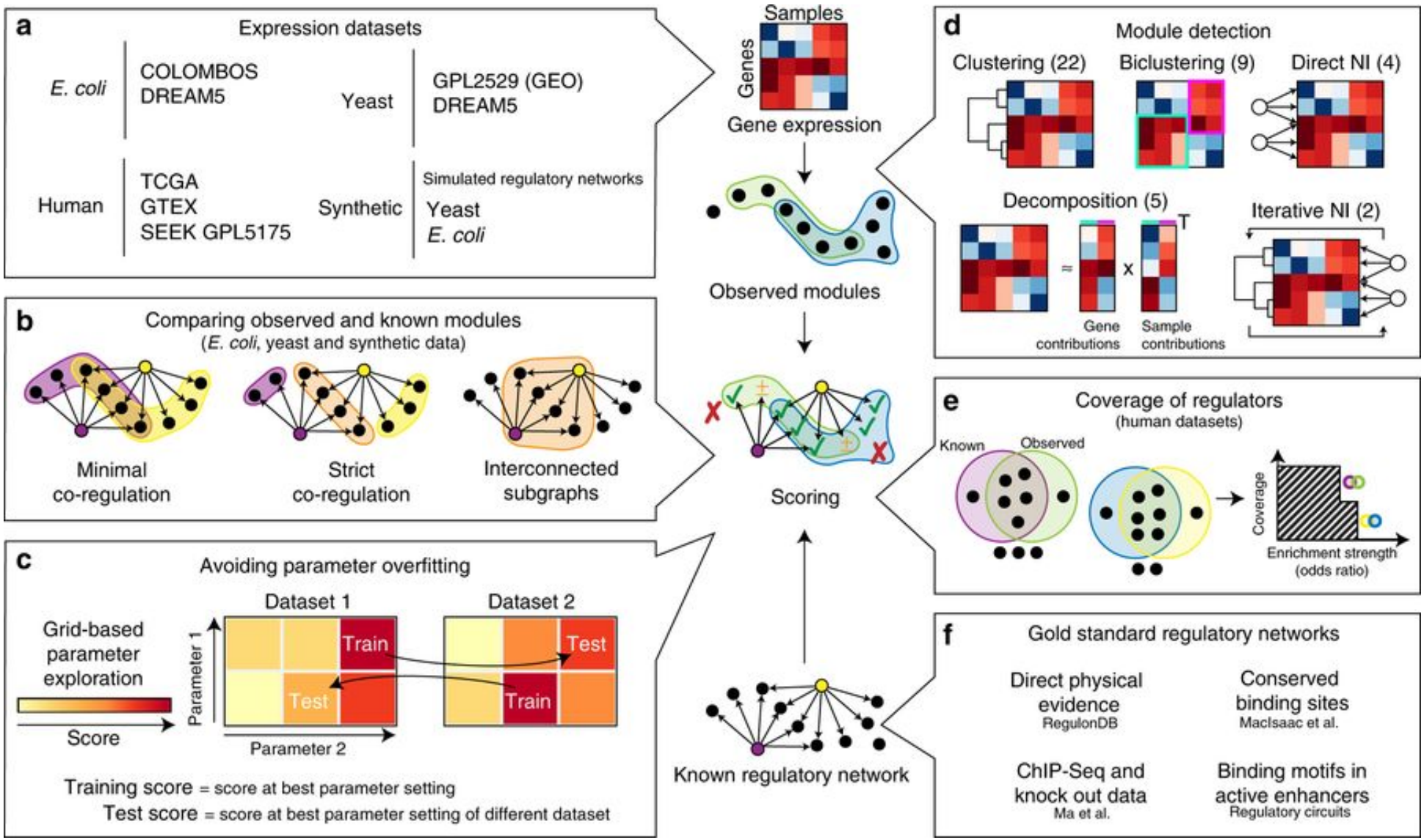
(d) direct network inference

(e) iterative network inference.

(d) and (e) also model the regulatory relationships between the genes.

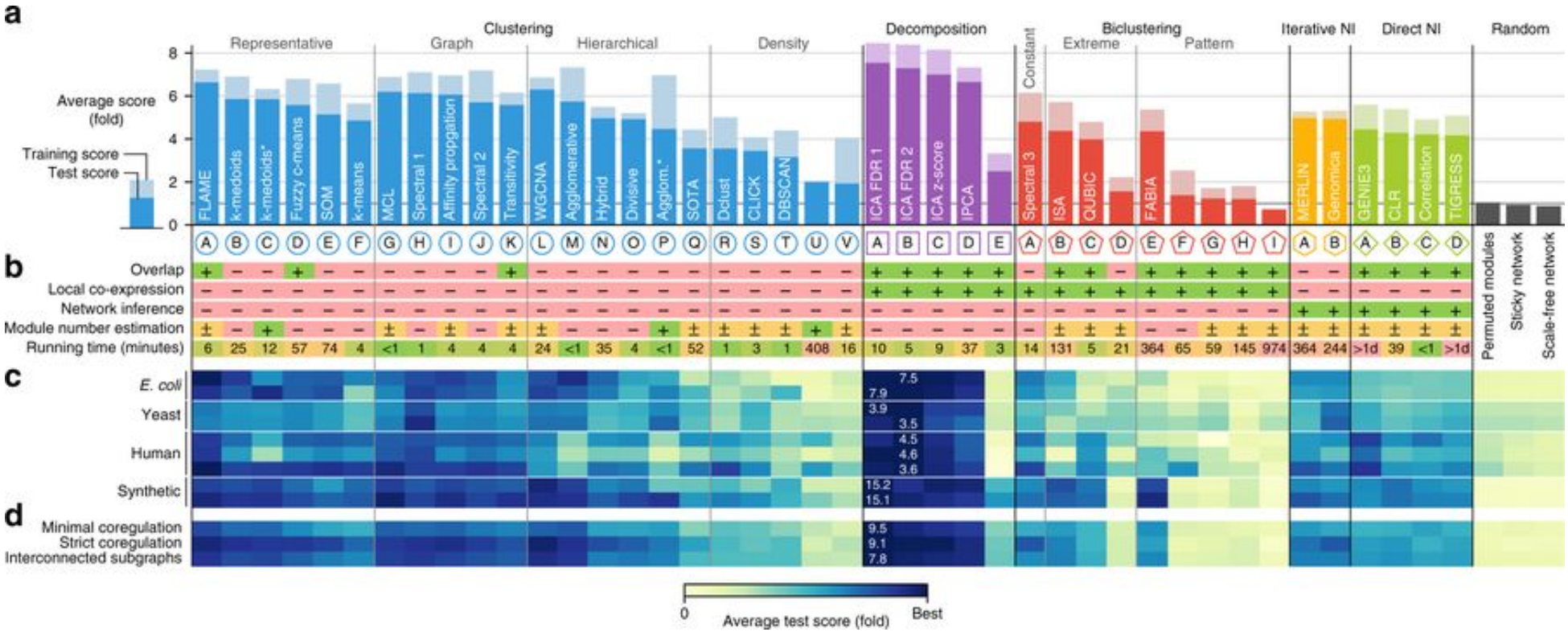
Saelens et al. *Nature Commun.* 9, 1090 (2018)

Module detection methods



Saelens et al. *Nature Commun.* 9, 1090 (2018)

Module detection methods: performance



ICA-based decomposition methods work best in detecting co-expression modules that overlap with known regulatory modules.

Saelens et al. *Nature Commun.* 9, 1090 (2018)

ICA method:
https://www.ece.ucsb.edu/wcsl/courses/ECE594/594C_F10Madhow/comon94.pdf
https://www.cs.helsinki.fi/u/ahyvarin/papers/NN00new.pdf

Independent Component Analysis (ICA)

ICA decomposes the expression data matrix X into a number of "components" (k = 1,2,..K), each of which is characterised by an **activation pattern** over genes (S_k) and another over samples (A_k)

$$X = \sum_{k=1}^{K} S_k \otimes A_k + E$$

in such a way that the gene activation patterns ($S_1, S_2, \ldots, S_K$) are as statistically independent as possible while also minimising the residual "error" matrix E

In the above formula, $\otimes$ denotes the Kronecker tensor product.

While ICA also provides a linear decomposition of the data matrix, the requirement of statistical independence implies that the data covariance matrix is decorrelated in a *non-linear* fashion, in contrast to PCA where the decorrelation is performed linearly.

