

# Multiomic integration and analysis through graphs

Rui Benfeitas

NBIS - National Bioinformatics Infrastructure Sweden  
Science for Life Laboratory, Stockholm  
Stockholm University

[rui.benfeitas@scilifelab.se](mailto:rui.benfeitas@scilifelab.se)



# Overview

---

## Overall objective

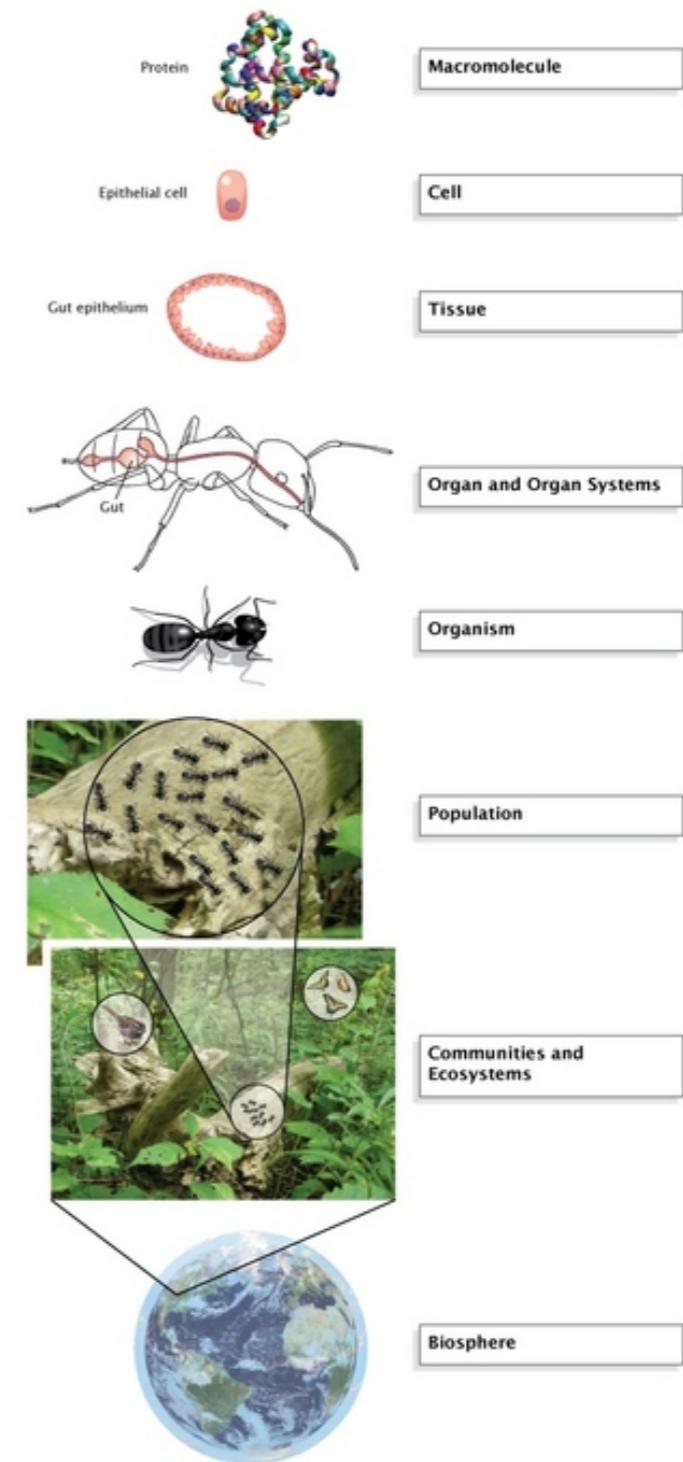
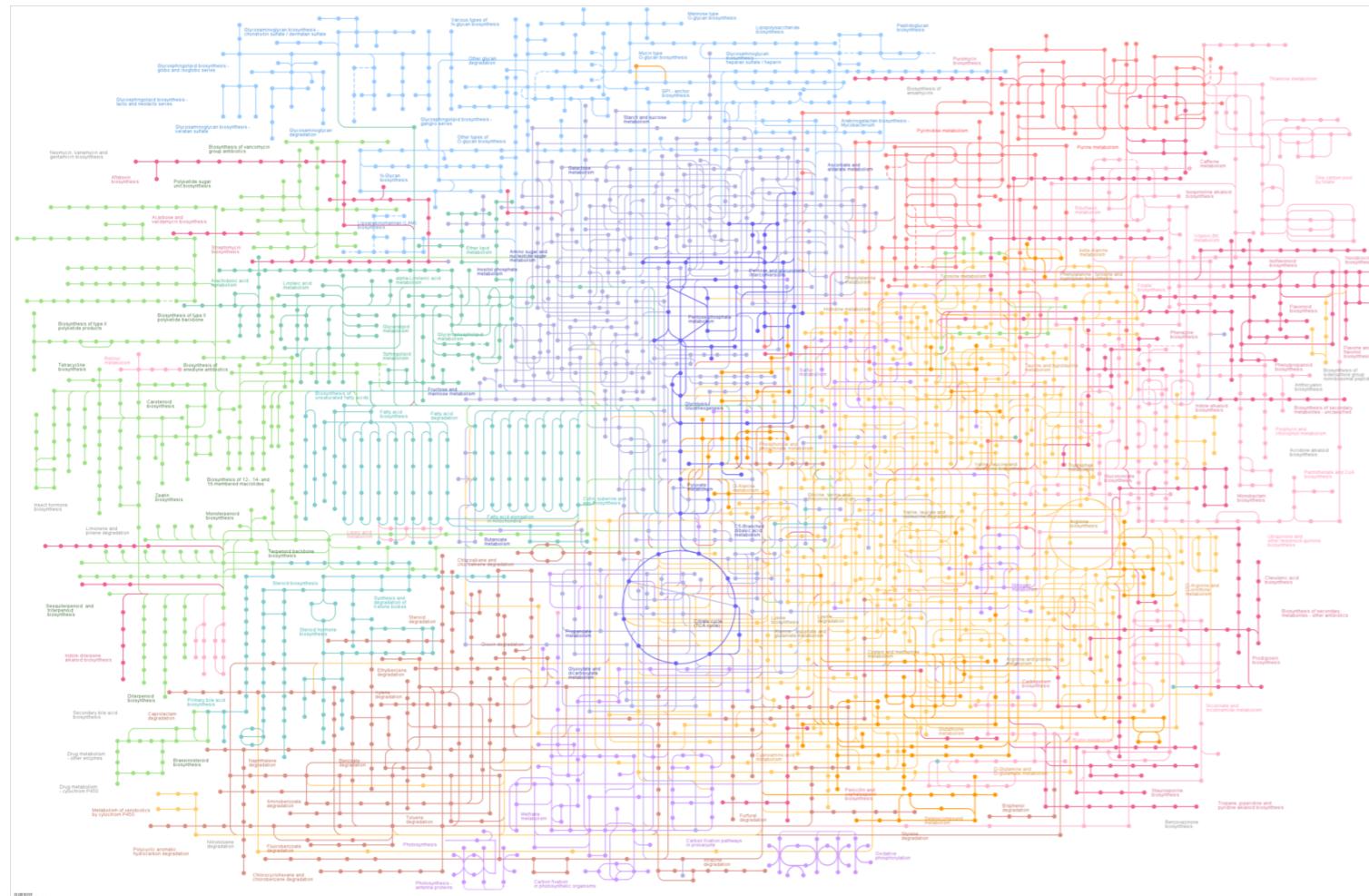
1. Introduction to network analysis
2. Terminology
3. Network construction
4. Key network properties
5. Community analysis
6. Workshop

This lecture is a condensed version of that taught at the [Omics Integration and Systems Biology](#) course

# Introduction

- 1. Introduction**
- 2. Terminology
- 3. Network construction
- 4. Key properties
- 5. Community analysis
- 6. Workshop

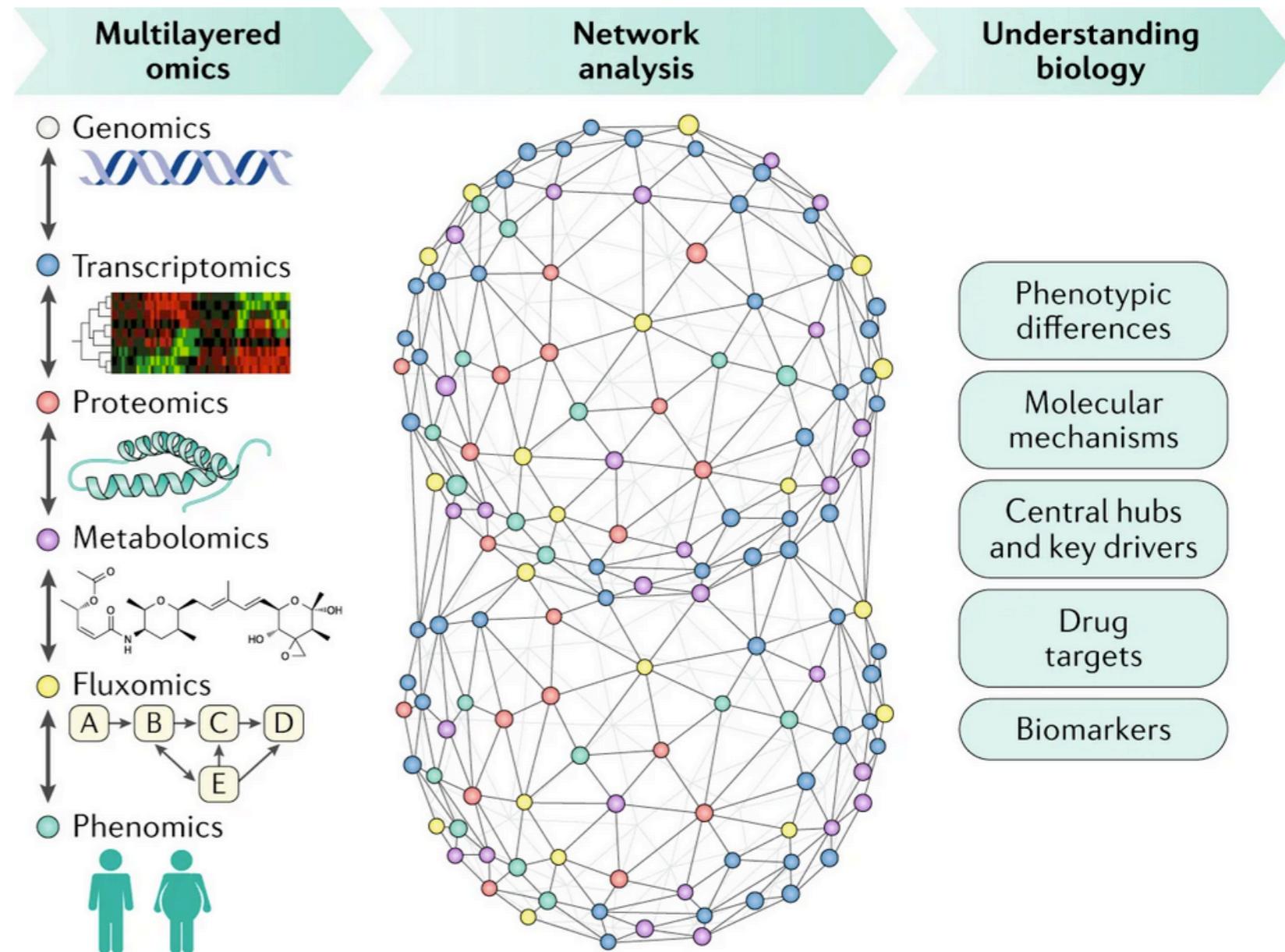
# Biological complexity under attack



# How to tackle biological complexity?

## Integrative approaches, and global patterns

- Feature association
- Modeling (e.g. GEMs)
- Network analysis



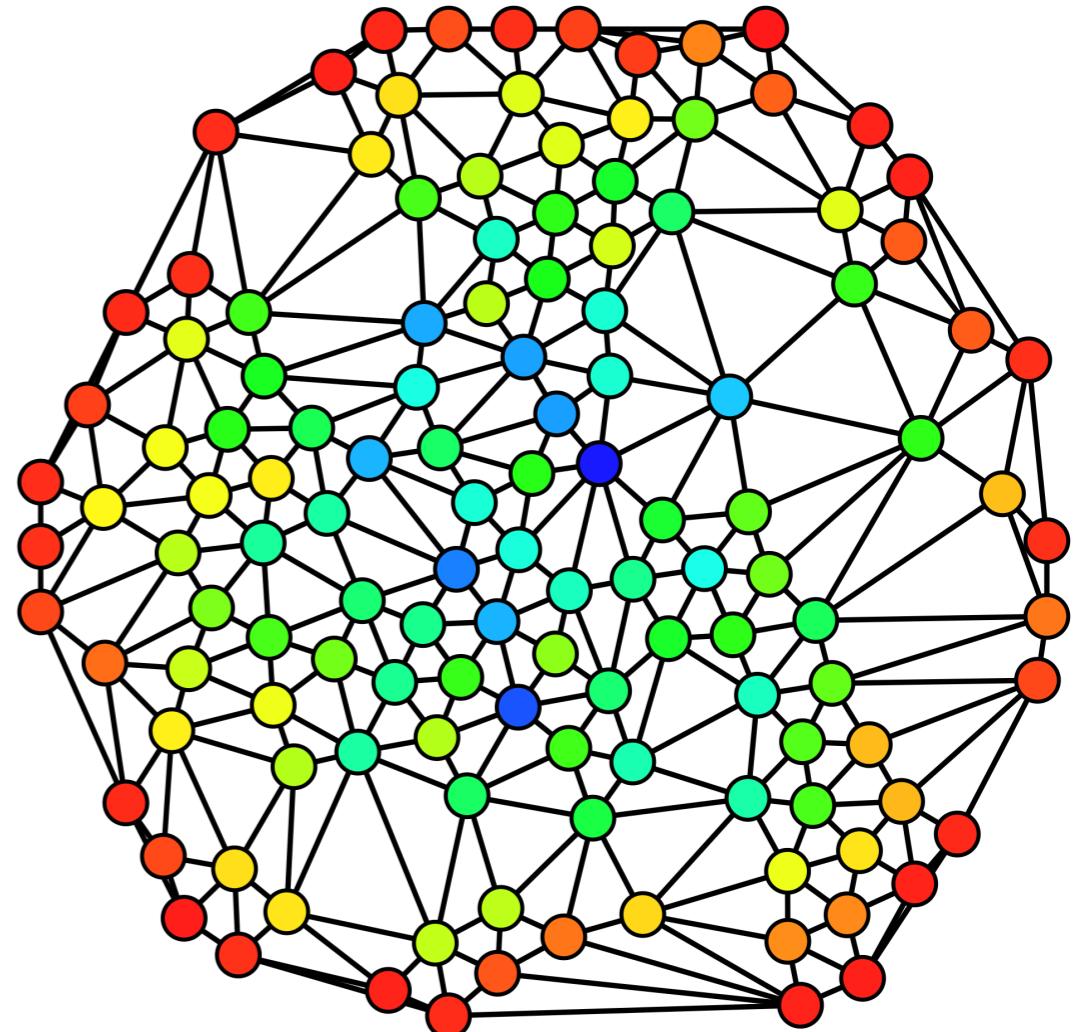
# What are networks?

---

Networks are representations of complex systems

Give us insight not easily achieved by other approaches:

- Comprehensive
- Coordinated (e.g. from diff. exp)
- Single- vs multiomic responses



# What are biological networks?

---



# What are biological networks?

---

Protein - Protein interaction (PPI) networks

Transcription-factor regulatory networks

Gene - gene co-expression networks

Signal transduction networks

Transcription-Factor Regulatory networks

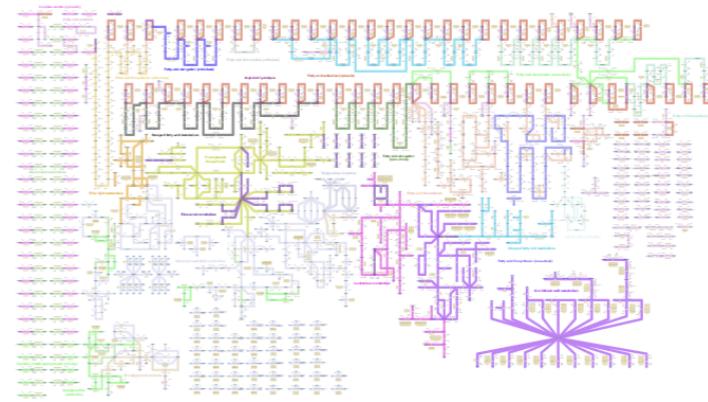
Drug-disease association networks

Aim

Functional / mechanistic characterisations

# What are biological networks?

Metabolite - Enzyme - Signal - Genes (GEMs)

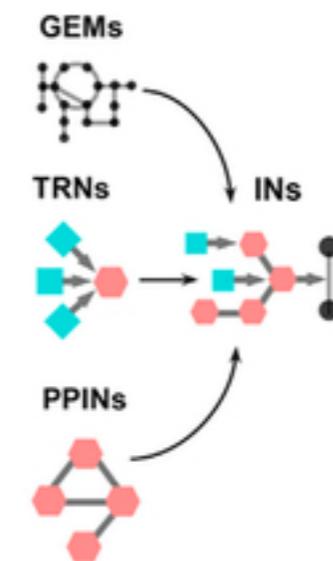
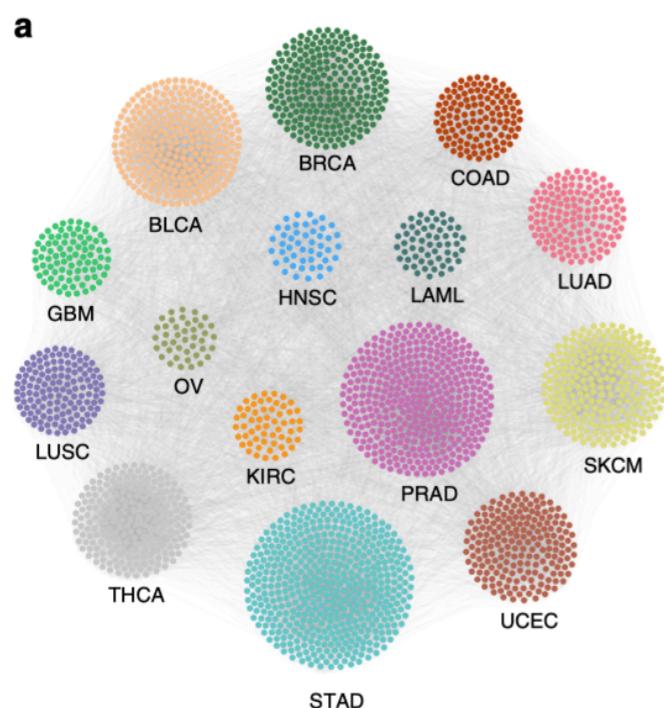
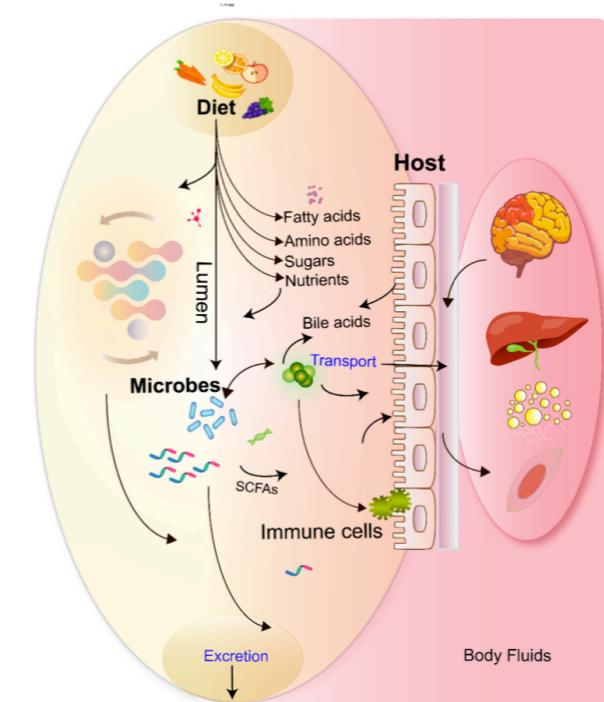
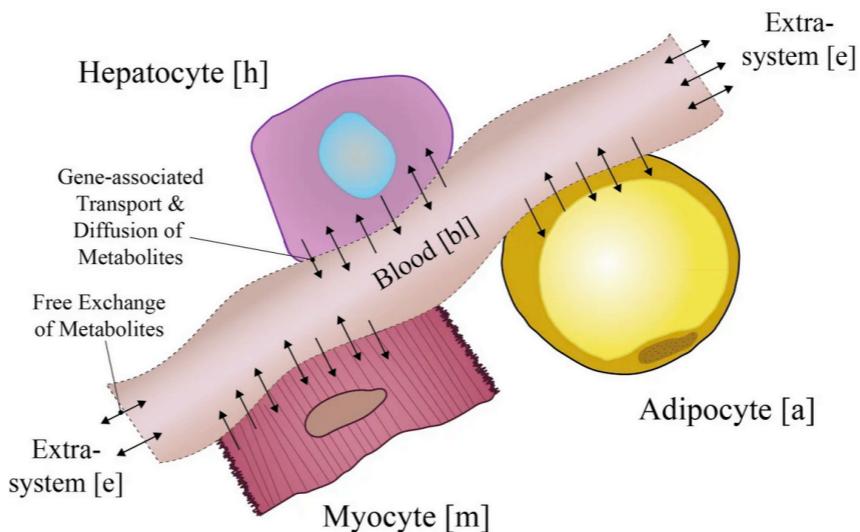


Multi-tissue networks

Multi-species networks

Cancer/disease networks

Integrated networks



# Terminology and initial properties in graph analysis

1. Introduction
- 2. Terminology**
3. Network construction
4. Key properties
5. Community analysis
6. Workshop

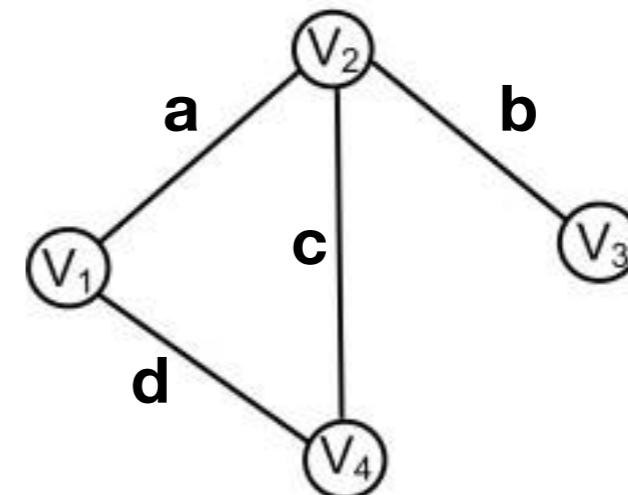
# Graphs, nodes, edges

---

**Graph G** consists of a set of **nodes** ( $V$ ) interconnected by **edges** ( $E$ )

**Nodes** sometimes called **vertices**

Two connected nodes: **neighbours**



# Simple vs multigraphs

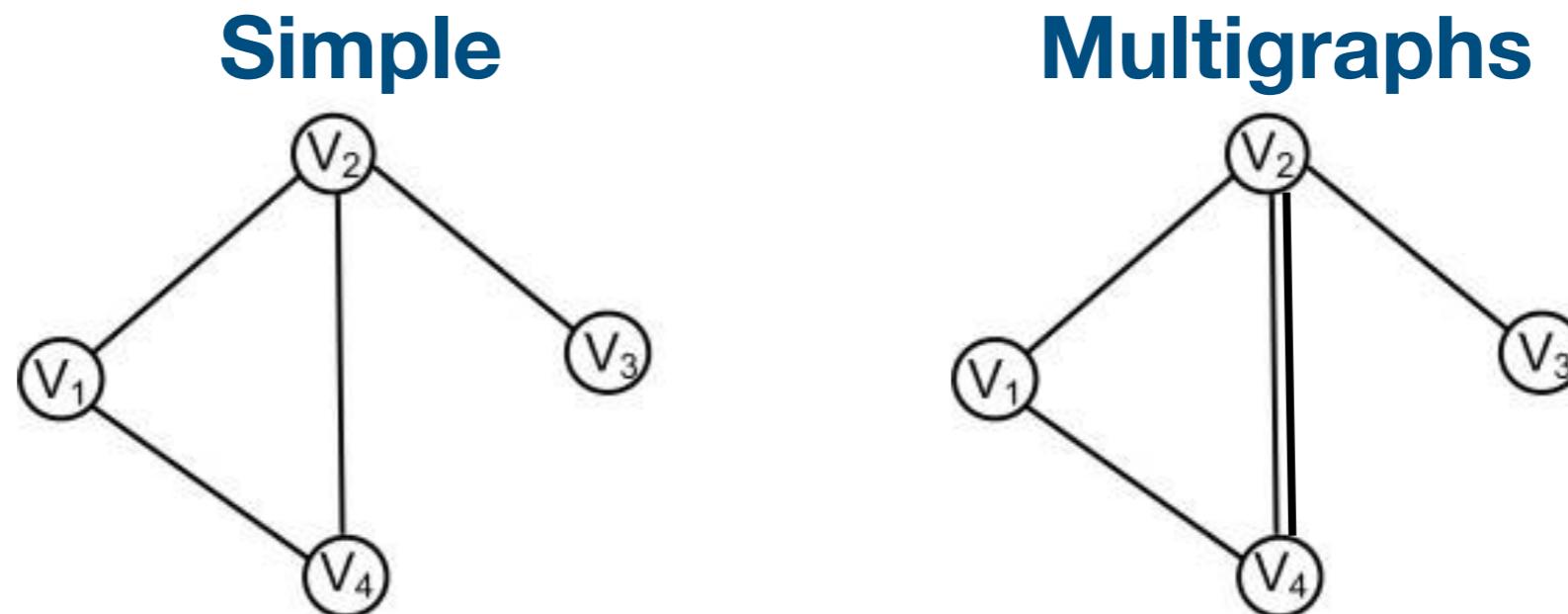
---

**Multigraphs** contain parallel edges

**Multi-edged** connections indicate different properties

Example: PPI

- Experimental evidence for interaction
- Co-expression



# Hypergraphs

---

**Hypergraphs** contain edges that connect any number of nodes

**Reaction 1:**  $A \rightarrow B + C$

**Reaction 2:**  $B + C \rightarrow D$

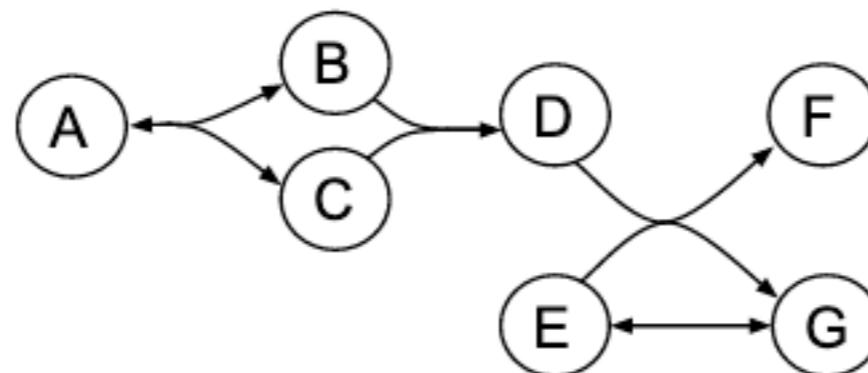
**Reaction 3:**  $D + E \rightarrow F + G$

