Motifs in Biological Networks

Proteomes Interactomes and Biological Networks

November 26, 27 and 29, 2019

Emidio Capriotti
http://biofold.org/

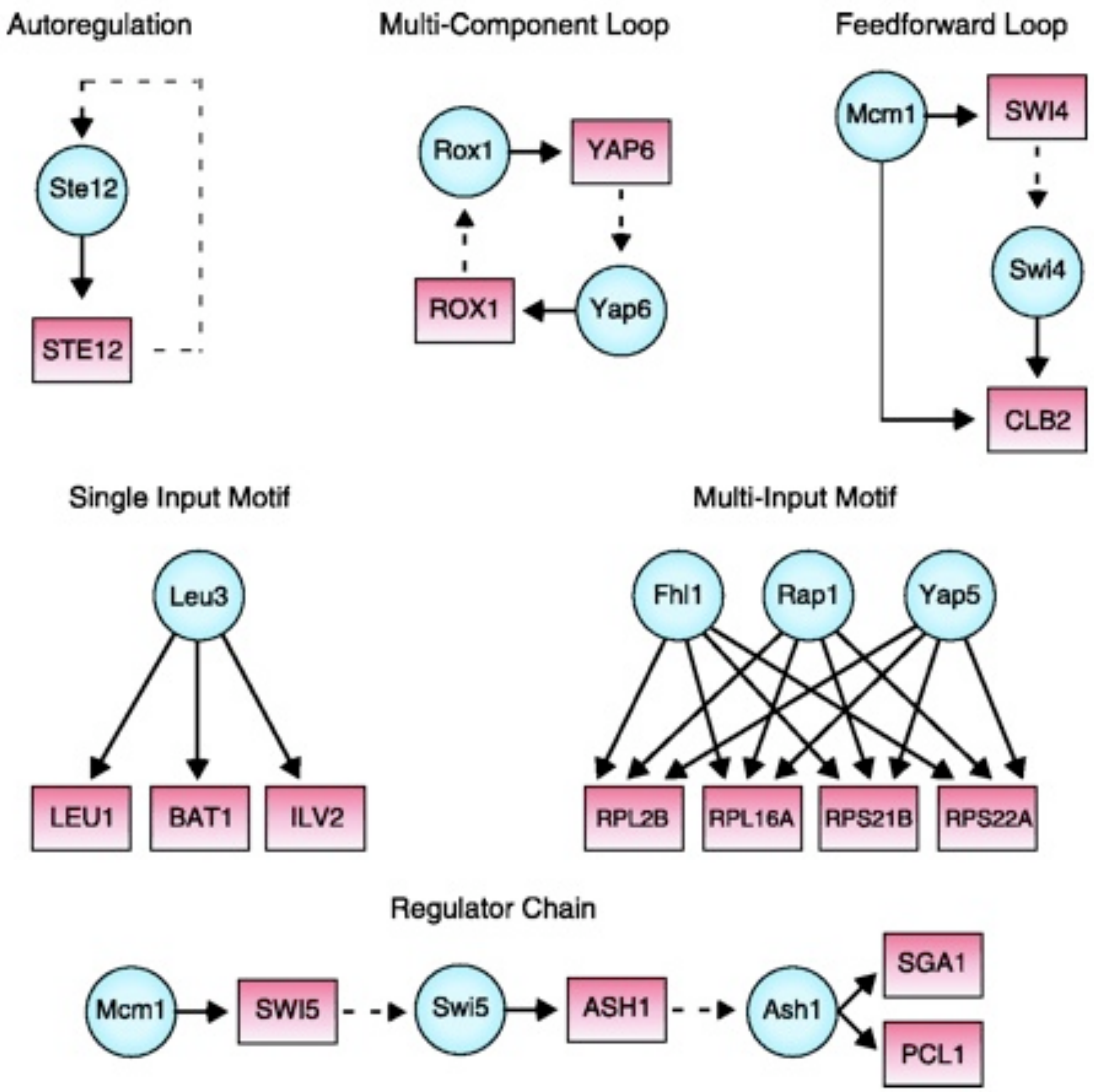
Department of Pharmacy and
Biotechnology (FaBiT)
University of Bologna

Biomolecules
Folding and
Disease

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
A.D. 1088

Network Motifs

Network analysis is important for detecting network motifs, which are recurrent and statistically significant sub-graphs or patterns.