Saelens et al. *Nature Commun.* 9, 1090 (2018)

Kronecker tensor product

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}.$$

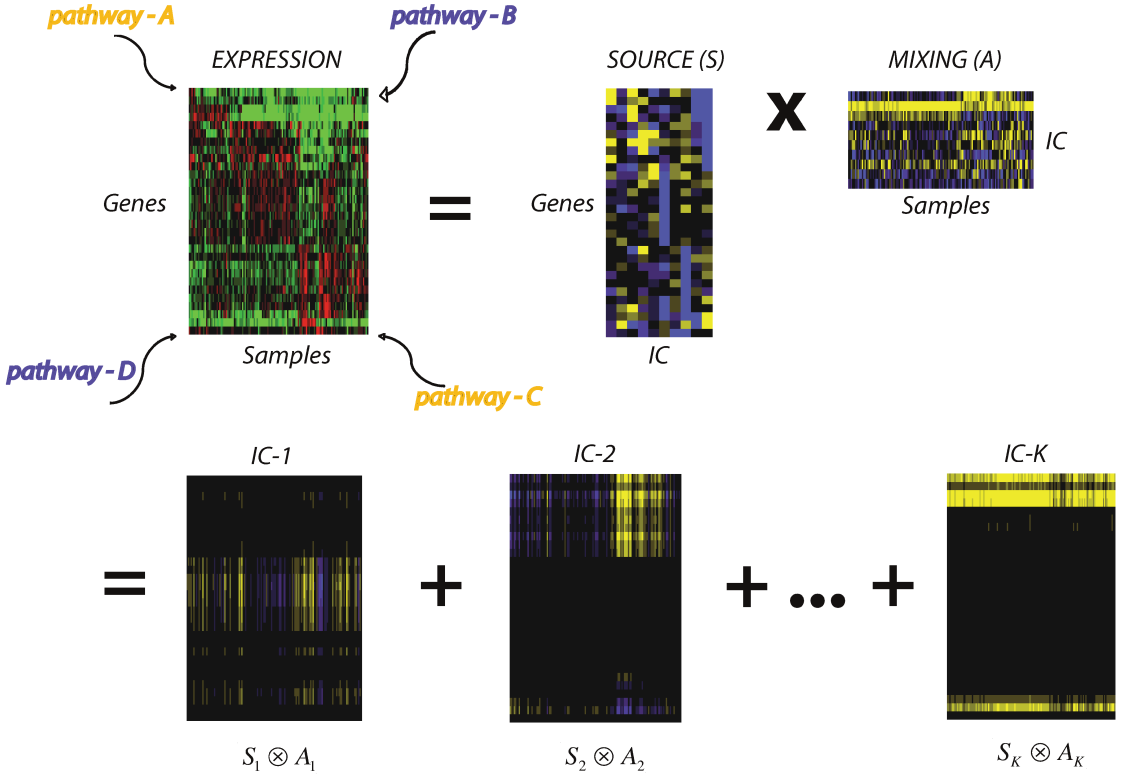
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 5 & 2 \cdot 0 & 2 \cdot 5 \\ 1 \cdot 6 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 7 \\ 3 \cdot 0 & 3 \cdot 5 & 4 \cdot 0 & 4 \cdot 5 \\ 3 \cdot 6 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}.$$

www.wikipedia.org

ICA model of gene expression

In the ICA model, the gene expression matrix is decomposed into the product of a "source" matrix *S* and a "mixing" matrix *A*.

K is the number of inferred independent components (IC) to which pathways and regulatory modules map.



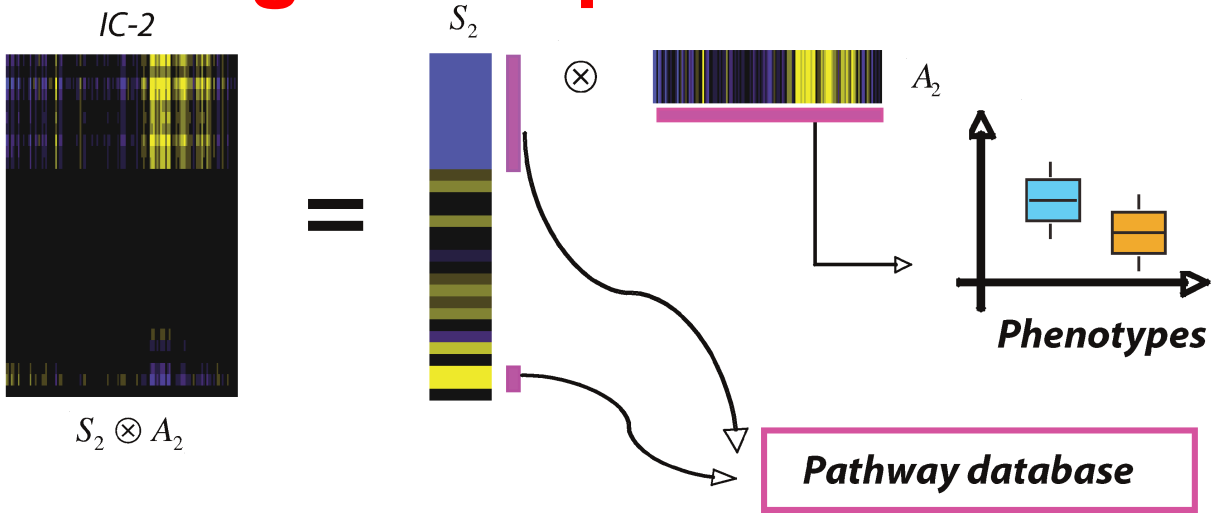
The columns of *S* describe the activation levels of genes in the various inferred independent components

The rows of *A* give the activation levels of the independent components across tumor samples.

The product of *S* and *A* can be written as a sum over the *IC* submatrices *IC*-1,*IC*-2,...*IC*-*K*.

Teschendorff AE, et al. (2007) PLoS Comput Biol 3: e161.

ICA model of gene expression



The *IC–k*–submatrix is obtained by multiplying the *k*-th column of *S*, S_k, with the *k*-th row of *A*, A_k.

The genes with the largest absolute weights in S_k are selected and tested for enrichment of biological pathways, while the distribution of weights in A_k are tested for discriminatory power of phenotypes.

Colour codes for heatmaps:

red, overexpression;

green, underexpression;

blue, upregulation;

yellow, downregulation.

Teschendorff AE, et al. (2007) PLoS Comput Biol 3: e161.

9.5 Network Motifs

Network motifs in the transcriptional regulation network of *Escherichia coli*

Shai S. Shen-Orr[1], Ron Milo[2], Shmoolik Mangan[1] & Uri Alon[1,2]

Nature Genetics **31** (2002) 64

RegulonDB + hand-curated literature evidence

→ break down network into motifs

→ statistical significance of the motifs?

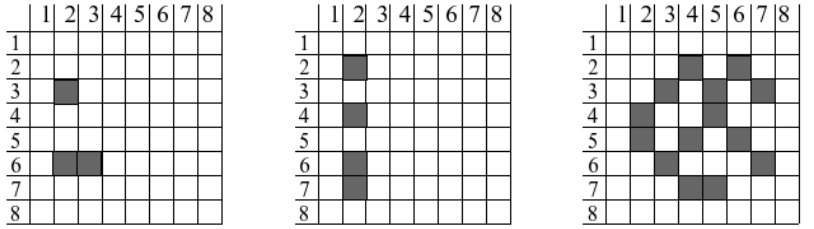
→ behavior of the motifs <=> location in the network?

Detection of motifs

Represent transcriptional network as a connectivity matrix M such that $M_{ij} = 1$ if operon j encodes a TF that transcriptionally regulates operon i and $M_{ij} = 0$ otherwise.

Scan all $n \times n$ submatrices of M generated by choosing n nodes that lie in a connected graph, for $n = 3$ and $n = 4$.

Submatrices were enumerated efficiently by recursively searching for nonzero elements.



Connectivity matrix for causal regulation of transcription factor j (row) by transcription factor i (column). Dark fields indicate regulation.
(Left) Feed-forward loop motif. TF 2 regulates TFs 3 and 6, and TF 3 again regulates TF 6.
(Middle) Single-input multiple-output motif.
(Right) Densely-overlapping region.

For $n = 3$, the only significant motif is the **feedforward loop**.
For $n = 4$, only the **densely overlapping regulation** motif is significant.
SIMs and multi-input modules were identified by searching for identical rows of M.

Shen-Orr et al. Nature Gen. 31, 64 (2002)

Motif Statistics

Compute a p-value for submatrices representing each type of connected subgraph by comparing # of times they appear in real network vs. in random network.

Table 1 • Statistics of occurrence of various structures in the real and randomized networks

Structure	Appearances in real network	Appearances in randomized network (mean ± s.d.)	*P* value
Coherent feedforward loop	34	4.4 ± 3	$P < 0.001$
Incoherent feedforward loop	6	2.5 ± 2	$P \sim 0.03$
Operons controlled by SIM (>13 operons)	68	28 ± 7	$P < 0.01$
Pairs of operons regulated by same two transcription factors	203	57 ± 14	$P < 0.001$
Nodes that participate in cycles*	0	0.18 ± 0.6	$P \sim 0.8$

*Cycles include all loops greater than size 1 (autoregulation). *P* value for cycles is the probability of networks with no loops.

Listed motifs are highly **overrepresented** compared to randomized networks

No cycles ($X \rightarrow Y \rightarrow Z \rightarrow X$) were identified, but this was not statistically significant in comparison to random networks

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

Generate Random Networks

For a stringent comparison to randomized networks, one generates networks with precisely the same

- number of operons,
- interactions,
- TFs and
- number of incoming and outgoing edges for each node

as in the real network (here the one from *E. coli*).

One starts with the real network and repeatedly swaps randomly chosen pairs of connections (*X1* → *Y1*, *X2* → *Y2* is replaced by *X1* → *Y2*, *X2* → *Y1*) until the network is well randomized.