**Reaction 4:**  $E \rightarrow G$

**Reaction 5:**  $B + C \rightarrow A$

**Reaction 6:**  $G \rightarrow E$

(a) Reaction network

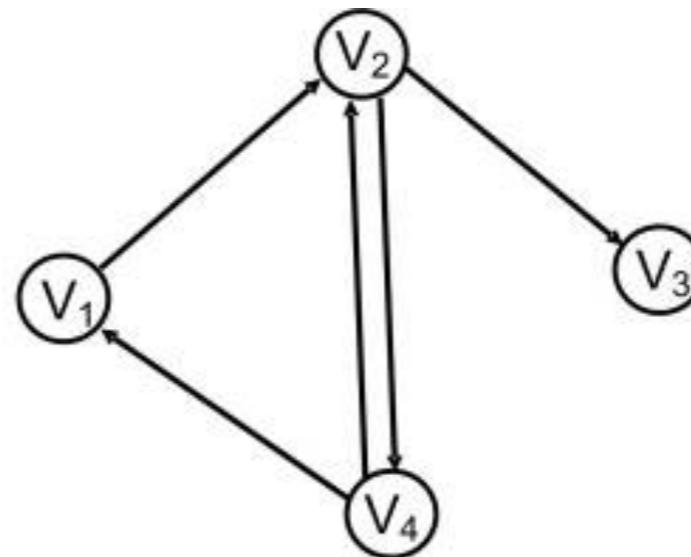
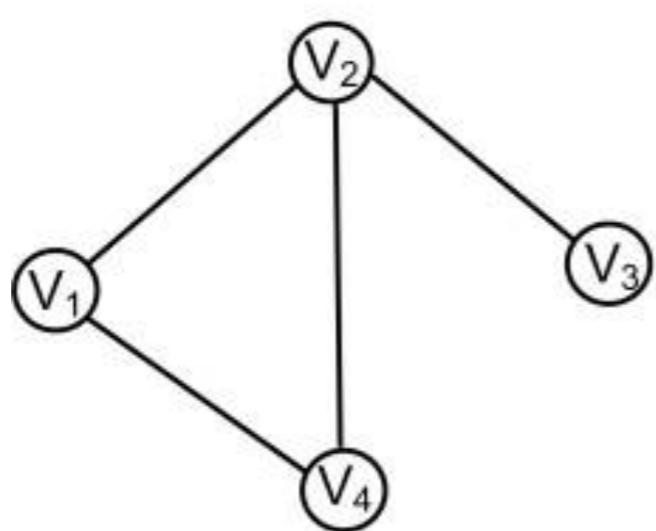


# Directed vs undirected graphs

---

Examples:

- **Undirected graphs:** co-expression networks
- **Directed graphs:** metabolic networks



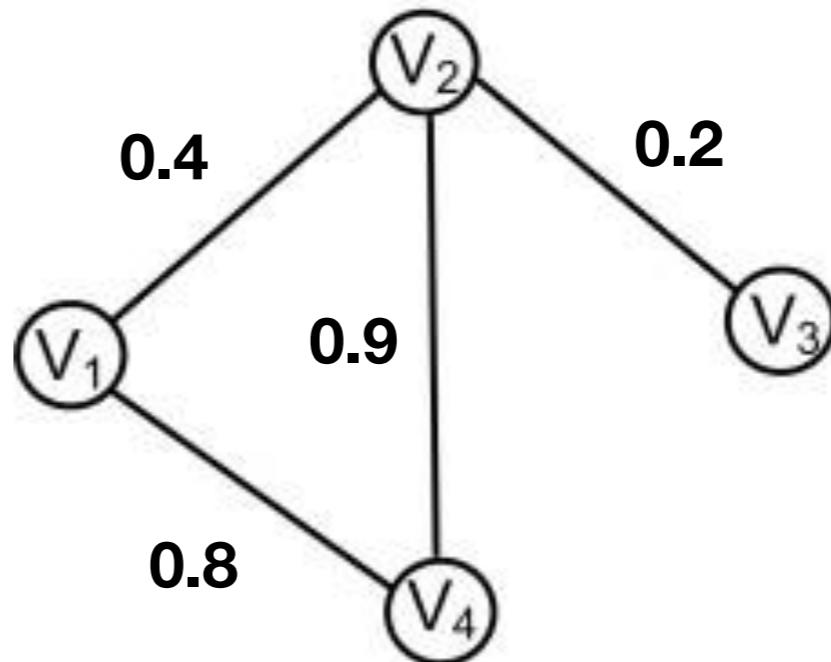
# Weighted vs unweighted graphs

---

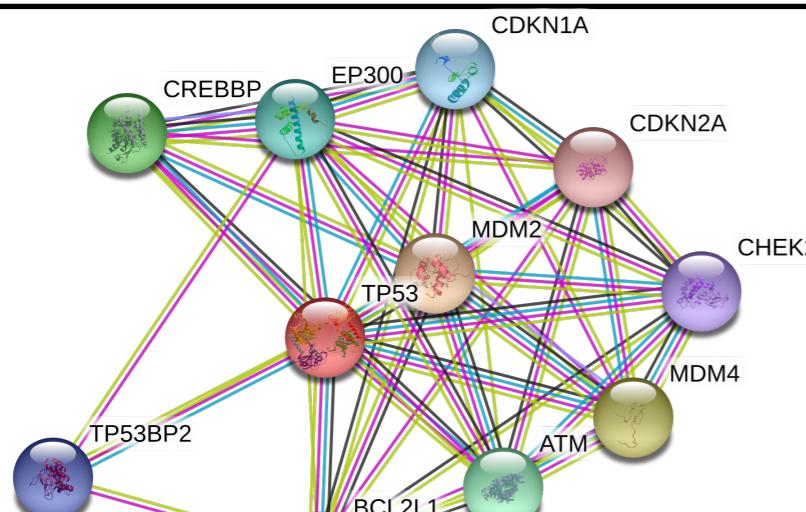
**Weighted edges** associate a value to an interaction between two nodes. E.g. weights ~ confidence in the interaction.

Negative weights?

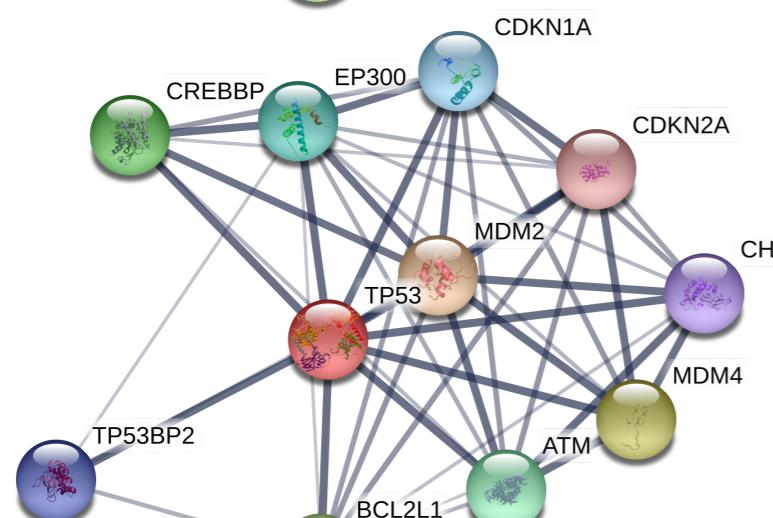
E.g. weighted co-expression networks



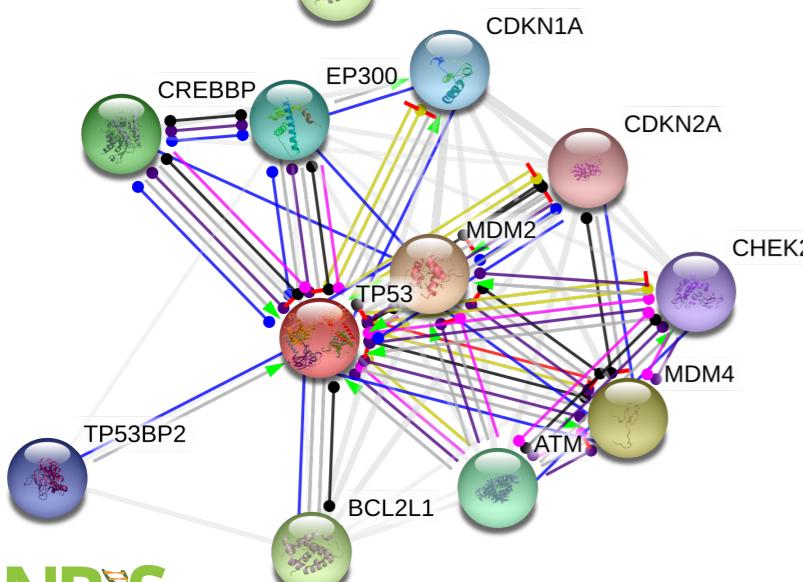
# STRING-db.org: TP53



Multi-edged



Weighted multi-edged



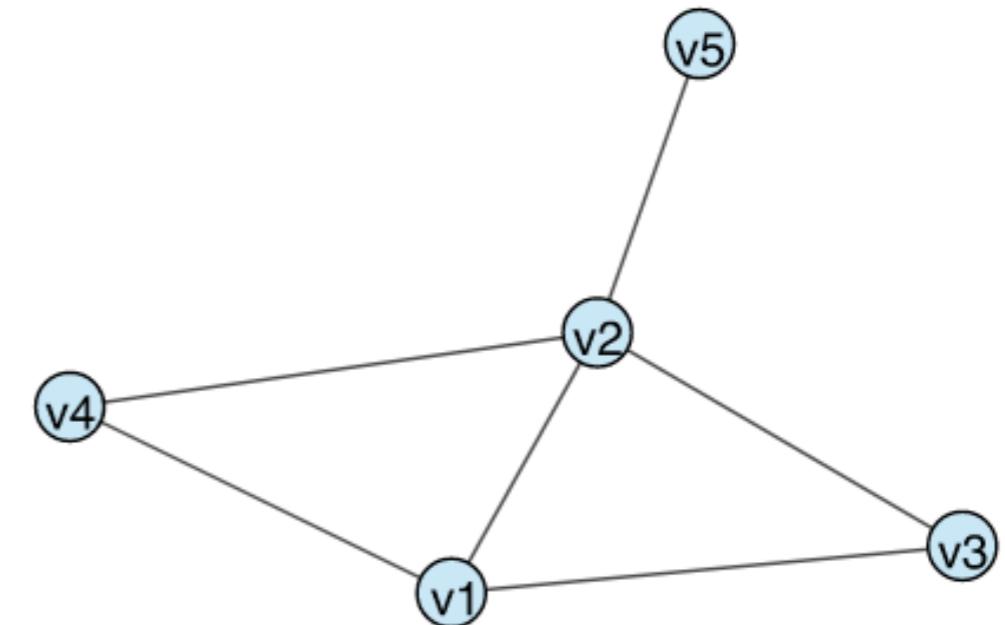
Multi-edged directed



# Adjacency matrix (undirected graphs)

Vertex association  
(undirected network)

| n1 | n2 |
|----|----|
| v1 | v2 |
| v1 | v4 |
| v2 | v4 |
| v2 | v3 |
| v2 | v5 |
| v1 | v3 |



Adjacency matrix is symmetric

|    | v1 | v2 | v3 | v4 | v5 |
|----|----|----|----|----|----|
| v1 | 0  | 1  | 1  | 1  | 0  |
| v2 | 1  | 0  | 1  | 1  | 1  |
| v3 | 1  | 1  | 0  | 0  | 0  |
| v4 | 1  | 1  | 0  | 0  | 0  |
| v5 | 0  | 1  | 0  | 0  | 0  |

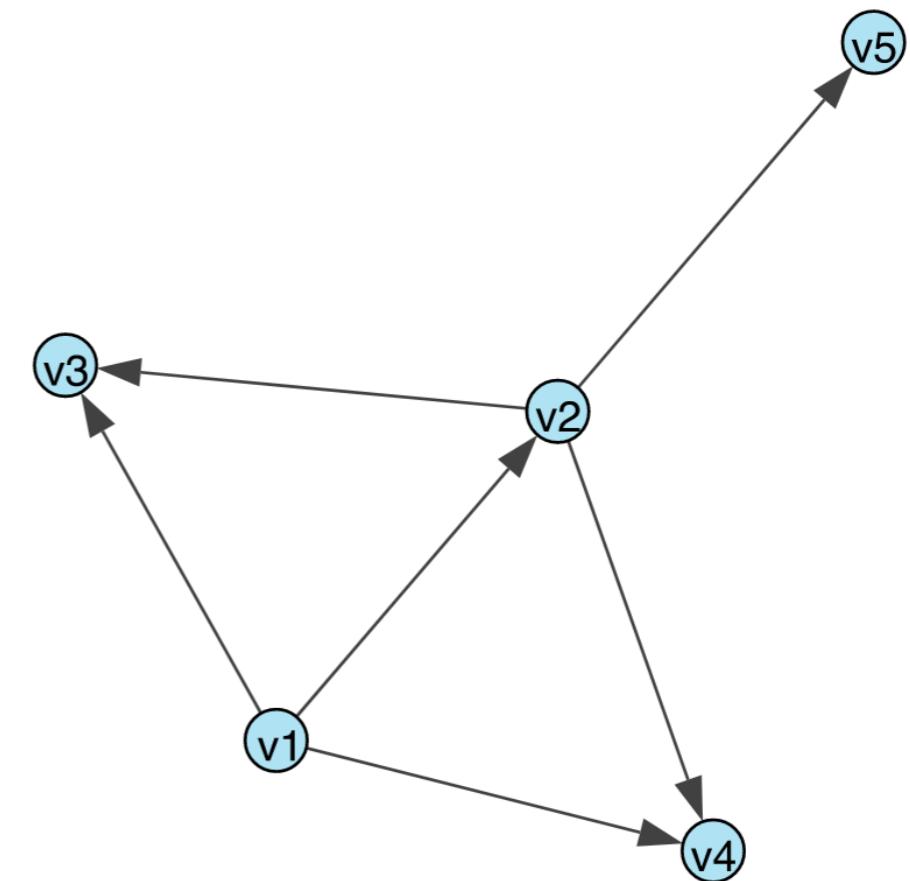
Upper triangular

Lower triangular

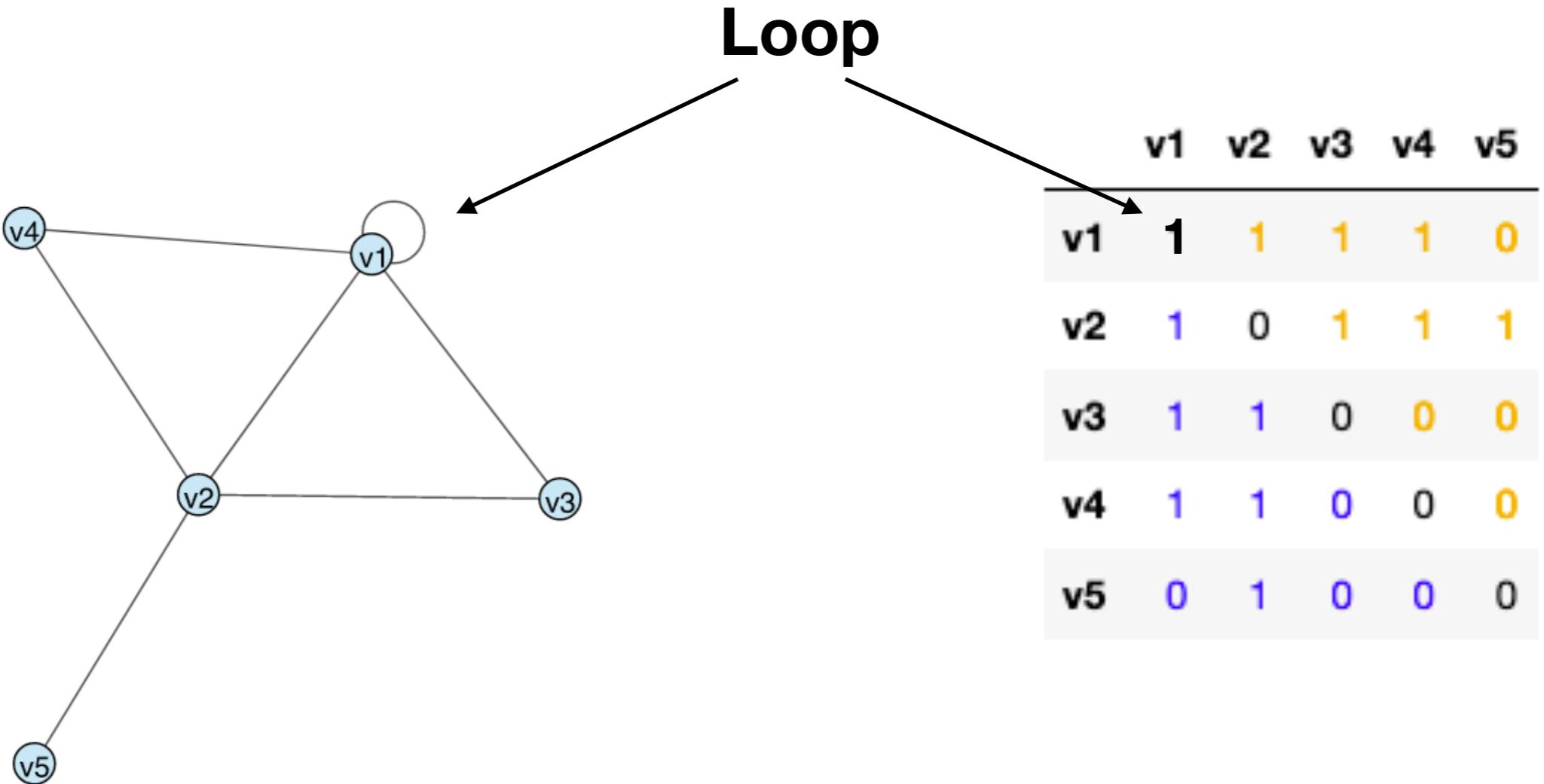
Diagonal

# Adjacency matrix (directed graphs)

|        |   | Target |    |    |    |    |
|--------|---|--------|----|----|----|----|
|        |   | v1     | v2 | v3 | v4 | v5 |
| Source |   | v1     | v2 | v3 | v4 | v5 |
| v1     | 0 | 1      | 1  | 1  | 0  |    |
| v2     | 0 | 0      | 1  | 1  | 1  |    |
| v3     | 0 | 0      | 0  | 0  | 0  |    |
| v4     | 0 | 0      | 0  | 0  | 0  |    |
| v5     | 0 | 0      | 0  | 0  | 0  |    |



# Graphs may contain self-loops



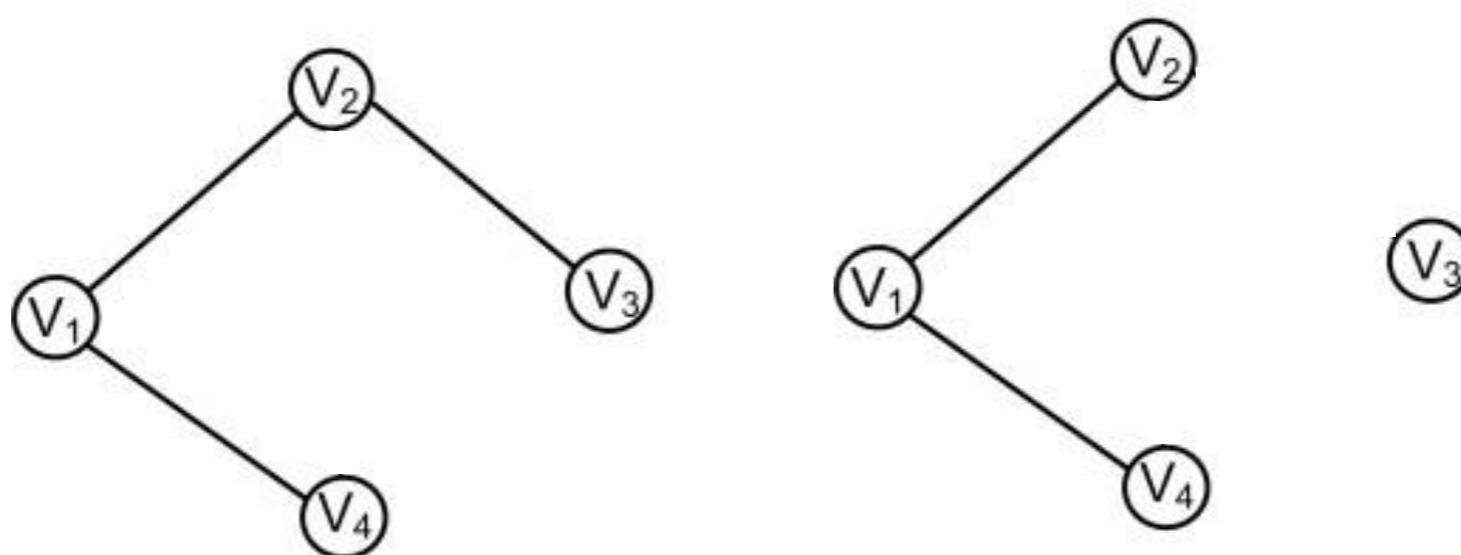
Examples of **self-loops** are auto-regulatory mechanisms  
(feed-back loops)