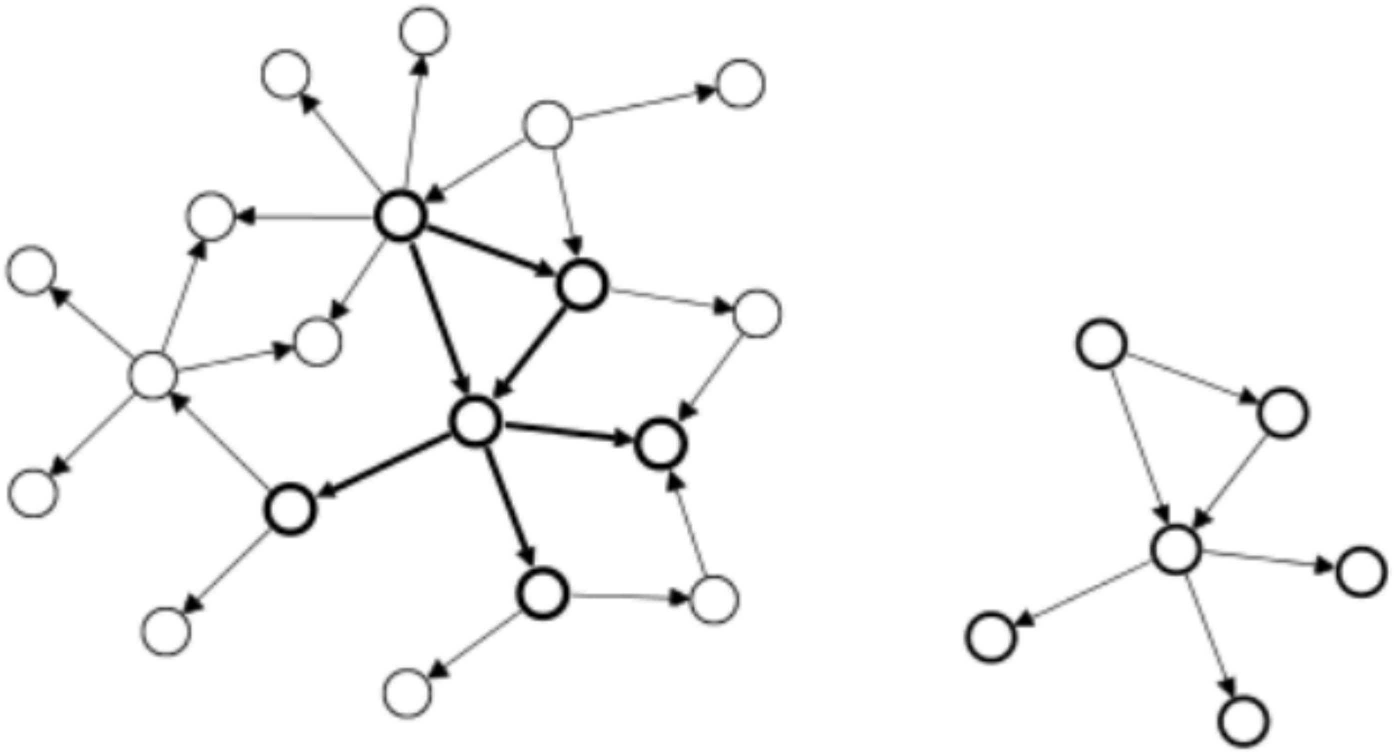
http://compbio.pbworks.com/

Motif Matching

A match of a motif G’ in the target graph G = (V, E) is a subgraph G” = (V”, E”) which is isomorphic to motif G’

Two graphs G’ and G” are isomorphic if there is a bijective mapping between the edge and vertex identities

i.e. G‘ is transformed to G” by changing the vertex and edge identities



Problem Complexity

The complexity of graph isomorphism is in the ’grey area’ of complexity:

- It belongs to NP class of problems (problems where solution is easy to verify once found)
- It is not known if graph isomorphism belongs to P class of problems (problems that can be solved efficiently)
- It is not known if graph isomorphism is NP-complete (problems that are believed to be hard ot solve but easy to verify)
- Subgraph isomorphism, checking if a subgraph G” that is isomorphic to given graph G’ exists in a larger graph G, is known to be NP-complete
- No hope for really fast algorithms for finding motifs.

