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?

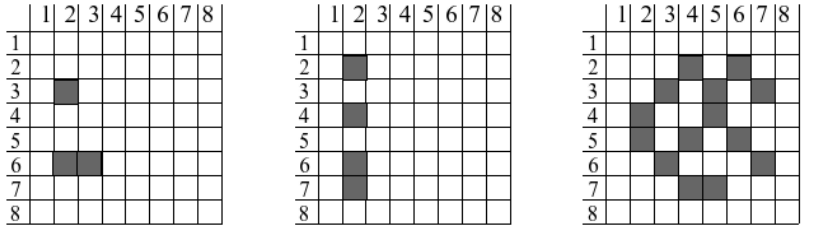
→ behavior of the motifs <=> location in the network?

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

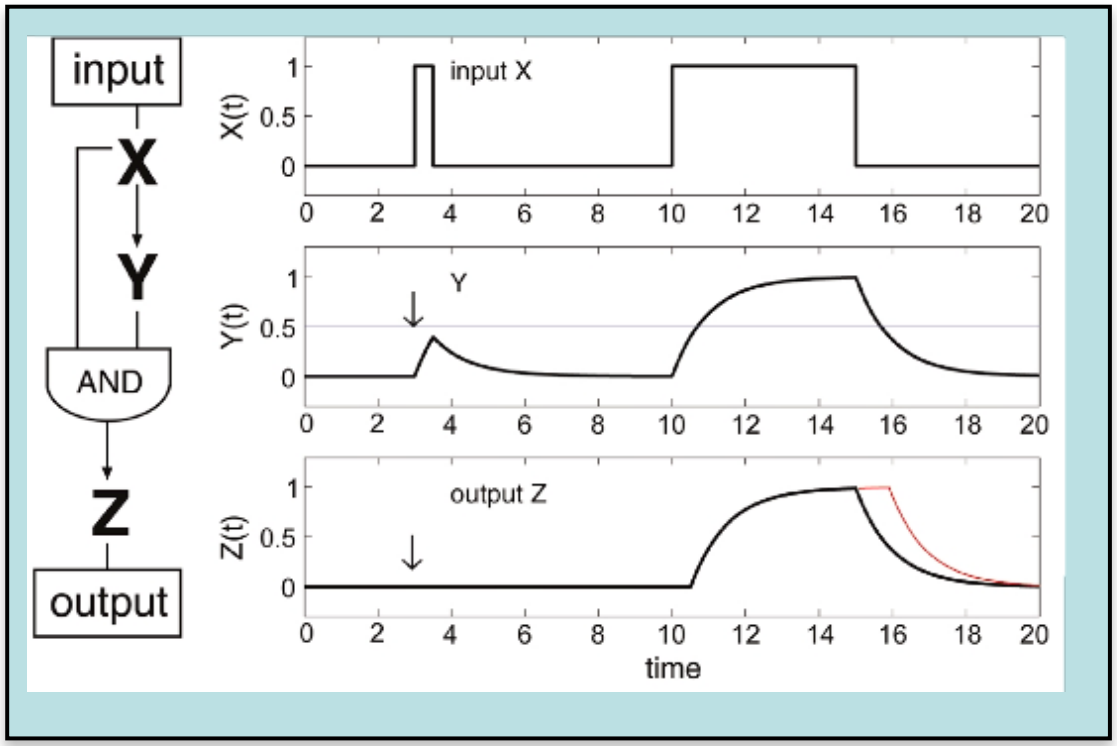
This yields networks with precisely the same number of nodes with *p* incoming and *q* outgoing nodes, as the real network.