# Connected vs disconnected networks

---

**Connected network:** there is at least 1 path connecting all nodes in a network

**Disconnected network:** some of the nodes are unreachable

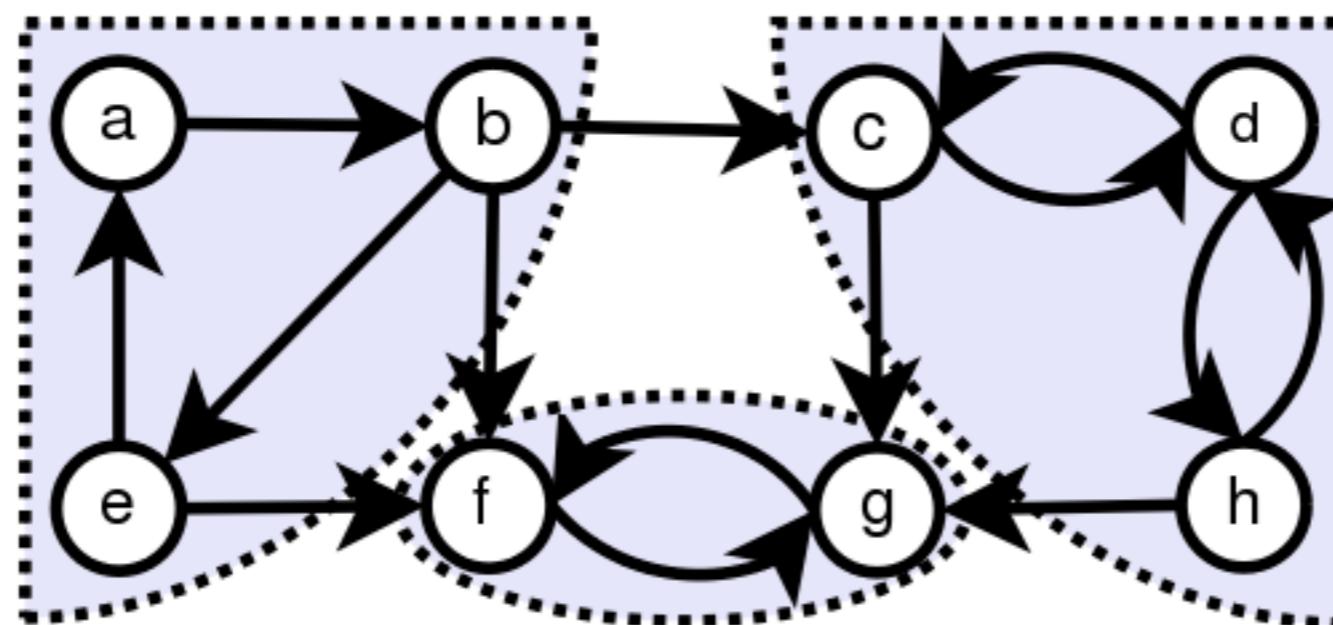


# Connected components

---

**Connected components** are those where all nodes of each subgraph are connected.

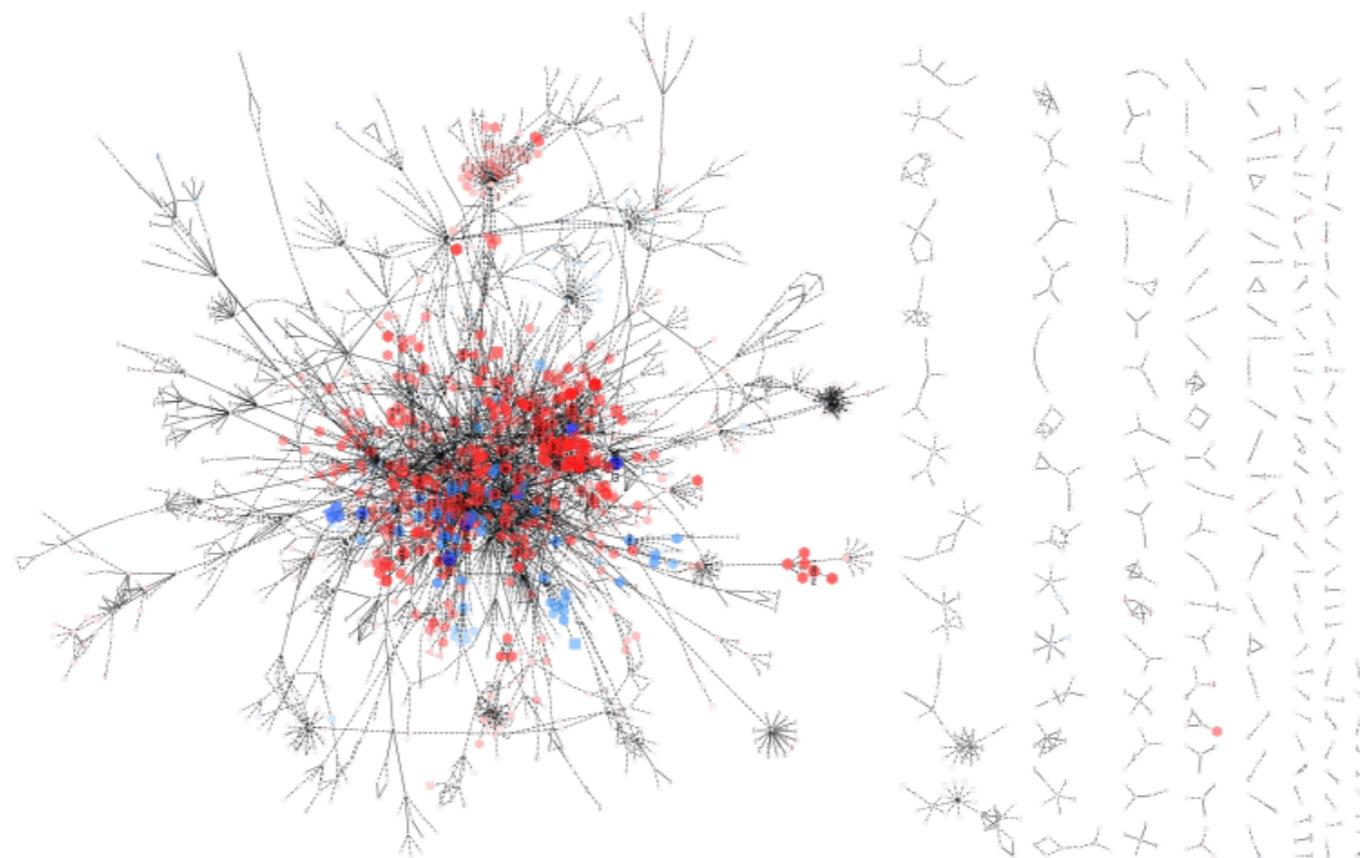
## Weak vs strong components



# Connected components

---

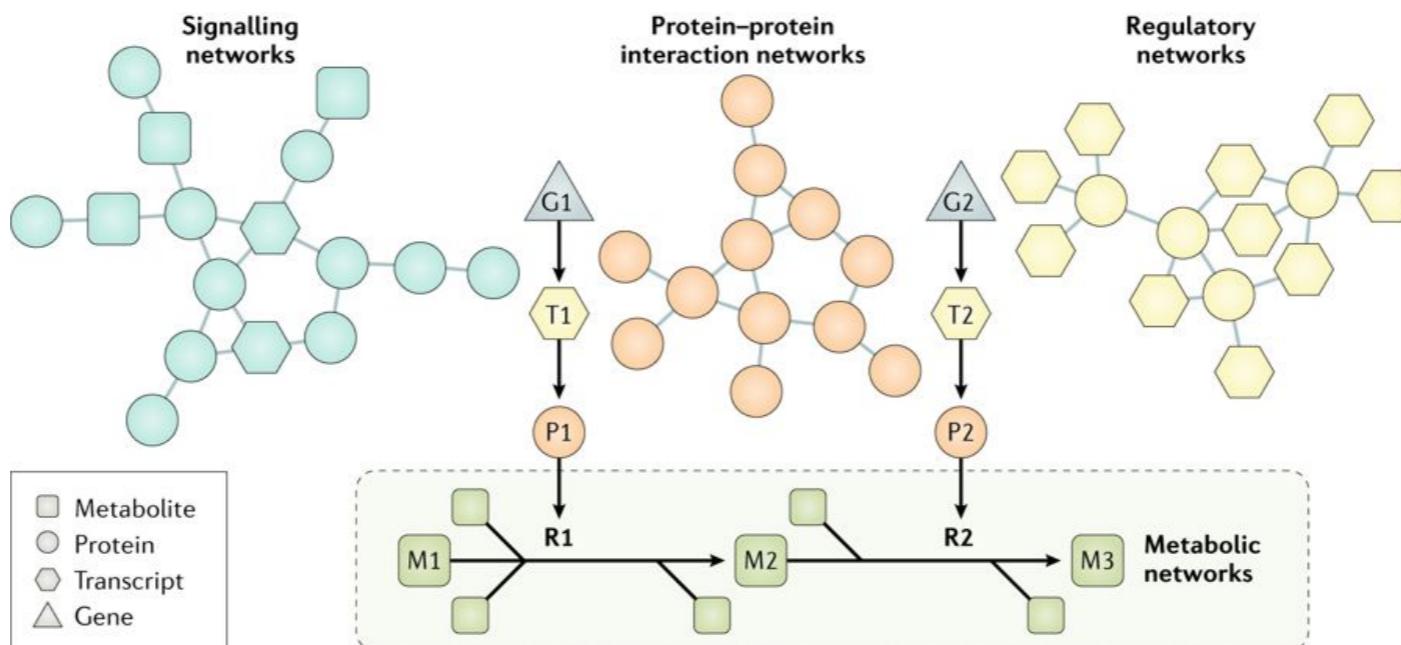
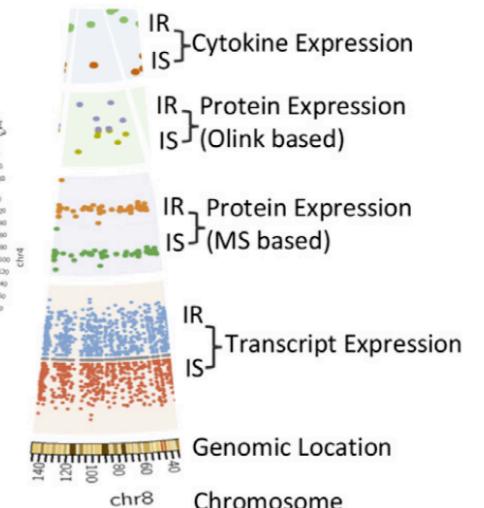
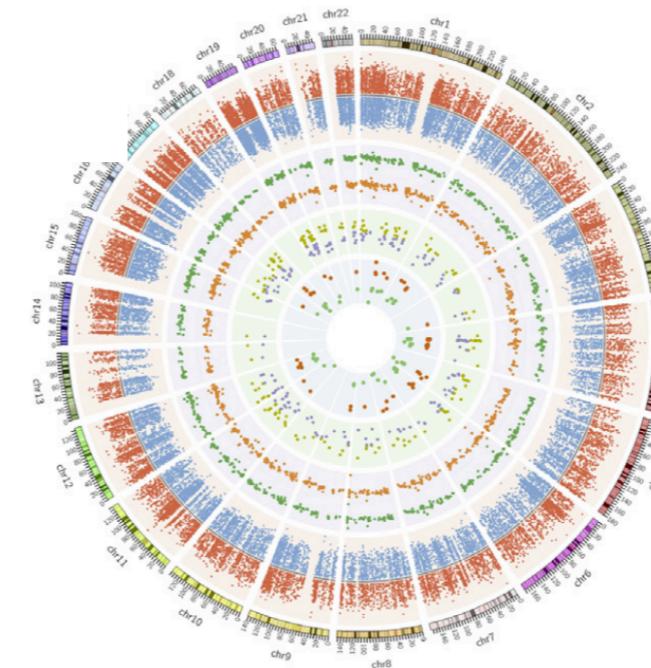
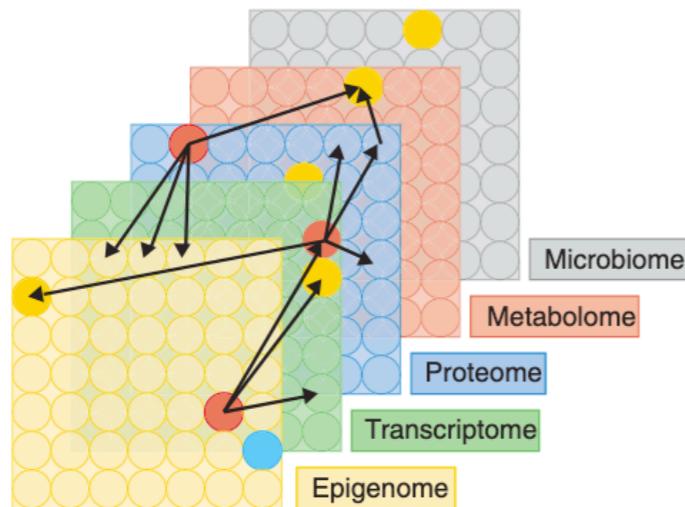
In biological networks, often the most insightful properties come from the **largest connected component**



# Biological network inference

# Building networks

How to go from raw data to a graph-tractable format?



Hasin 2017  
Piening 2018  
Mardinoglu 2018

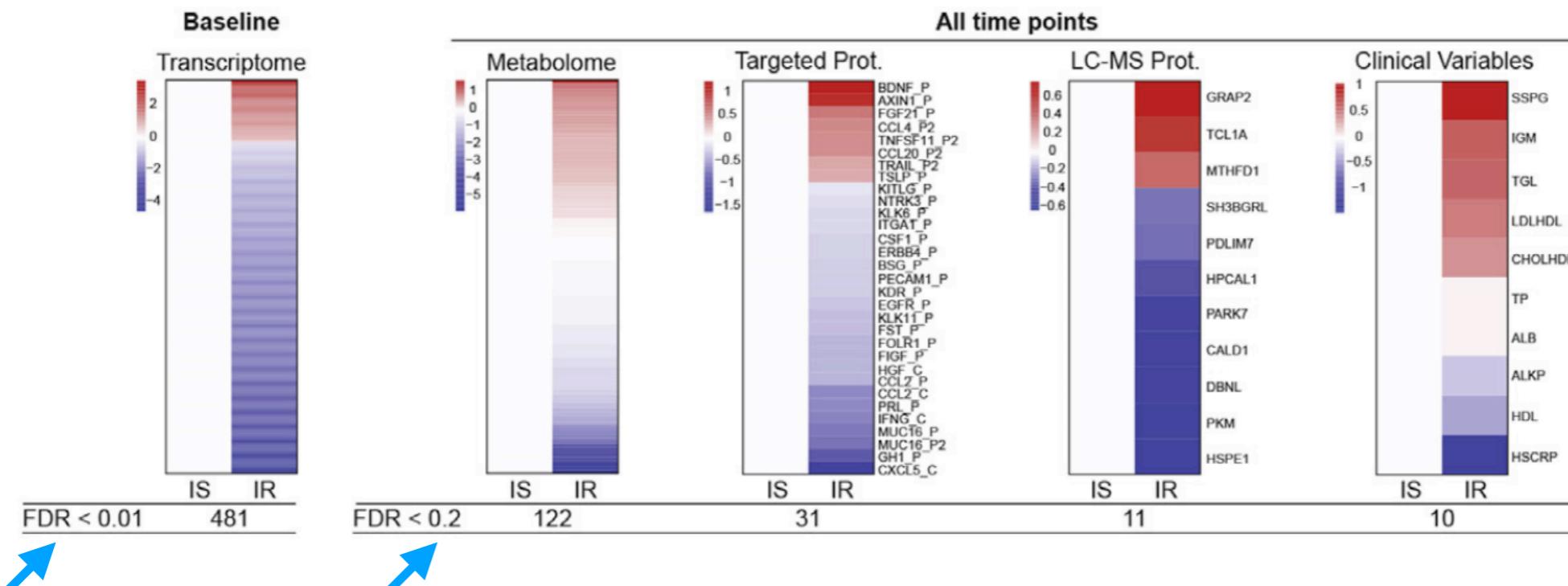
# Important considerations

Importance of applying standard statistics and false discovery corrections

Importance of power and sample size

Statistical power and significance in high throughput studies

Confounding factors may be a problem: maximise technical variation, minimise biological signal



Clarke 2011

Krzywinski 2013

Sham 2014

Nygaard 2016

Piening 2018

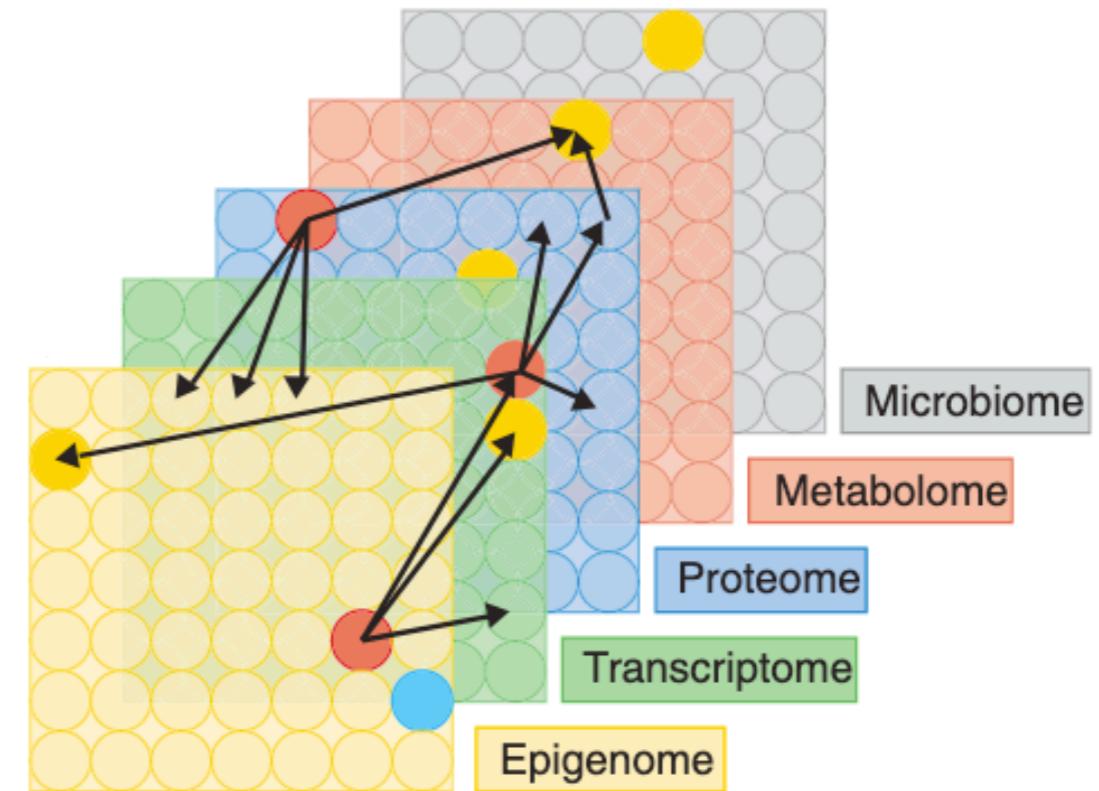
other refs as links

# Interomic vs Intraomic networks

Networks may be build for individual omics or for their integration

Should I even integrate different omics? What is my biological question?

- Do I want to analyse vertical relationships between features?
- Why integrate omics with different coverage such as transcriptomic and proteomic data
- What to do when samples have unequal data points?
- Do I want to extract functional properties?
- Am I predicting biomarkers?

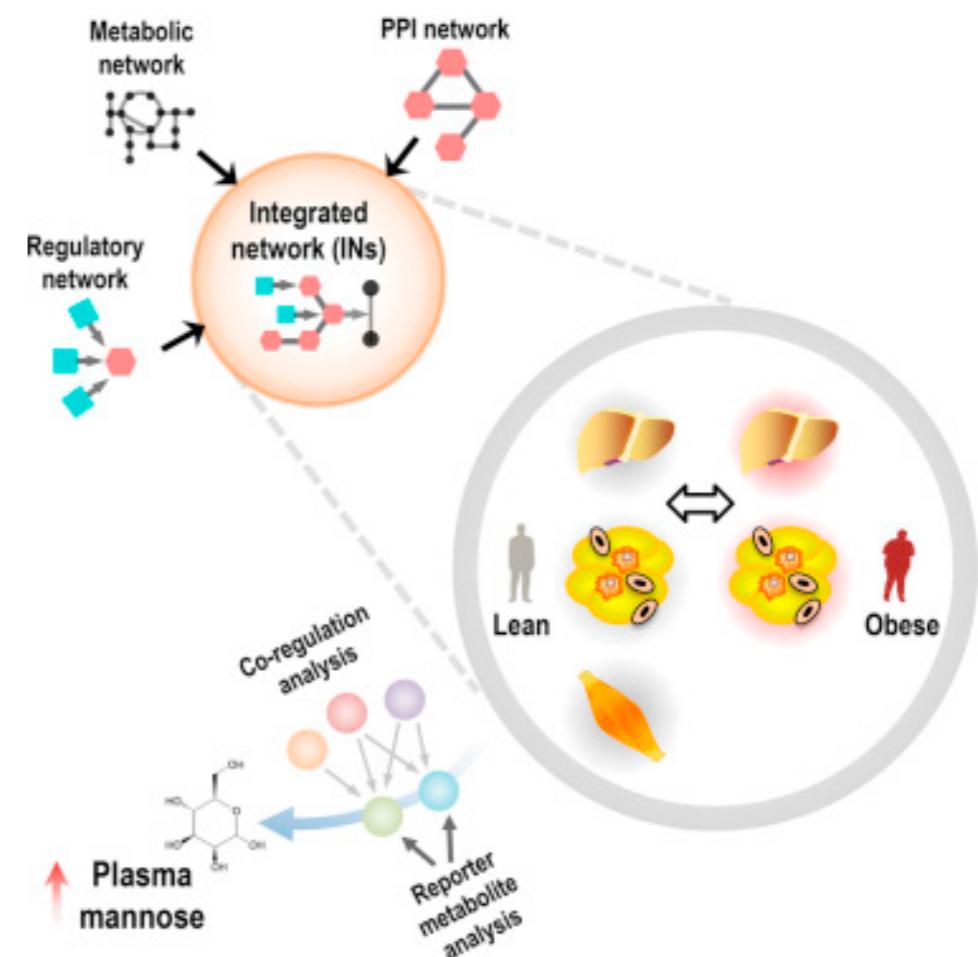


# Interomic vs Intraomic networks

Multi-modal networks may be generated from different sources

- Transcription-factor - Gene (DNAseq, ChIPseq)
- Gene-gene (Co-expression, PPI, GEMs)
- Gene-metabolite (GEM)
- Metabolite-metabolite (GEM)

## Integrated Networks



# Different approaches for network inference

---

- 1. Feature association
  - 2. K-nearest neighbour graph (k-NNG) construction
  - 3. Latent-factor extraction
  - 4. Pathway-based
  - 5. Genome-scale metabolic models
  - 6. Network deconvolution
- |  |                                       |
|--|---------------------------------------|
|  | <b>Data-driven</b>                    |
|  | <b>No prior info</b>                  |
|  | <b>Based on available information</b> |
|  | <b>Filter indirect effects</b>        |

# 1. Association analysis

---

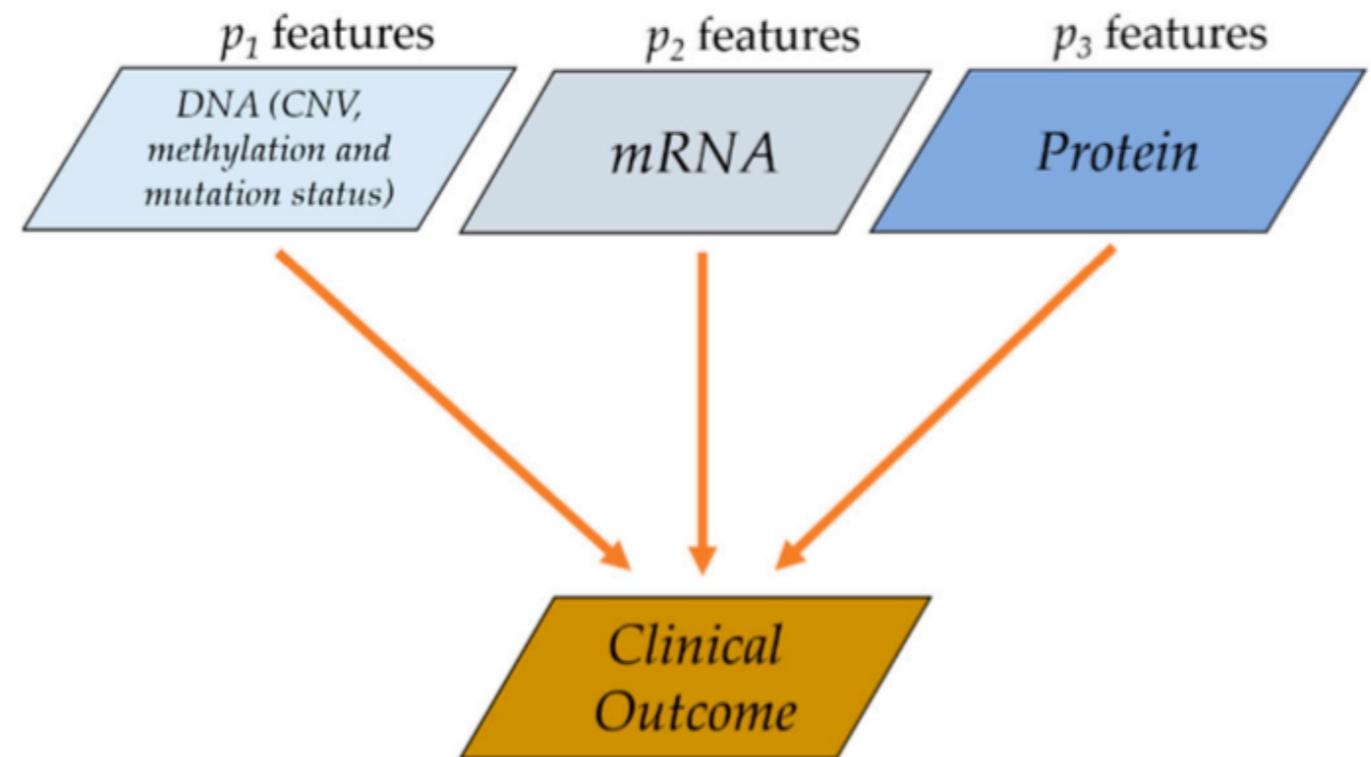
Balanced dataset, standard pre-processing of each omics needed