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

Generate Random Networks

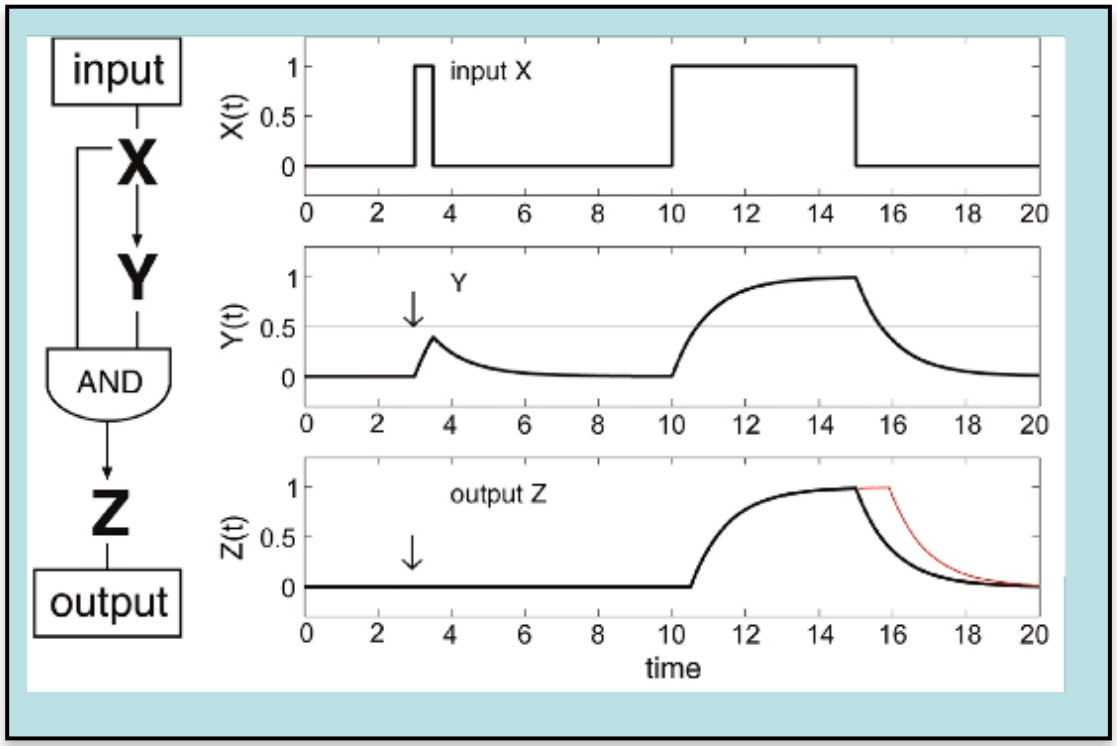
This yields networks with precisely the same number of nodes with *p* incoming and *q* outgoing nodes, as the real network.

The corresponding randomized connectivity matrices, *Mrand*, have the same number of nonzero elements in each row and column as the corresponding row and column of the real connectivity matrix *M*:

$$\sum_i Mrand_{ij} = \sum_i M_{ij} \quad \text{and} \quad \sum_j Mrand_{ij} = \sum_j M_{ij}$$

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

FFL dynamics



In a **coherent** FFL:
X **and** Y activate Z

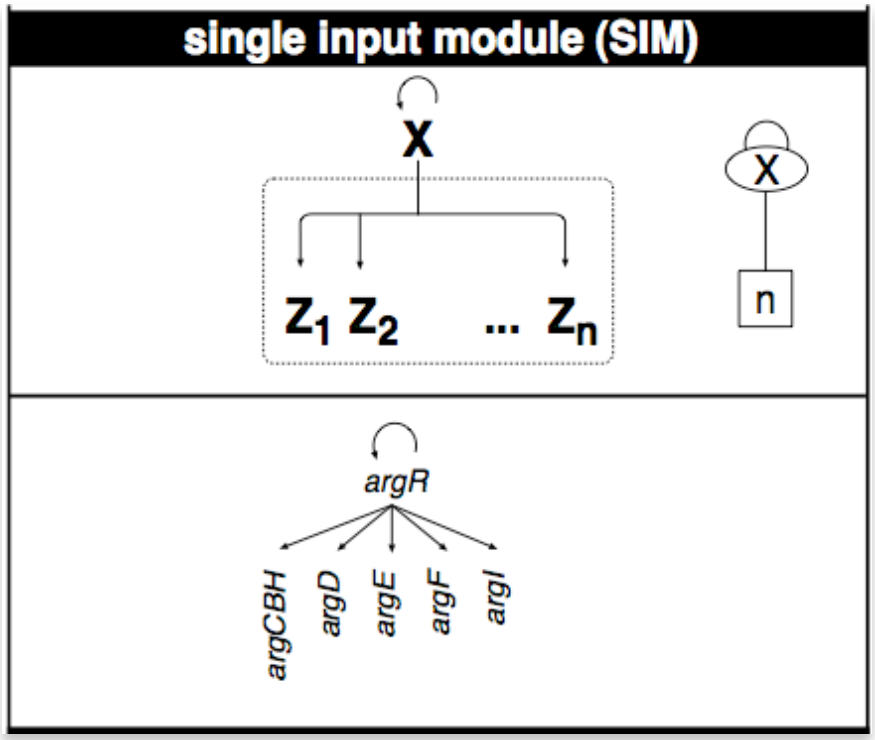
Dynamics:
- input activates X
- X activates Y (delay)
- (X && Y) activates Z

Delay between X and Y → signal must persist longer than delay
(see lecture 12, slide 31)
→ reject transient signal, react only to **persistent** signals
→ enables fast shutdown

Helps with **decisions** based on **fluctuating signals.**

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

Motif 2: Single-Input-Module



Set of operons controlled by a single transcription factor

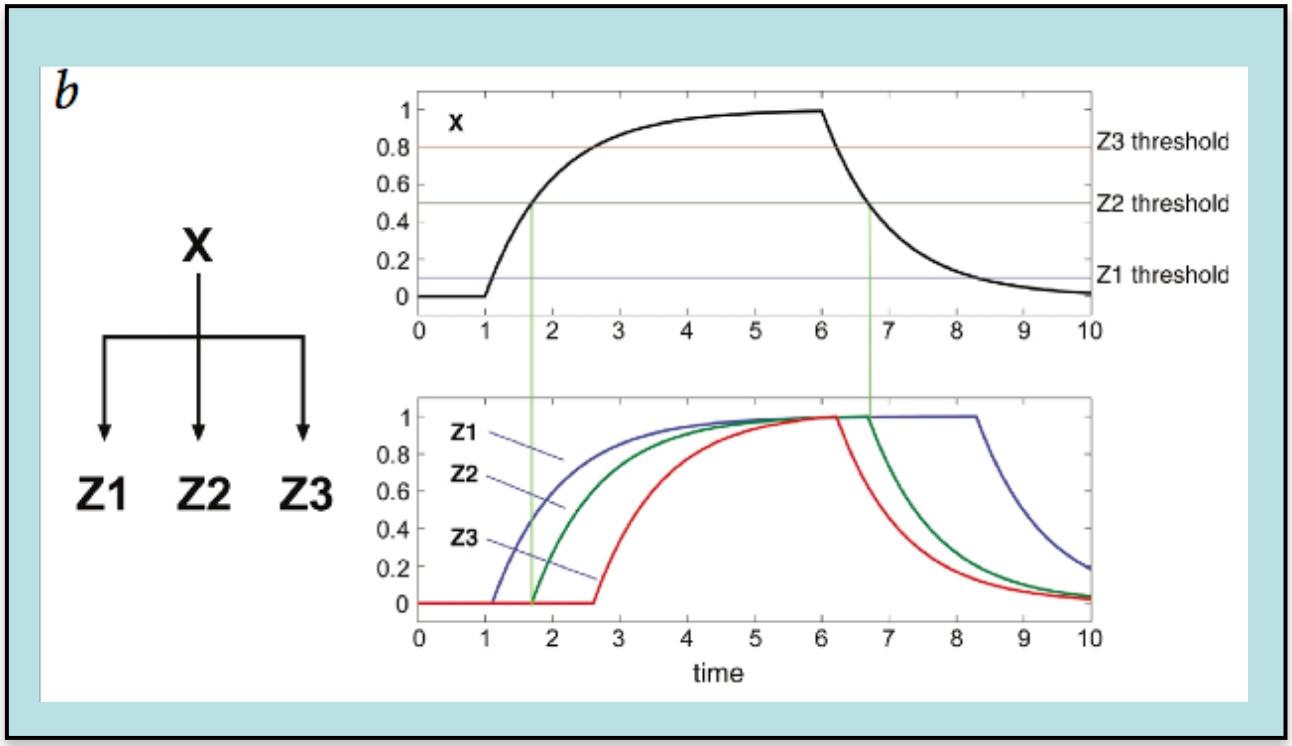
- same sign
- no additional regulation
- control is usually autoregulatory (70% vs. 50% overall)

Example for this in *E. coli*:
arginine biosynthetic operon *argCBH* plus other enzymes of arginine biosynthesis pathway.

Mainly found in genes that code for **parts** of a protein **complex** or metabolic **pathway**
→ produces components in comparable amounts (stoichiometries).

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

SIM-Dynamics



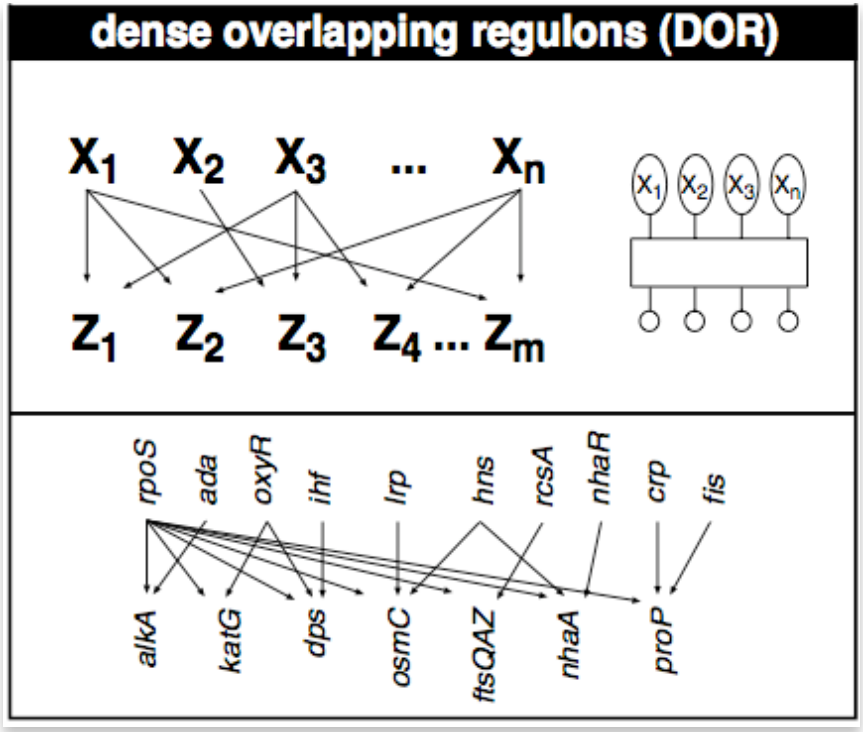
If different thresholds exist for each regulated operon:

→ first gene that is activated is the last that is deactivated

→ well defined temporal ordering (e.g. flagella synthesis) + stoichiometries

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

Motif 3: Densely Overlapping Regulon



Dense layer between groups of TFs and operons
→ much denser than network average (≈ community)

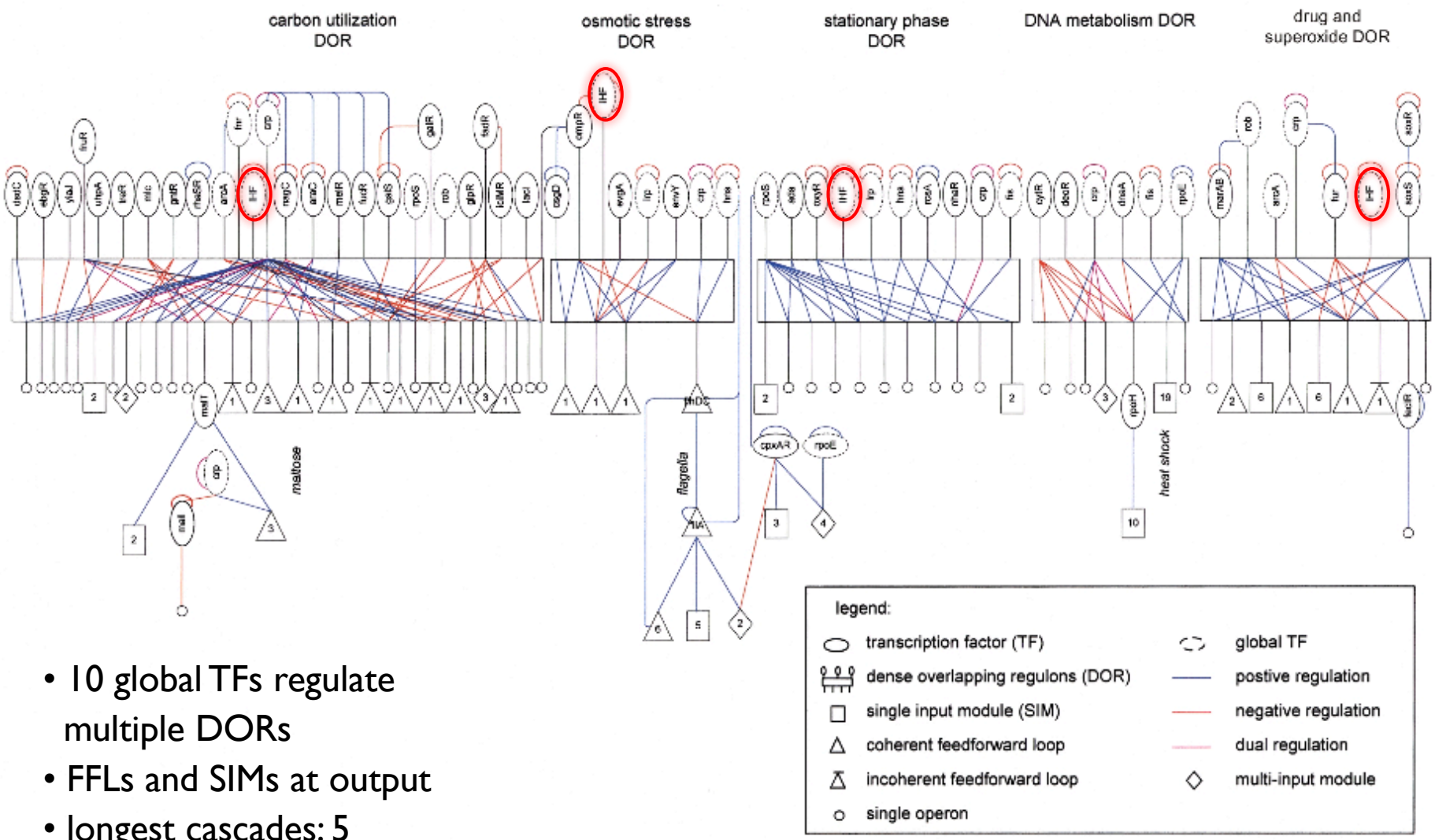
Usually each operon is regulated by a different combination of TFs.

Main "**computational**" **units** of the regulation system

Sometimes: same set of TFs for group of operons → "multiple input module"

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

Network with Motifs



- 10 global TFs regulate multiple DORs
- FFLs and SIMs at output
- longest cascades: 5 (flagella and nitrogen systems)

Shen-Orr et al., *Nature Genetics* **31** (2002) 64

9.6 Key pathway miner algorithm

The key-pathway miner algorithm solves the problem of finding key pathways at the level of labeled graphs (Alcaraz 2012).

Key pathways: connected sub-networks where most of the components are active/expressed/methylated in most conditions.

The algorithm can either output only the best solution found or multiple top solutions.

For a labeled graph $G = (V, E, d)$ of vertices V and edges E, there also exists a **labeling function** d: $V \rightarrow$ |N.

Alcaraz et al. (2012),
Integrative Biology 4, 756-764.

9.6 Key pathway miner algorithm

Let $k, l \in \mathbb{N}$.

The (k, l)-*KeyPathway problem* determines a connected subset $U \subseteq V$ of maximal cardinality which contains at most k elements $u \in U$ with $d(u) \leq l$.

Any set U fulfilling these two conditions is termed a (k, l)-*component*.

Any vertex $v \in V$ for which $d(v) \leq l$ is termed an **exception vertex**.

Vertices of the graph represent biological entities (e.g. genes or proteins); edges stand for interactions between two such entities, *e.g.* a protein–protein interaction.

The labels on a vertex v denote the number of situations were v is active/expressed/methylated *etc*.

Alcaraz et al. (2012),
Integrative Biology 4, 756-764.

9.6 Key pathway miner algorithm

In a preprocessing stage, one generates an auxiliary labeled graph $C(G, l)$ that serves to reduce the problem size and to help in steering the algorithm to more promising regions of the search space.

$C(G, l)$ is the l-component graph that is deduced from G in the following way:
- The vertex set of $C(G, l)$ contains all **exception vertices** of G.

- Two exception vertices are linked by an **edge** in $C(G, l)$ if they are connected by a path in G which does not contain exception vertices as inner vertices.

- For any subset $U \subseteq V$ of exception vertices, $S(U)$ is defined as the set of all vertices $v \in V$ that can be reached in G from an element of U without visiting an exception vertex that does not belong to U.

Intuitively, one simply needs to select a **connected set** of k exception vertices U in $C(G, l)$ to construct a (k, l)-component of G, namely $S(U)$.

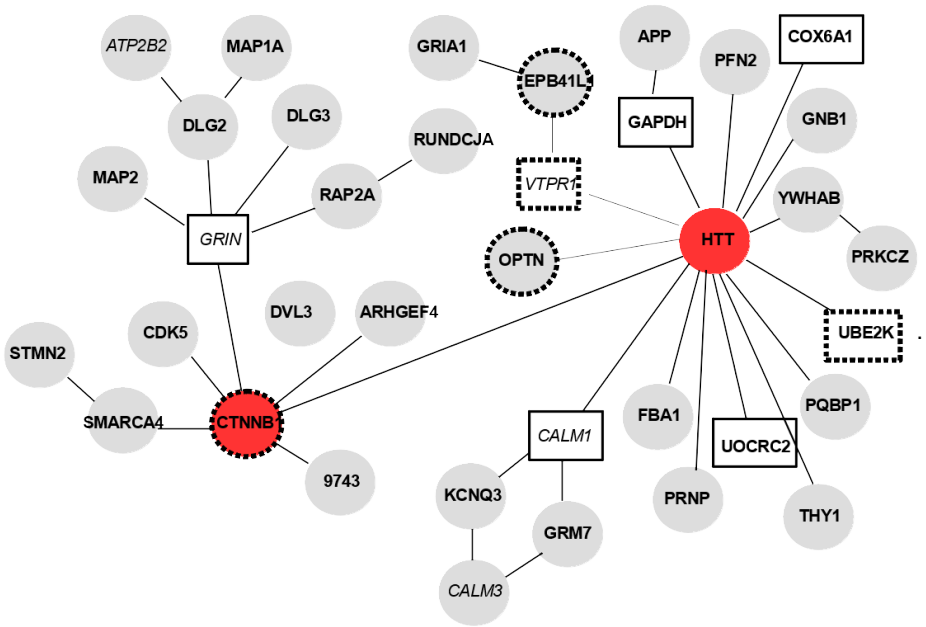
Alcaraz et al. (2012),
Integrative Biology 4, 756-764.

9.6 Key regulator genes

For this, the Key-pathway miner algorithm applies a greedy principle. For every vertex u, a set W_u is iteratively constructed that begins with $W_u = \{u\}$.

At every iteration step, one adds a vertex v from $C(G, I)$ to W_u that is adjacent to W_u in $C(G, I)$ and which maximizes $|S(W_u \cup \{v\})|$.