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

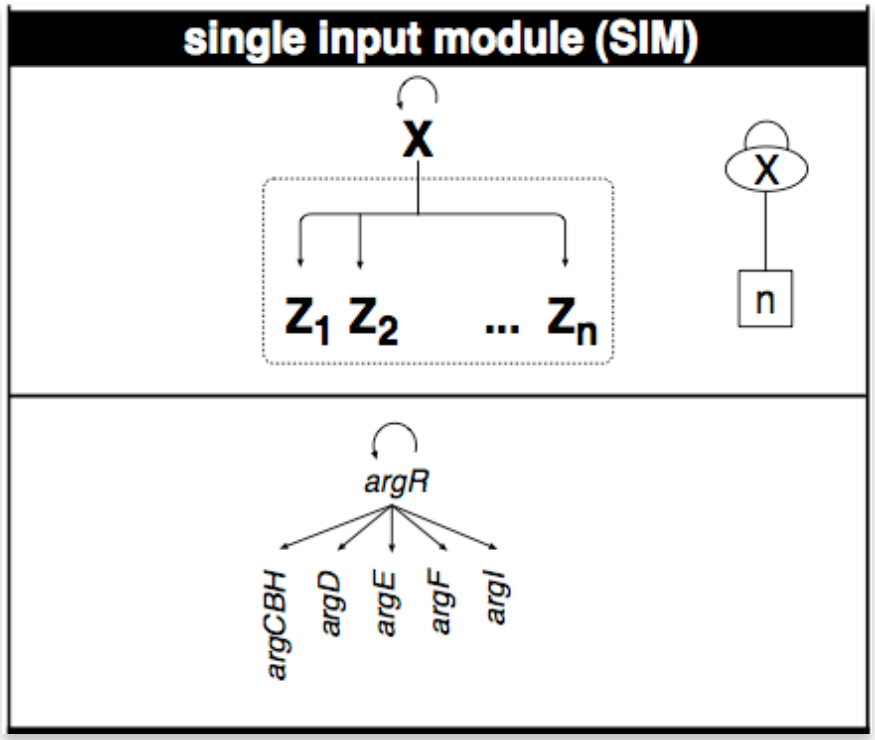
Dynamics:
- input activates X
- X activates Y (delay)
- (X && Y) activates Z

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

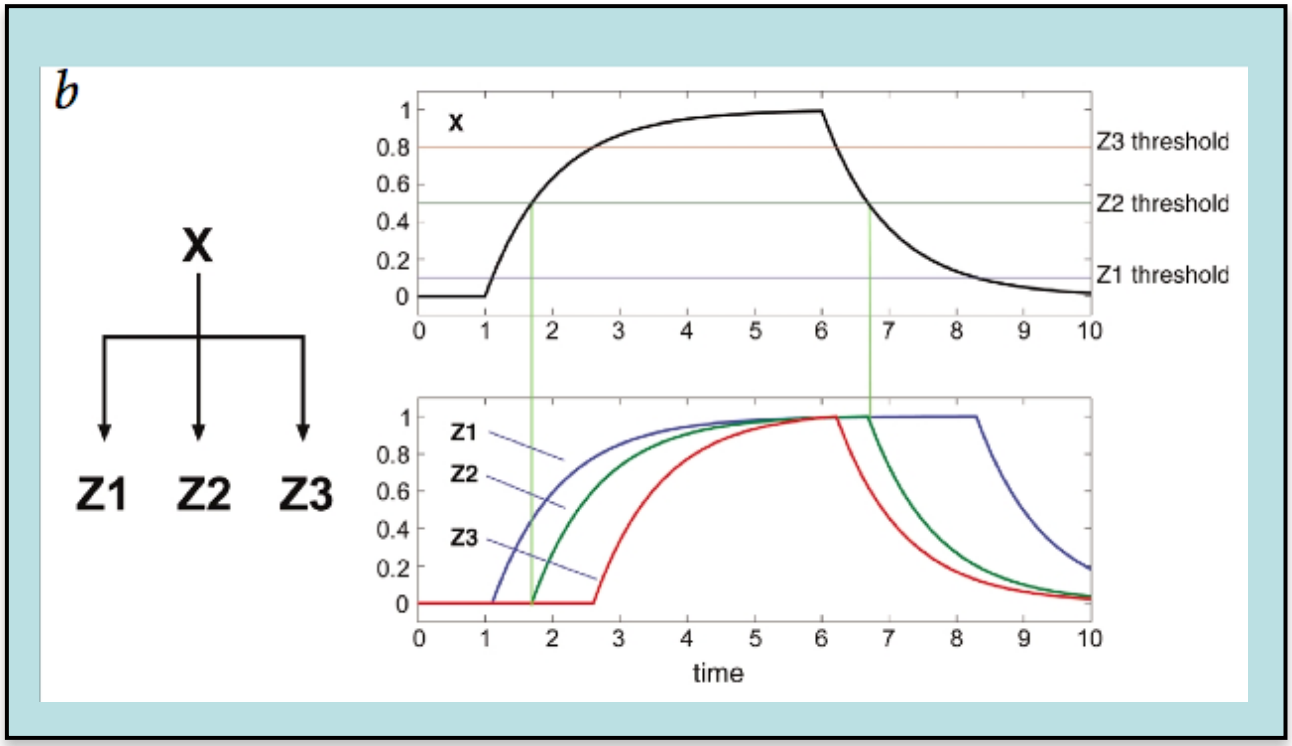
- same sign
- no additional regulation
- control is usually autoregulatory (70% vs. 50% overall)