Statistical Significance

A motif is a statistically overrepresented pattern of local interactions in the network

- Overrepresentation = occurring more frequently than expected by chance

- The motif has emerged several times therefore it has been conserved in the evolution of the network

- The rationale is that overrepresentation may denote possible conservation of the function

Significance tests

The statistical significance can be tested calculating the z-score of the presence of the motif on a set of randomly generated graphs obtained with

- Generation of random networks with the Erdos-Renyi algorithm

- Random shuffling of the edges

Detection of Motifs

Networkx allow to select a subgraph of a the whole graph and verify if two graphs are isomorphic

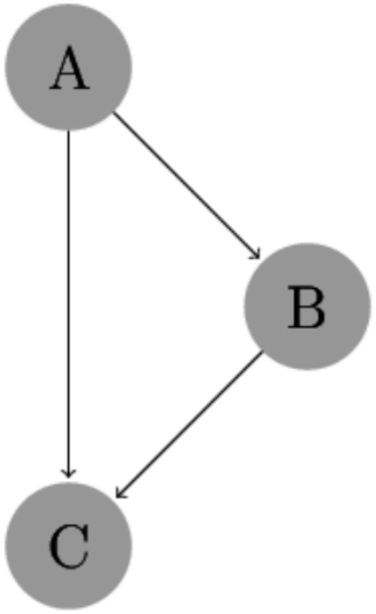
```
>>> g = nx.Graph().add_nodes_from([(1,2),(1,3)])

>>> mot = nx.Graph().add_nodes_from([(“A”,”B”)])

>>> g1 = g.subgraph([1,2])

>>> nx.is_isomorphic(g1,mot)
```

Exercise

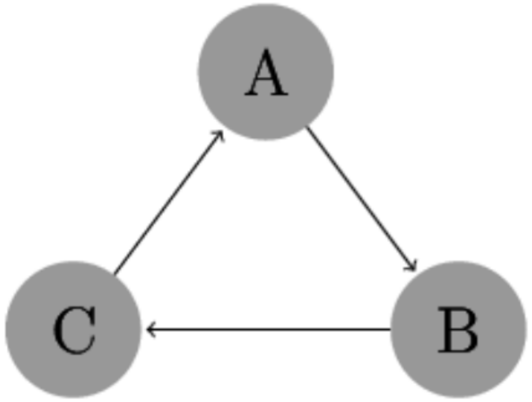
Given the Feed Forward Loop (FFL) and 3-Cycle write the code to detect the motif in the graph with 6 nodes and 8 edges.

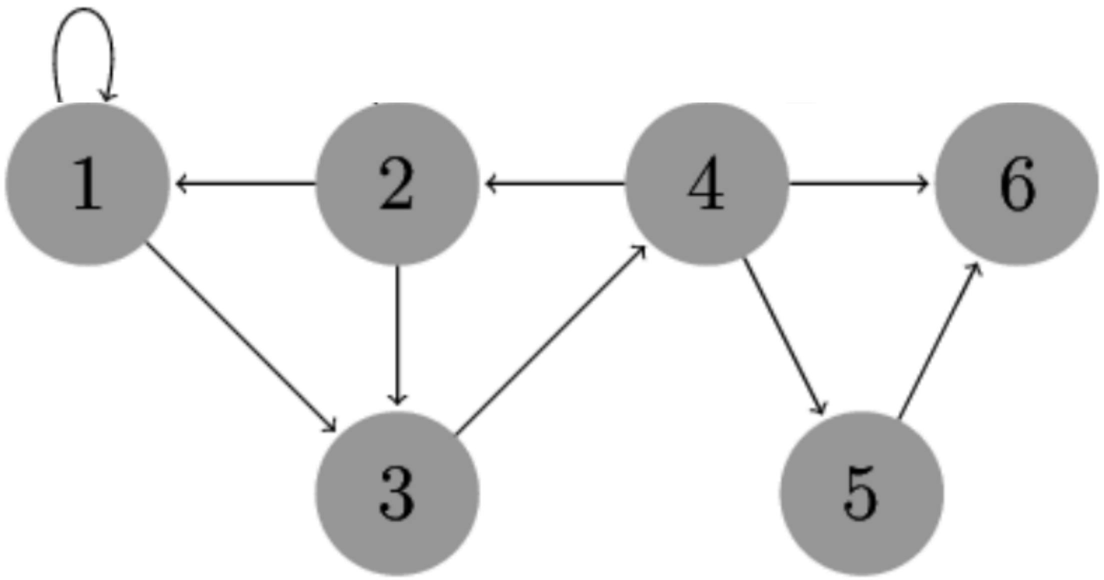
Calculate occurrence on random network and the z-score.

