Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots.

So far, the Universe is winning. - Unknown
Biological Networks

Questions:
- Which are the most common motifs among biological networks?
- What are the information processing functions of these motifs?
- What is the difference between a sub graph and a motif?
- How can we detect motifs?
Revision: Solving Differential Equations

- How do we solve \[ \frac{dY}{dX} = AY \]?
- How about \[ \frac{dY}{dX} = B - AY \]?
- Can you plot Y Vs X in each of these cases?
Sub graph Patterns

Let us consider all possible patterns which can appear in directed networks with:
- size = 3
- size = 4
- Etc.

Such combinations are called ‘sub graph patterns’ - all of them may not occur in a given network.

Number of sub graph patterns increases if we consider the ‘activating / inhibiting’ nature of a directed link.
Three Node sub graph patterns
Four Node Sub graph patterns
Network Motifs

What are “Network Motifs”? 

- Network Motifs are defined as patterns of interconnections that recur in many different parts of a network at frequencies much higher than those found in randomized networks.

Why do we need them?

- To help us understand how biological networks work.
- Exact forecasting of operation and reaction in the network under given situations.
The concept of “network motifs” was first proposed by Uri Alon’s group.
Schematic View of Network Motif Detection

A) Real network

B) Randomized networks

motif:
Detecting Network Motifs: Zscores

- Let us consider $N$ number of random networks with the same size etc.
- What is the average number of a given sub graph pattern in the random networks?
- What is the standard deviation?
- The Zscore of a sub graph pattern can be calculated as:

$$Z_i = \left( \frac{N_{i}^{\text{real}} - < N_{i}^{\text{rand}} >}{\text{std}(N_{i}^{\text{rand}})} \right)$$

- If $|z_i| > 3$, then the sub graph pattern can be considered a motif.
Detecting Network Motifs

- Edges easily lost/added
- Compare real networks to randomized networks
- Patterns that occur more often in real networks = Network motifs
Detecting Network Motifs (Cont.)

- $N$ nodes

- possible pairs of nodes: $[N(N-1)] + N = N^2$

- edge position is occupied: $p = \frac{E}{N^2}$
Examples for motifs

- **FeedForward Loop**
  Found in neural networks.
  It seems to be used to neutralize “Biological Noise”. That is, it controls pulses.

- **Single-Input Module**
  Implemented in gene control networks
Examples for motifs

- Parallel paths

Found in neural networks, food webs etc.
(and not so much in gene networks)
Biological Network motifs

- Autoregulation (AR)
- Feed Forward Loops (FFL)
- Regulating and Regulated Feedback Loops (RFL)
- Cascade

Single Input Model (SIM)
Dense Overlapping Regulon (DOR)
BiFan
Diamond
Autoregulation
AS A NETWORK MOTIF
Auto regulation

- Regulation of a gene by its own gene product
- How does it look in the graph?

- E. coli network:
  - 40 self edges
  - 34 repressors
  - 6 activators
Cont.) (Autoregulation)

- **Probability for self edge:**  $P_{self} = 1/N$

- **Expected number of self edges:**  $<N_{self}>_{rand} \sim E \cdot P_{self} \sim E/N$

- **Standard deviation:**  $\sigma_{rand} \sim \sqrt{E/N}$
Cont.) (Autoregulation

- Number of self edges:

<table>
<thead>
<tr>
<th>Network Type</th>
<th>$\langle N_{self} \rangle_{\text{rand}} \sim 1.2$, $\sigma_{\text{rand}} \sim \sqrt{1.2} \sim 1.1$</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E. coli network</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Conclusion: Self edges are a network motif

- But... why?
Negative Autoregulation

\[ pX \rightarrow \text{geneX} \rightarrow \text{protein X} \]
Negative Autoregulation - Response time

► Reminder:
  ► Logic input function:
    ► $X(t) \sim \beta t$ when $X < K$ and $X \ll \beta/\alpha$

► Steady-state level:
  $X_{st} = K$

► Response time:
  \[
  \beta T_{1/2}^{(n.a.r.)} = \frac{K}{2} \Rightarrow T_{1/2}^{(n.a.r.)} = \frac{K}{2\beta}
  \]
## Negative Autoregulation

Response time (Cont.)

