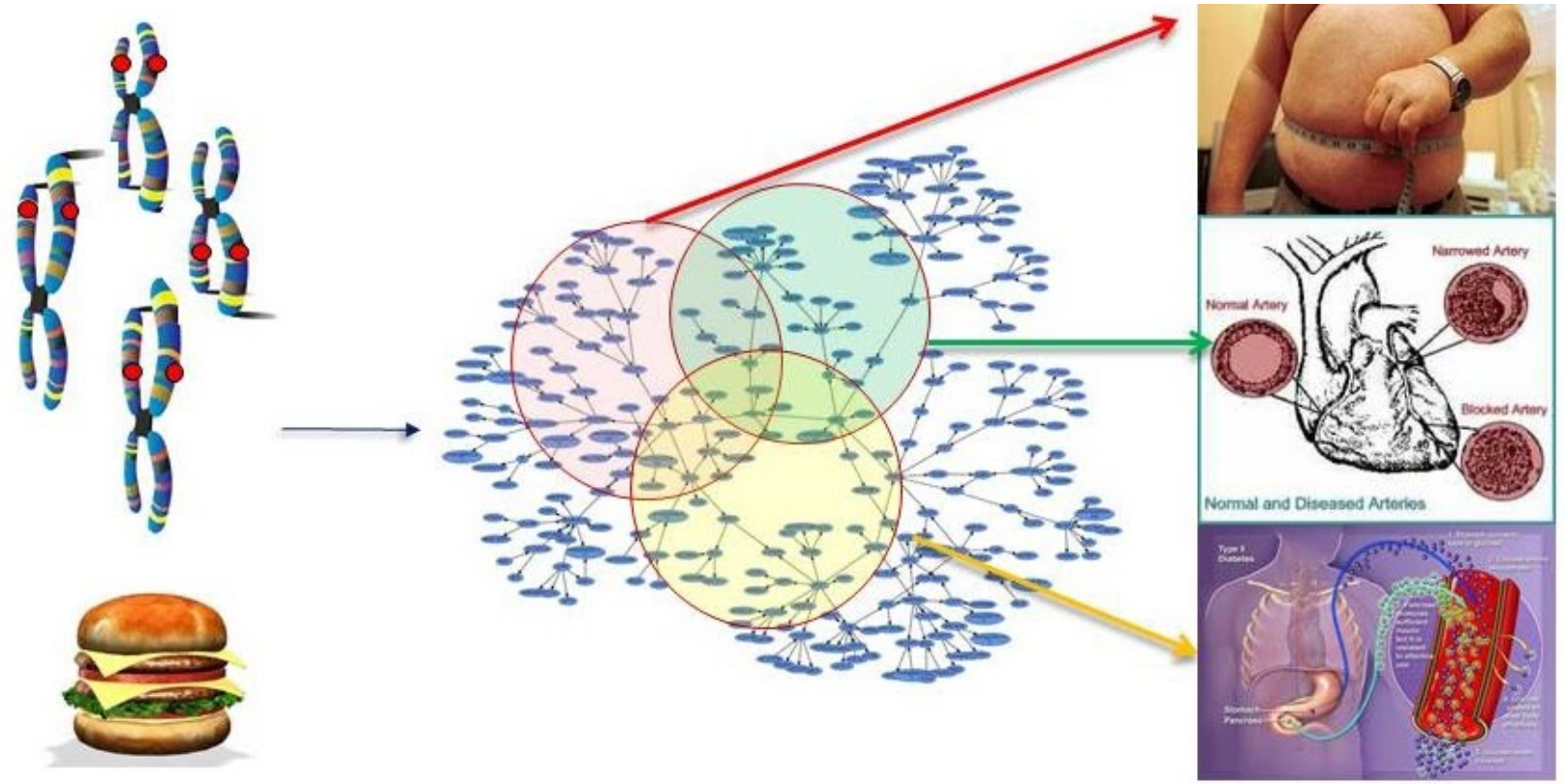
Biological Network Analysis and Gene-Gene Interaction Networks

Assist. Prof. Zeyneb Kurt
Department of Computer Engineering,
Yildiz Technical University,
Istanbul, Turkey
28/11/2019

Content

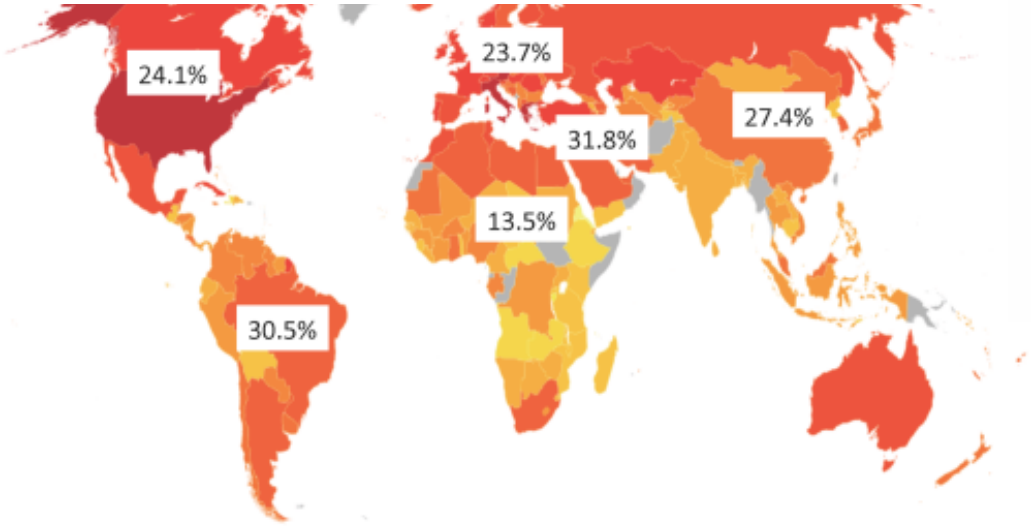
- Rationale of "**biological network analysis**"
- Inferring gene-gene interaction networks
- Co-regulation (co-expression) networks
- Weighted correlation network analysis of genes (WGCNA)
- Bayesian networks
- Real life examples of gene networks usage