Standardization needed to make omic datasets comparable

Common approach: compute correlations between different features

- Spearman
- Pearson

Extend known associations



# 1. Association analysis

---

Easy to interpret

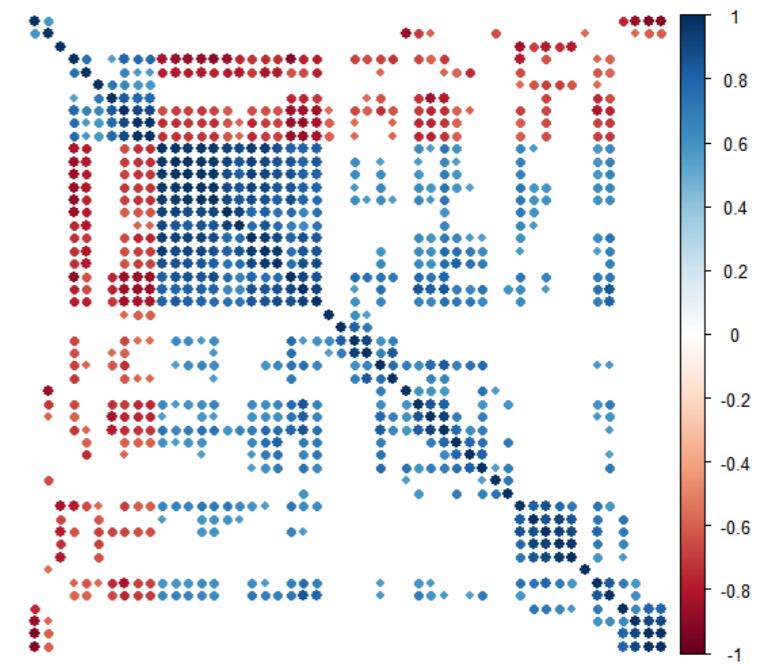
Unweighted vs weighted ( $-1 \leq \rho \leq 1$ )

Unbalanced networks

**Prone to type I errors**

- FDR vs Bonferroni
- Correlation coefficient cutoff

Need adjustment to possible confounding factors



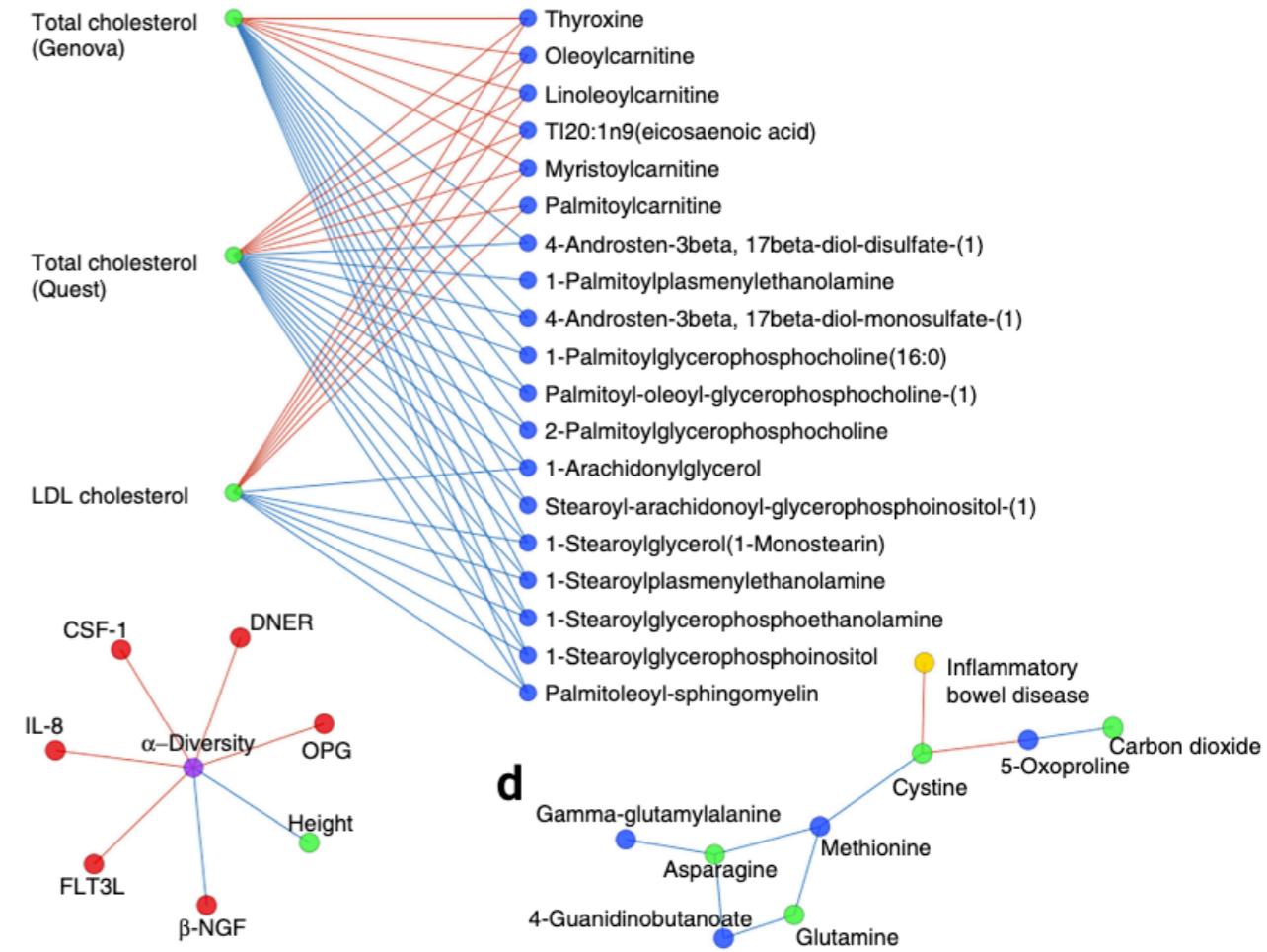
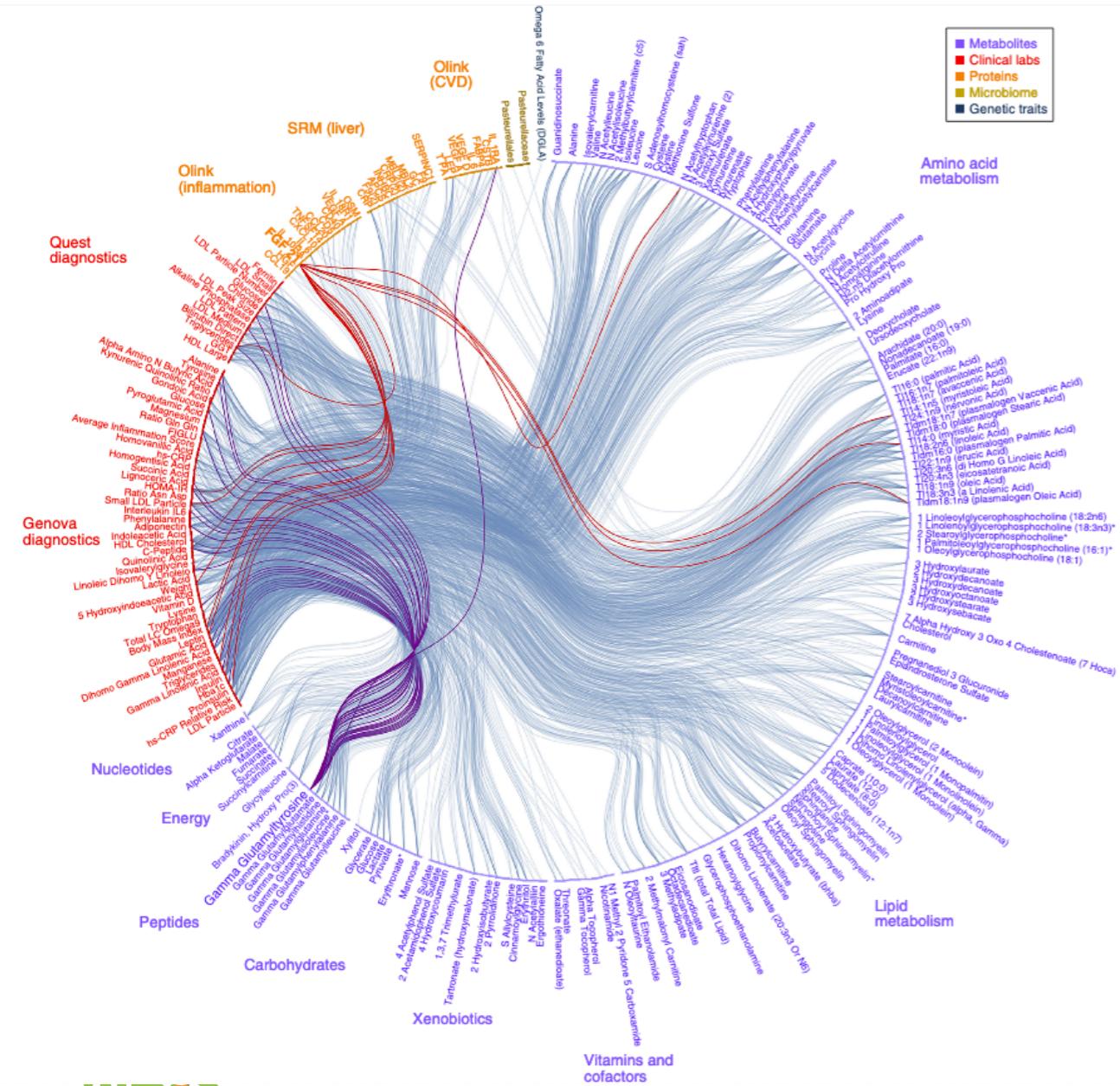
# 1. Association analysis

Adjusting for confounding factors: partial correlation analysis

Still considers linear relationship between variables

Below:

- gender and age are known confounding factors
- feature regression on confounding factors, followed by correlation on the residuals of each model



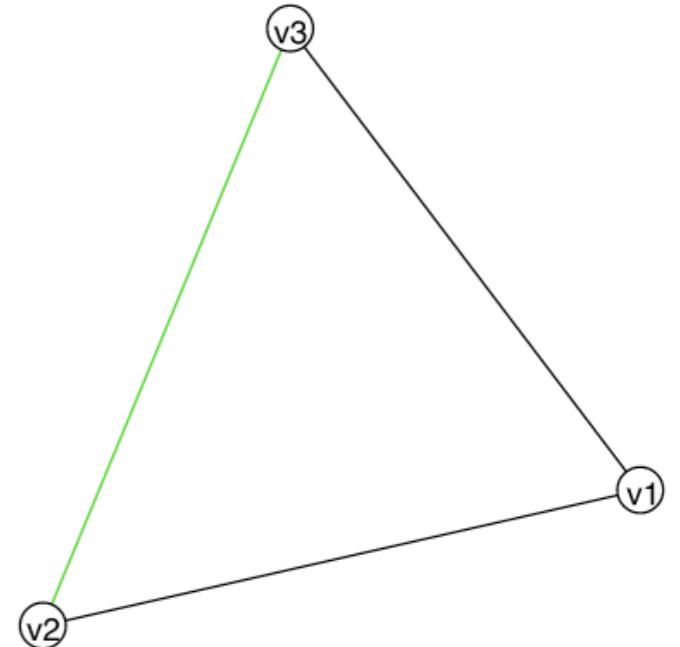
## 2. $k$ -nearest neighbour graph

---

1. For each pair of features  $(u, v)$ , compute a distance metric:

- Correlation
- Euclidean
- Jaccard
- ...

2. For each feature, select the *closest  $k$*  neighbours



Efficiency (not scalable, compute all neighbours for every node)

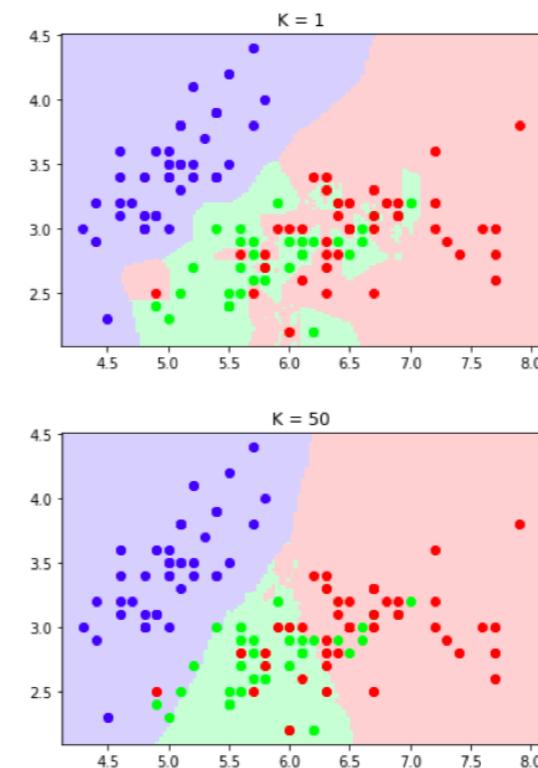
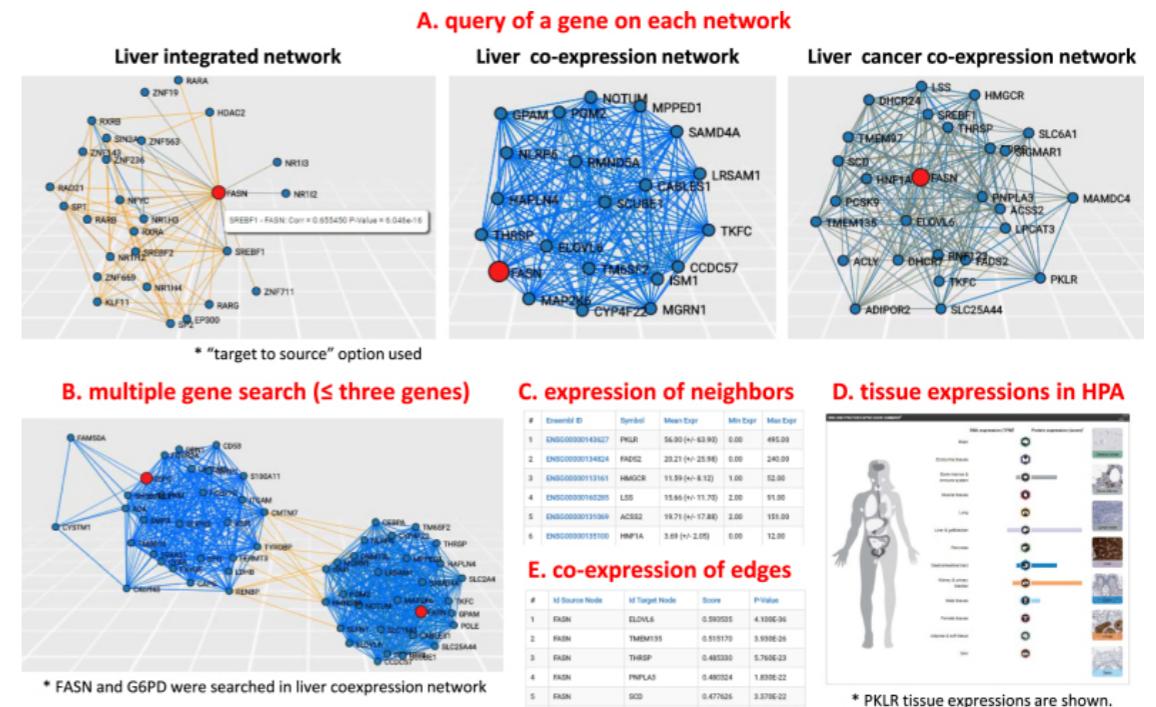
Generates well-structured graph

Simple as it reduces the number of features

# 2. $k$ -nearest neighbour graph

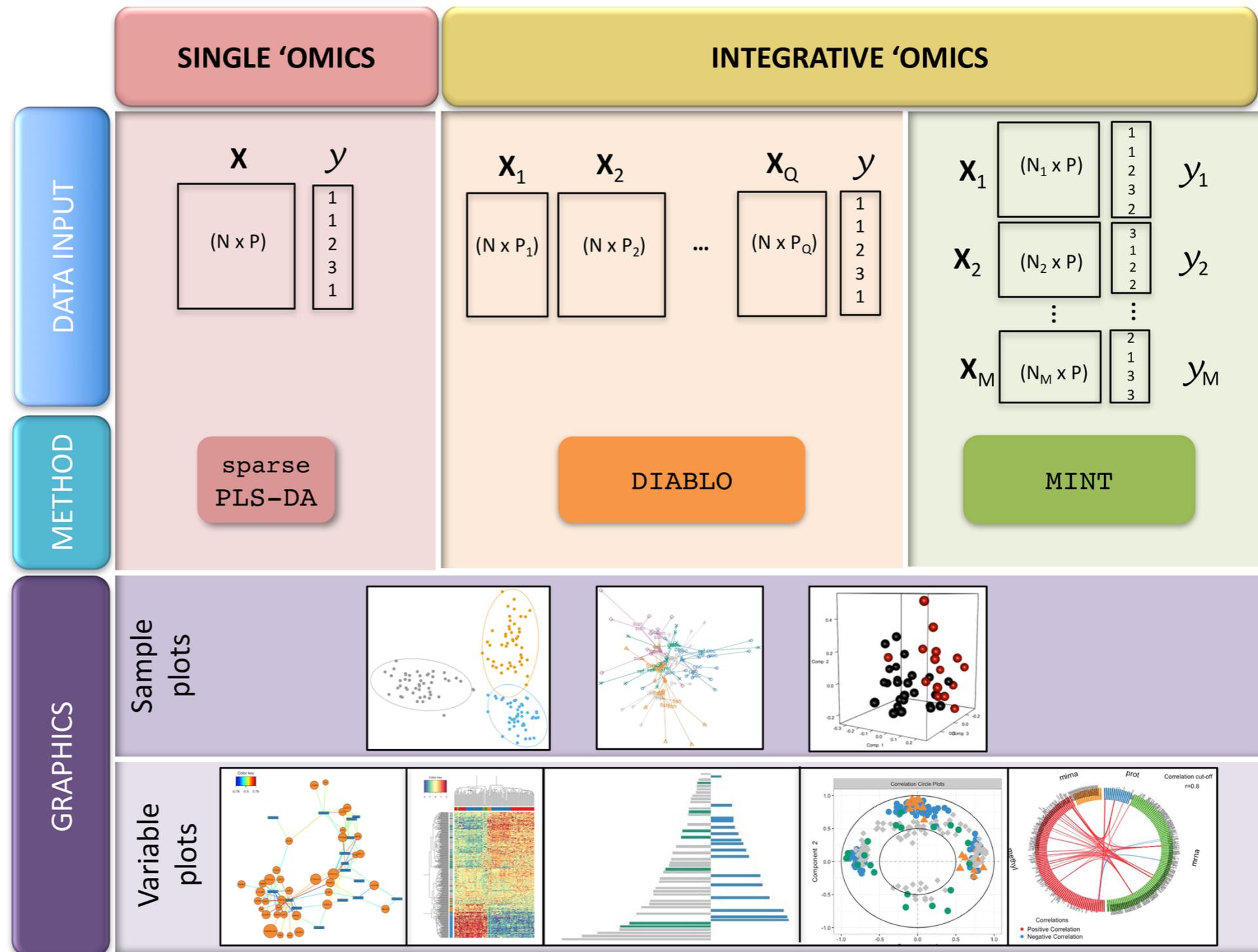
High  $k$  is smooth, but biased (underfitting)

Low  $k$  is accurate, but noisy (overfitting)



# 3. Latent-factor extraction

## Mixomics



# 4. Knowledge-based graph creation

---

## Database-derived

- PPI
- TFRN
- Metabolic Atlas
- ...

## Many reference databases

KEGG

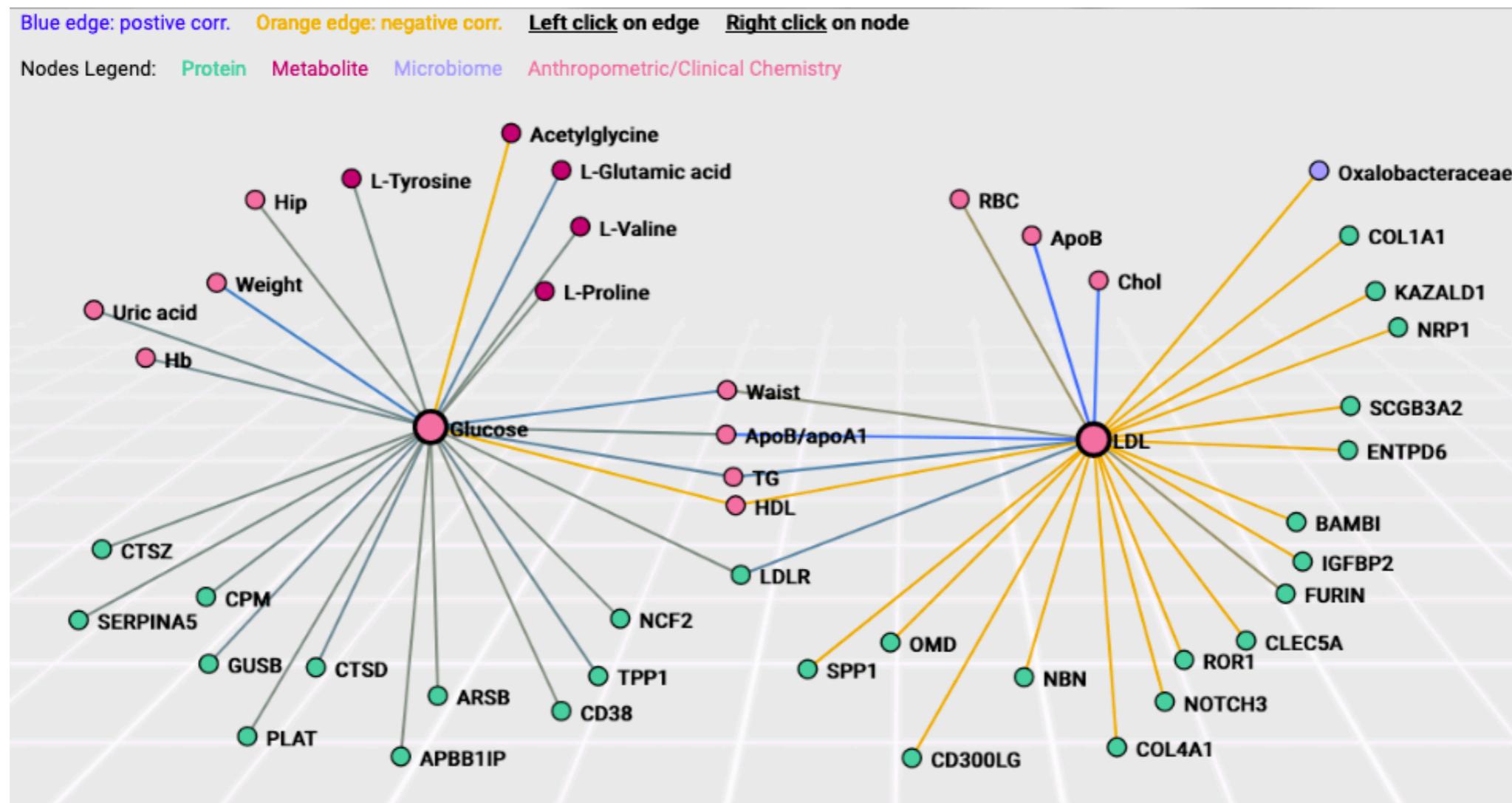
Reactome

WikiPathways

MSigDB

# 4. Knowledge-based graph creation

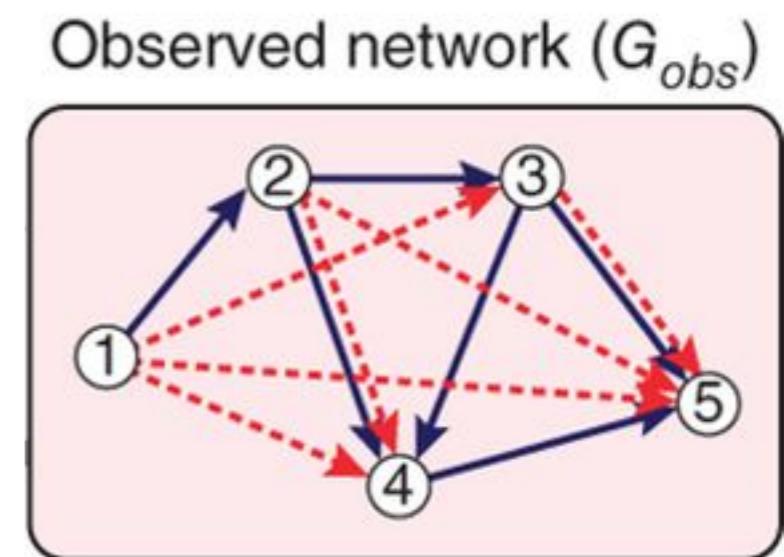
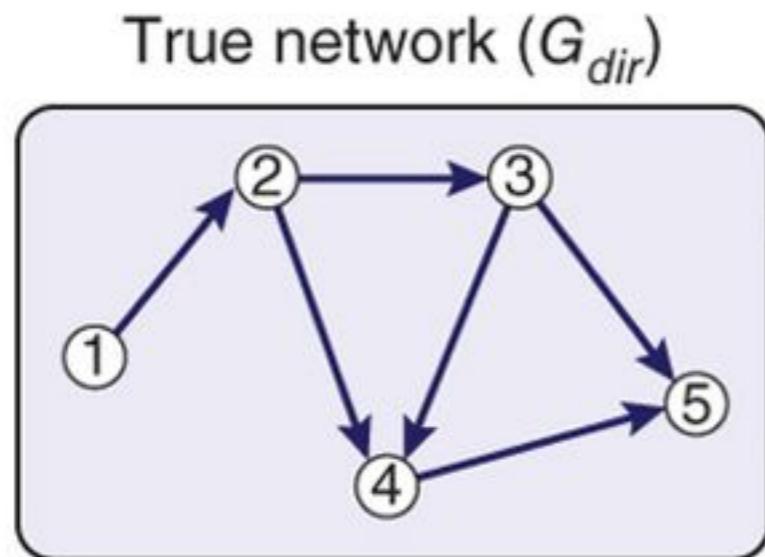
## Multi-omic biological networks



## 4. Knowledge-based graph creation

How to overlay your data based on known interactions?