### Response time comparison:

<table>
<thead>
<tr>
<th>Simple regulation</th>
<th>Negative autoregulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_{st} = \frac{\beta_{\text{simple}}}{\alpha_{\text{simple}}} )</td>
<td>( X_{st} = K )</td>
</tr>
<tr>
<td>( K = \frac{\beta_{\text{simple}}}{\alpha_{\text{simple}}} )</td>
<td></td>
</tr>
<tr>
<td>( T_{\text{simple}}^{1/2} = \frac{\log 2}{\alpha_{\text{simple}}} )</td>
<td>( T_{\text{1/2}}^{(\text{n.a.r.})} = \frac{K}{2\beta} )</td>
</tr>
<tr>
<td>( \frac{T_{\text{1/2}}^{(\text{n.a.r.)}}}{T_{\text{1/2}}^{\text{simple}}} = \frac{K\alpha_{\text{simple}}}{2 \log 2\beta} = \frac{\beta_{\text{simple}}}{2 \log 2\beta} )</td>
<td></td>
</tr>
</tbody>
</table>
Negative Autoregulation - Response time (Cont.)

\[
\frac{T_{1/2}^{(n.a.r.)}}{T_{1/2}^{\text{simple}}} = \frac{K\alpha_{\text{simple}}}{2 \log 2\beta} = \frac{\beta_{\text{simple}}}{2 \log 2\beta}
\]
Negative Autoregulation - Robustness

- Production rate ($\beta$) fluctuates over time

<table>
<thead>
<tr>
<th>Simple regulation</th>
<th>Negative autoregulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{st} = \frac{\beta_{simple}}{\alpha_{simple}}$</td>
<td>$X_{st} = K$</td>
</tr>
</tbody>
</table>
THE FEED FORWARD LOOP (FFL) AS A NETWORK MOTIF
The Coherent (a) and Incoherent (b) feedback Loops

- With the Coherent FFL, Y activates Z
- With the Incoherent FFL, Y inhibits Z
Three nodes subgraphs

- 13 possible three-nodes patterns
- Which ones are motifs?
Cont.) (Three nodes subgraphs

- Sub graph G with n nodes and g edges

- $N^2$ possibilities to place an edge

- Probability of an edge in a given direction between a given pair of nodes: $p = E/N^2$
Cont.) (Three nodes subgraphs

- Mean number of appearances:
  \[ <N_G> \approx a^{-1} N^n p^g \]

- Mean connectivity: \( \lambda = \frac{E}{N} \) -> \( p = \frac{\lambda}{N} \)

\[ <N_G> \approx a^{-1} \lambda^g N^{n-g} \]
Cont.) (Three nodes subgraphs

How \( \langle N_G \rangle \) scales with the network size?

\[ \langle N_G \rangle \sim N^{n-g} \]

Triangle-shaped patterns (3 nodes and 3 edges):

- Feed-forward loop
  \[ \langle N_{FFL} \rangle \sim \lambda^3 N^0 \]

- 3-node feedback loop (cycle)
  \[ \langle N_{3\text{loop}} \rangle \sim \frac{1}{3} \lambda^3 N^0 \]
Cont.) (Three nodes subgraphs

<table>
<thead>
<tr>
<th></th>
<th>FFL</th>
<th>3LOOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>E. coli</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>Random net</td>
<td>1.7</td>
<td>0.6</td>
</tr>
</tbody>
</table>

☐ FFL is the only motif of the 13 three- node patterns
FFL- Structure

▶ E. coli example:
FFL- Structure (Cont.)

COHERENT FFL

- Coherent type 1
- Coherent type 2
- Coherent type 3
- Coherent type 4

INCOHERENT FFL

- Incoherent type 1
- Incoherent type 2
- Incoherent type 3
- Incoherent type 4
Relative abundance of FLL types in yeast and E. coli:
FFL- Structure (Cont.)