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



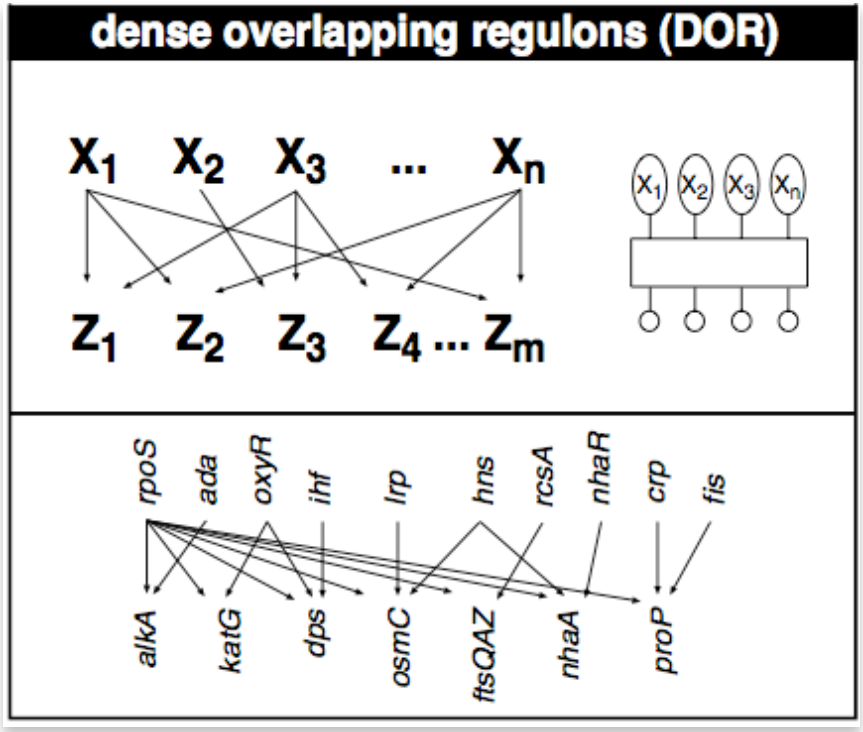
If different thresholds exist for each regulated operon:

→ 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)