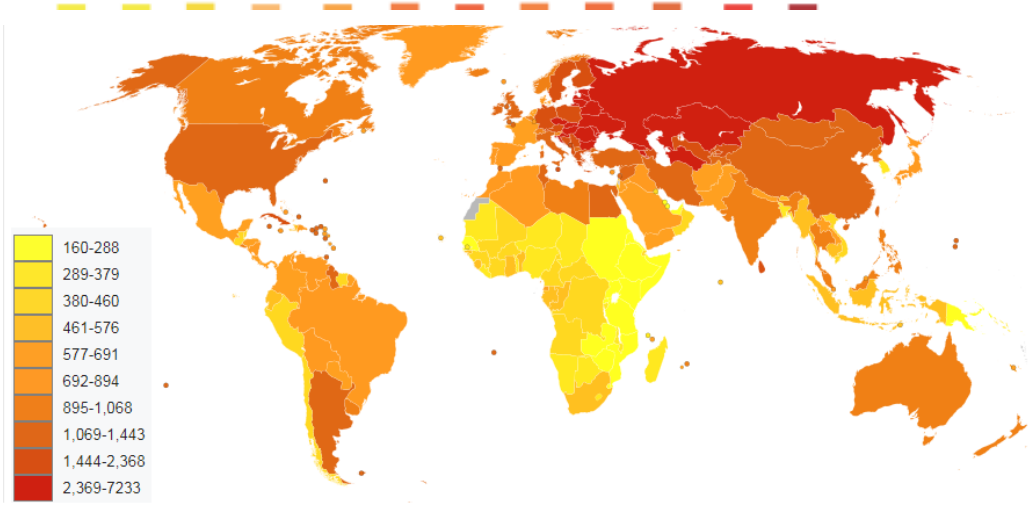
Hypothesis



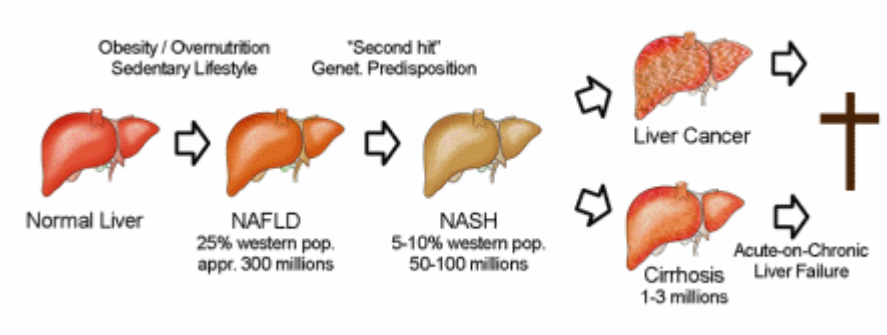
DNA variations/ environmental → Drive changes in network states → Variations in disease phenotypes and drug

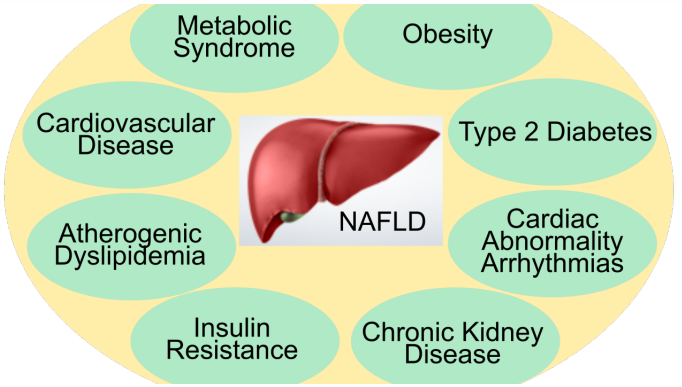
What can gene networks provide us?
To study common human diseases such as NAFLD and CAD..

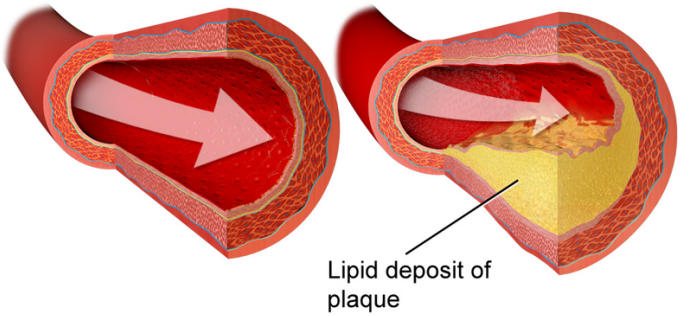


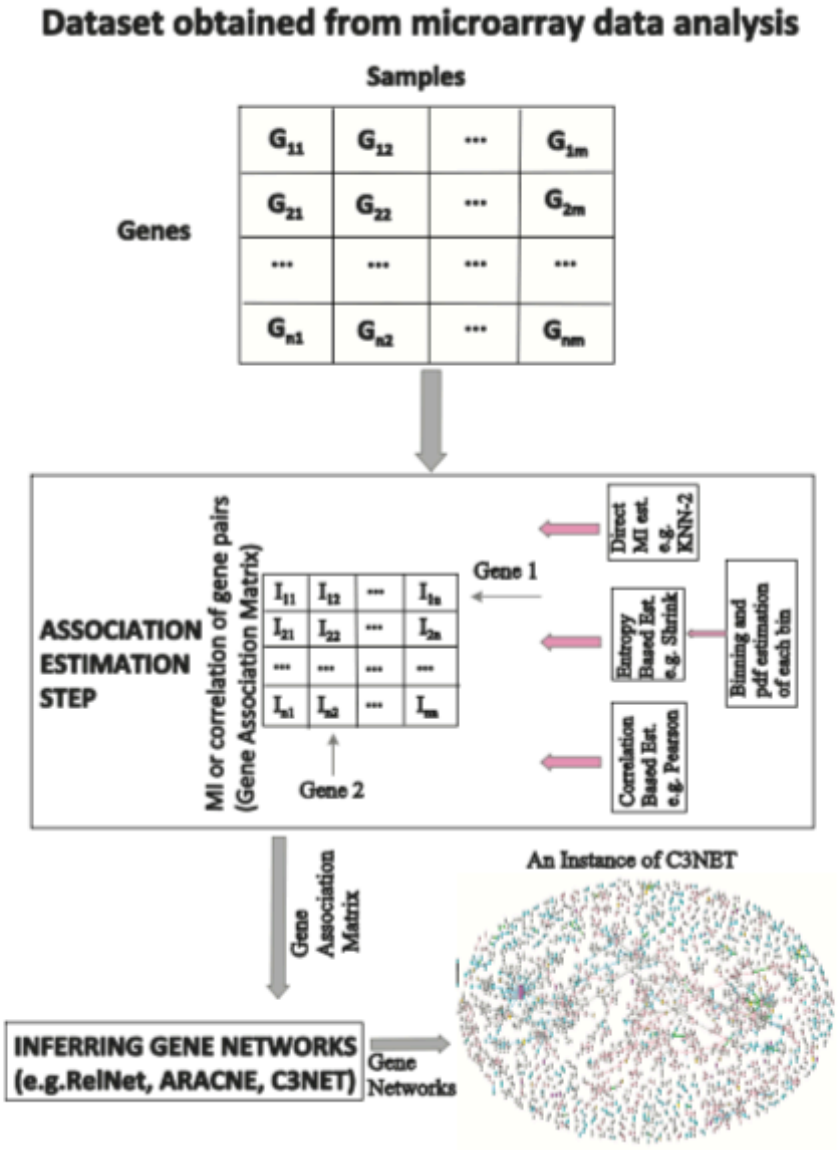


Deaths from CAD in 2012 per million persons.
Statistics from WHO, grouped by deciles.