- Logic function
- AND logic
- OR logic

- X and Y respond to external stimuli
Coherent Type-1 FFL – AND logic

- $X^*$ appear, $X$ rapidly changes to $X^*$
- $X^*$ binds to gene $Z$, but cannot activate it
- $X^*$ binds to gene $Y$, and begins to transcript it
- $Z$ begins to be expressed after $T_{on}$ time, when $Y^*$ crosses the activation threshold $K_{yz}$
Feed forward Loop: Example
Coherent Type-1 FFL – AND logic (Cont.)

- **definition:**
  - **ON step** - Sx moves from absent to saturated state
  - **OFF step** - Sx moves from saturated to absent state

- Sy is present continuously
The Time Delay in Z

We can notice that after the Signal $S_x$, there is a delay $T_{ON}$ before $Z$ increases.
Coherent Type-1 FFL – AND logic (Cont.)

On step-

- $Y^*(t) = Y_{ST}(1-e^{-\alpha_y t})$

- $Y^*(T_{ON}) = Y_{ST}(1-e^{-\alpha_y T_{ON}}) = K_{yz}$

- $T_{ON} = 1/\alpha_y \log\left[1/(1-K_{yz}/Y_{st})\right]$
Coherent Type-1 FFL – AND logic (Cont.)
No Time Delay when switching OFF

However, there is no such time delay when $S_x$ switches off. For this reason, this motif is called a
The Coherent FFL as a filter of pulses

We can note that Coherent FFL shows no response to ‘slim’ pulses, but lets through ‘fat’ pulses. It is
Coherent Type-1 FFL – OR logic

- Delay for OFF Steps of Sx
- Flagella system of E. coli:
  - $T_{OFF} = 1$ hour
The Incoherent FFL as a pulse generator

The Incoherent FFL, on the other hand, can work as a ‘pulse generator’. Notice that the signal in $S_x$ results in a ‘pulse’ in $Z$. 

![Graph showing the relationship between $S_x$, $Y$, and $Z$ over time.](image)
Incoherent Type-1 FFL-Dynamics (Cont.)

- Dynamic equation of Z:
  - $Y^* < K_{yz}$
    - $\frac{dZ}{dt} = \beta_z - \alpha_z Z$
    - $Z_m = \frac{\beta_z}{\alpha_z}$
    - $Z(t) = Z_m (1 - e^{-\alpha Z t})$
  - $Y^* > K_{yz}$
    - $\frac{dZ}{dt} = \beta'_z - \alpha'_z Z$
    - $Z_{st} = \frac{\beta'_z}{\alpha'_z}$
    - $Z(t) = Z_{st} + (Z(T_{rep}) - Z_{st}) e^{-\alpha (1 - T_{rep})}$
    - $Y^*(T_{rep}) = Y_{ST} (1 - e^{-\alpha y T_{rep}}) \Rightarrow T_{rep} = \frac{1}{\alpha_y} \ln \left[ 1/(1 - K_{yz}/Y_{ST}) \right]$
The ‘Height’ of the Pulse depends on the threshold value.
Incoherent Type-1 FFL-Example (Galactose)
Summary of Coherent and Incoherent FFLs.

- Note that the Coherent FFL introduces a delay in Z.
- Coherent FFL - allows the signals through only if they have a certain width.
- Now consider the case where the And function is replaced by the Or function.
- This also introduces a delay, but now on the ‘OFF’ signal rather than the ‘On’ signal.
What do we mean by coding standards and best practices?

- Good coding standards and practices are necessary to ensure software quality

- Coding – Aesthetic issues
  - Naming the variables
  - Capitalization
  - Modularity
  - Language specific practices
  - How important are these? Important enough to make your project collapse...

- Formal methods to ensure software Quality
- Software Engineering Process Models
Properties

- Coherent FFL - Sign sensitive delay element
- And a filter
- Incoherent FFL - Pulse generator
The SIMs are common in sensory transcription networks:

- Genes from a same Pathway (Arginine synthesis).
- Genes responding to stress (DNA repair).
- Genes that assemble a same biological machine (ribosomal genes).
The SIMs can generate temporal programs of expression Eg: Last In First Out Order.