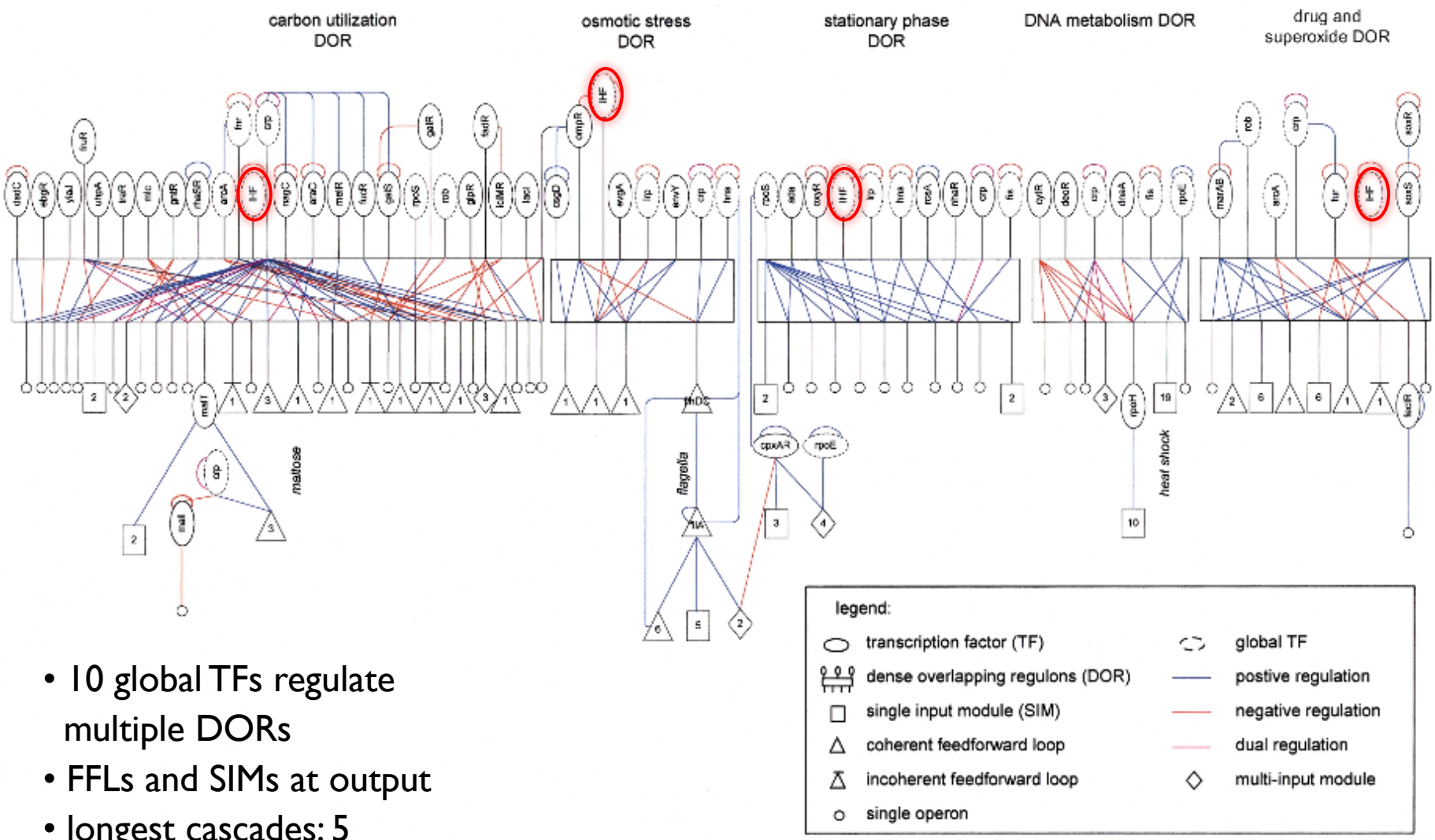
Usually each operon is regulated by a different combination of TFs.

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 $S(U)$.

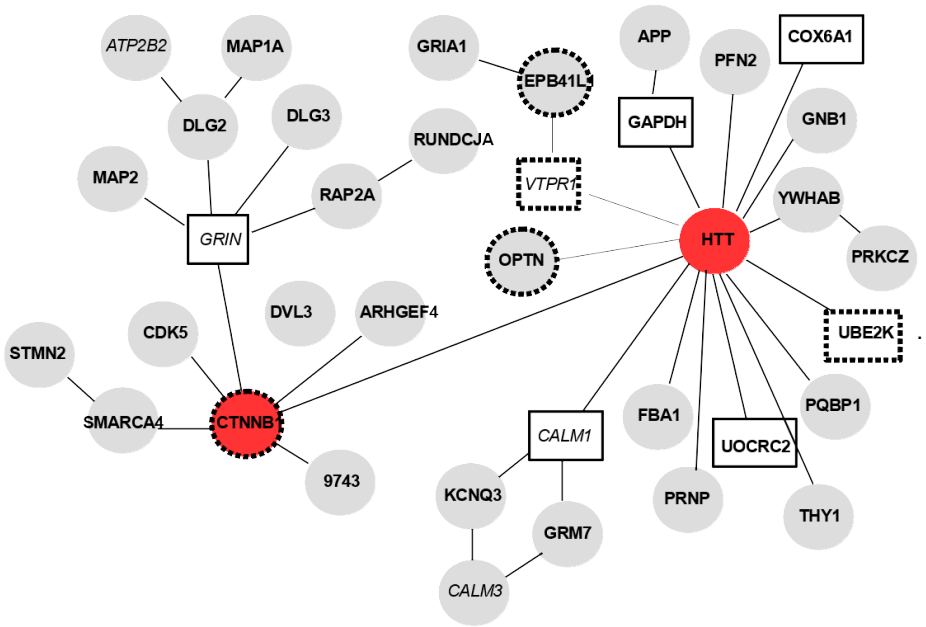
Alcaraz et al. (2012),
Integrative Biology 4, 756-764.