Feed Forward Loop



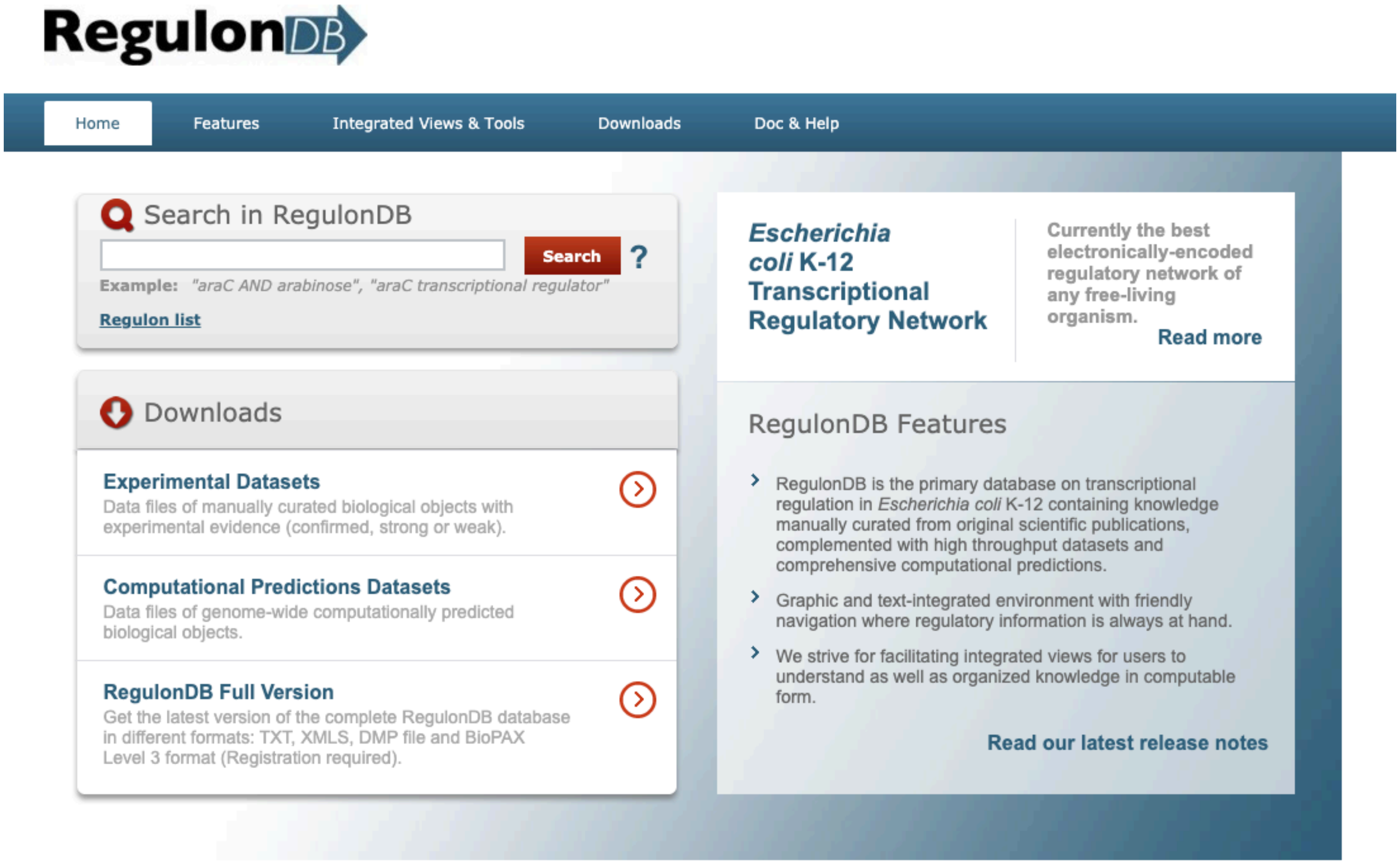
3-Cycle





RegulonDB

Database of Escherichia coli K-12 Transcriptional Regulatory Network



Search in RegulonDB

Example: *"araC AND arabinose", "araC transcriptional regulator"*

Regulon list

Downloads

Experimental Datasets
Data files of manually curated biological objects with experimental evidence (confirmed, strong or weak).

Computational Predictions Datasets
Data files of genome-wide computationally predicted biological objects.

RegulonDB Full Version
Get the latest version of the complete RegulonDB database in different formats: TXT, XMLS, DMP file and BioPAX Level 3 format (Registration required).

***Escherichia coli* K-12 Transcriptional Regulatory Network**

Currently the best electronically-encoded regulatory network of any free-living organism.
Read more

RegulonDB Features

- RegulonDB is the primary database on transcriptional regulation in *Escherichia coli* K-12 containing knowledge manually curated from original scientific publications, complemented with high throughput datasets and comprehensive computational predictions.
- Graphic and text-integrated environment with friendly navigation where regulatory information is always at hand.
- We strive for facilitating integrated views for users to understand as well as organized knowledge in computable form.

Read our latest release notes

http://regulondb.ccg.unam.mx/

Regulation Data

The regulation data includes information about the transcription factors (TF) that activate or repress the expression of the genes with associated supporting evidences.

```