**Last-In First-Out (LI FO) Program**
Topological Generalization of network Motifs

- Three node Sub graphs - 13
- Four Node Sub graphs - 199
- Five Node Sub graphs – Over 9000
- Seven Node Subgraphs – Million

Fortunately, Most Biological Network shows families of dominant motifs

- Eg: Transcription Networks Show
  - Feed Forward Loops (FFL)
  - Dense Overlapping Regulons (DOR)
  - Single Input Modules (SIM)
Generalizing motifs – Role replication
First-In First-Out (FIFO) Program

\[ K_{xz1} > K_{xz2} > K_{xz3} \quad K'_{xz1} < K'_{xz2} < K'_{xz3} \]

July 2013
N. H. N. D. de Silva
FIFO program

- The FFL works as a FIFO program here because of the OR functions at Zi
- Notice that threshold levels have to be reversed for Y and Z
Aside: E Coli Flagella – Technological wonder

Single cell, 1 micron length
Contains only \( \sim 1000 \) protein types at any given moment

still: Amazing technology

![Diagram of E Coli Flagella](image.png)
FIFO program is governed by a FFL.
Multi-input FFL in Neuronal Networks

Nose Touch → FLP
Noxious Chemicals → Nose Touch → ASH

AVD

AVA

Backward movement

July 2013

N. H. N. D. de Silva
Dense Overlapping Regulon (DOR)
DORs

- The Genes in each DOR have a shared Global function
  - Such as stress response
  - Nutrient metabolism
  - Biosynthesis of key classes of cellular components
- The DOR forms the backbone of networks global structure
- So can we use motifs to simplify networks?
How do Network Motifs Integrate?

A Master Regulators Layer (lots of Auto-Reg.)

Where are the X→Y→Z?

A single DOR Layer

FFLs and SIMs are integrated within DORs

The E. coli Transcription Network (partial)
Compare with...
DO Rs in transcription networks

- Form single layers - do not form cascades
- No ‘Single line’ cascades
  - Rate limited networks tend not to employ such cascades
  - Cascades are found in networks with interactions that are rapid compared to the timescale in which the network needs to function.
Network Motifs in developmental transcription networks

- These are not rate limited
Developmental Transcription Networks

Drosophila melanogaster (parallel)

The TF expression profile in a developing Drosophila embryo
Developmental Transcription Networks

Two-node Feedback Loops - Locking

- Both $X$ AND $Y$ are ON at the same time.
- Genes regulated by $X$ and $Y$ belong to the same tissue (or strip).

- $X$ OR $Y$ is ON at a given time.
- Genes regulated by $X$ and $Y$ belong to different tissues (strips).
Developmental Transcription Networks

Regulating Feedback Loops

Double Positive Loops

Regulated Feedback Loops

Double Negative Loops
Developmental Transcription Networks

Regulated Feedback Loops as a Memory Element

![Diagram of regulated feedback loops](image)
Developmental Transcription Networks

Cascades

\[ X \rightarrow Y \rightarrow Z \]

\[ X \rightarrow Y \rightarrow Z \]

\[ [X] \]

\[ [Y] \]

\[ [Z] \]

\[ [X] \]

\[ [Y] \]

\[ [Z] \]

Time

July 2013

N. H. N. D. de Silva
Feed Back Loops in developmental transcription networks

- X transcriptionally activates X.
- Y inhibits X.
- Z transcriptionally activates Y.
- X forms a complex with Y.
- X phosphorylates Y.

Power $\rightarrow$ Heater $\rightarrow$ Temperature

Thermostat

(Fast) Protein-Protein Interactions
(Slow) Transcriptional Interactions
Feed Back Loops Produce Oscillation (remember from Control theory?)

Cdc20 oscillator controls Cell Cycle

Mutation of the Drosophila CWO gene
The critically damped response of the oscillator is described by the equation:

$$x = e^{-\gamma t} [x_0 + (\nu_0 + \gamma x_0)t]$$

which is a combination of an exponential and a linear term.