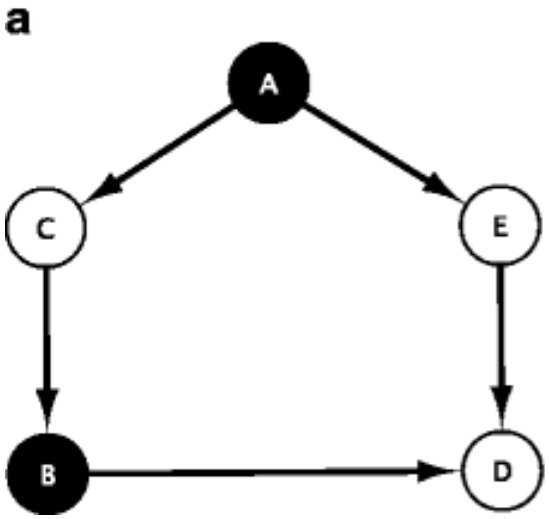
The iterations are stopped when $|W_u| = k$. The algorithm returns $S(W_u)$ of maximal size found for some u.



Largest subnetwork identified as down-regulated in the caudate nucleus of huntington disease patients found by the key pathway miner algorithm for $k = 2$.
Red nodes represent exception nodes,
squared nodes: genes of the Huntington's disease KEGG pathway,
nodes with dashed borders : *HTT* modifiers, nodes with italic font : part of the calcium signaling pathway.

Alcaraz et al. (2012),
Integrative Biology 4, 756-764.

Identification of Master regulatory genes



A vertex *u* **dominates** another vertex *v* if there exists a directed arc *(u,v)*.

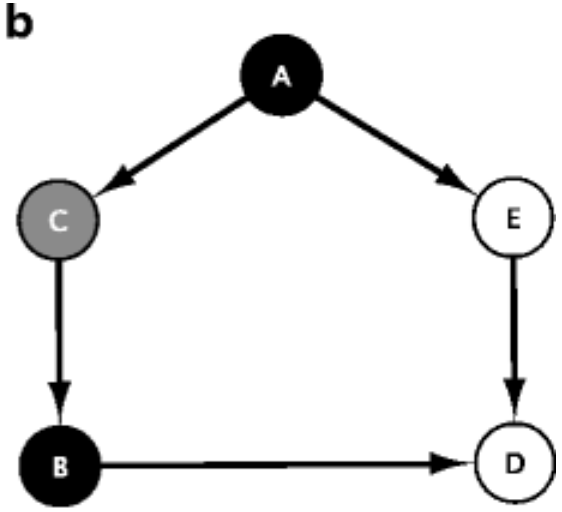
Idea: find a **set of dominator nodes** of minimum size that controls all other vertices.

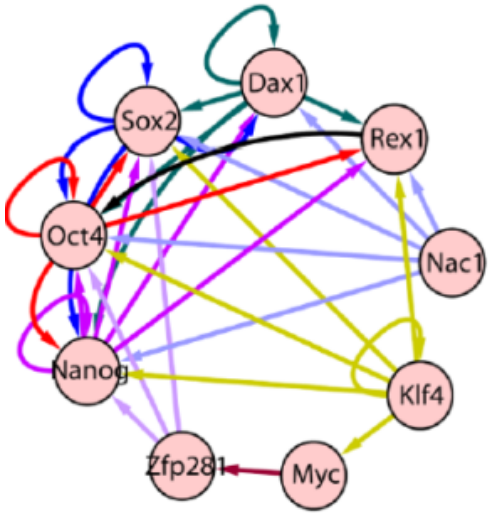
In the case of a GRN, a directed arc symbolizes that a transcription factor regulates a target gene.

In the figure, the MDS nodes {*A*,*B*} are the dominators of the network. Together, they regulate all other nodes of the network (*C*, *E*, *D*).

Nazarieh et al. BMC Syst Biol 10:88 (2016)

Identification of Master regulatory genes





Core pluripotency network,
Kim et al. Cell (2008)

The nodes of a MDS can be spread as isolates nodes over the entire graph.
However, e.g. the set of core pluripotency factors is tightly connected (right).

Idea: find a **connected dominating set of minimum size** (MCDS).

(Left) the respective set of MCDS nodes (*black and gray*).
Here, node *C* is added in order to preserve the connection
between the two dominators *A* and *B* to form an MCDS

ILP for minimum dominating set

Aim: we want to determine a set D of minimum cardinality such that for each $v \in V$, we have that $v \in D$ or that there is a node $u \in D$ and an arc $(u,v) \in E$.

Let $\delta^-(v)$ be the set of incoming nodes of v such that $(u,v) \in E$,
x_u and x_v are binary variables associated with u and v.

We select a node v as dominator if its binary variable x_v has value 1, otherwise we do not select it.

$$\text{minimize} \quad \sum_{v \in V} x_v$$

$$\text{subject to} \quad x_u + \sum_{v \in \delta^-(u)} x_v \geq 1 \quad \forall u \in V$$

$$x_v \in \{0,1\} \qquad \forall v \in V$$

With the GLPK solver, the runtime was less than 1 min for all considered networks.

Nazarieh et al. BMC Syst Biol 10:88 (2016)

ILP for minimum connected dominating set

A minimum connected dominating set (MCDS) for a directed graph G = (V,E) is a set of nodes $D \subseteq V$ of minimum cardinality that is a dominating set and additionally has the property that the graph *G[D]* induced by *D* is **weakly connected**, i.e. such that in the underlying undirected graph there exists a path between any two nodes of *D* that only uses vertices in *D*.

This time we will use two binary valued variables y_v and x_e .
y_v indicates whether node *v* is selected to belong to the MCDS.
x_e for the edges then yields a tree that contains all selected vertices and no vertex that was not selected.

$$\text{minimize} \quad \sum_{v \in V} y_v$$

$$\text{subject to} \quad \sum_{e \in E} x_e = \sum_{i \in V} y_i - 1$$

This guarantees that the number of edges is one less than the number of vertices. This is necessary (but not sufficient) to form a (spanning) tree.

ILP for minimum connected dominating set

$$\text{minimize} \quad \sum_{v \in V} y_v$$

$$\text{subject to} \quad \sum_{e \in E} x_e = \sum_{i \in V} y_i - 1$$

$$\sum_{e \in E(S)} x_e \leq \sum_{i \in S \setminus \{j\}} y_i \qquad \forall S \subset V, \forall j \in S$$

Second constraint

→ selected edges imply a **tree**.

(Note that this defines an exponential number of constraints for all subgraphs of V!)

$$y_u + \sum_{v \in \delta^-(u)} y_v \geq 1 \qquad \forall u \in V$$

Third constraint

→ node set forms **dominating set**.

$$y_v \in \{0, 1\} \qquad \forall v \in V$$

$$x_e \in \{0, 1\} \qquad \forall e \in E$$

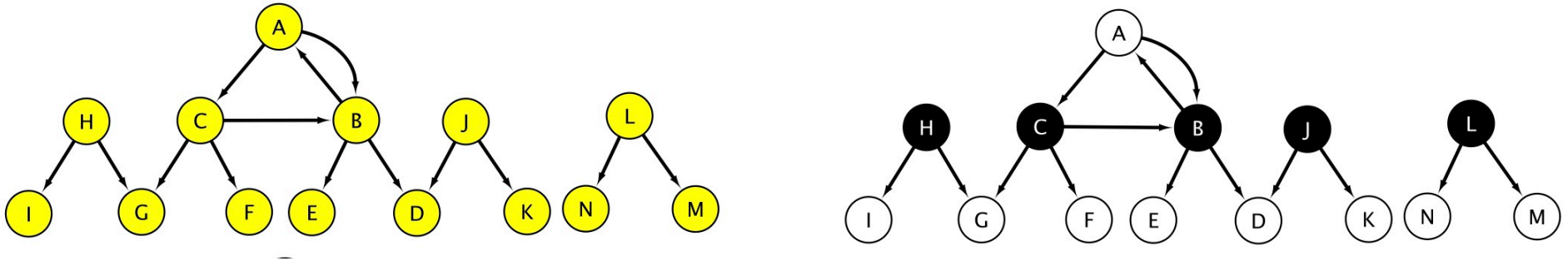
For dense graphs, this yields a quick solution.

However, for sparse graphs, the running time may be considerable.

Here we used an iterative approach for the second constraint.

Nazarieh et al. BMC Syst Biol 10:88 (2016)

Example MDS



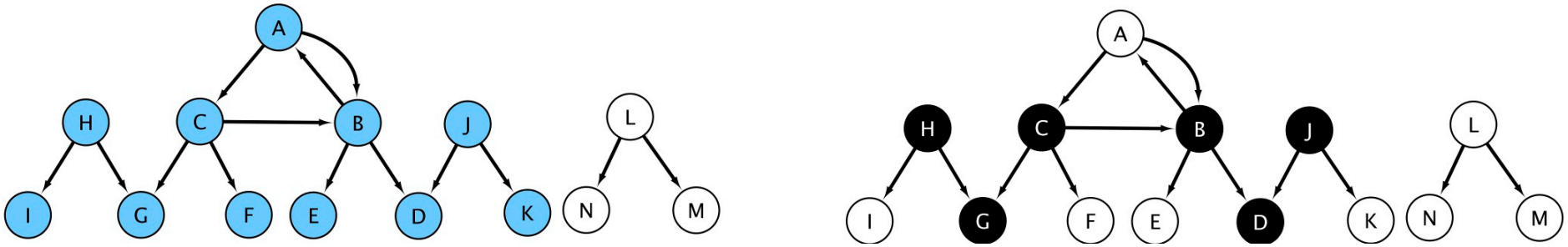
(Left) this toy network includes 14 nodes and 14 edges.

(Right) The dark colored nodes {*J, B, C, H, L*} are the dominators of the network obtained by computing a MDS.