Dataset obtained from microarray data analysis
Samples
Genes
ASSOCIATION ESTIMATION STEP
MI or correlation of gene pairs (Gene Association Matrix)
Gene 1
Gene 2
Direct MI est. e.g. KNN-2
Entropy Based Est. e.g. Shrink
Binning and pdf estimation of each bin
Correlation Based Est. e.g. Pearson
Gene Association Matrix
INFERRING GENE NETWORKS (e.g.RelNet, ARACNE, C3NET)
Gene Networks
An Instance of C3NET

Weighted correlation network analysis (WGCNA)*

***Citation for WGCNA summary:**

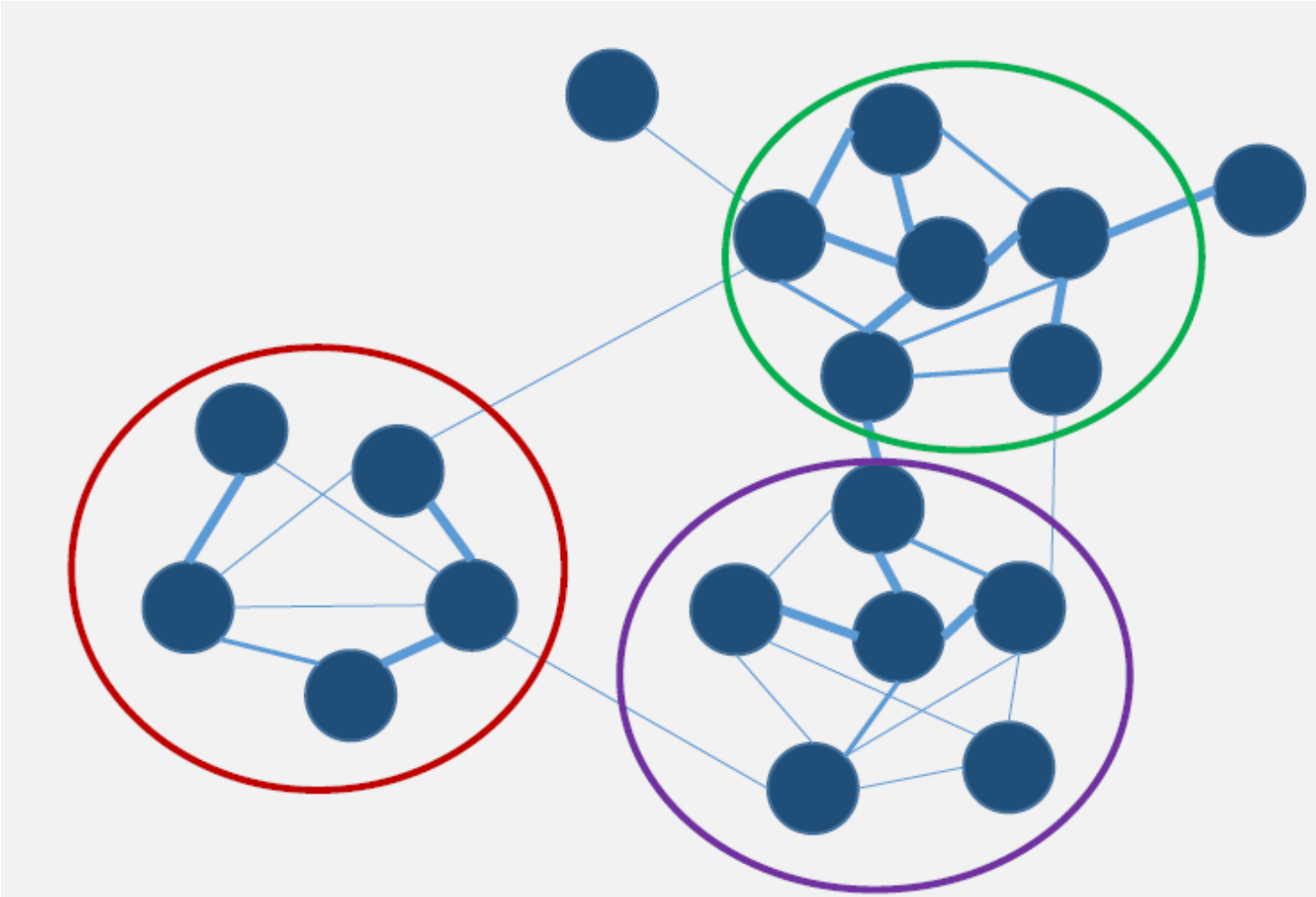
SIB course, Nov. 15-17, 2016, “Introduction to Biological Network Analysis” by Leonore Wigger and with Frédéric Burde tand Mark Ibberson

Overview of WGCNA

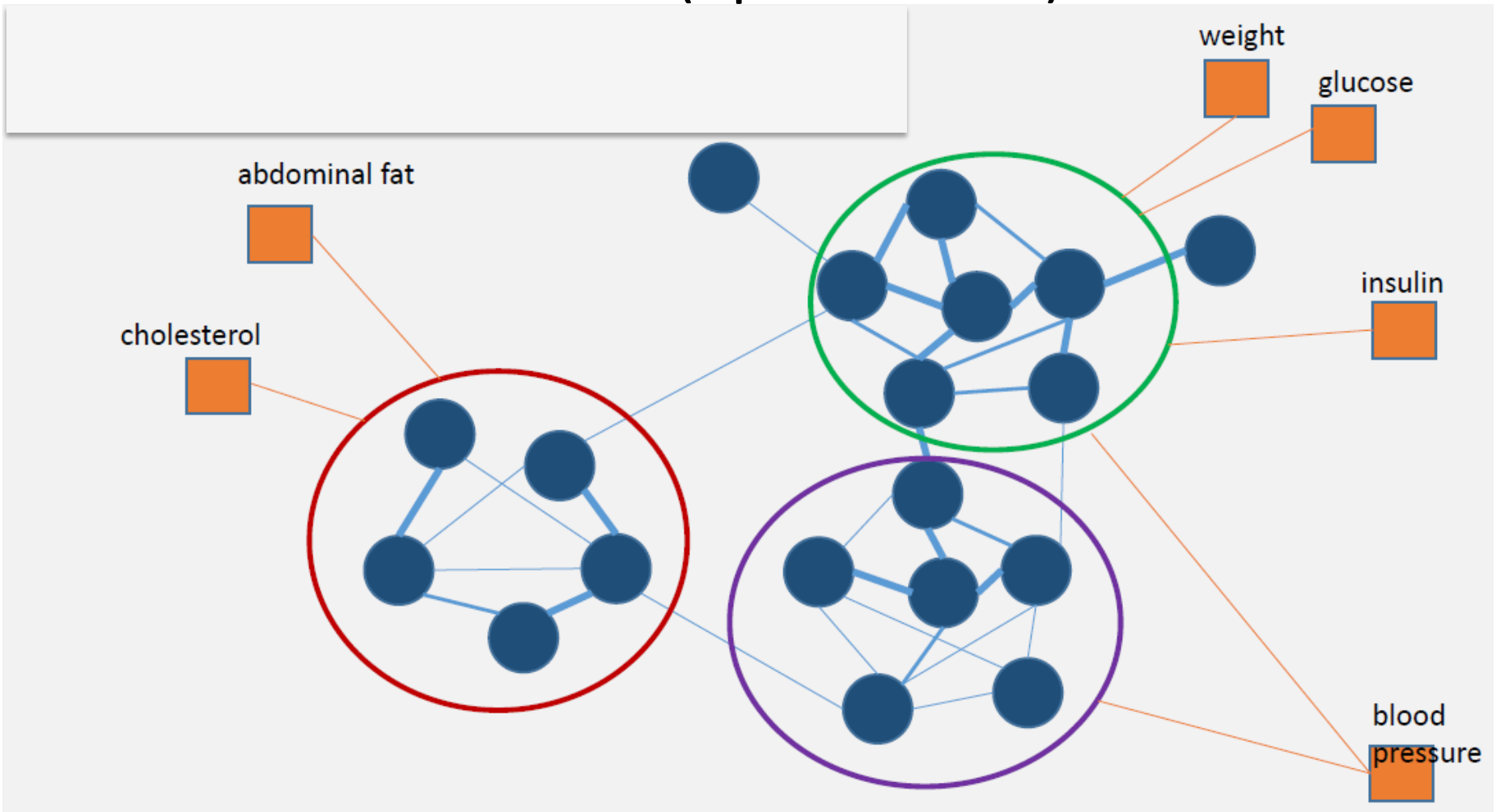
- **Theory 1:** Weighted correlation network, split into modules
- **Theory 2:** Identify modules and genes of interest
- **Input data:**
  - Gene expression data (microarray or RNA-Seq)