- Filter your predicted interactions based on known information? (intersection)
- Add interactions that are not found in the reference networks?
- Simply consider interactions based on physical presence considering the reference networks?

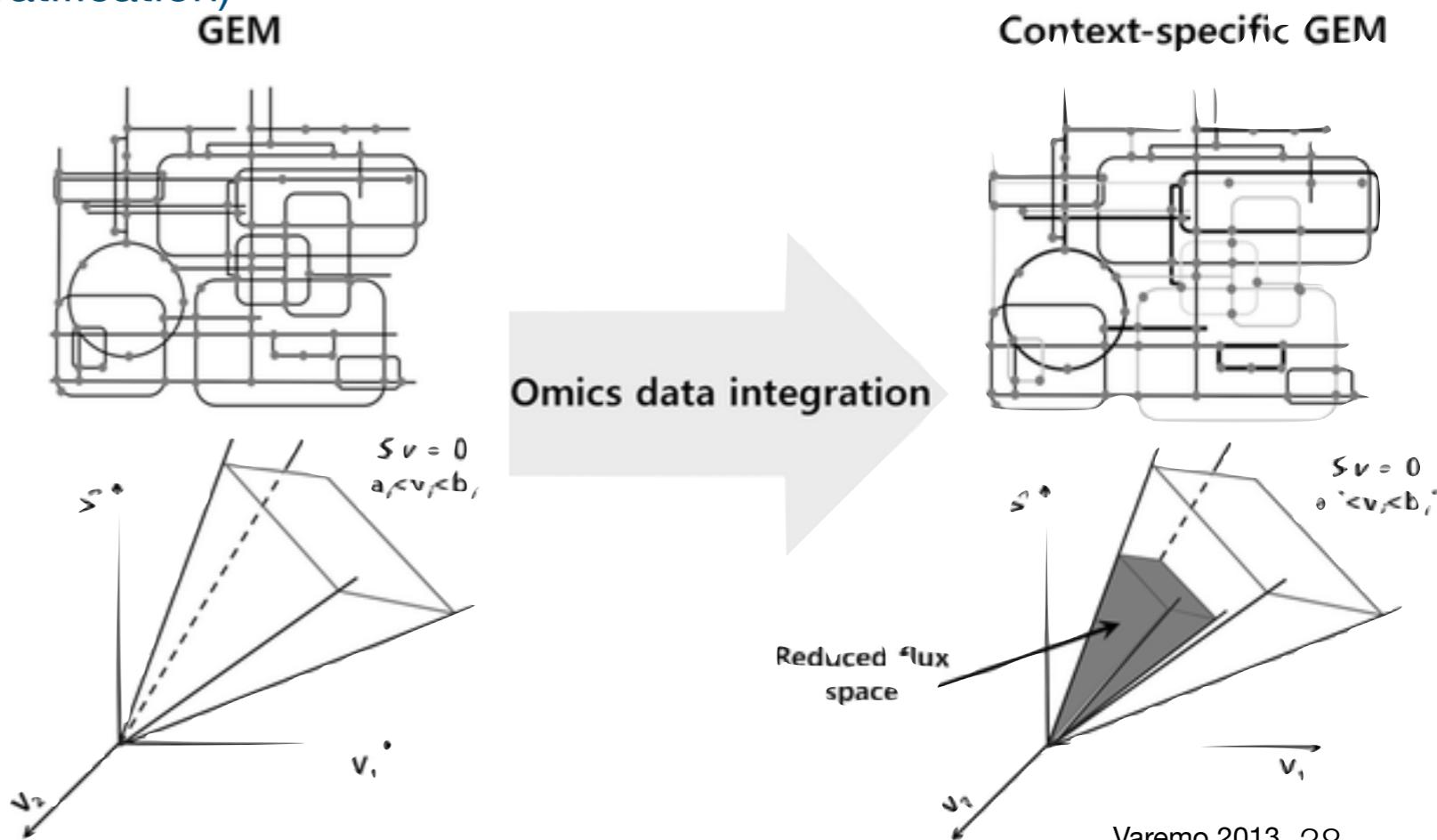


## 5. Using genome-scale metabolic models for graph creation

GEMs may be used to find such missing relationships, but there is a coverage issue

The overall strategy follows

1. Integration and flux prediction
2. Compute metabolite-reaction-gene relationships
3. Extract relevant relationships (met-met, met-gene, ...)
- 3b. Exclude unnecessary interactions (e.g. cofactors)
4. Downstream analysis (e.g. topology, stratification)

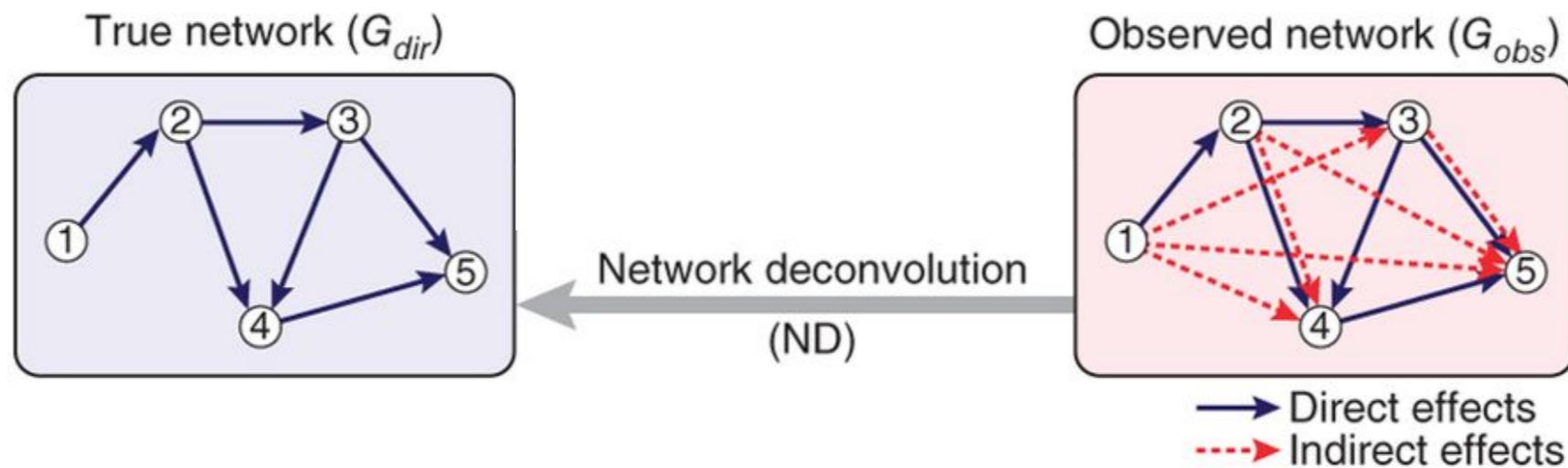


# 6. Network deconvolution

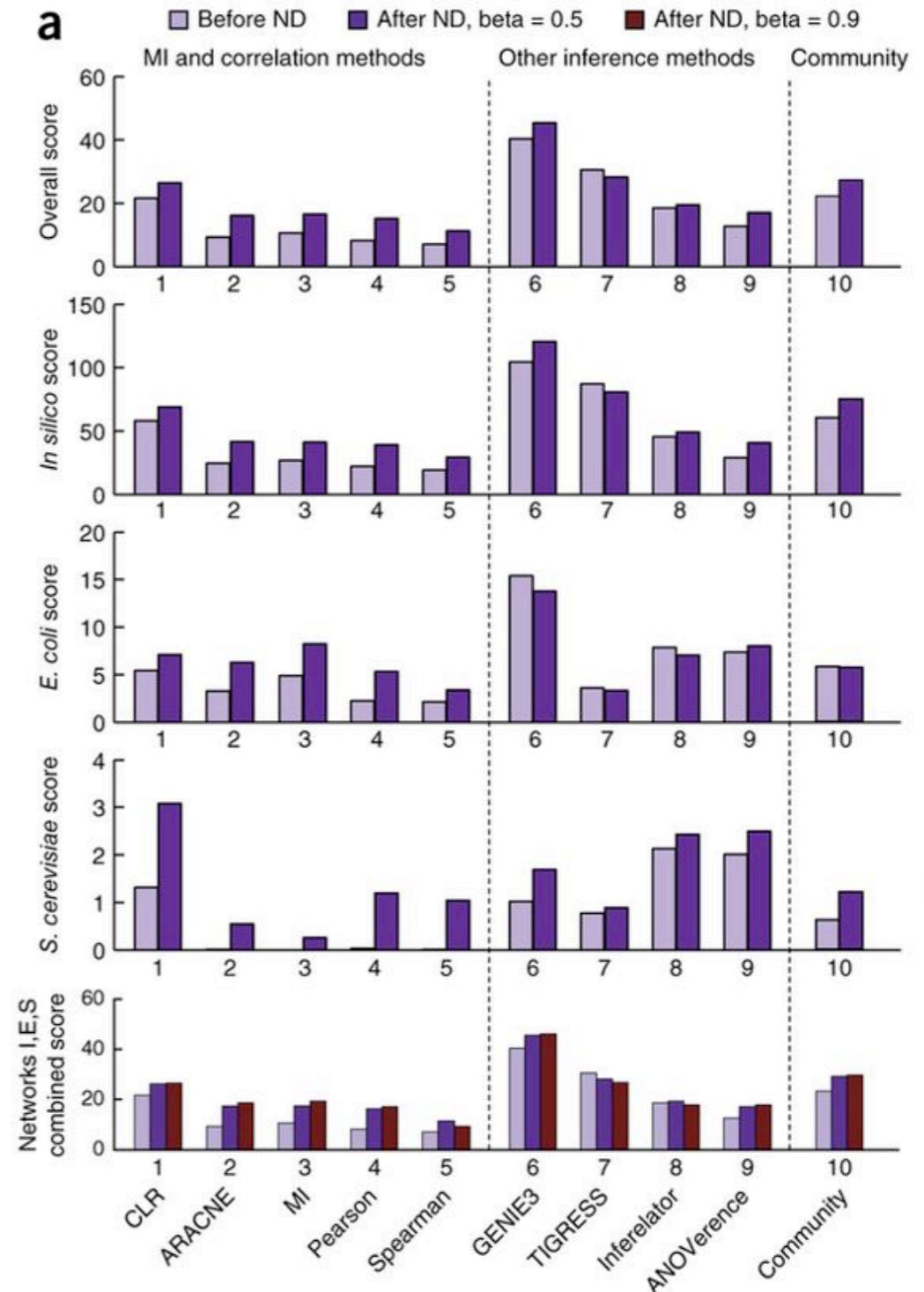
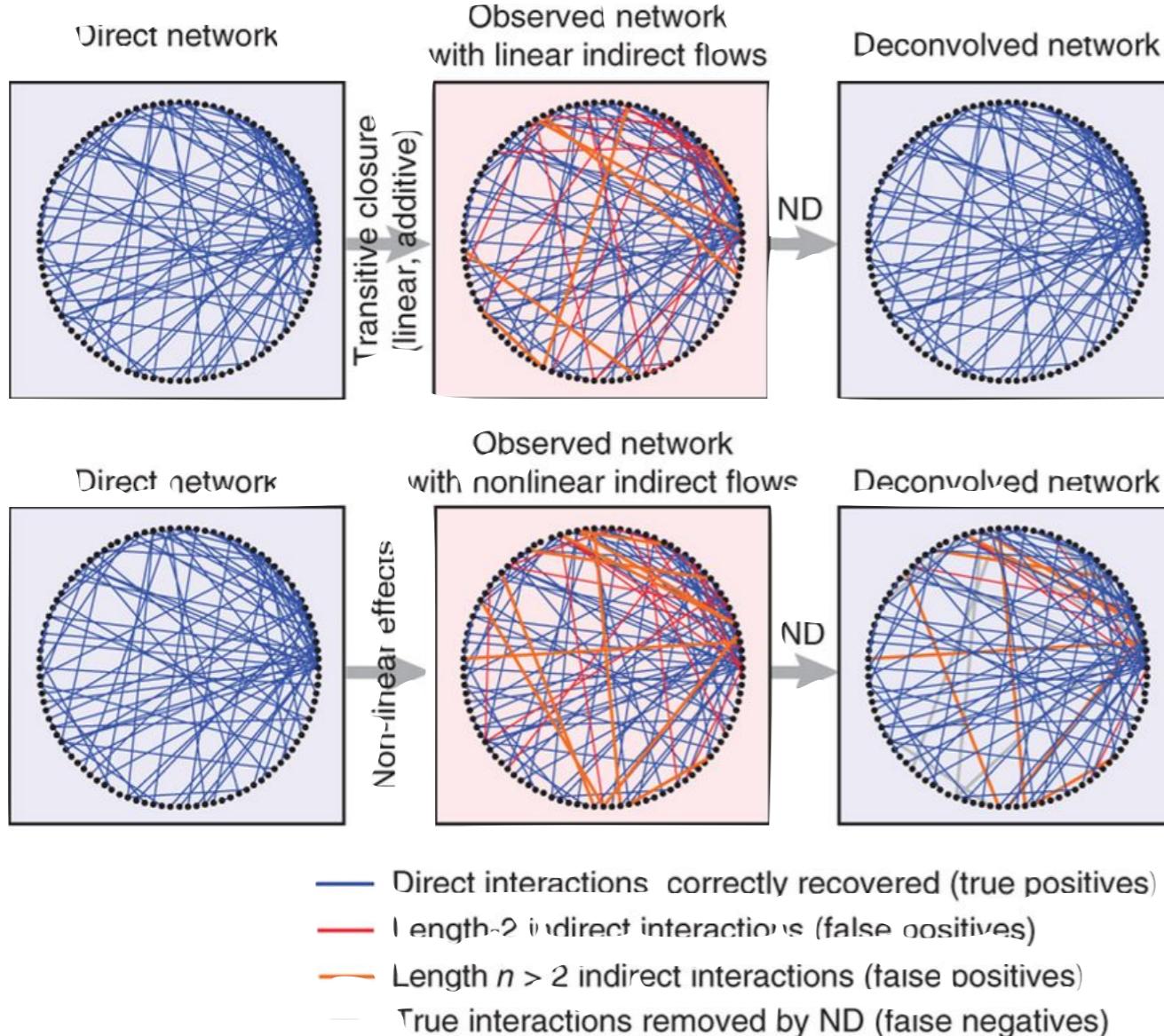
Biology is **noisy**, which may result in edges that are not true

- $1 \rightarrow 2$
- $2 \rightarrow 3$
- $1 \rightarrow 3$

Direct and indirect effects



# 5. Network deconvolution



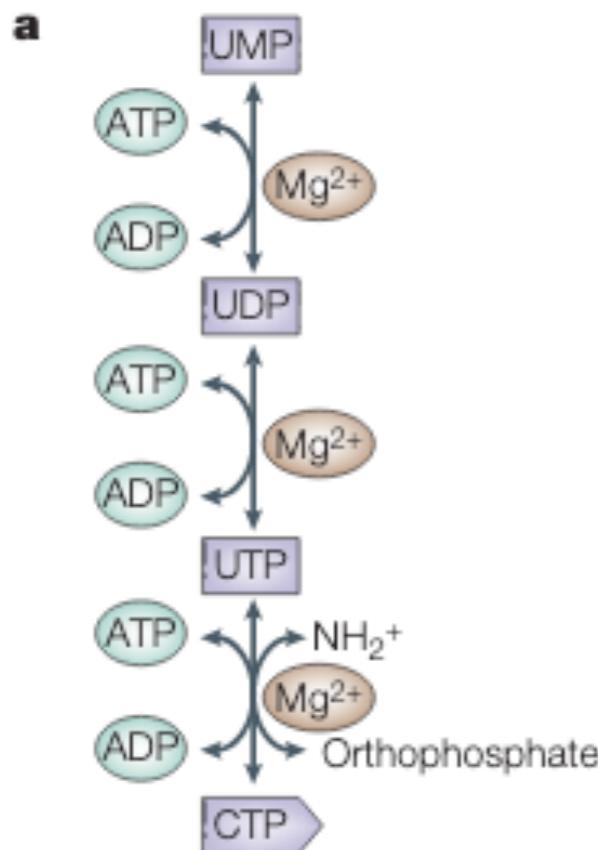
# Key network properties

1. Introduction
2. Terminology
3. Network construction
- 4. Key properties**
5. Community analysis
6. Visualization
7. Workshop

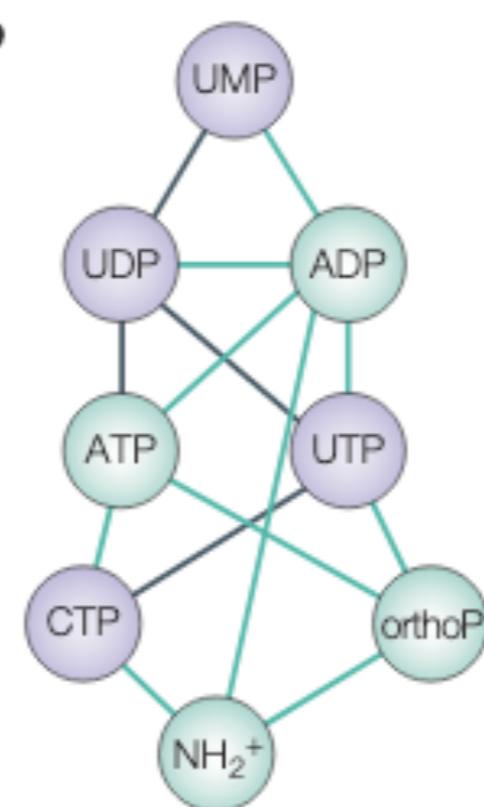
# 1. Network representations

## Representations of a metabolic network: pyrimidine metabolism

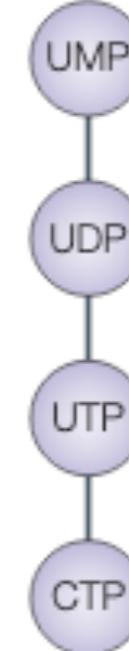
### Metabolism



Graph representation:  
metabolites and co-factors



metabolite-metabolite  
association



Other representations: Protein-Protein, Protein-Metabolite, ...

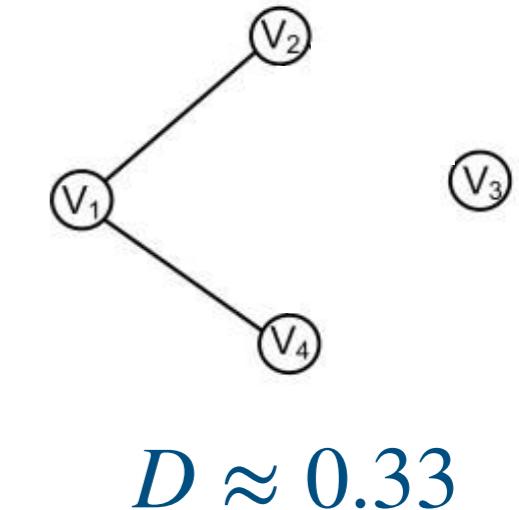
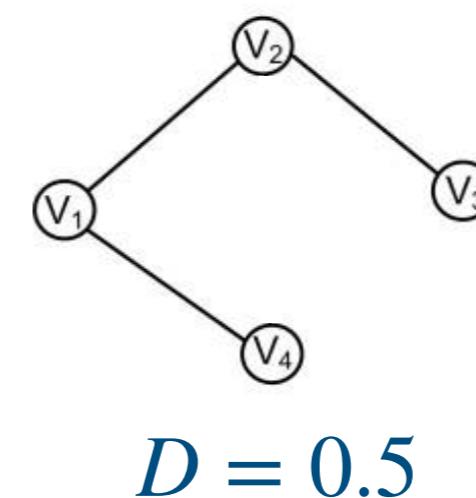
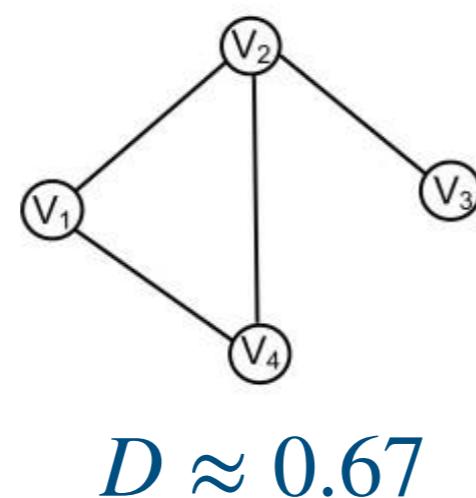
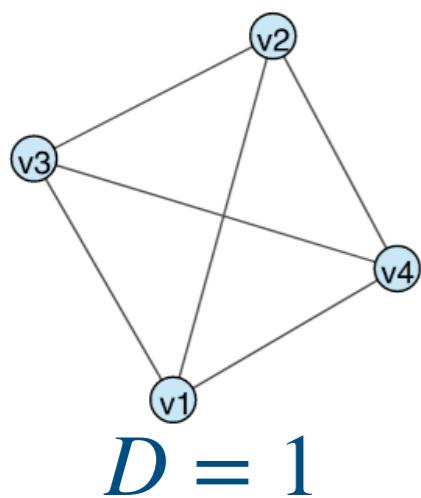
## 2. Network density

---

A **dense graph** is a graph where the number of edges approximates the maximum possible number of edges for the given node number.

$$0 \leq D \leq 1$$

Higher density indicates higher associations in the network, which implies lower resilience to changes.



## 2. Biological network density

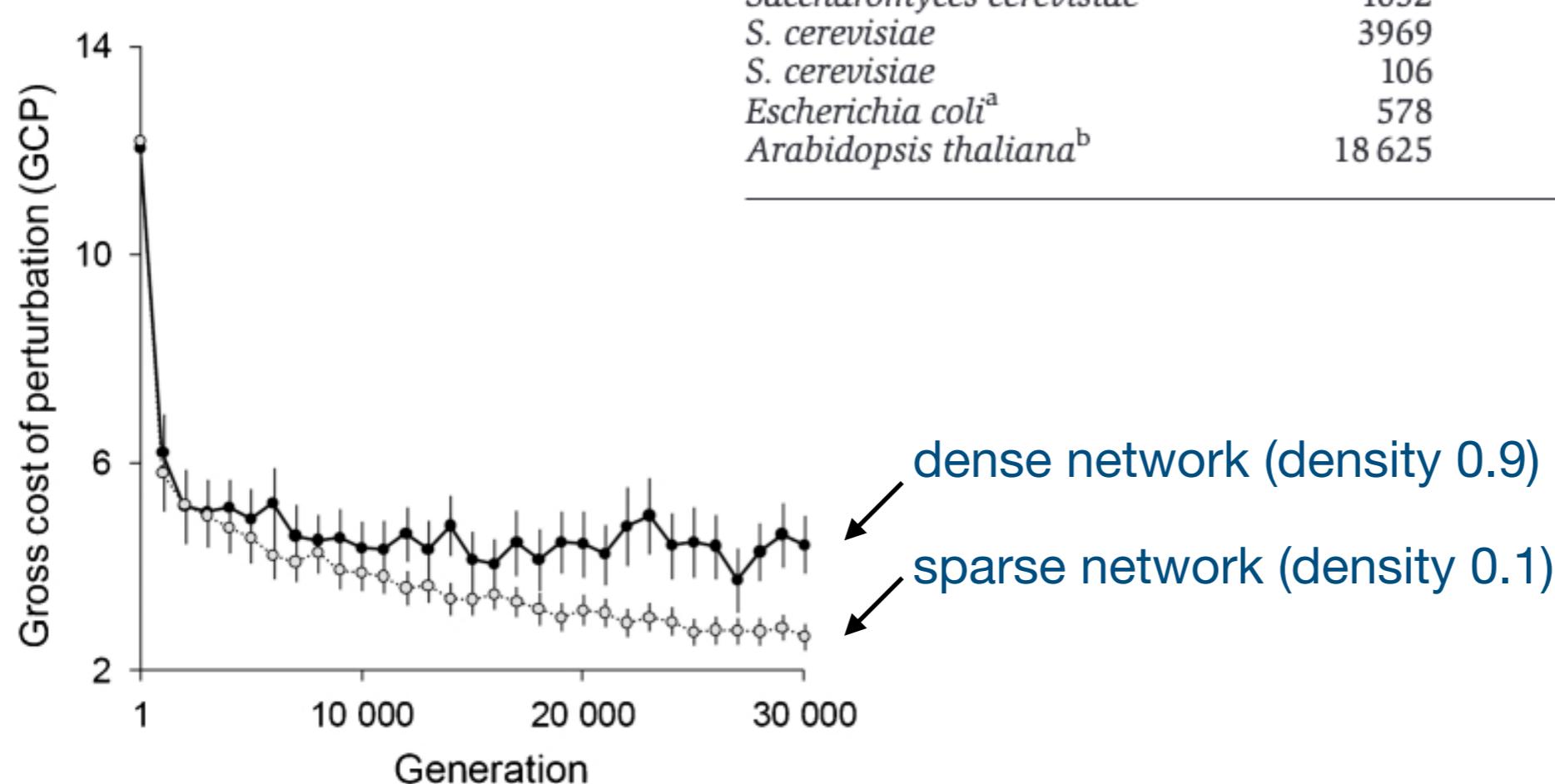
Evolutionary analysis of biological networks indicates general sparsity

Network structure must balance robustness to mutation, stochasticity and environmental queues

Sparse networks show higher robustness when accounting for costs and benefits of complexity