Nazarieh et al. BMC Syst Biol 10:88 (2016)

Example MCDS

b



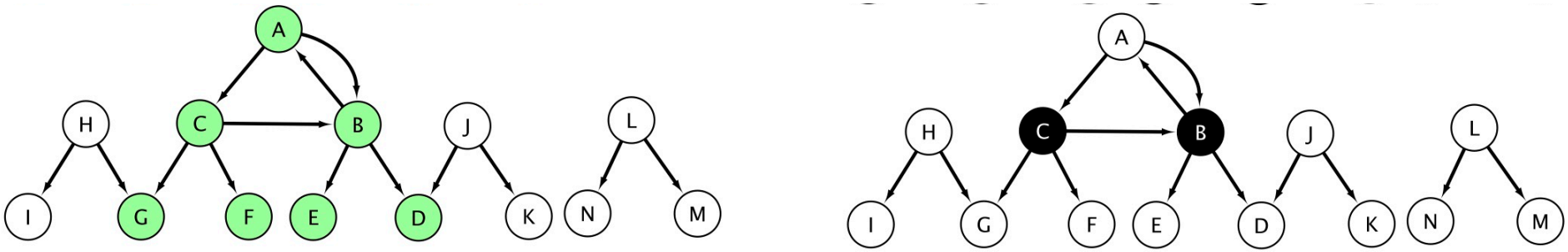
(Left) The nodes colored blue make up the **largest connected component** (LCC) of the underlying **undirected** graph.

(Right) MCDS nodes for this component are {*J, D, B, C, G, H*}.

Nazarieh et al. BMC Syst Biol 10:88 (2016)

Example MCDS

c



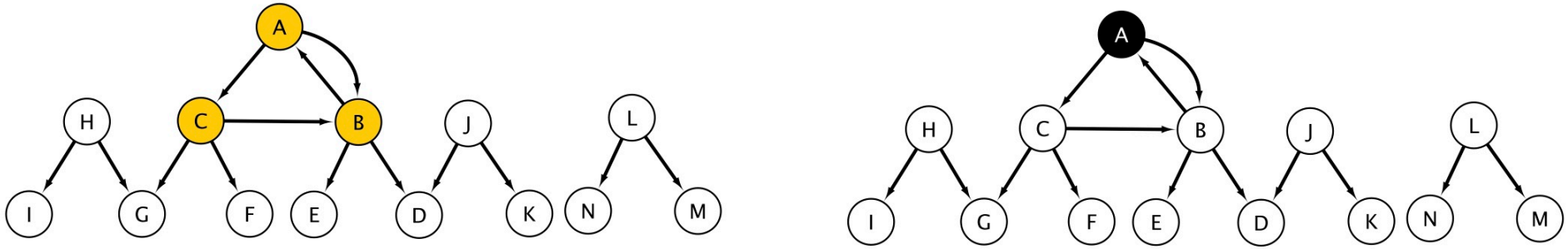
(Left) The green colored nodes are elements of **the largest connected component** underlying the **directed** graph.

(Right) The two nodes {*B*, *C*} form the MCDS for this component.

Nazarieh et al. BMC Syst Biol 10:88 (2016)

MCDS of the strongly connected component

d



(Left) The nodes colored orange show the LSCC in the network.

(Right) The node *A* is the only element of the MCDS

Nazarieh et al. BMC Syst Biol 10:88 (2016)

Studied networks: RegulonDB (E.coli)

This GRN contains 1807 genes, including 202 TFs and 4061 regulatory interactions. It forms a general network which controls all sorts of responses which are needed in different conditions.

Due to the sparsity of the network, its MDS contains 199 TFs.

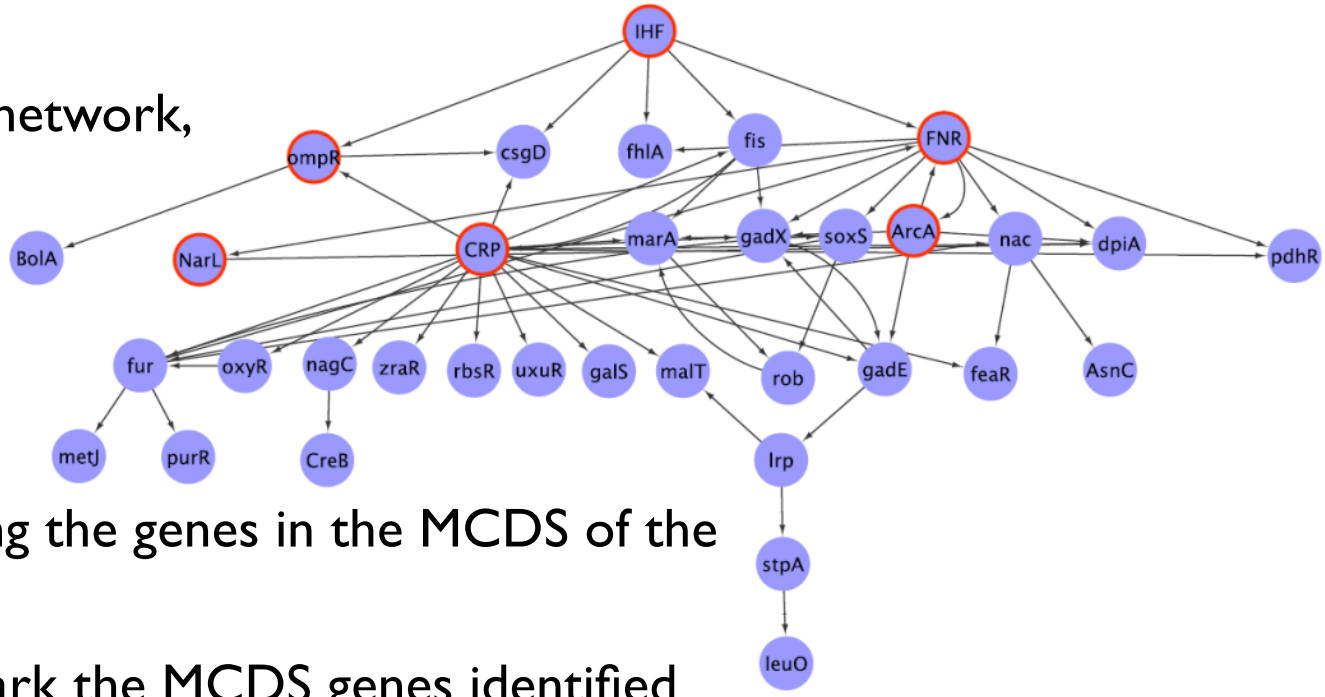


Figure: Connectivity among the genes in the MCDS of the LCC of the *E.coli* GRN.
The red circle borders mark the MCDS genes identified as global regulators by Ma *et al.* (see lecture V12, slide 7).

Periodic genes in cell cycle network of yeast

Take regulatory data from Yeast Promoter Atlas (YPA).
It contains 5026 genes including 122 TFs.

From this set of regulatory interactions, we extracted a cell-cycle specific subnetwork of 302 genes that were differentially expressed along the cell cycle of yeast (MA study by Spellman et al. Mol Biol Cell (1998)).

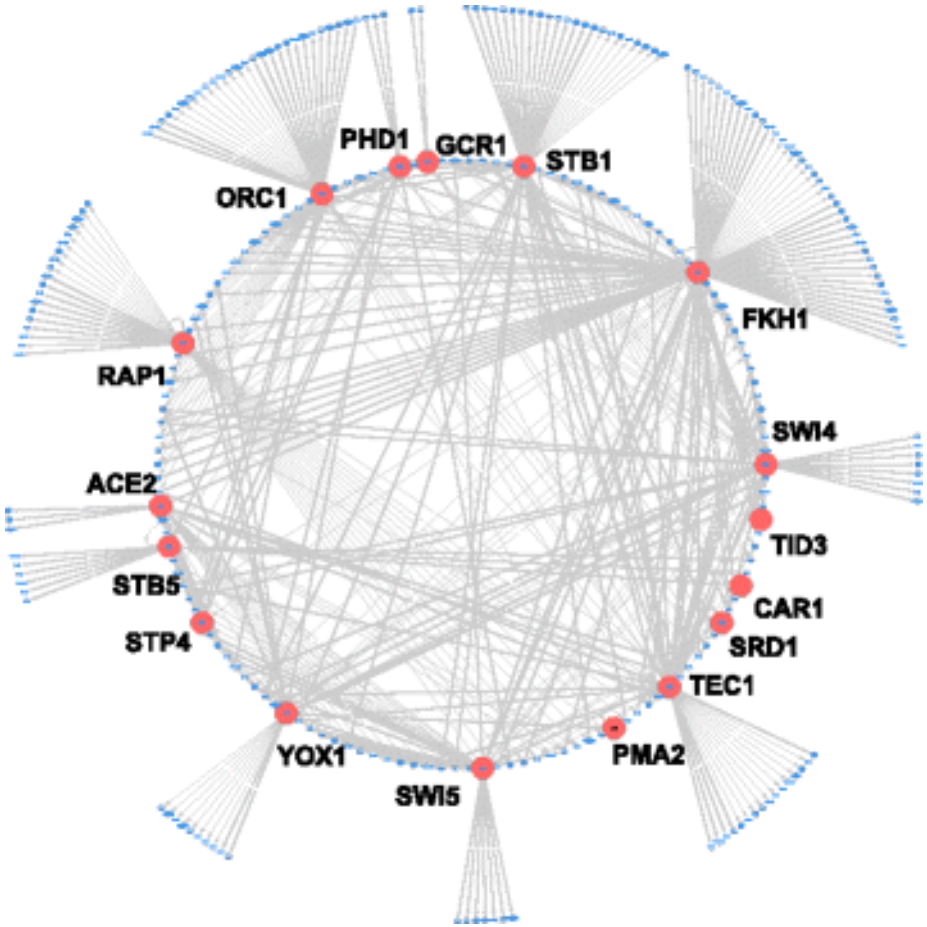
Nazarieh et al. BMC Syst Biol 10:88 (2016)

MCDS of cell cycle network of yeast

Tightly interwoven network of 17 TFs and target genes that organize the cell cycle of *S. cerevisiae*.