  Recommendation: at least 20 individuals
  - Clinical/phenotypical traits from the same individuals (optional) e.g. weight, insulin level, glucose level

Aims of WGCNA:
Inferring a gene-gene similarity network

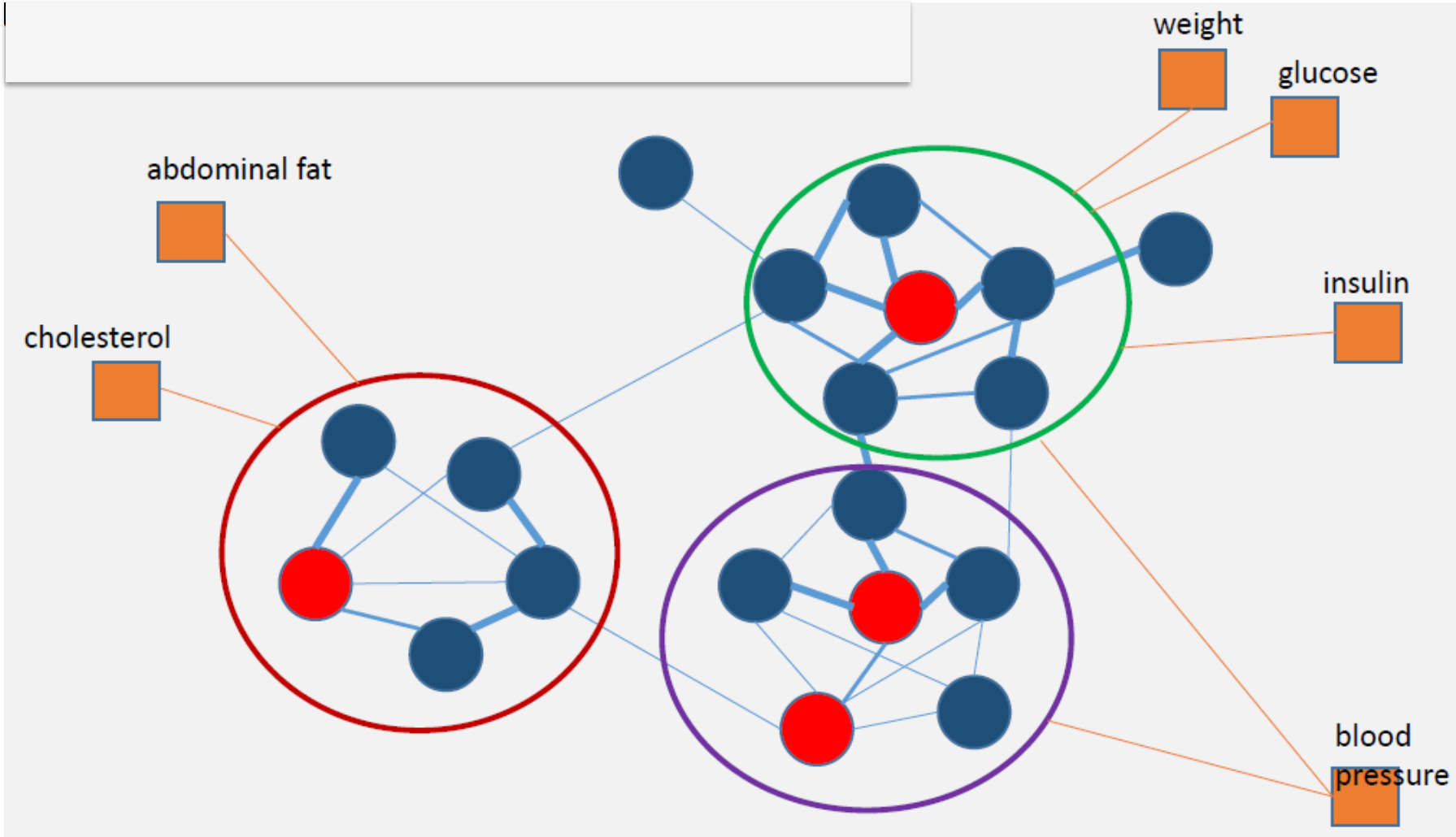


Aims: Correlate phenotypic traits to gene modules* (optional aim)



*** "Modules" found in WGCNA:** Groups or clusters of co-expressed genes with similar expression profiles over a large group of individuals