# Release: 10.6.2 Date: 10-04-2019
# ______________________________________________#
# Columns:
#  (1) Transcription Factor (TF) name
#  (2) Gene regulated by the TF (regulated gene)
#  (3) Regulatory effect of the TF on the regulated gene (+ activator, - repressor, +- dual, ? unknown)
#  (4) Evidence that supports the existence of the regulatory interaction
#
AcrR    acrA    -       [BCE, BPP, GEA, HIBSCS] Strong
AcrR    acrB    -       [BCE, BPP, GEA, HIBSCS] Weak
AcrR    acrR    -       [AIBSCS, BCE, BPP, GEA, HIBSCS] Weak
AcrR    flhC    -       [GEA, HIBSCS]   Weak
AcrR    flhD    -       [GEA, HIBSCS]   Weak
AcrR    marA    -       [BPP, GEA, HIBSCS]      Strong
AcrR    marB    -       [BPP, GEA, HIBSCS]      Strong
AcrR    marR    -       [BPP, GEA, HIBSCS]      Strong
AcrR    micF    -       [AIBSCS]        Weak
AcrR    soxR    -       [BPP, GEA, HIBSCS]      Strong
```

http://regulondb.ccg.unam.mx/menu/download/datasets/files/network_tf_gene.txt

Nodes ad Edges

With networkx we can assign attributes to nodes and edges

```
>>> G=nx.DiGraph()

>>> G.add_node(1, color=‘blue’)

>>> G.add_node(2, color=‘red’)

>>> G.add_edge(1, 2, sign=‘+’)

>>> G.node[1]

>>> G.edge[1][2]
```

Matches Node and Edges

Matches can be performed based on node and edges attributes

```
>>> import networkx.algorithms.isomorphism as iso

>>> em=so.categorical_edge_match(‘sign'=‘+’)

>>> nm=iso.categorical_node_match(‘color'=‘red’)

>>> nx.is_isomorphic(G1,G2,edge_match=em, node_match=nm)
```

Exercise

Write a program to analyze the RegulonDB network considering only data with strong supporting information.

- Find the TF that regulates more genes (activation and suppression)
- Find the gene that is regulated by more TFs
- Match a graph that contains a TF activating three genes.