Shown on the circumference of the outer circle are 164 target genes that are differentially expressed during the cell cycle and are regulated by a TF in the MCDS (shown in the inner circle).

The inner circle consists of the 14 TFs from the heuristic MCDS
and of 123 other target genes that are regulated by at least two of these TFs

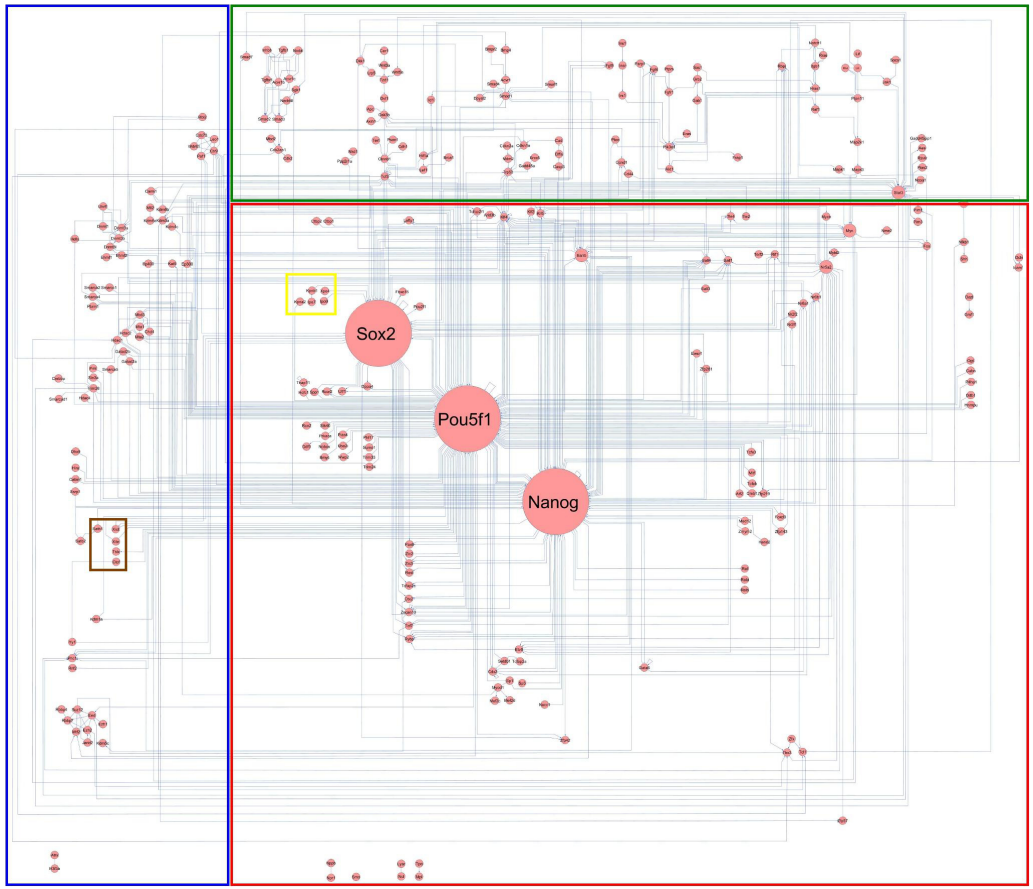


Nazarieh et al. BMC Syst Biol 10:88 (2016)

Studied networks: PluriNetwork

PluriNetWork was manually assembled as an interaction/regulation network describing the molecular mechanisms underlying pluripotency.

It contains 574 molecular interactions, stimulations and inhibitions, based on a collection of research data from 177 publications until June 2010, involving 274 mouse genes/proteins.



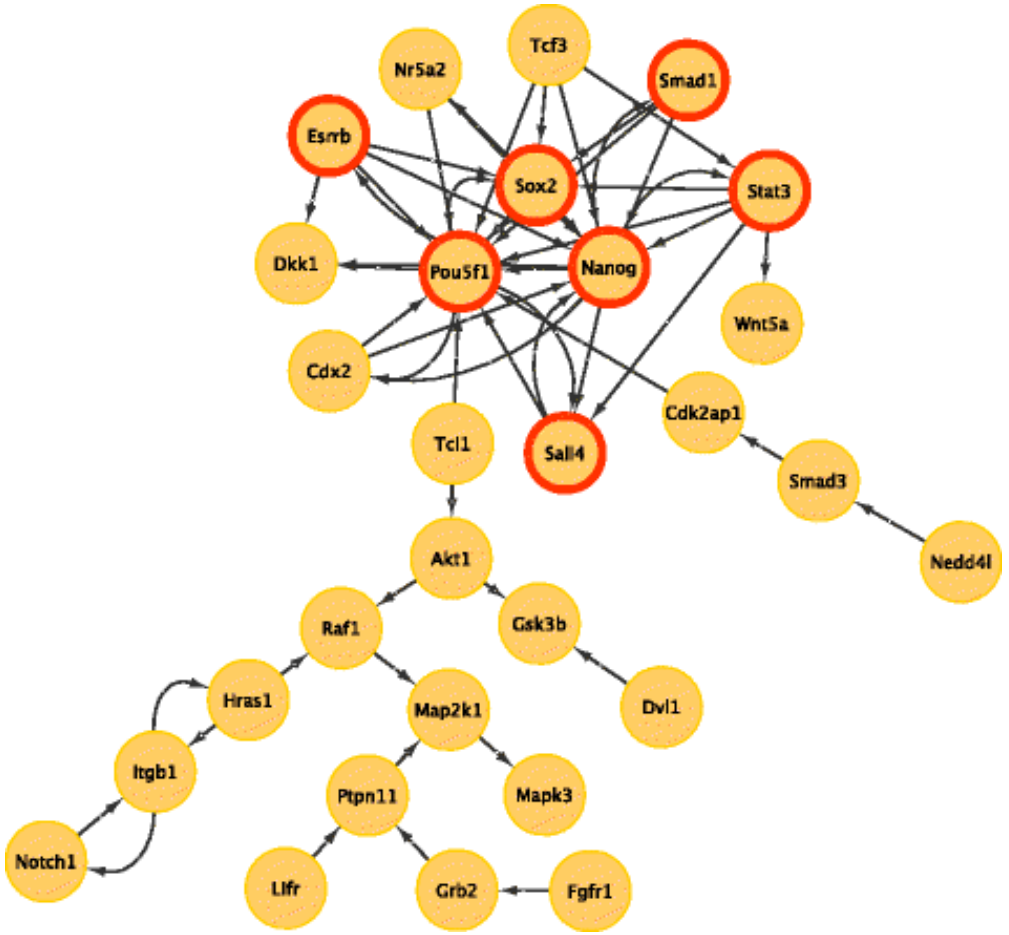
Som A, et al. (2010) PLoS ONE 5: e15165.

MCDS of mouse pluripotency network

Connectivity among TFs in the heuristic MCDS of the largest strongly connected component of a GRN for mouse ESCs.

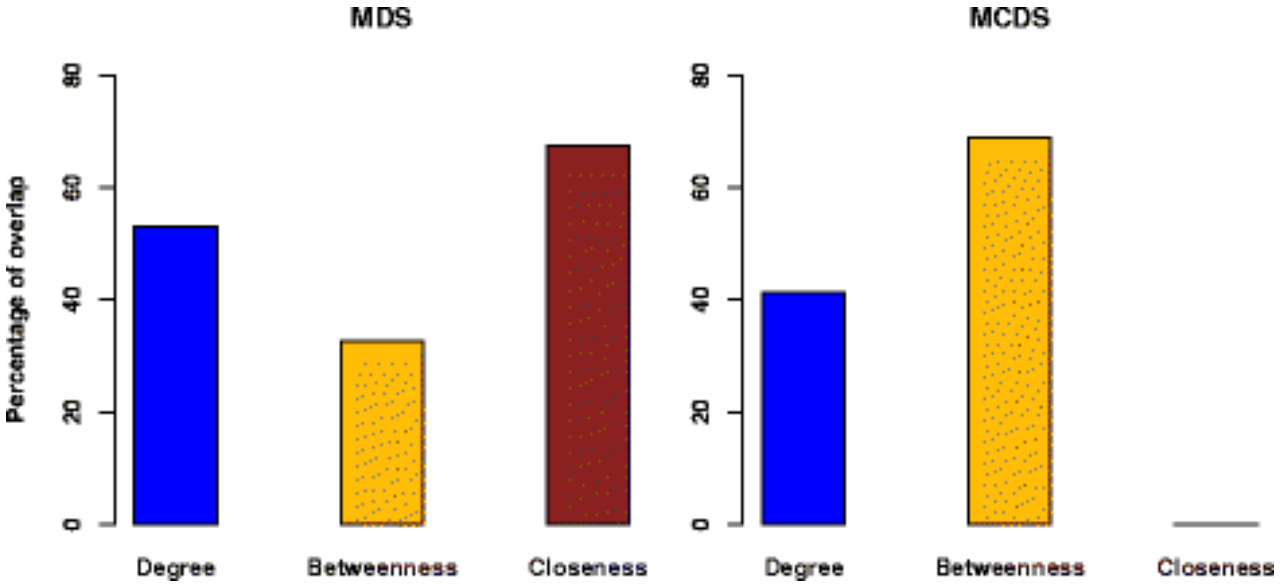
The red circle borders mark the 7 TFs belonging to the set of master regulatory genes identified experimentally.