Aims: Identify “key driver” genes in modules

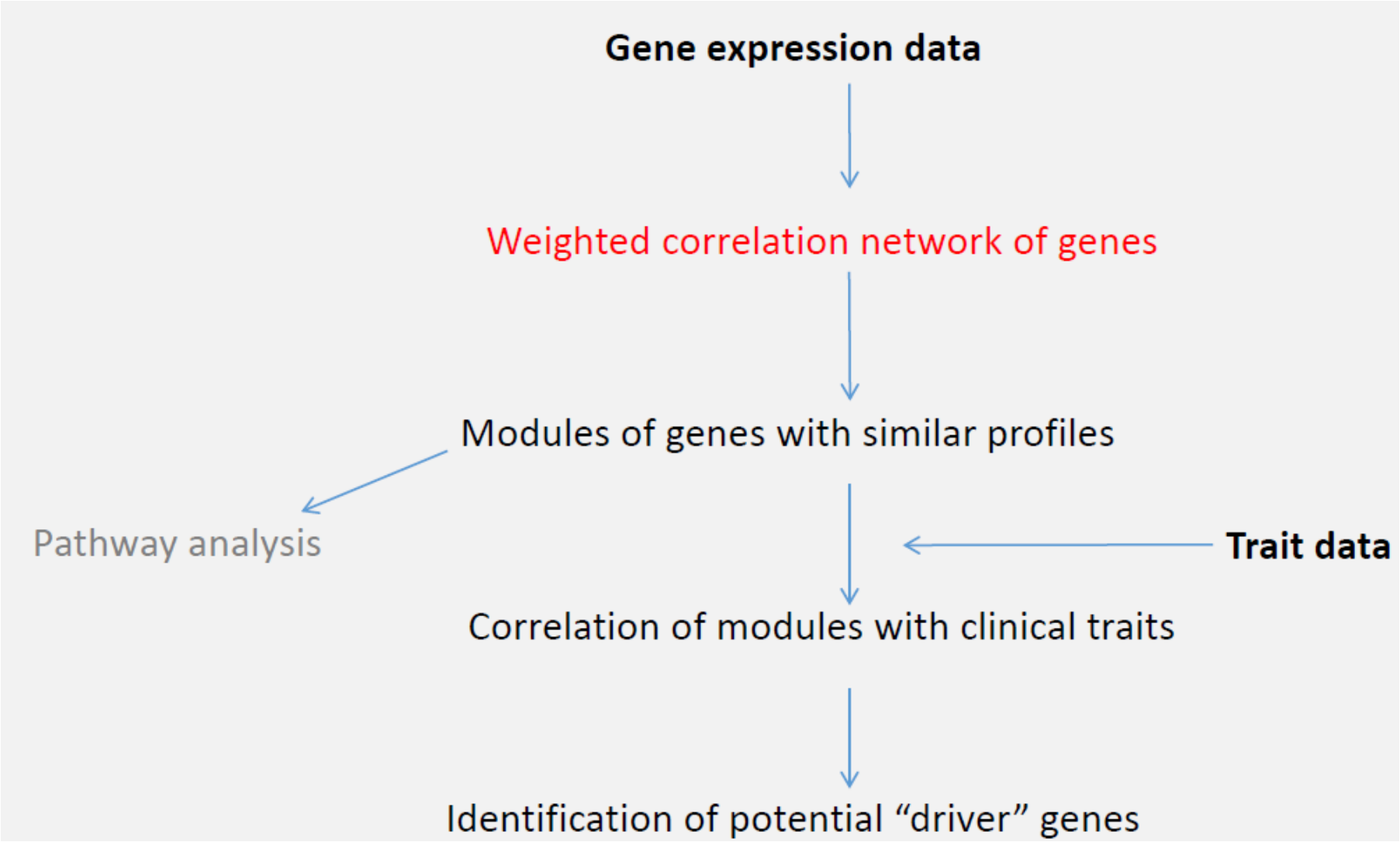


Rationale

- Genes with **similar expression patterns are of interest because they may be**
  - tightly co-regulated
  - functionally related
  - members of the same pathway

- WGCNA encourages hypotheses about genes based on their close network neighbors.

Workflow


Gene expression data
Weighted correlation network of genes
Modules of genes with similar profiles
Pathway analysis
Trait data
Correlation of modules with clinical traits
Identification of potential “driver” genes

But first: preprocess the gene expression data

- **Remove outlier samples e.g.** by creating dendrogram for samples (take transpose of the expression matrix for this) and identifying the outliers

- **Remove the lowly expressed genes** by eliminating the genes that do not have expression levels > 0 at least for 50% of all samples

- **Normalize gene expression by log2(expr+1).** We need to transform expression values with logarithm base 2, since the correlation measures (e.g. Pearson coefficient) assume that the expression values of each gene are normally distributed (approximately) across the samples.

- **Extract expression table with the most variable probes/genes:** e.g. find the top ~15,000 highly variable genes based on the "coefficient of variation" (cv) measure.

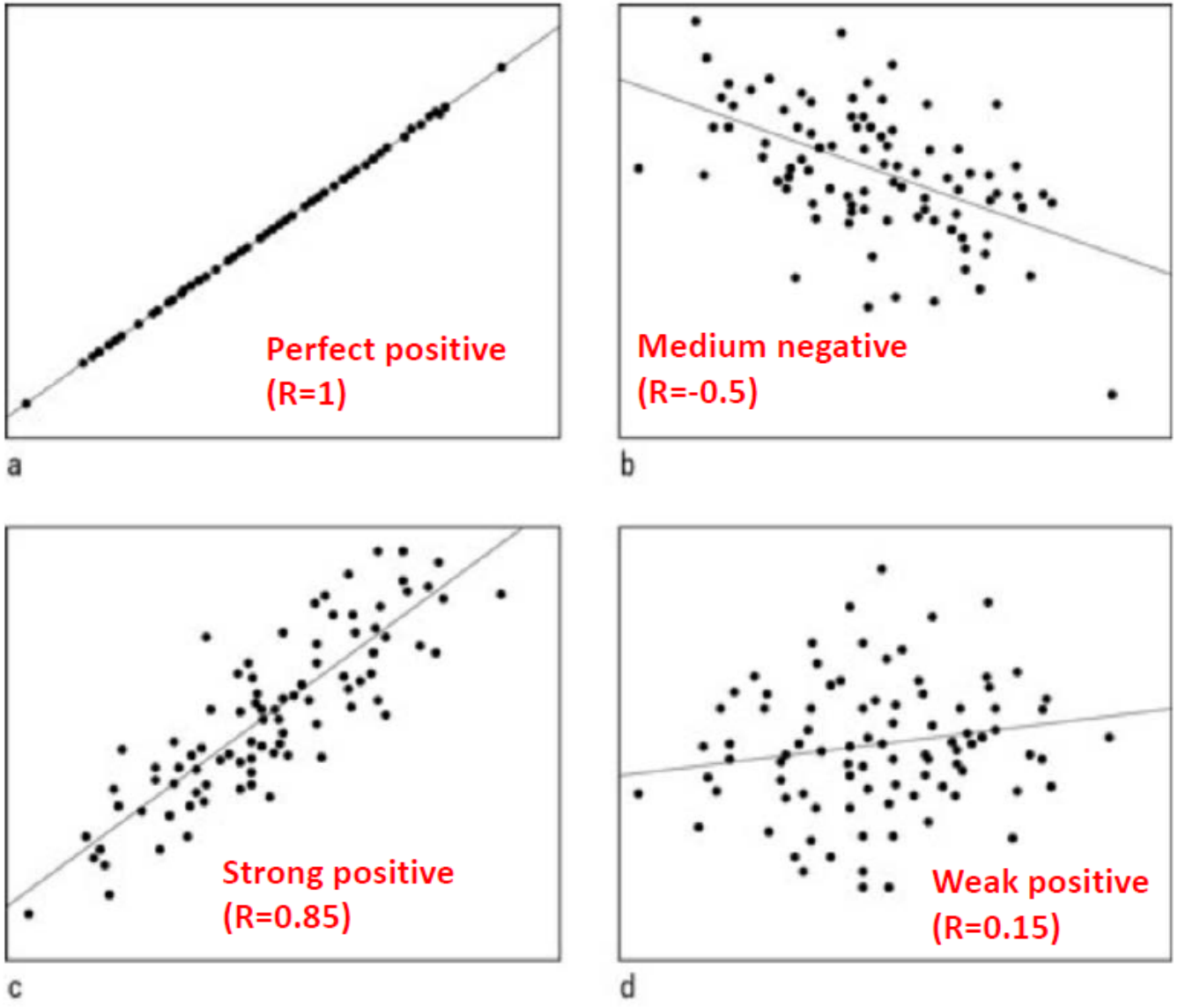
But first: preprocess the gene expression data

