Reminder from the first lecture
Protein-Protein binding
IntAct Database (Dec 2015)
Interactions: 577,297  Proteins: 89,716

Baker’s yeast *S. cerevisiae* (only nuclear proteins shown)
From S. Maslov, K. Sneppen, Science 2002

Worm *C. elegans*
From S. Lee et al., Science 2004
Basic concepts of network analysis
Degree of a node – its # of neighbors

\[ K_1 = 4 \]

\[ K_2 = 2 \]
Directed networks have in- and out-degrees.

In-degree
$K_{in}=2$

Out-degree
$K_{out}=5$
How to find “important” nodes?

• By their degree
• Hubs = important
• Example: Google’s PageRank
How Google PageRank algorithm works?

• Google was solving the following problem in mid-1990s: too many websites match a typical search query: need to rank websites.

• Other popular search engines (e.g. Altavista) count the # of times a query word appears in website’s text. Websites respond by putting lots of invisible words

• One could rank the importance of webpages by number of hyperlinks pointing to it (in-degree $K_{in}$) but:
  – Too democratic: It doesn’t take into account the importance of webpages sending hyperlinks
  – It’s easy to trick and artificially boost the rank

• Google’s solution: simulate the behavior of many “random surfers” and then count the number of times they visited each webpage = it’s PageRank
  – Popular pages send more surfers your way $\rightarrow$ the PageRank weight is proportional to $K_{in}$ but weighted by popularity
PageRank algorithm is Google’s $2.8T$ idea

- PageRank assigns to every webpage an importance score $G_i$
- The meaning of $G_i$ – how often random surfers visit this website
- To determine solves a self-consistent Eq.:
  \[ G_i \sim \sum_j T_{ij} G_j \]
  Here
  \[ T_{ij} = A_{ij} / K_{out}(j) \]
  is the normalized adjacency matrix
- It finds the principal eigenvector (the one with the largest eigenvalue).
Problem with PageRank algorithm and how Google solved it

- Problem: surfers can be trapped in infinite loops with one or more entrances and no exits
- Model with random jumps mimicking surfers getting bored when following a chain of links
  \[ G_i \sim (1-\alpha) \sum_j T_{ij} G_j + \alpha \sum_j G_j \]
- \( \alpha=0.15 \) meaning that an average web surfer (circa 1995) on average jumped around \( 1/\alpha \approx 6 \) webpages before going somewhere else
How to find “important” nodes?

• By their connectivity
• Connectors = important
• Betweenness-centrality
Betweenness centrality: definition

• Take a node i
• There are \((N-1)\times(N-2)/2\) pairs of other nodes
• For each pair find the shortest path on the network
• If more than one shortest path, sample them equally
• Betweenness-centrality \(C(i)\) ~ the number of shortest paths going through node i
How is it connected to expression data analysis?
T-cell expression data

- The matrix contains 47 expression samples from Lukk et al, Nature Biotechnology 2011
- All samples are normal T-cells from different individuals
- Only the top 3000 genes with the largest variability were used
- The value is log2 of gene’s expression level in a given sample as measured by microarray technology

A global map of human gene expression

Margus Lukk, Misha Kapushesky, Janne Nikkilä, Helen Parkinson, Angela Goncalves, Wolfgang Huber, Esko Ukkonen & Alvis Brazma

Affiliations | Corresponding author


Although there is only one human genome sequence, different genes are expressed in many different cell types and tissues, as well as in different developmental stages or diseases. The structure of this ‘expression space’ is still largely unknown, as most transcriptomics experiments focus on sampling small regions. We have constructed a global gene expression map by integrating microarray data from 5,372 human samples representing 369 different cell and tissue types, disease states and cell lines. These have been compiled in an online resource (http://www.ebi.ac.uk/gxa/array/U133A) that allows the user to search for a gene of interest and
Correlated pairs
plausible biological connection based
on short description

\[ g_1 = 1994; \quad g_2 = 188; \quad \text{group 1} \]
\[ g_1 = 2872; \quad g_2 = 1269; \quad \text{group 2} \]
\[ g_1 = 1321; \quad g_2 = 10; \quad \text{group 3} \]
\[ g_1 = 886; \quad g_2 = 819; \quad \text{group 4} \]
\[ g_1 = 2138; \quad g_2 = 1364; \quad \text{group 5} \]

no obvious biological common function
\[ g_1 = 1 + \text{floor(rand.} \times 3000); \quad g_2 = 1 + \text{floor(rand.} \times 3000); \]
\[ \text{disp}([g_1, \ g_2]) \]
To analyze correlations in expression for all pairs of genes:

Co-expression networks
How to construct a co-expression network?