The MCDS genes were functionally significantly more homogeneous than randomly selected gene pairs of the whole network (p = 6.41e-05, Kolmogorov-Smirnow test).



Nazarieh et al. BMC Syst Biol 10:88 (2016)

Overlap with most central nodes



Percentage overlap of the genes of the MDS and MCDS with the list of top genes (same size as MCDS) according to 3 centrality measures. Shown is the percentage of genes in the MDS or MCDS that also belong to the list of top genes with respect to degree, betweenness and closeness centrality

MDS nodes tend to be central in the network (high closeness) and belong to the most connected notes (highest degree).

When considering only outdegree nodes in the directed network, most of the top nodes of the MCDS have the highest overlap with the top nodes of the degree centrality and the betweenness centrality
(→ connector nodes).

Nazarieh et al. BMC Syst Biol 10:88 (2016)

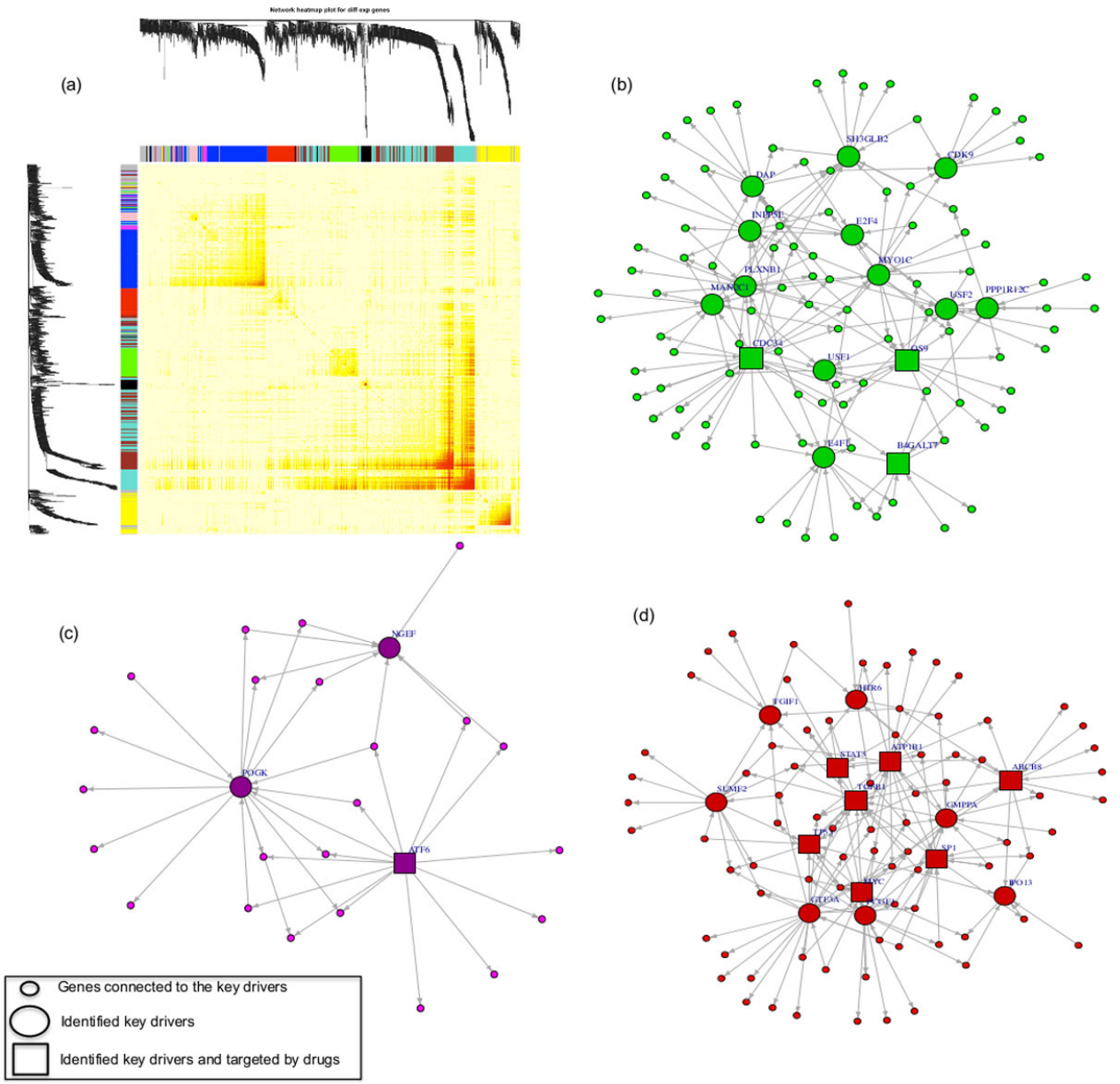
Breast cancer network

Analyze breast cancer data from TCGA →
ca. 1300 differentially expressed genes.

Hierarchical clustering of co-expression network yielded 10 segregated network **modules** that contain between 26 and 295 gene members.

Add regulatory info from databases Jaspar, Tred, MSigDB.

(b) – (d) are 3 modules.



Hamed et al. BMC Genomics 16 (Suppl5):S2 (2015)

Breast cancer network

The MDS and MCDS sets of the nine modules
contain 68 and 70 genes, respectively.

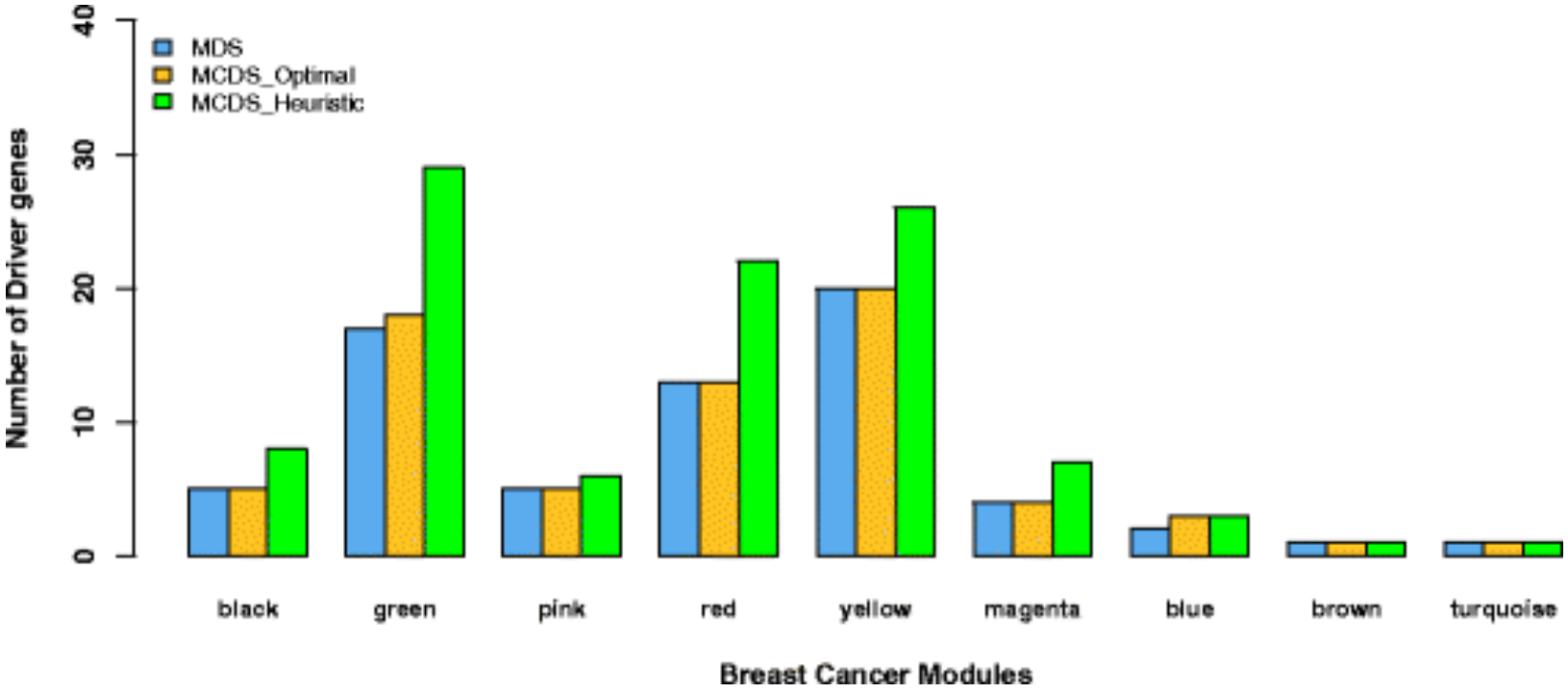
Intersect the proteins encoded by these genes with the targets
of anti-cancer drugs.

20 of the 70 proteins in the MCDS are known drug targets
(p = 0.03, hypergeometric test against the network
with 1169 genes including 228 drug target genes).

Also, 16 out of the 68 proteins belonging to the MDS genes
are binding targets of at least one anti-breast cancer drug.

Nazarieh et al. BMC Syst Biol 10:88 (2016)

|MDS| ≤ |MCDS|



Number of MCDS genes determined by the heuristic approach or by the ILP formulation and in the MDS.

Shown are the results for 9 modules of the breast cancer network

Nazarieh et al. BMC Syst Biol 10:88 (2016)

Summary

Today:

- network co-expression modules are best identified by ICA

- Network **motifs**: FFLs, SIMs, DORs are overrepresented
  → different functions, different temporal behavior

- Key pathway miner algorithm determines key genes.

- MDS and MCDS identify candidate master regulatory genes
  → who reliable are they when applied to noisy and incomplete data?

Next lecture V15:

- Epigenetics, analysis of DNA methylation data