- Remove the lowly expressed genes:
  - Min_Exp_level=0
  - Gene_present<-matrix()
  - for (i in 1:nrow(Expr_Mat))
    Gene_present[i]
    = (sum(Expr_Mat[i,] > Min_Exp_level) /ncol(Expr_Mat))>=0.5
  - Expr_Mat <- Expr_Mat[Gene_present,]
- Transform expression values with logarithm base 2 (normalization):
  - Expr_Mat = log2(Expr_Mat + 1)
- Extract expression table with the most variable probes/genes:
  - cv=NULL
  - a=Expr_Mat
  - for (m in 1:nrow(a)){
    cv_individual = cv(a[m,]) ; cv = rbind(cv,cv_individual)
    }
  - a_sort = with(data.frame(a), data.frame(a)[order(-cv),])
  - # Keep the top e.g. 90 percentile, which provides us to have ~15,000 genes:
  - perc <- 0.90
  - a_keep <- a_sort[1:(perc*nrow(a_sort)),]
  - Expr_Mat <- a_keep

Construct weighted correlation network

- **Correlation:** A statistical measure for the extent to which two variables fluctuate together.
- **Positive correlation:** variables increase/decrease together
- **Negative correlation:** variables increase/decrease in opposing direction
- **Caution:** There is no causality here!!

Correlation examples (Pearson correlation, R^2)



Scatterplots with correlations of a) +1.00; b) –0.50; c) +0.85; and d) +0.15.

http://www.dummies.com/how-to/content/how-to-interpret-a-correlation-coefficient-r.html

Correlation measures implemented in WGCNA

- Pearson
- Spearman
- Kendall's tau
- Biweight midcorrelation (bicor)

Choosing a correlation method

$$a_{i,j} = |cor(i, j)|^{\beta}$$

- **Fastest, but sensitive to outliers:** Pearson correlation, cor(x), "standard" measure of linear correlation
- **Less sensitive to outliers but much slower:** Biweight mid-correlation, bicor(x), robust, recommended by the authors for most situations [needs modification for correlations involving binary/ categorical variables]
- Spearman correlation, cor(x, method="spearman"), rank-based, works even if relationship is not linear (but the relationship expected to be monotonous), less sensitive to gene expression differences [can be used as-is for correlations involving binary/ categorical variables]

- **Default correlation method in WGCNA**: cor(Pearson).
- **Caveat:** use it only if there are no outliers, or for exercises/tutorials.

Adjacency matrix calculation

- Compute a correlation raised to a power between every pair of genes (i,j): $a_{i,j} = |cor(i,j)|^{\beta}$
- Effect of raising correlation to a power:
  - Amplifies disparity between strong and weak correlations
  - Example: Power term $\beta = 4$

Correlations		Adjacencies
$cor(i,j) = 0.8$	→	$\|0.8\|^4 = 0.4096$
$cor(k,l) = 0.2$	→	$\|0.2\|^4 = 0.0016$
0.8/0.2: 4-fold difference	→	0.4096/0.0016: 256-fold difference

Strong corr.

Weak corr.

Adjacency matrix calculation – cont'd

Adjacencies

Compute a correlation raised to a power between every pair of genes (i, j)

$$a_{i,j} = |cor(i, j)|^{\beta}$$

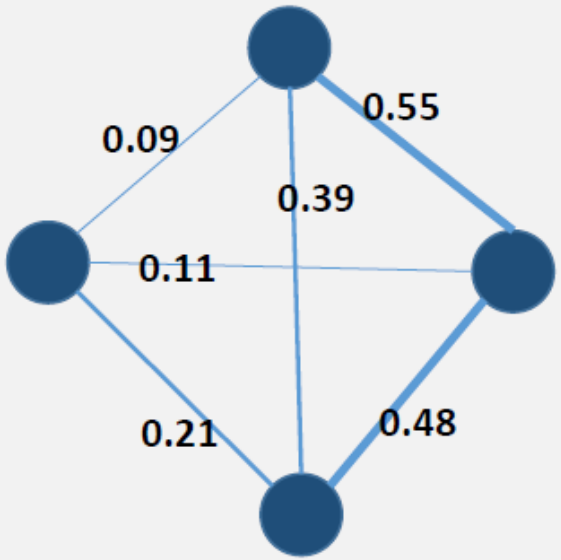
Adjacency matrix of 4 genes

$a_{i,j}$	gene1	gene2	gene3	gene4
gene1	1	0.55	0.39	0.09
gene2	0.55	1	0.48	0.11
gene3	0.39	0.48	1	0.21
gene4	0.09	0.11	0.21	1

Network

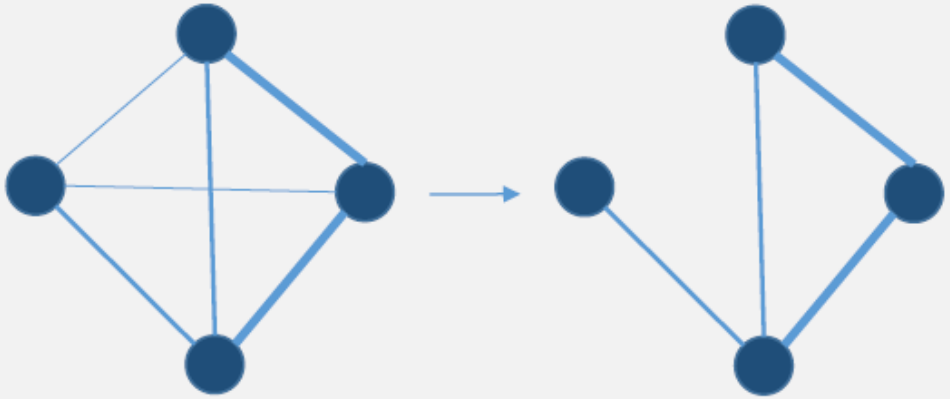
Construct a fully connected network;
Genes as nodes, $a_{i,j}$ as edge weights.

high correlation – strong connection
low correlation – weak connection

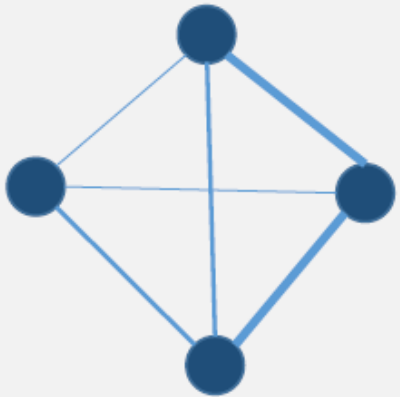


Adjacency matrix calculation – cont'd

For visualizations, set a threshold on edge weight and **remove the weakest links.**



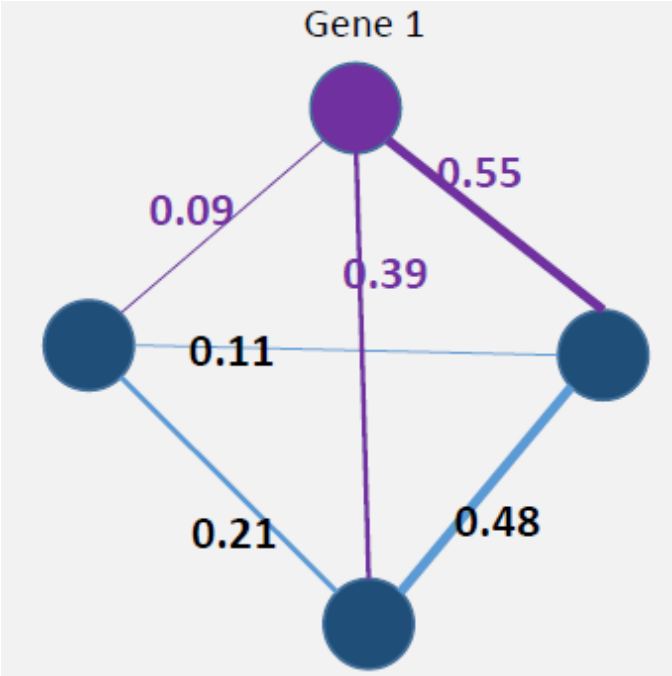
In most computations, work with all edges of the **fully connected network**.