Oscillator with resonant frequency 10 rad/s started from rest. After Barger & Olsson.
Cell Signaling networks

- What are cell signaling networks?
Signal Transduction Cascades
Popular Motifs in Signal Transduction Cascades

Generalization of DOR

Multi-layer Perceptrons (multi-DORs)
Multi Layer Perceptrons in Signal Transduction Cascades
Dynamics of Signal Transduction Cascades

At Steady State,

\[
\frac{dY}{dt} = v_1 X_1 Y_o + v_2 X_2 Y_o - \alpha Y_p
\]

\[
\frac{dY}{dt} = 0
\]

\[
\frac{Y_p}{Y} = f\left(w_1 X_1 + w_2 X_2\right)
\]

\[
w_2 = \frac{v_2}{\alpha} , \quad w_1 = \frac{v_1}{\alpha}
\]

\[
w_1 X_1 + w_2 X_2 > 1
\]

Activation Threshold

\[
w_2 = 0.7 \quad w_1 = 0.7
\]
Dynamics of Signal Transduction Cascades

"AND" gate

\[ w_2 = 0.7 \quad w_1 = 0.7 \]

"OR" gate

\[ w_2 = 2 \quad w_1 = 2 \]
Dynamics of Signal Transduction Cascades

\[
Z_p / Z = f(w_{z_1} Y_1 + w_z Y_2)
\]

“OR” gate

“AND” gate
Dynamics of Signal Transduction Cascades
Summary

- Network motifs can function in several biological processes (sensory systems, development).
  - different time scales (milliseconds, cell generations).

- Network motifs can produce temporal programs (LIFO, FIFO, oscillation).

- Motifs within a network may be arranged in organized structures (perceptrons, interlocking FFL).

- It is possible to understand network topology better by reducing the network to motif topology.
Graphlets
FURTHER READING
2.1) Graphlet degree distribution agreement between two networks

2.1) Graphlet degree distribution agreement between two networks

Graphlet Degree (GD) vectors, or “node signatures”

\[ \text{GDV}(u) = (2, 1, 1, 0, 0, 1) \]

2.1) Graphlet degree distribution agreement between two networks

Signature Similarity Measure between nodes $u$ and $v$

- $o_i$ is number of orbits that affect orbit $i \in \{0, \ldots, 72\}$
- $w_i = 1 - \frac{\log(o_i)}{\log(73)}$
- Distance between the $i^{th}$ orbits of nodes $u$ and $v$ is
  \[ D_i(u, v) = w_i \times \frac{|\log(u_i+1) - \log(v_i+1)|}{\log(\max\{u_i, v_i\} + 2)} \]
- The total distance between nodes $u$ and $v$ is
  \[ D(u, v) = \frac{\sum_{i=0}^{72} D_i}{\sum_{i=0}^{72} w_i} \]
- The signature similarity between nodes $u$ and $v$ is
  \[ S(u, v) = 1 - D(u, v) \]

Software that implements many of these network properties and compares networks with respect to them: **GraphCrunch**  
http://bio-nets.doc.ic.ac.uk/graphcrunch/
Software that implements many of these network properties and compares networks with respect to them: *GraphCrunch*
http://bio-nets.doc.ic.ac.uk/graphcrunch2/
Software that implements many of these network properties and compares networks with respect to them:

**GraphCrunch**

http://bio-nets.doc.ic.ac.uk/graphcrunch2/
Another Software: Cytoscape
http://www.cytoscape.org/

GraphletCounter: A software tool and Cytoscape plugin for computing graphlet degree signatures of nodes and motifs in biological networks

GraphletCounter is an open-source software tool for computing the graphlet signatures of nodes and motifs in biological networks. GraphletCounter can operate on its own or as a plugin to the network analysis environment Cytoscape. GraphletCounter computes the graphlet signatures of individual nodes or of motifs, which can be specified by files generated by the motif-finding tool infolder. It displays graphlet signatures visually within Cytoscape, and can output graphlet data for integration with larger workflows. Graphlet signatures were introduced by the Prud'homme group at the University of California at Irvine; please see the following references:


Also refer to the official home page for information on motif finding and a description of its output file format.