- Start with a matrix of log2 of expression levels of N genes in K samples (conditions): for our T-cell data N=3000, K=47

- For each of N(N-1)/2 pairs of genes i and j calculate the correlation coefficient \( \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \) of gene levels across K samples

- Put a threshold, e.g. \( \rho_{ij} > 0.85 \), or otherwise select the most correlated pairs of genes (~4500 in our case). Now you have a weighted network.

- Identify densely interconnected functional modules in this network.

- Modules can be used to infer unknown functions of genes via “Guilt by Association” principle.
How to install Gephi software for network analysis?

• Install Gephi from: https://gephi.org/users/download/

• One of the common problems with installation is the version of Java on your computer. One possible solution is here: https://github.com/gephi/gephi/issues/1787.

Sometimes after installation Gephi may complain that it cannot find java version 1.8 or higher. In this case you need to go to C:\Program Files\Gephi-0.9.2\etc

Open file gephi.conf using notepad.exe (MS Word does not work!).

Add a line jdkhome="C:\Program Files (x86)\Java\jre1.8.0_231"

(the numbers in ...jre1.8.0_231 may be changed to reflect the actual directory where Java is installed on your computer).

If JDK is not installed on your computer, you need to install it first from https://www.java.com/en/download/win10.jsp"
Co-expression network analysis exercise

• Start Gephi and open
  coexpression_network_random_start.gephi

• Run “Layout” ➔ Fruchterman Reingold ➔ Speed 10.0

• Run “Average degree”, “Network diameter”, “Modularity” in the Statistics tab in the right panel.

• Color nodes by “modularity class”:
  Appearance ➔ Nodes ➔ Partition ➔ Palette Icon ➔ Modularity class

• Size nodes first by “degree”.
  Appearance ➔ Nodes ➔ Ranking ➔ Multiple Circles Icon ➔ Degree
  – If the nodes are too small, select “Min size”: 10 and “Max size”: 80
  – Nodes in large tightly connected clusters have large degree

• Then size nodes by “betweenness-centrality”
  Appearance ➔ Nodes ➔ Ranking ➔ Multiple Circles Icon ➔ Betweenness-centrality
  – Large circles are “coordinator” genes connecting different co-expressed clusters to each other. Potentially biologically interesting
Disease-disease similarity network

- Based on the table summarizing all current medical knowledge of genes implicated in diseases:
  - Rows: 516 common human diseases
  - Columns: 25,000 human genes
  - Matrix element $D_{i\alpha} = 1$ if the gene $\alpha$ is known to be involved in the disease $i$. 0 – otherwise

- Constructed disease-disease similarity network:
  - Weight of the edge - # of shared genes between two diseases
  - Easy to construct: the adjacency matrix $A$ of the network is simply $A = D \cdot D^+$
Disease network analysis exercise

• Start Gephi and open disease_disease_random_start.gephi
• Run “Layout” → Fruchterman Reingold → Speed 10.0
Observe how clusters emerge.
• Run “Average degree”, “Network diameter”, “Modularity” analysis tools in the right panel.
• Color nodes with medical term: “disorder class”
Appearace → Nodes → Partition → Palette Icon → Disorder class
• Then color nodes by “modularity class”. See how well it agrees with the previous color.
Appearance → Nodes → Partition → Palette Icon → Modularity class
• Size nodes first by “degree”.
Appearance → Nodes → Ranking → Multiple Circles Icon → Degree
  – Which disease has the largest degree?
• Size nodes by “betweenness centrality”
Appearance → Nodes → Ranking → Multiple Circles Icon → Degree
  – Which diseases have the largest betweenness-centrality?
These “connector” diseases linking different diseases clusters to each other. They highlight potentially interesting connections between diseases