Connectivity (degree) in a weighted network

- **Connectivity of a gene**: Sum of the weights of all edges connecting to this gene
- Example: Connectivity of gene 1 is:

  0.55 + 0.39 + 0.09 = 1.03

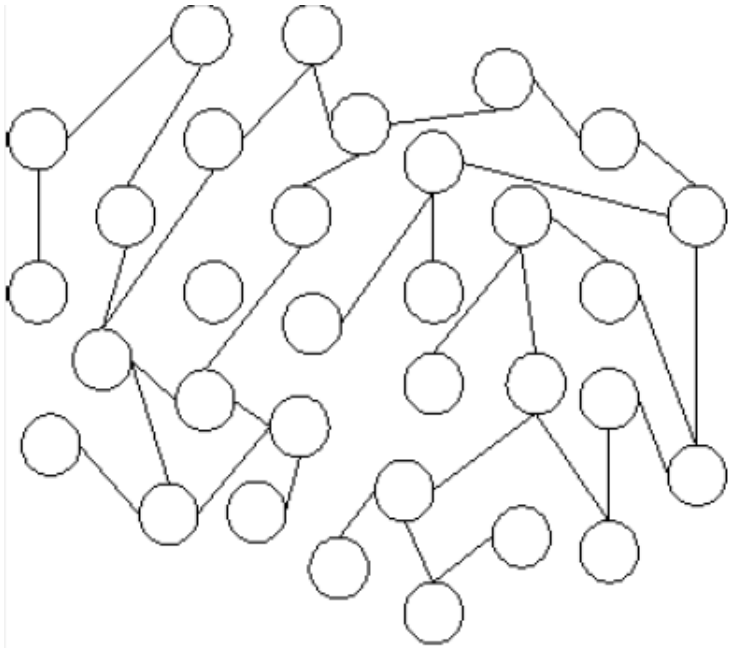


Picking a power term

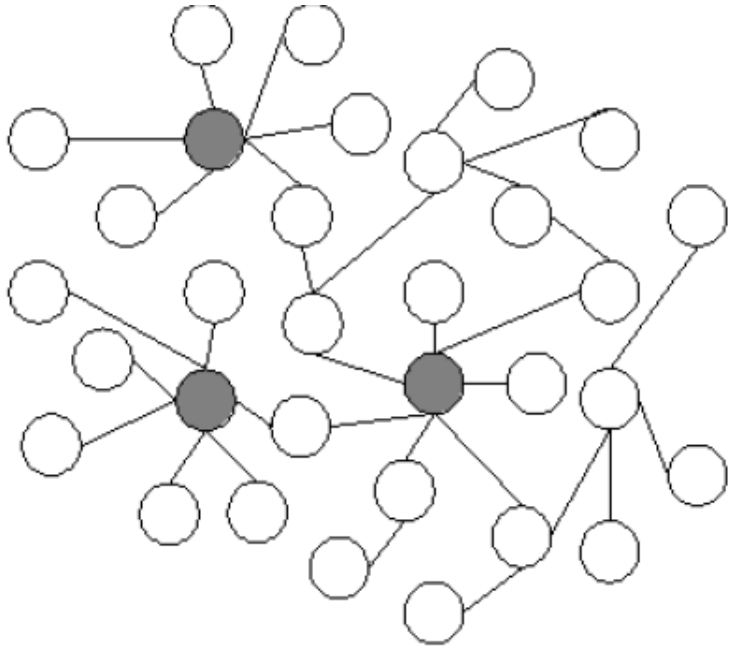
$$a_{i,j} = |cor(i, j)|^{\beta}$$

- **Selection criterion:** Pick lowest possible β that leads to an approximately *scale-free network topology:*
  - few nodes with many connections (”hubs”)
  - many nodes with few connections

- Degree distribution follows a power law:
  - the probability for a node of having *k* connections is k^{γ}



(a) Random network



(b) Scale-free network

Source: Carlos Castillo: Effective Web Crawling, PhD Thesis, University of Chile, 2004

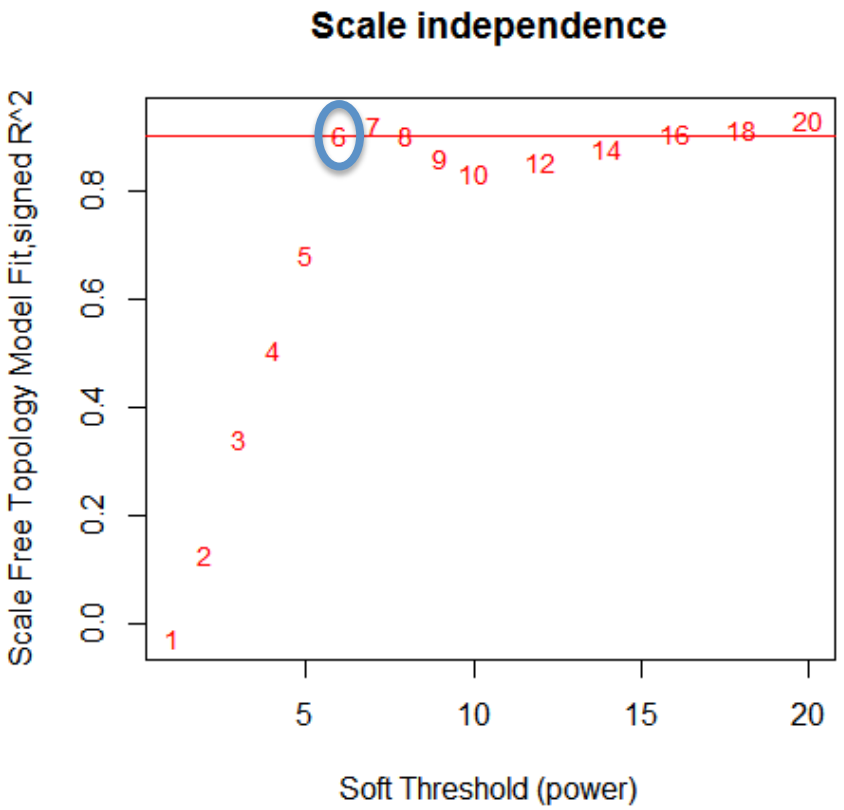
Why scale-free network topology?