UPDATE: GraphletCounter now has a project website at Google code. Go there to check out the source code, and add your own improvements.
Examples of signatures and signature similarities:

Examples of signatures and signature similarities:

Examples of signatures and signature similarities:

![Signatures of proteins with similarities above 0.90](image)

Examples of signatures and signature similarities:

*Statistically significant threshold at ~85%

Later we will see how to use this and other techniques to link network structure with biological function.
Generalize Degree Distribution of a network

The degree distribution measures:
• the number of nodes “touching” $k$ edges for each value of $k$
For each of these 73 automorphism orbits, we count:

- the number of nodes touching a particular graphlet at a particular orbit.

⇒ The spectrum of 73 “graphlet degree distributions (GDDs)” for $G_0, \ldots, G_{29}$ measuring local structural properties of a network.

Illustration of comparing $i^{th}$ GDDs of two networks:

There are 73 such GDD agreements, one for each graphlet orbit. The total agreement between two networks is the average of these 73 GDD agreements.

Network “Agreement”

- \( d^j_G(k) \) is the number of nodes in \( G \) touching \( j^{th} \) orbit \( k \) times
- scale sample distribution \( d^j_G(k) \) as \( S^j_G(k) = \frac{d^j_G(k)}{k} \)
- normalize the distribution w.r.t. its total area \( T^j_G = \sum_{k=1}^{\infty} S^j_G(k) \)
- \( \Rightarrow \) “normalized distribution” is \( N^j_G(k) = \frac{S^j_G(k)}{T^j_G} \)
- “distance” between networks \( G \) and \( H \) at orbit \( j \) is:
  \[ D^j(G, H) = \left( \sum_{k=1}^{\infty} [N^j_G(k) - N^j_H(k)]^2 \right)^{\frac{1}{2}} \]
- \( j^{th} \) GDD agreement is: \( A^j(G, H) = 1 - D^j(G, H) \)
- The agreement between two networks \( G \) and \( H \) is:
  \[ A_{arirth}(G, H) = \frac{1}{73} \sum_{j=0}^{72} A^j(G, H), \text{ or } A_{geo}(G, H) = \left( \prod_{j=0}^{72} A^j(G, H) \right)^{\frac{1}{73}} \]

This is called Graphlet Degree Distribution (GDD) Agreement between networks \( G \) and \( H \).
Software that implements many of these network properties and compares networks with respect to them: GraphCrunch
http://bio-nets.doc.ic.ac.uk/graphcrunch/
Software that implements many of these network properties and compares networks with respect to them:

*GraphCrunch*

http://bio-nets.doc.ic.ac.uk/graphcrunch2/

**GraphCrunch 2:**

Software tool for network modeling, alignment and clustering

Oleksii Kuchaiev, Aleksandar Stevanovic, Wayne Hayes, Natasa Przulj

About GraphCrunch 2

GraphCrunch 2 is a new version of our GraphCrunch software tool for network analysis, modeling and alignment. It automates tasks of finding the best fitting model for the network data, pairwise comparisons of networks, alignment of two networks using GRAAL algorithm, and provides capabilities of clustering network nodes based on their topological surrounding in the network. Furthermore, GraphCrunch 2 is highly efficient in computing these tasks, utilizing a parallelization architecture to speed up computations by running tasks in parallel using all available cores and processors in the computer.

Designed with a user friendly, easy and intuitive drag and drop interface, GraphCrunch 2 allows users previously unfamiliar with GraphCrunch to quickly become proficient with typical use case scenarios.

GraphCrunch 2 allows users to stop and resume long computations, provides for reuse of previously computed results without recomputing them again and even recover from hardware malfunctions/execution interruptions without any loss of data or any unnecessary recomputations of previously finished tasks.

For more detailed description of GraphCrunch 2 and its capabilities, see the BMC Bioinformatics paper.
References

- Slides prepared by Dr. Mahendra Piraveenan for this class in previous years.
- “An Introduction to Systems Biology: Design Principles of Biological Circuits’ - Uri Alon
- “Network properties” by Dr. Nataša Pržulj, Department of Computing, Imperial College London
  (http://www.doc.ic.ac.uk/~natasha/course/docs/network_properties_and_models.ppt)
- “An Introduction To System Biology” - Uri Alon - Presented by Nitsan Chrizman