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 $|S(W_u \cup \{v\})|$.

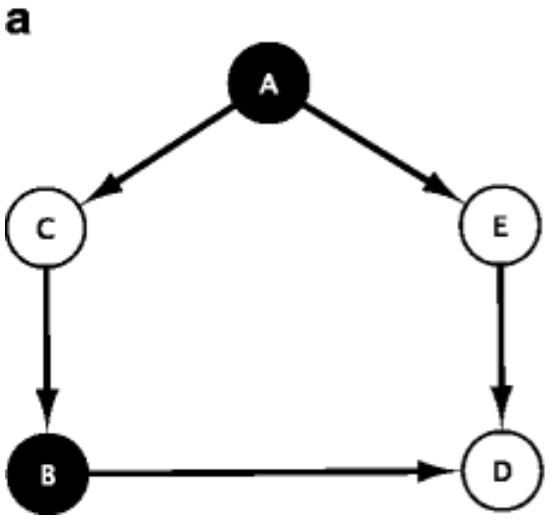
The iterations are stopped when $|W_u| = k$. The algorithm returns $S(W_u)$ of maximal size found for some u.



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)*.

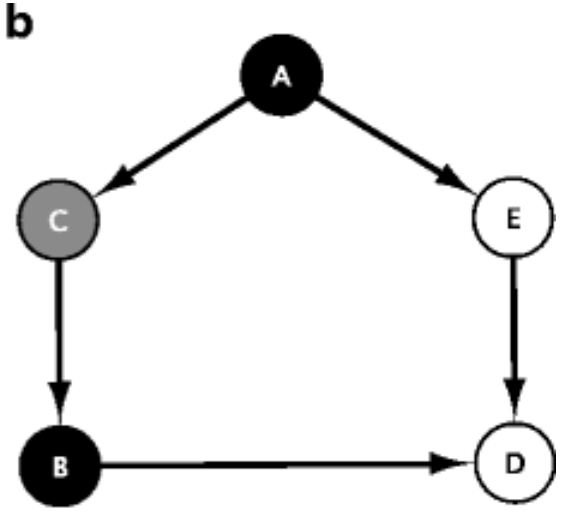
Idea: find a **set of dominator nodes** of minimum size that controls all other vertices.

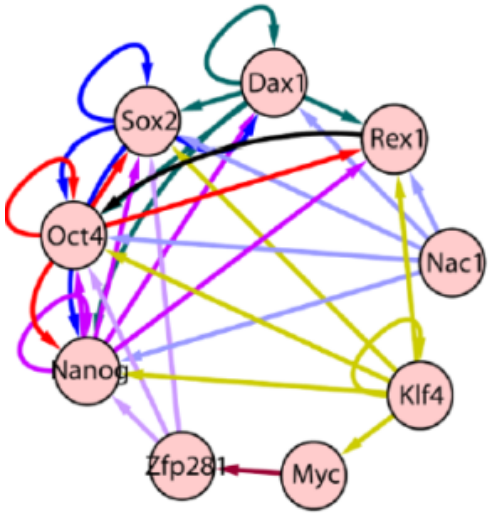
In the case of a GRN, a directed arc symbolizes that a transcription factor regulates a target gene.

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)