**Table I** Biological networks are sparsely connected

| Organism                                 | Interactions | Genes | D      | K    |
|--|--------------|-------|--------|------|
| <i>Drosophila melanogaster</i>           | 29           | 14    | 0.148  | 2.07 |
| <i>D. melanogaster</i>                   | 45           | 25    | 0.072  | 1.8  |
| Sea urchin                               | 82           | 44    | 0.0065 | 1.86 |
| <i>Saccharomyces cerevisiae</i>          | 1052         | 678   | 0.0023 | 1.55 |
| <i>S. cerevisiae</i>                     | 3969         | 2341  | 0.0007 | 1.7  |
| <i>S. cerevisiae</i>                     | 106          | 56    | 0.0338 | 1.9  |
| <i>Escherichia coli</i> <sup>a</sup>     | 578          | 423   | 0.0032 | 1.37 |
| <i>Arabidopsis thaliana</i> <sup>b</sup> | 18 625       | 6760  | 0.0004 | 2.75 |

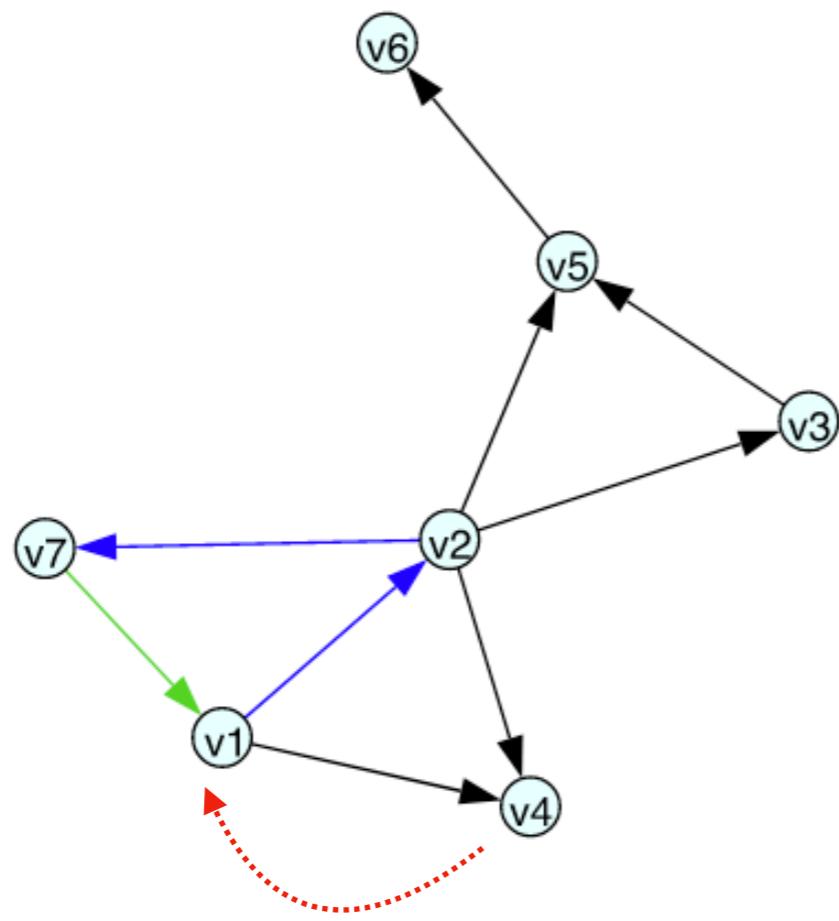


# 3. Paths

---

Distance between nodes is measured in path length

In directed graphs, the shortest path between  $(a, b) \neq (b, a)$



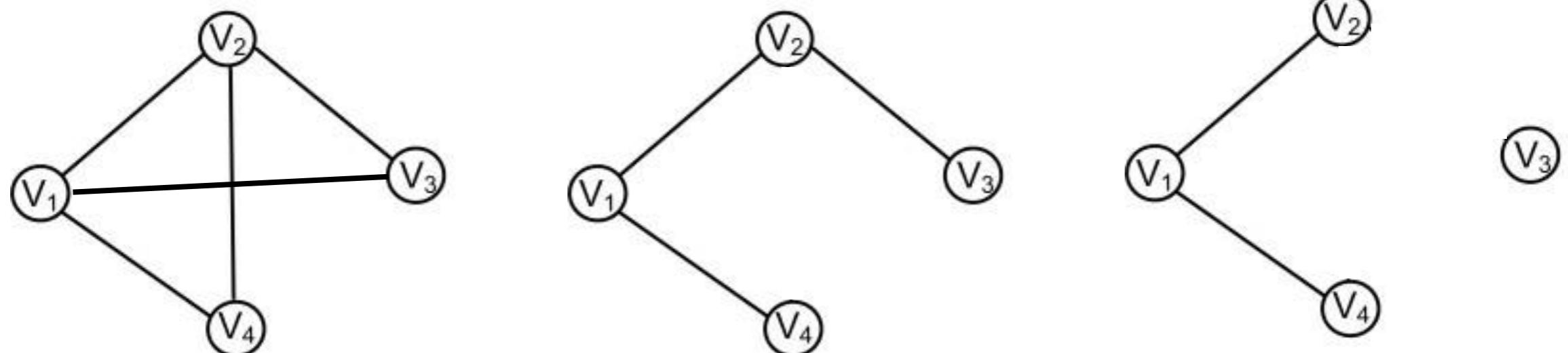
|    | v1  | v2  | v4  | v3  | v5  | v7  | v6  |
|----|-----|-----|-----|-----|-----|-----|-----|
| v1 | 0.0 | 1.0 | 1.0 | 2.0 | 2.0 | 2.0 | 3.0 |
| v2 | 2.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 |
| v4 | inf | inf | 0.0 | inf | inf | inf | inf |
| v3 | inf | inf | inf | 0.0 | 1.0 | inf | 2.0 |
| v5 | inf | inf | inf | inf | 0.0 | inf | 1.0 |
| v7 | 1.0 | 2.0 | 2.0 | 3.0 | 3.0 | 0.0 | 4.0 |
| v6 | inf | inf | inf | inf | inf | inf | 0.0 |

## 4. Connectivity

---

**Node connectivity**  $\kappa(G)$ : minimum number of **nodes** whose removal renders the network disconnected

**Edge connectivity**  $\lambda(G)$ : minimum number of **edges** whose removal renders the network disconnected



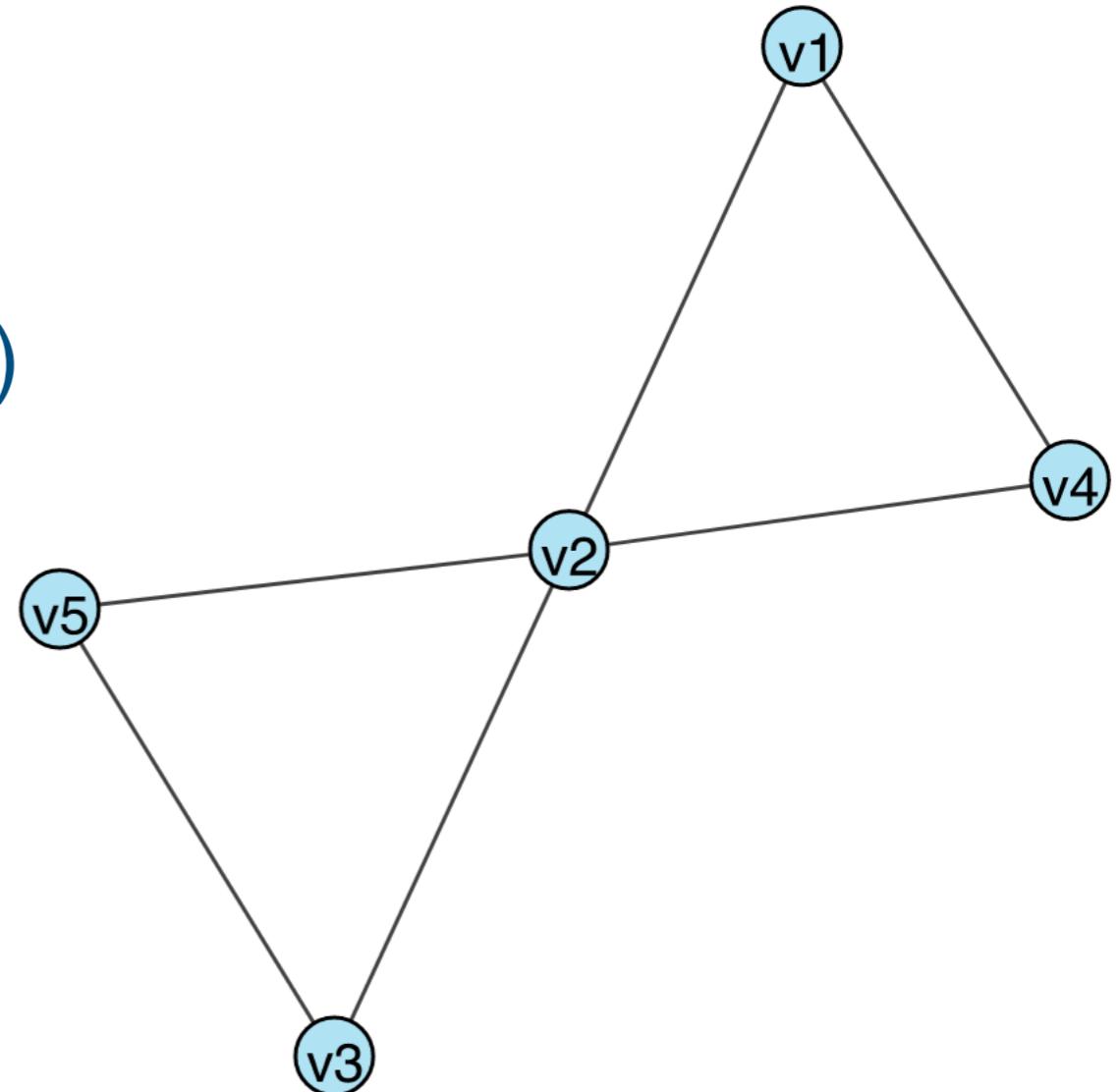
## 4. Connectivity

---

$\kappa(G) = 1$ ; **cut:**  $v_2$

$\lambda(G) = 2$ ; **bridge:** (  $(v_2, v_1)$  &  $(v_2, v_4)$  )

**Local connectivity** may also be computed for any given pair of vertices  
(e.g.  $v_3, v_1$ :  $v_2$  and associated edges)

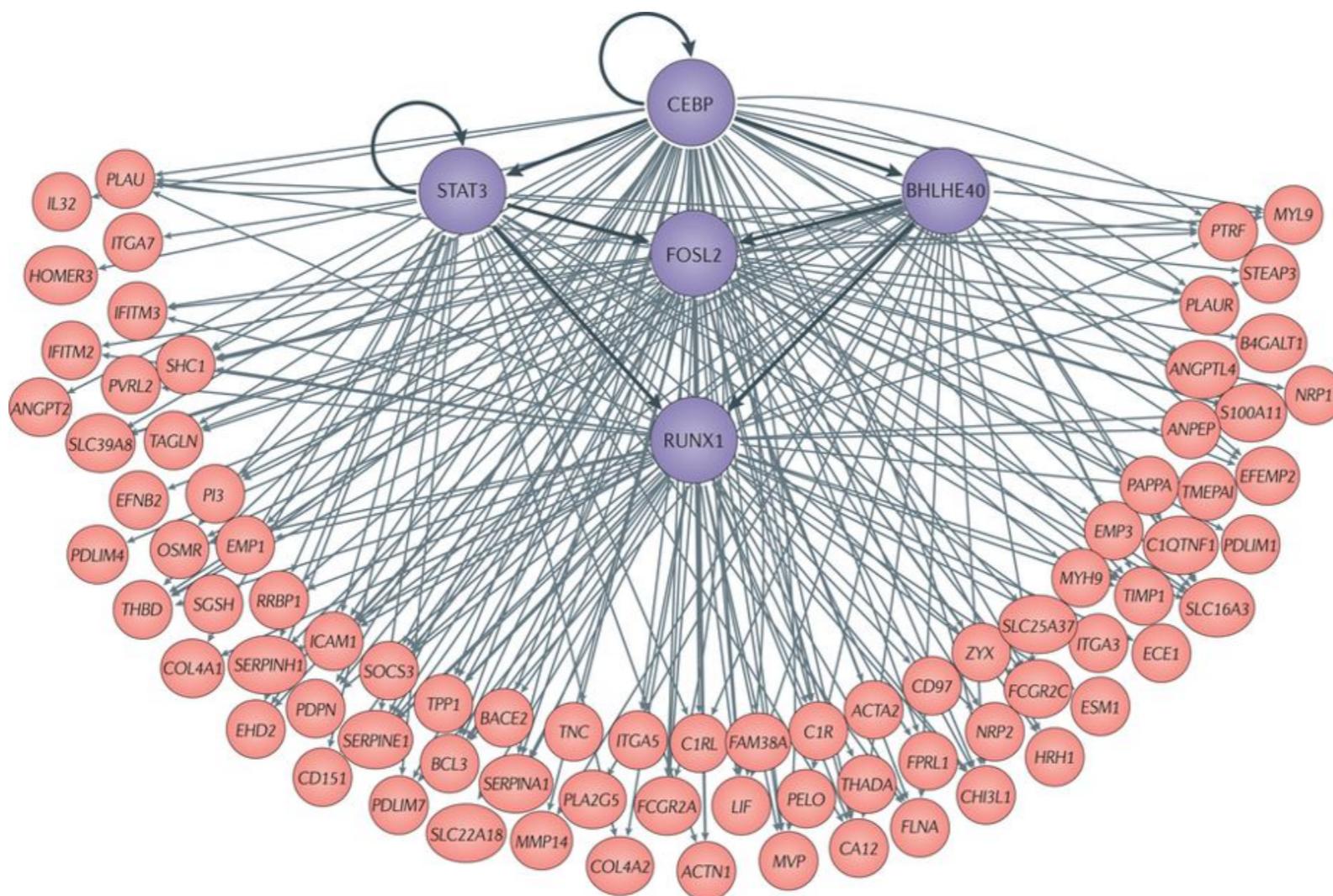


# 5. Centrality

Indicate the most central nodes in a network

Central nodes may act as **hubs** and have important roles

Example: Transcription Factor Master Regulators



# 5. Centrality

---

There are many different measures of centrality:

- Degree
- Eccentricity
- Closeness
- Betweenness
- Eigenvector
- Katz
- PageRank
- Percolation
- Cross-clique

...

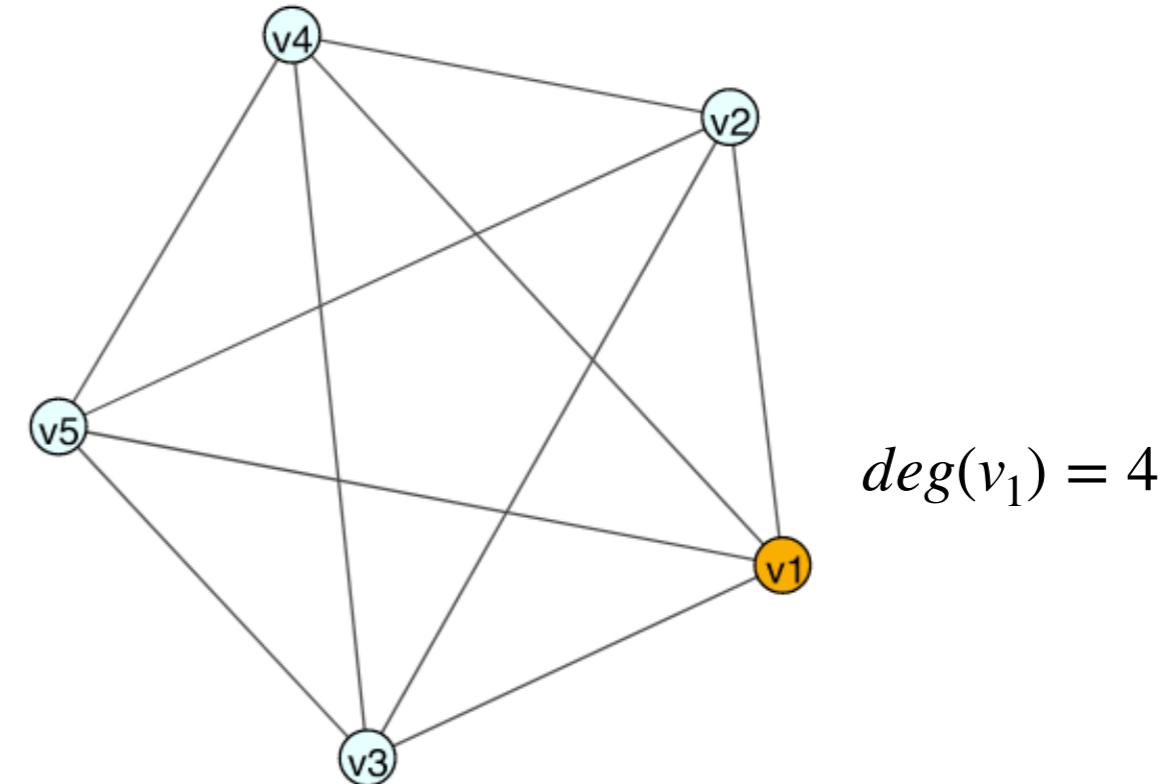
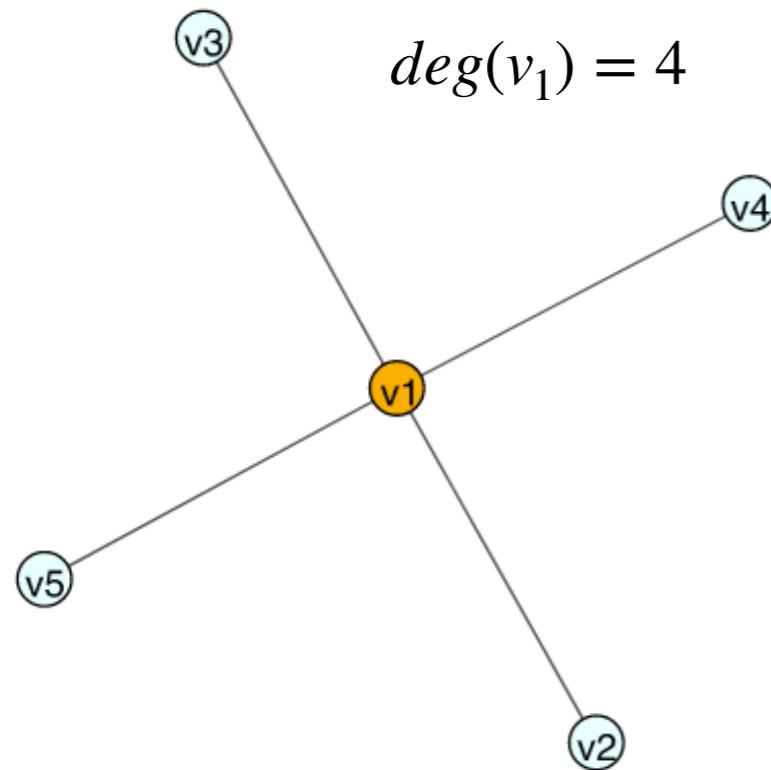
## 5. Centrality: degree centrality

---

Degree indicates the number of connections with a node

$$d(v) = |N(i)|$$

where  $N(i)$  is the number of 1st neighbours of a node.



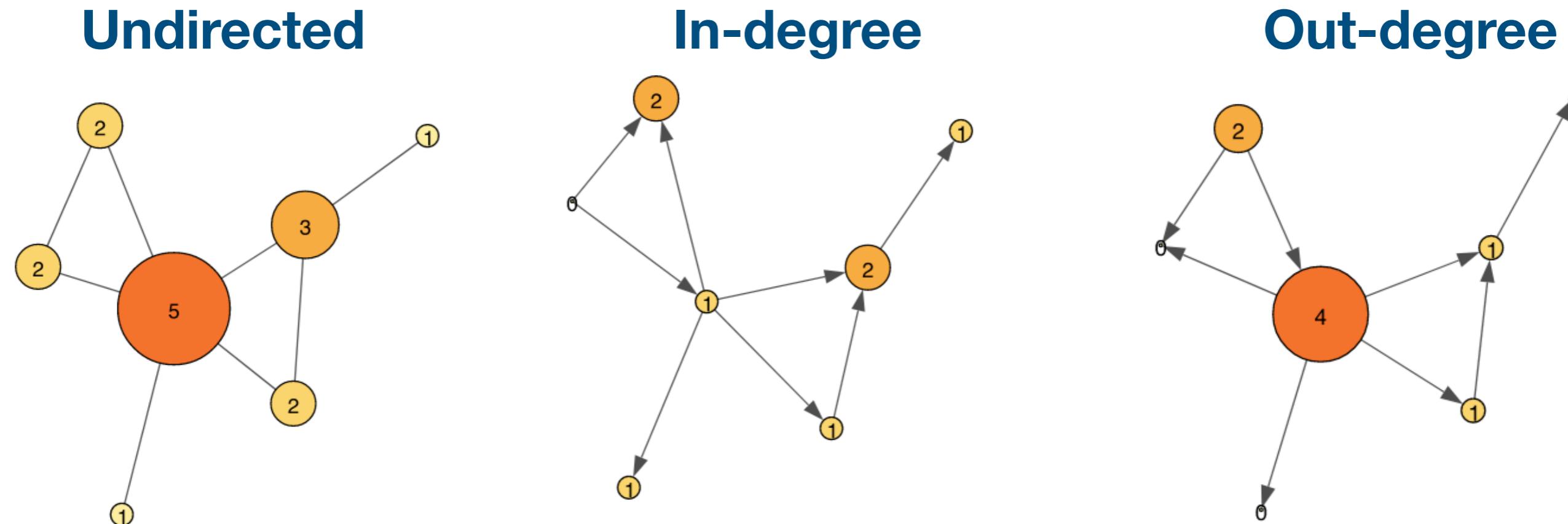
# 5. Centrality: degree centrality

Undirected networks vs directed networks

**In-degree vs Out-degree**

$$C_D(v_i) = \sum_{j=1}^N e_{ij}$$

Numbers indicate degree:



# 5. Centrality: degree centrality

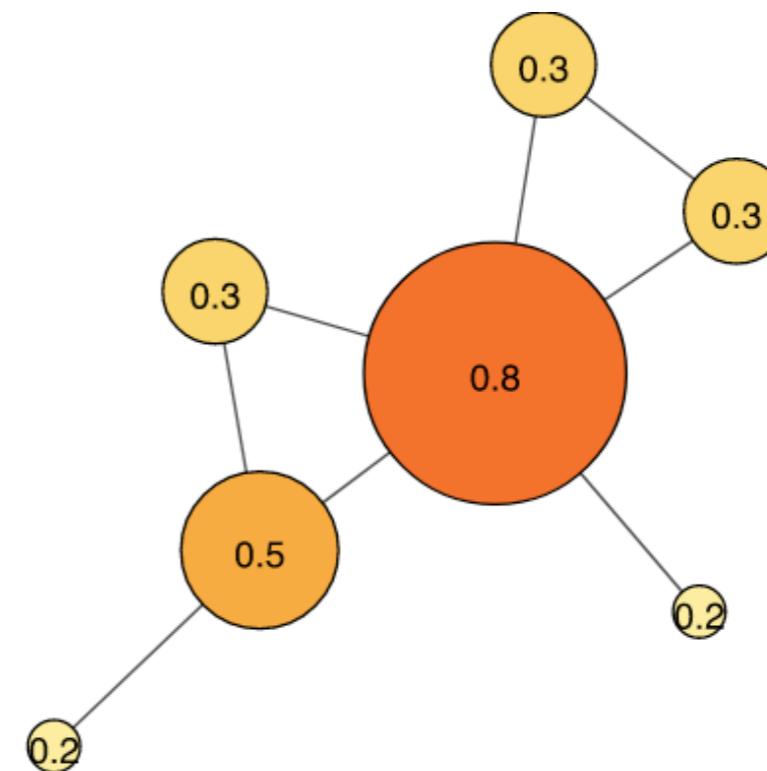
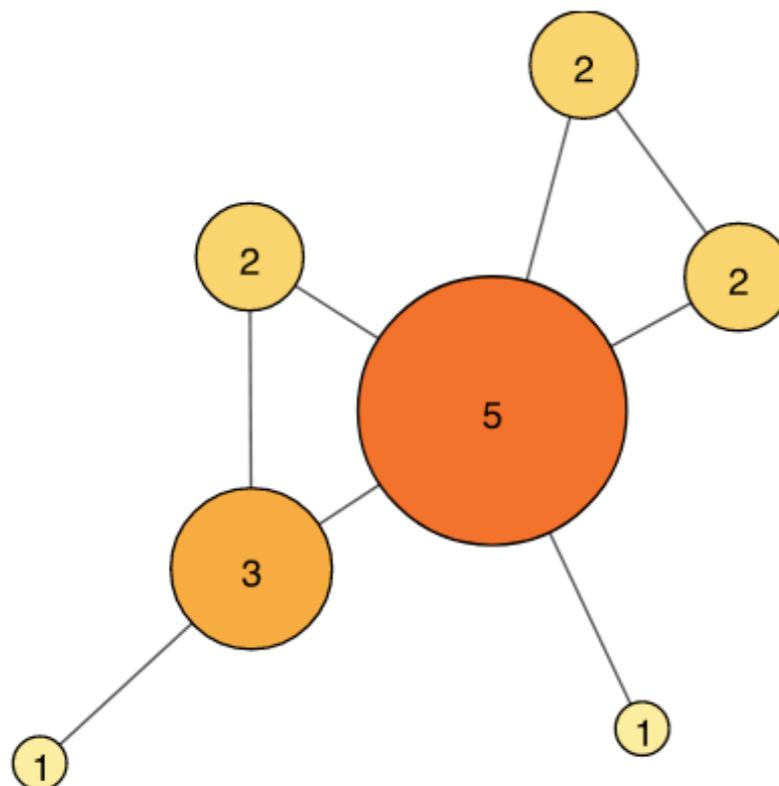
Degree centrality

$$C_D(v_i) = \sum_{j=1}^N e_{ij}$$

Normalized  
degree centrality

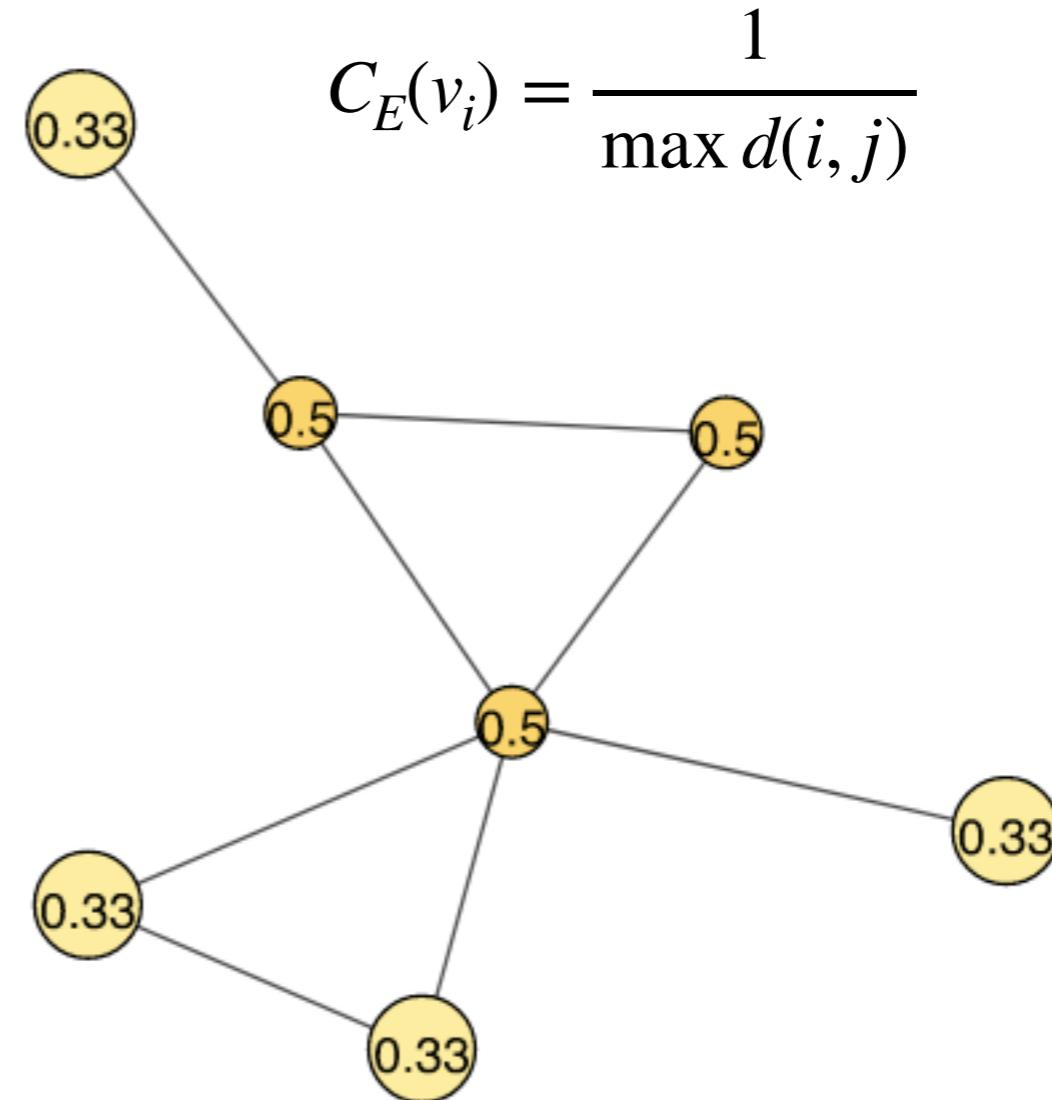
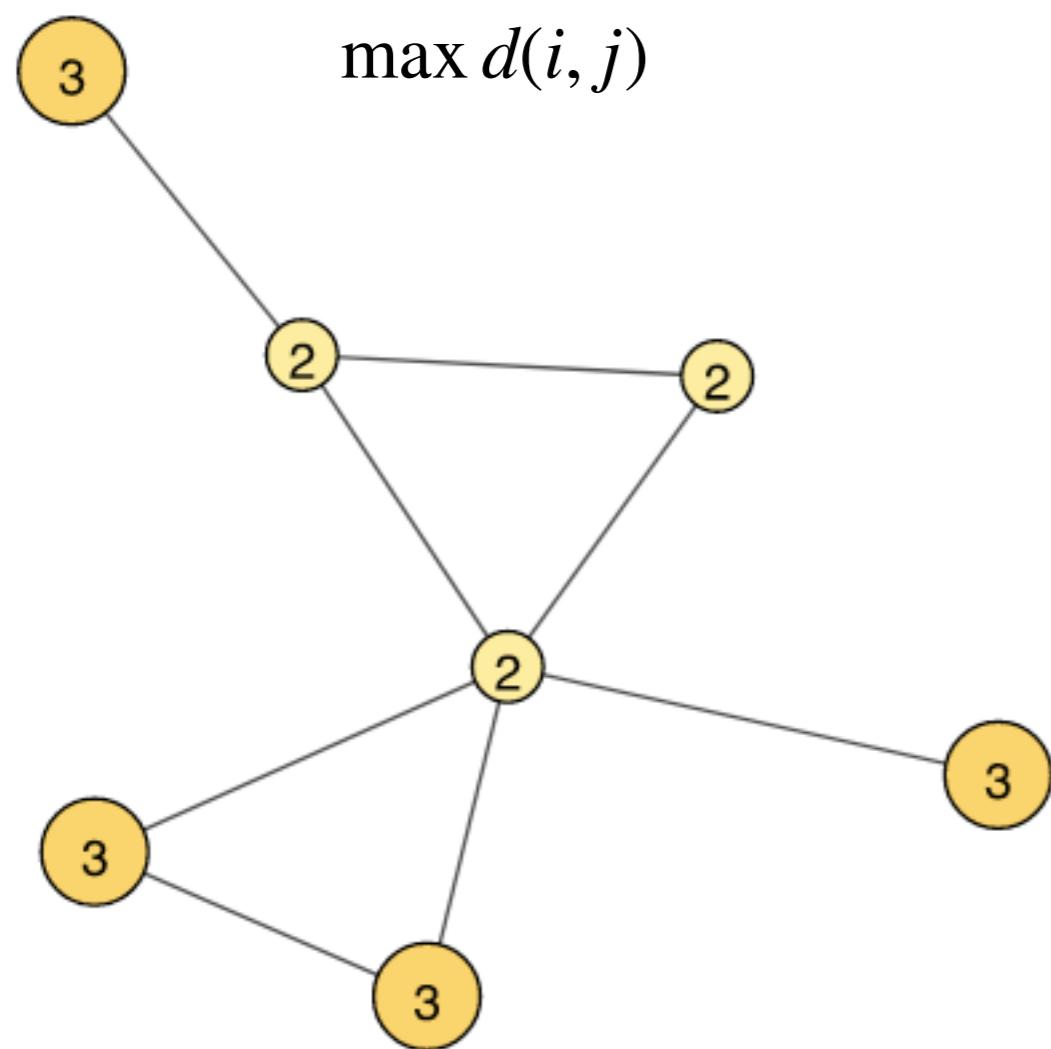
$$C_D(v_i) = \frac{\sum_{j=1}^N e_{ij}}{N - 1}$$

Centrality normalization allows for comparison between networks of different sizes



## 5. Centrality: eccentricity centrality

Eccentricity considers a node's max path to all other nodes



# 5. Centrality: limitations & influence

---

Node centrality does not necessarily imply **importance**

How to tackle this?

1. Complement with experimental observations
2. Compute multiple metrics and summarise joint observations
3. Measure **information transmission** rather than *connectiveness*
  - **Accessibility**
  - **Dynamic influence**
  - **Impact**
  - **Expected force**

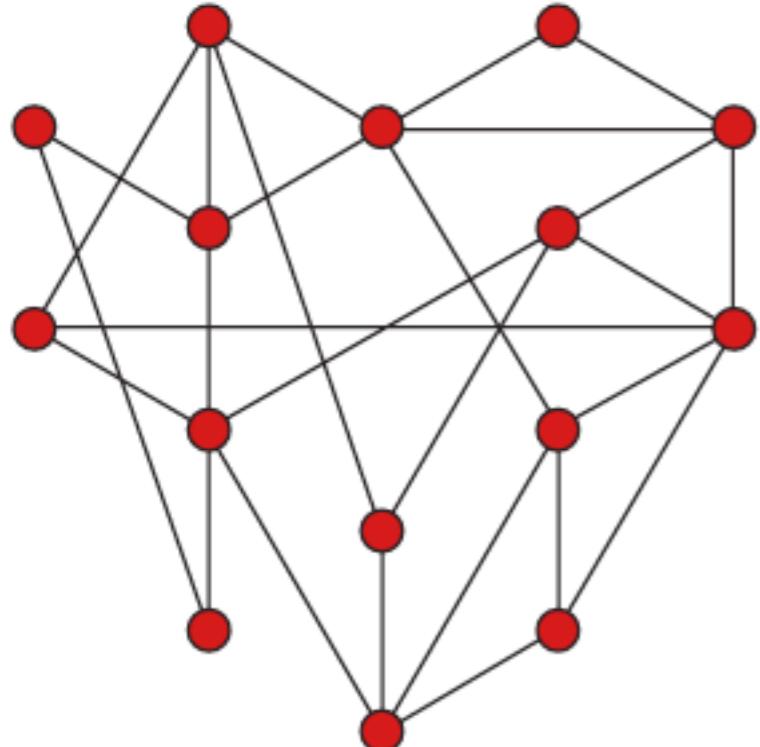
## 6. Degree distribution

---

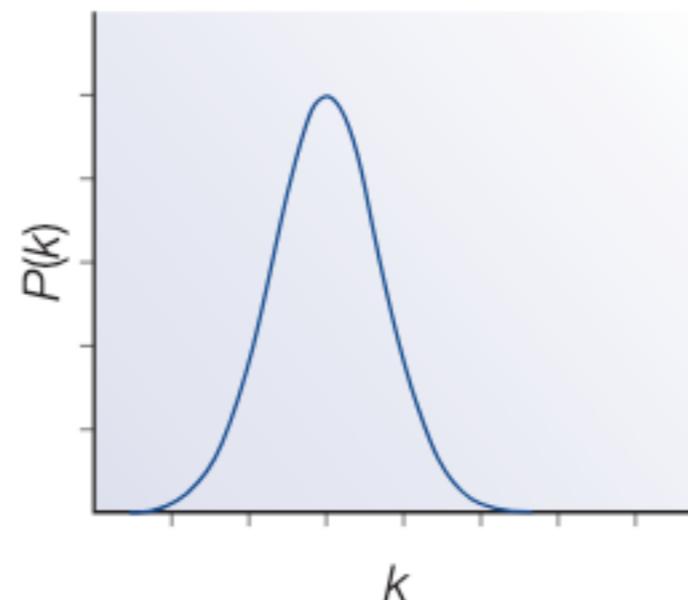
$P(k)$  gives the probability that a selected node has exactly  $k$  edges

Allows distinguishing different kinds of networks

**Random network**



**Poisson degree distribution**  
shows no highly connected nodes

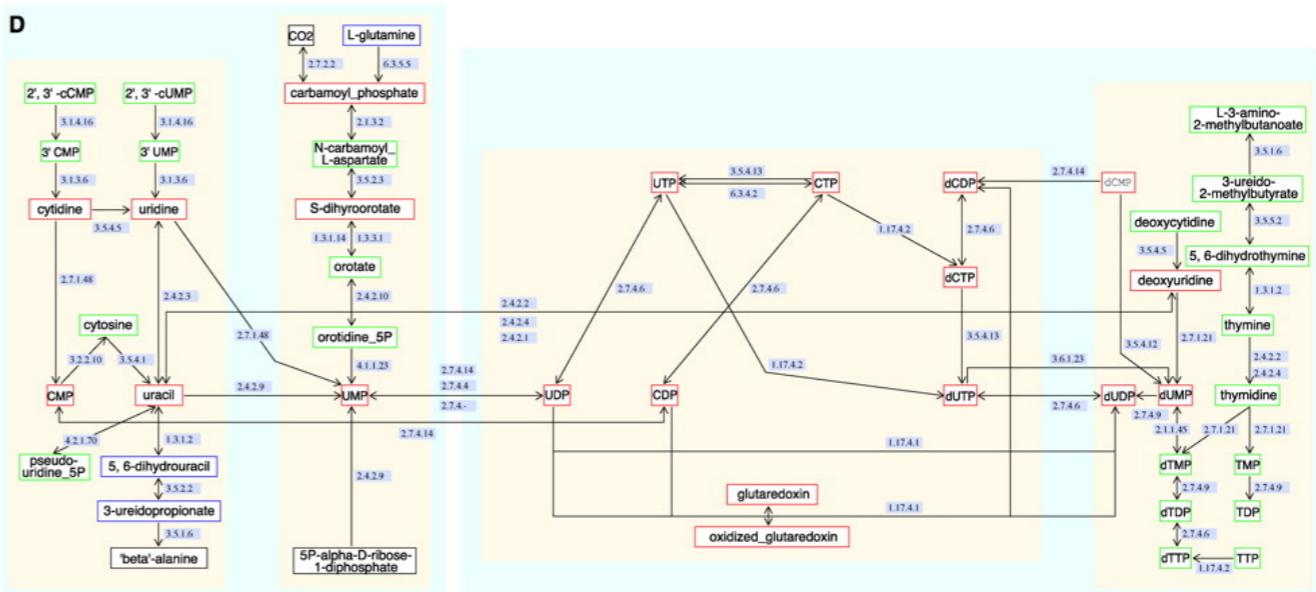
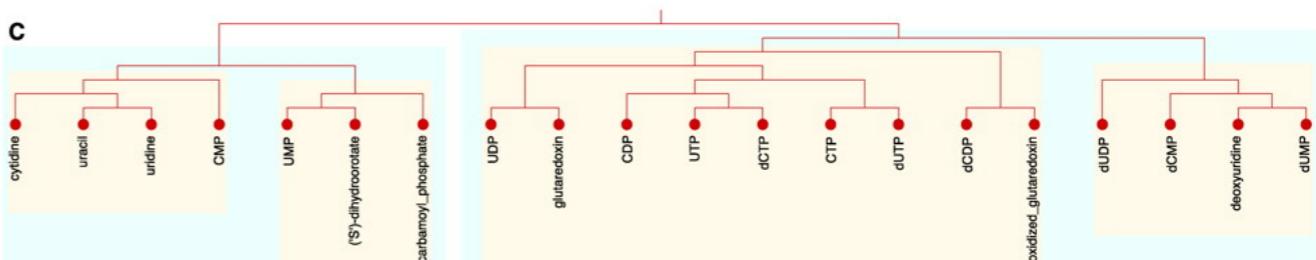
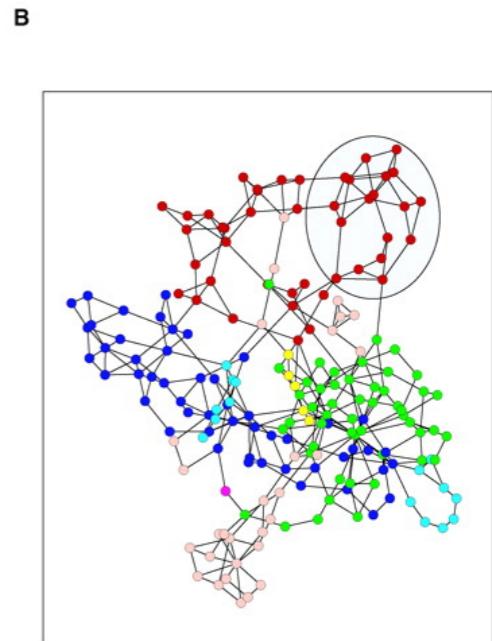


Most nodes have near  $\langle k \rangle$

# Metabolic networks show hierarchical topology

Metabolic networks of 43 organisms are organised into **small, tightly connected modules**

Their combination shows a hierarchical structure



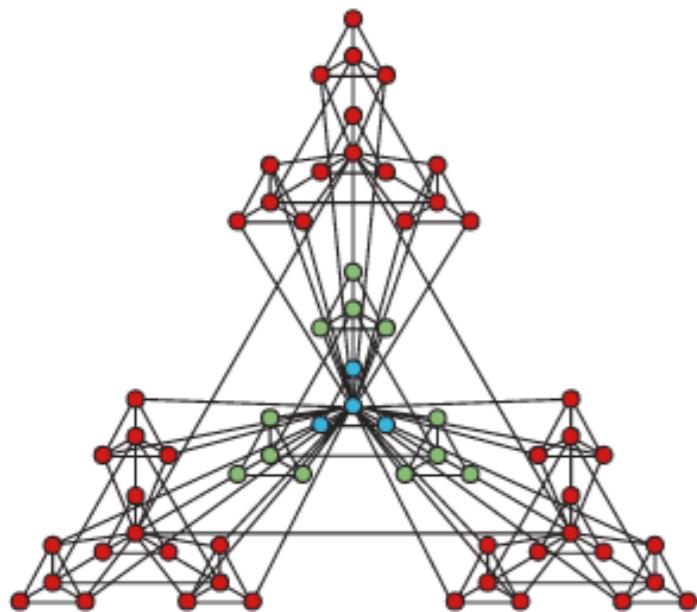
# 6. Degree distribution

Biological networks do not follow topology features of random networks.

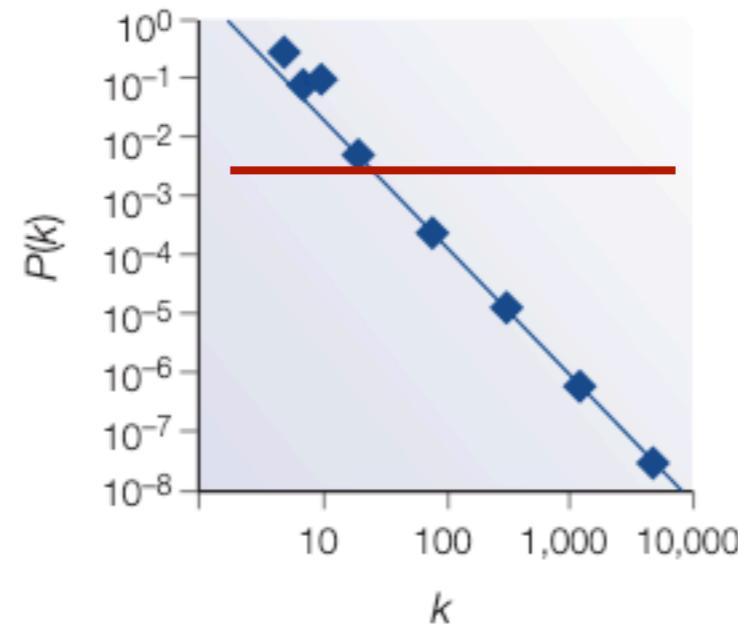
Degree distribution *follows* the power-law  $P(k) \propto k^{-\gamma}$

This allows for high robustness to node failure: removal of <80% nodes still retains paths between any two nodes

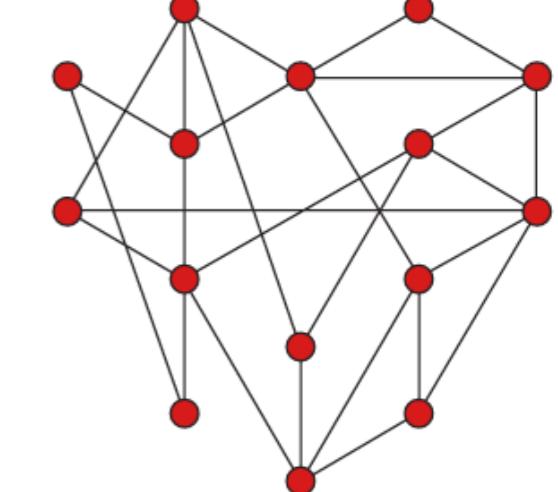
**Hierarchical network**



**Degree distribution**  
shows many with low degrees  
a few highly connected nodes



**Random network**



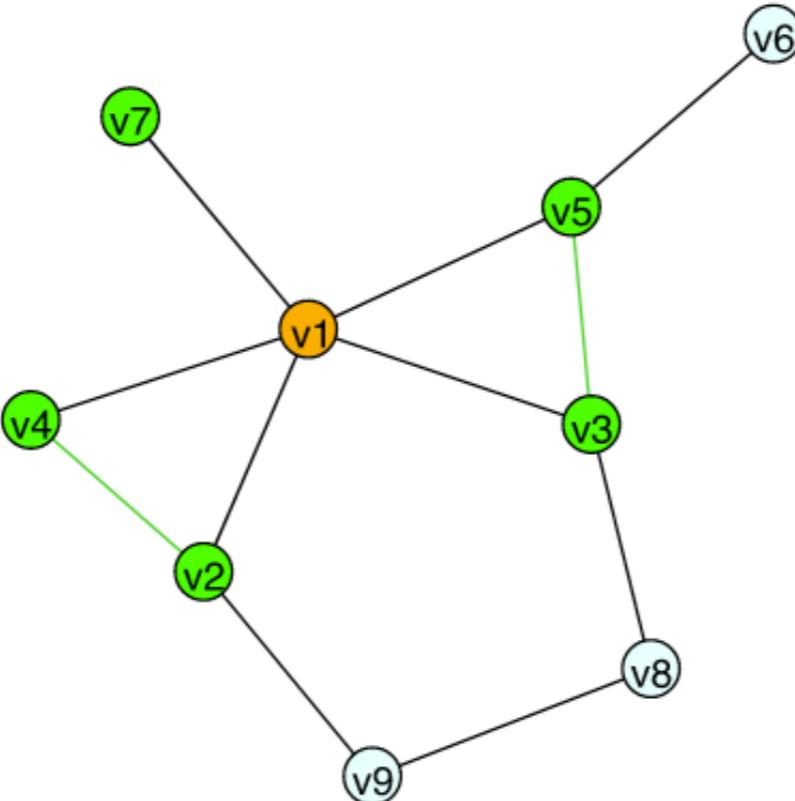
# 7. Clustering coefficient

---

Gives the **fraction of possible interconnections** for neighbours of node  $i$

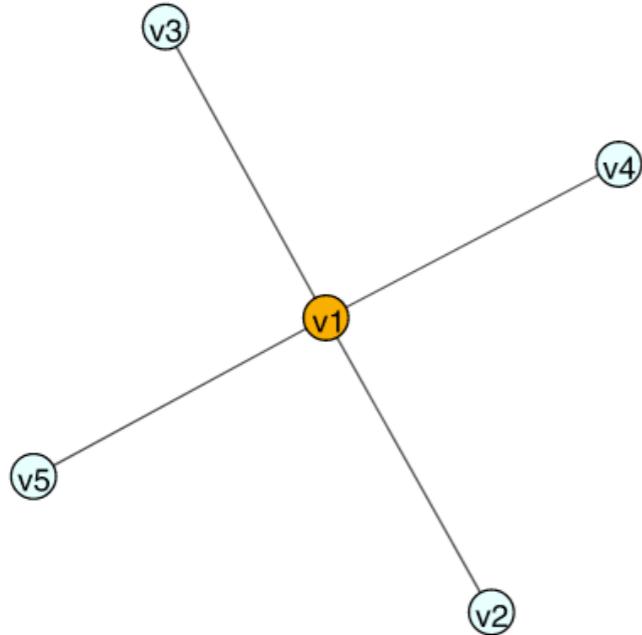
If node  $v_1$  is connected with  $v_2$  and  $v_3$ , it is very likely that  $v_2$  and  $v_3$  are also connected.

Takes into account degree of a node and the degree of its 1st neighbours

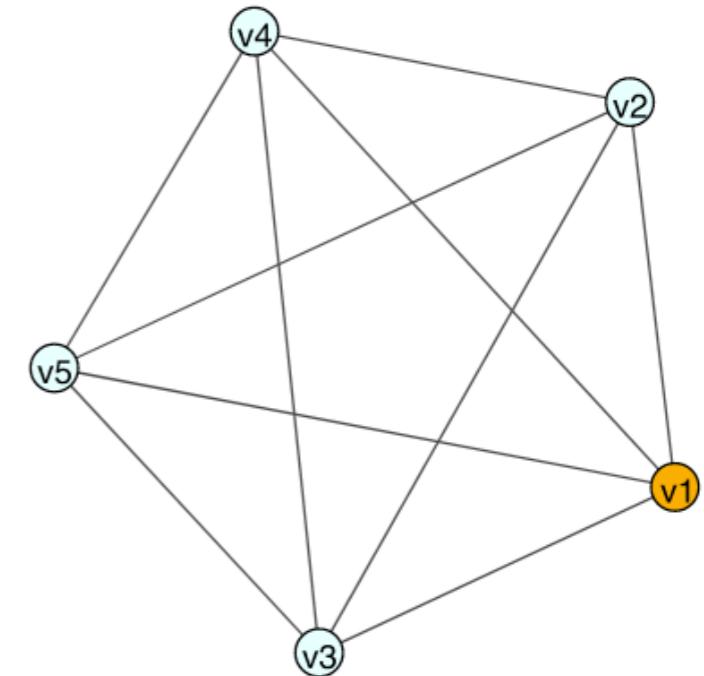


# 7. Clustering coefficient

---



$$0 \leq C_i \leq 1$$



The global clustering coefficient  $C(G)$  is simply the average of its clustering coefficients

# Overview

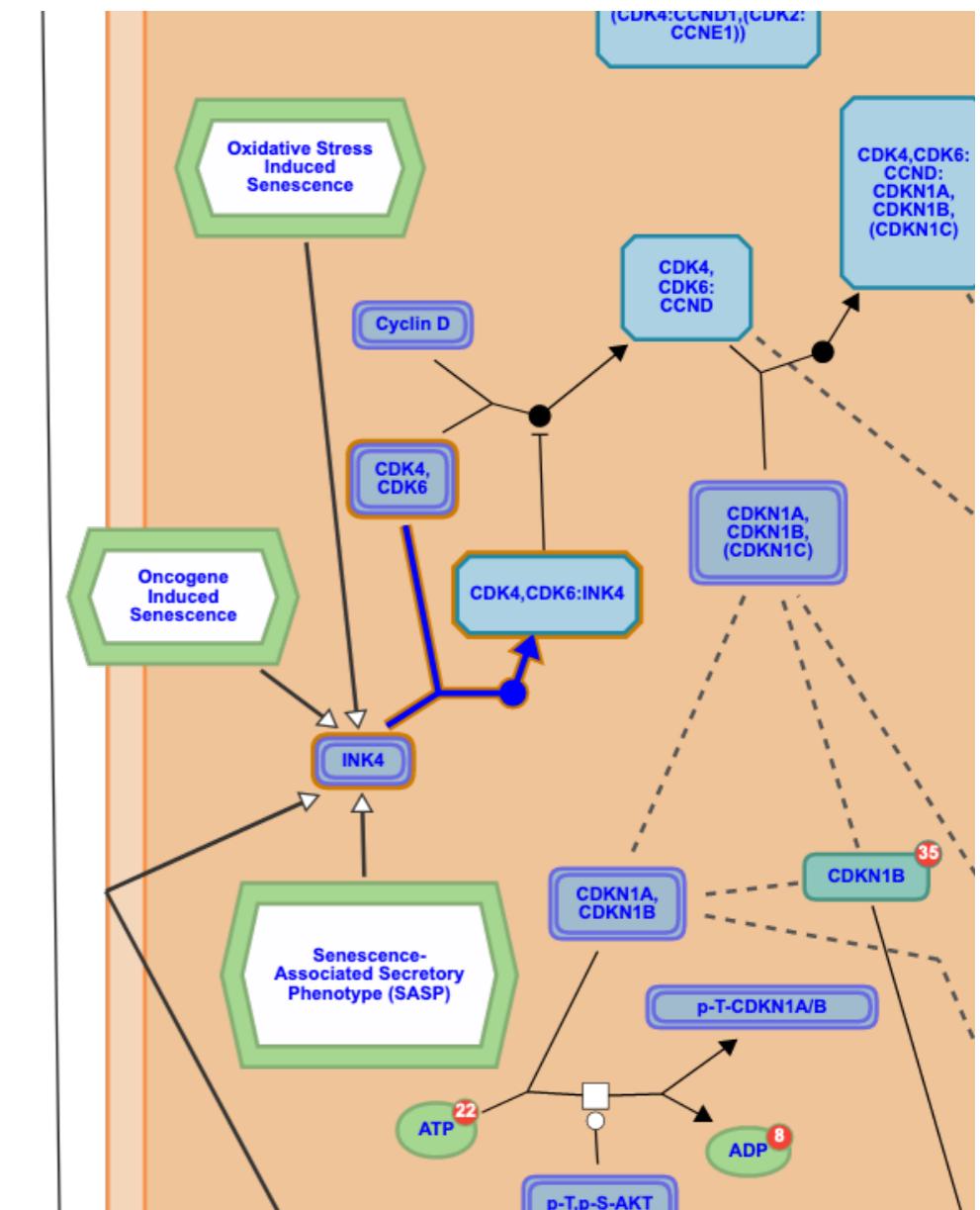
---

1. Introduction to network analysis
2. Terminology
3. Network inference
4. Key network properties
5. Community analysis
6. Workshop

# What are modules?

**Modules** are physically or functionally associated nodes that work together to achieve a distinct function

Protein complexes are physical modules



# What are modules?

In addition to physical or functional modules, one may identify other types of modules

**Topological:** derived from their high within-module degree

**Disease:** highly interconnected nodes associated with a disease response

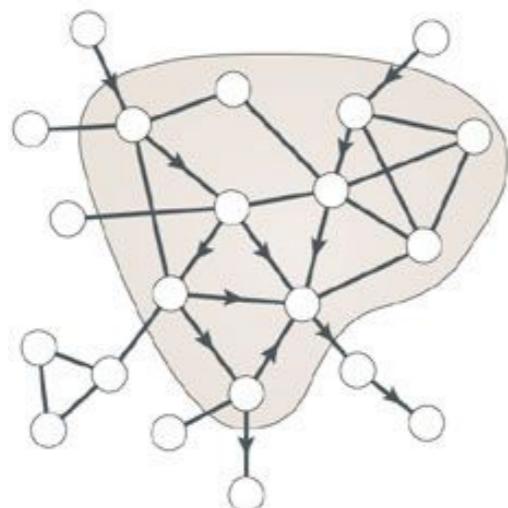
**Drug:** highly interconnected nodes associated with a drug response

**Subgroup:** highly interconnected nodes associated with a sample subgroup (e.g. cancer subtype)

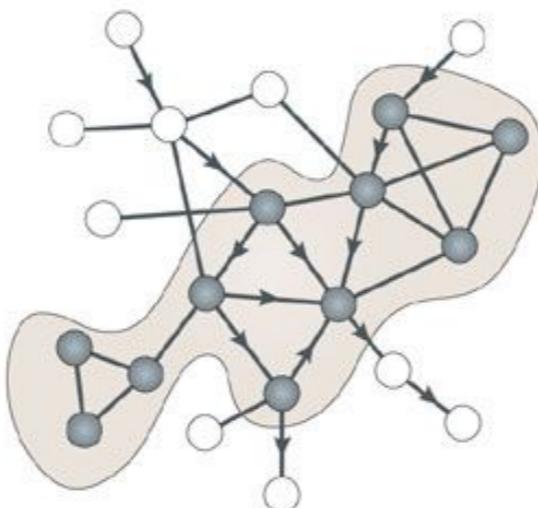
**Tissue-, cell-type-specific:** highly interconnected nodes associated with a specific tissue or cell type

Highly interlinked local regions of a network

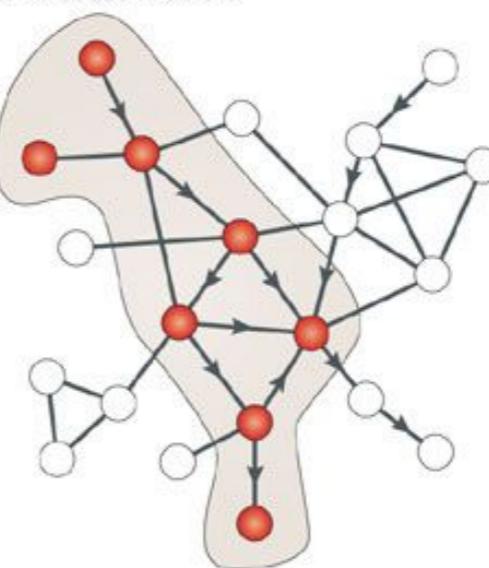
a Topological module



b Functional module



c Disease module



○ Topologically close genes (or products)

● Functionally similar genes (or products)

● Disease genes (or products)

— Bidirectional interactions

→ Directed interactions

# Modularity

**Modularity** is a property of the network

**Modularity (Q)** measures the tendency of a graph to be organised into modules

**Modules** computed by comparing probability that an edge is in a module vs what would be expected in a random network

$$Q \propto \sum_{s \in S} [(e_s) - (\text{expected } e_s)]$$

# edges in group  $s$

Random network with  
same number of nodes, edges and  
degree per node

$Q = 1$ : much higher number of edges than expected by chance

$-1 < Q < 1$        $Q = -1$ : lower number of edges than expected by chance

$Q > 0.3 - 0.7$  means significant community structure

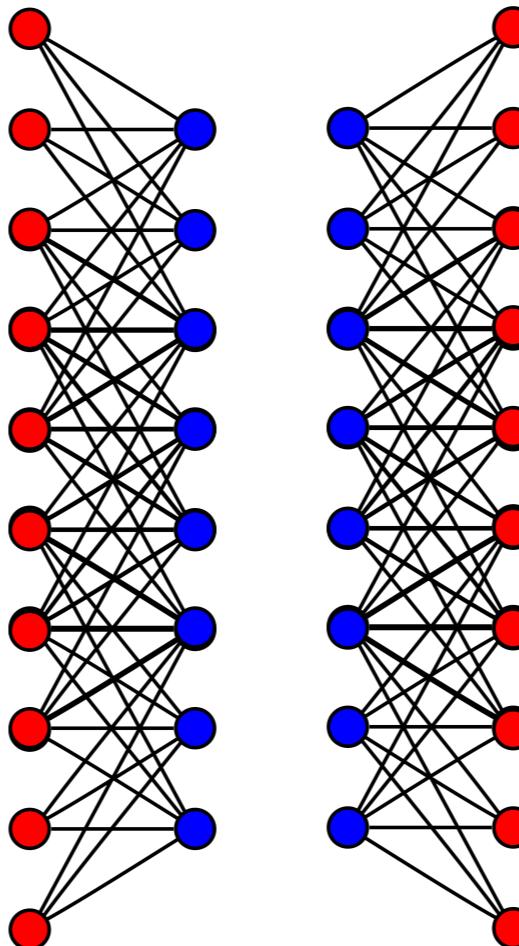
# Modularity

---

**Modularity** is different than **clustering coefficient**:

Take the following graph:

high Q but low connectivity (C)



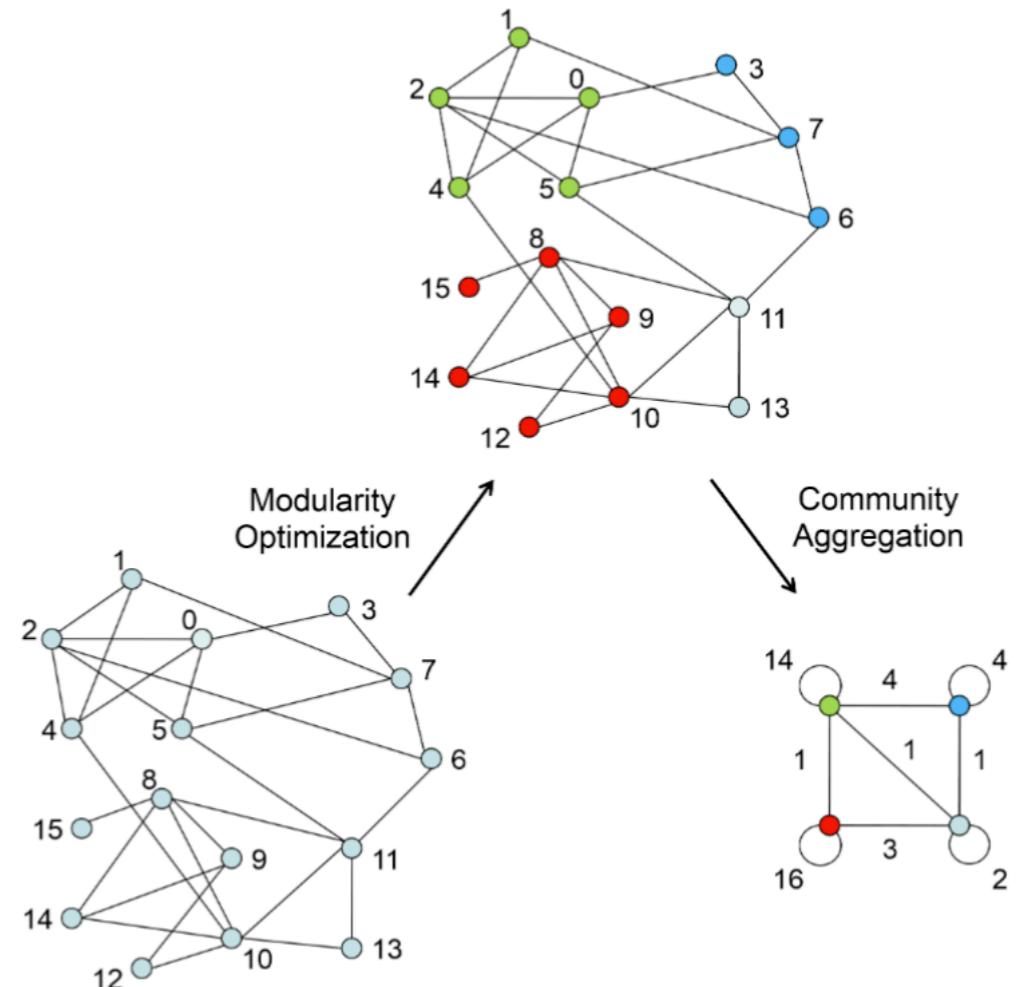
# Module detection: Louvain algorithm

## Phase 1: greedy modularity optimisation

1. Start with 1n/community
2. Compute  $Q$  by moving  $i$  to the community of  $j$
3. If  $\Delta Q > 1$ , node is placed in community
4. Repeat 1-3 until no improvement is found. Ties solved arbitrarily

## Phase 2: coarse grained community aggregation

5. Link nodes in a community into single node.
6. Self loops show intra-community associations
7. Inter-community weights kept
8. Repeat phase 1 on new network



# Community characterisation

---

Clustering coefficient and degree distribution

Enrichment analysis

**What do the nodes in each community have in common that could explain why they are in the same community?**

# Enrichment analysis

## MSigDB



GSEA  
Gene Set Enrichment Analysis

GSEA Home Downloads Molecular Signatures Database Documentation Contact

### Overview

**Gene Set Enrichment Analysis (GSEA)** is a computational method that determines whether *a priori* defined set of genes shows statistically significant, concordant differences between two biological states (e.g. phenotypes).

From this web site, you can:

- ▶ [Download](#) the GSEA software and additional resources to analyze, annotate and interpret enrichment results.
- ▶ [Explore the Molecular Signatures Database \(MSigDB\)](#), a collection of annotated gene sets for use with GSEA software.
- ▶ [View documentation](#) describing GSEA and MSigDB.

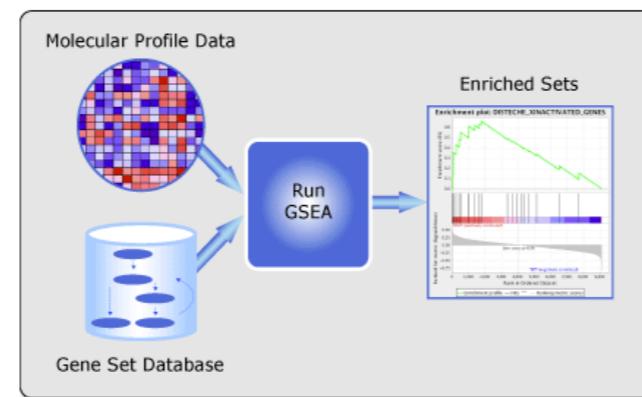
### What's New

20-Aug-2019: MSigDB 7.0 released. This is a major release that includes a complete overhaul of gene symbol annotations, Reactome and GO gene sets, and corrections to miscellaneous errors. See the [release notes](#) for more information.

20-Aug-2019: GSEA 4.0.0 released. This release includes support for MSigDB 7.0, plus major internal updates for Java 11 support and performance improvements. See the [release notes](#) for more information.

16-Jul-2018: MSigDB 6.2 released. This is a minor release that includes updates to gene set annotations, corrections to miscellaneous errors, and a handful of new gene sets. See the [release notes](#) for more information.

[Follow @GSEA\\_MSigDB](#)



### License Terms

GSEA and MSigDB are available for use under [these license terms](#).

Please [register](#) to download the GSEA software, access our web tools, and view the MSigDB gene sets. After registering, you can log in at any time using your email address. Registration is free. Its only purpose is to help us track usage for reports to our funding agencies.

### Contributors

GSEA and MSigDB are maintained by the [GSEA team](#). Our thanks to our many contributors. Funded by: National Cancer Institute, National Institutes of Health, National Institute of General Medical Sciences.



### Citing GSEA

To cite your use of the GSEA software, please reference Subramanian, Tamayo, et al. (2005, PNAS 102, 15545-15550) and Mootha, Lindgren, et al. (2003, Nat Genet 34, 267-273).

## Enrichr



Login | Register

21,153,478 lists analyzed

307,486 terms

154 libraries

Analyze What's New? Libraries Find a Gene About Help

| Gene-set Library                               | Terms | Gene Coverage | Genes per Term   |
|--|-------|---------------|--|
| Genes_Associated_with_NIH_Grants               | 32876 | 15886         | 9.0       |
| Cancer_Cell_Line_Encyclopedia                  | 967   | 15797         | 176.0     |
| Achilles_fitness_decrease                      | 216   | 4271          | 128.0     |
| Achilles_fitness_increase                      | 216   | 4320          | 129.0     |
| Aging_Perturbations_from_GEO_down              | 286   | 16129         | 292.0     |
| Aging_Perturbations_from_GEO_up                | 286   | 15309         | 308.0     |
| Allen_Brain_Atlas_down                         | 2192  | 13877         | 304.0     |
| Allen_Brain_Atlas_up                           | 2192  | 13121         | 305.0     |
| ARCHS4_Cell-lines                              | 125   | 23601         | 2395.0    |
| ARCHS4_IDG_Coexp                               | 352   | 20883         | 299.0     |
| ARCHS4_Kinases_Coexp                           | 498   | 19612         | 299.0     |
| ARCHS4_TFs_Coexp                               | 1724  | 25983         | 299.0     |
| ARCHS4_Tissues                                 | 108   | 21809         | 2316.0    |
| BioCarta_2013                                  | 249   | 1295          | 18.0      |
| BioCarta_2015                                  | 239   | 1678          | 21.0     |
| BioCarta_2016                                  | 237   | 1348          | 19.0    |
| BioPlex_2017                                   | 3915  | 10271         | 22.0    |
| ChEA_2013                                      | 353   | 47172         | 1370.0  |
| ChEA_2015                                      | 395   | 48230         | 1429.0  |
| ChEA_2016                                      | 645   | 49238         | 1550.0  |
| Chromosome_Location                            | 386   | 32740         | 85.0    |
| Chromosome_Location_hg19                       | 36    | 27360         | 802.0   |
| CORUM  | 1658  | 2741          | 5.0     |
| Data_Acquisition_Method_Most_Popular_Genes     | 12    | 1073          | 100.0   |
| dbGaP  | 345   | 5613          | 36.0    |
| DepMap_WG_CRISPR_Screens_Broad_CellLines_2019  | 558   | 7744          | 363.0   |
| DepMap_WG_CRISPR_Screens_Sanger_CellLines_2019 | 325   | 6204          | 387.0   |
| Disease_Perturbations_from_GEO_down            | 839   | 23939         | 293.0   |
| Disease_Perturbations_from_GEO_up              | 839   | 23561         | 307.0   |
| Disease_Signatures_from_GEO_down_2014          | 142   | 15406         | 300.0   |

# Enrichment analysis

---

Important databases with gene-sets:

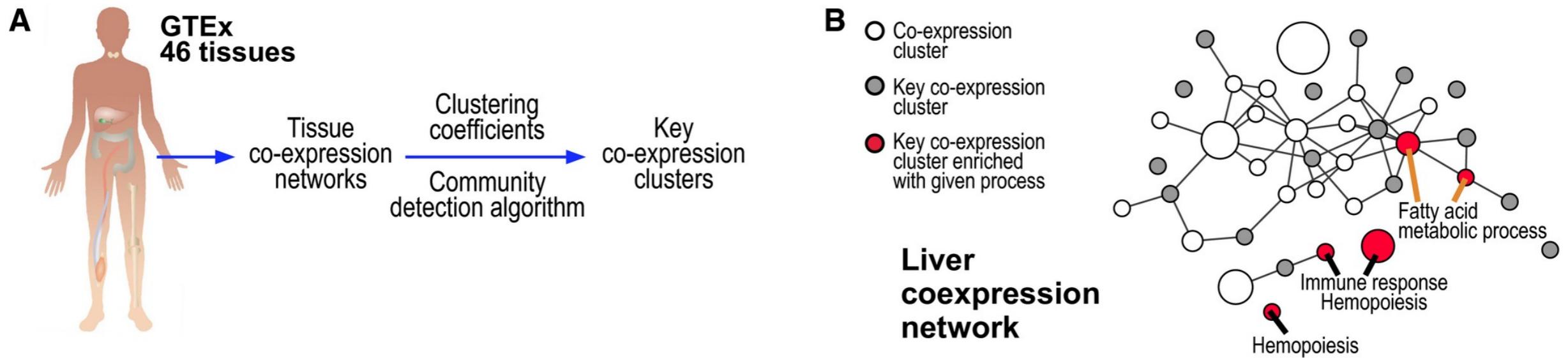
- [MSigDB](#) (gene)
- [Enrichr](#) (gene)
- [KEGG](#) (metabolite, gene)
- [DIANA](#) (miRNA)
- [MetaboAnalyst](#) (metabolite)
- [DAVID](#) (web)
- [Reactome](#) (web)

Creating custom sets and joint sets

Mapping your data to common IDs

- Easy for genes and proteins: use [DAVID](#), [Biomart](#), or [MyGene](#) (in [Python](#) or [R](#))
- Harder for other feature types

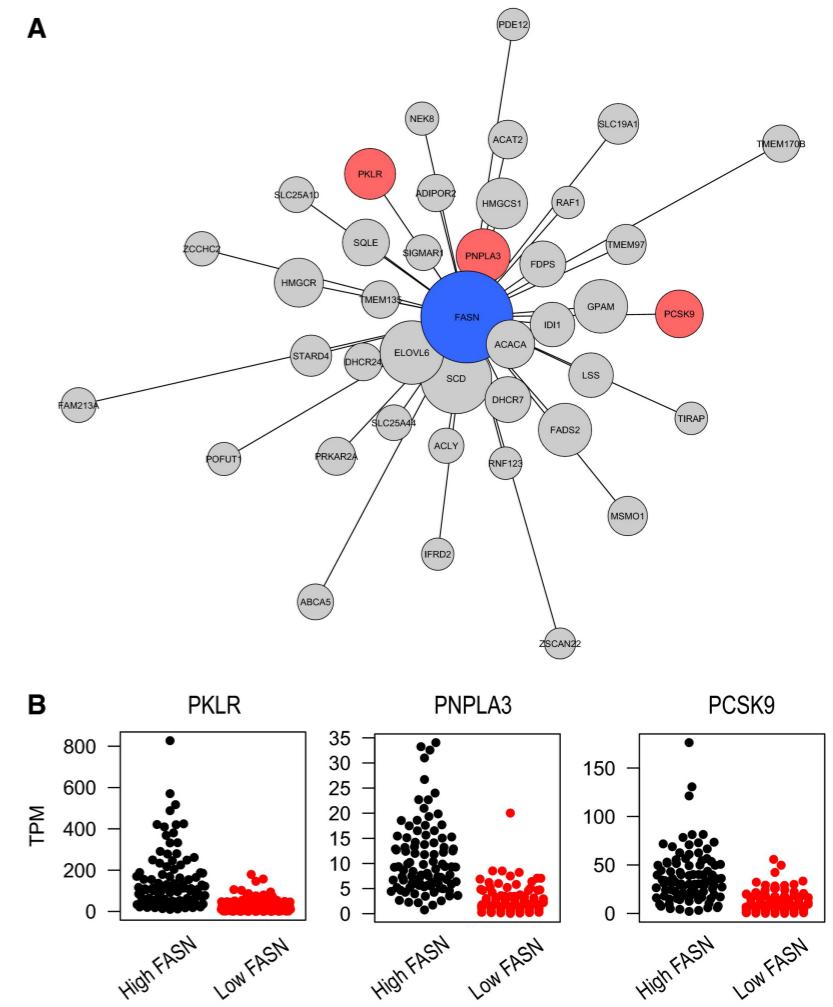
# Outcome



GO BPs from MSigDB

Network analyses identifies key stratifying genes

40-NN of FASN



# Workshop objective: Build and analyze a gene- metabolite association network

1. Introduction
2. Terminology
3. Network construction
4. Key properties
5. Community analysis
- 6. Workshop**

# Additional reading

---

- [Network Science](#) - A fascinating textbook on graph theory and network analysis.
- [Communication dynamics in complex brain networks](#) - Interesting discussion about whether and how network topology may be applied to study the brain networks.
- [A Systematic Evaluation of Methods for Tailoring Genome-Scale Metabolic Models](#) - General review and discussion on methods to use in genome-scale metabolic models.
- [Analysis of Biological Networks](#) - General introduction into biological networks, network notation, and analysis, including graph theory.
- [Multi-omics approaches to disease](#) - Introduction to how integrative approaches may be applied in disease

Additional references displayed as hyperlinks in each slide.