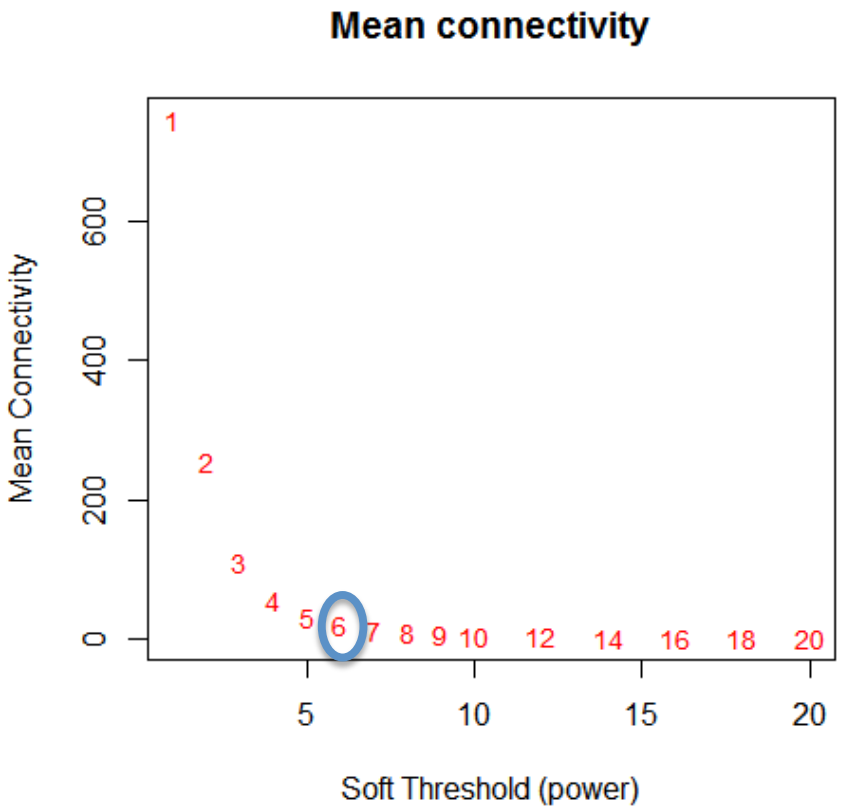
- Barábasi et al.* found many types of network in many domains to be approximately scale-free, including metabolic and protein interaction

- So, aim in WGCNA is: Building a biologically “realistic” network.

* Barabási, Albert-László; Bonabeau, Eric (May 2003). "Scale-Free Networks"(PDF). Scientific American. **288(5): 50–9.**

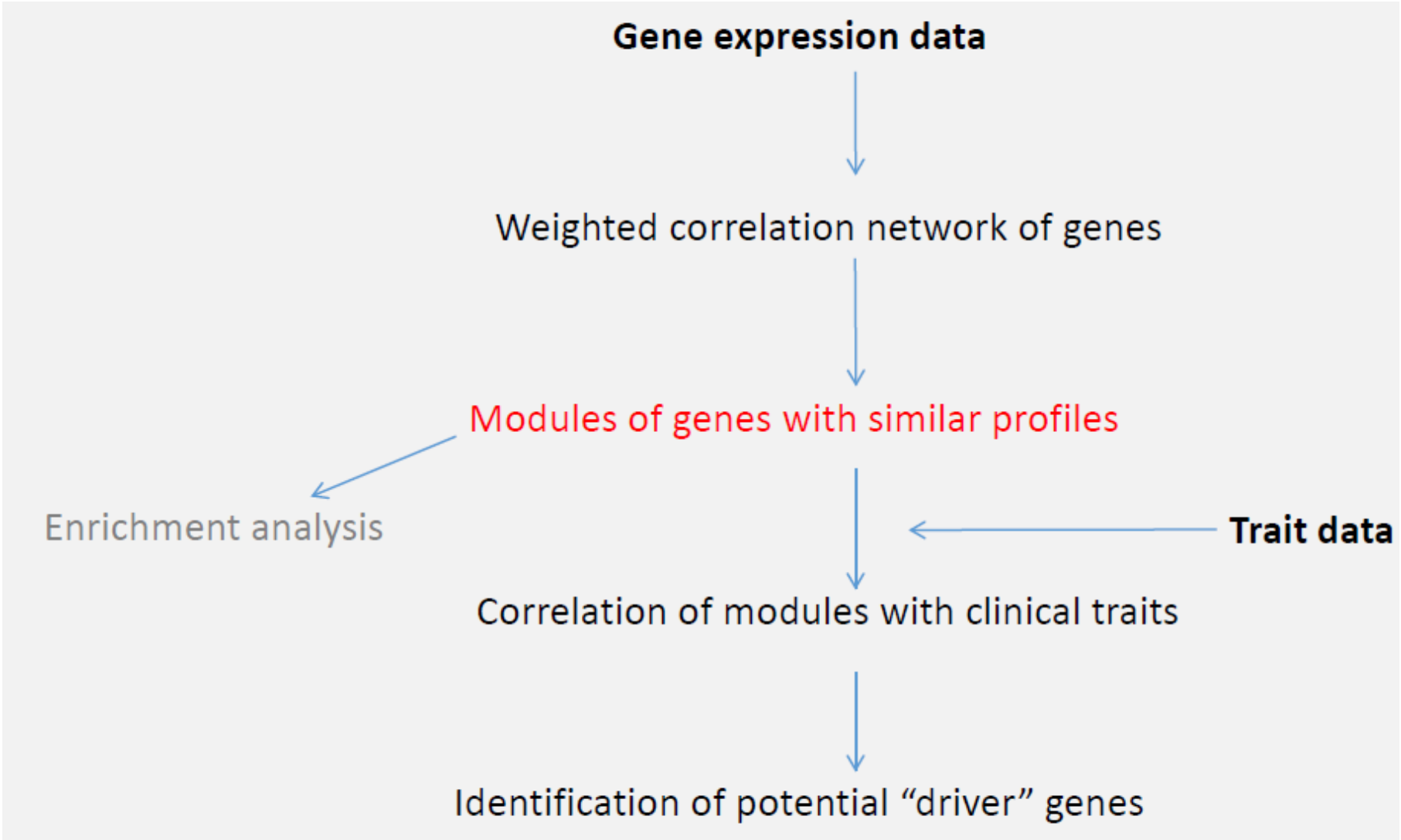
Pick a power term: Visual Aid in WGCNA





- **Left plot:** Choose power 6. Lowest possible power term where topology approximately fits a scale free network (on or above red horizontal line).
- **Right plot:** mean connectivity drops as power goes up. Must **not** drop too low

Next Step: Detect **modules** of co-expressed genes



4 steps to get from network to modules

1. Compute dissimilarity between genes: “topological overlap measure dissimilarity”

2. Perform hierarchical clustering of genes: obtain tree structure

3. Divide clustered genes into modules: cut tree branches

4. **Optional:** Merge very similar modules: use module “eigengenes”

Step 1: Compute dissimilarity between genes

- **Why we use Topological Overlap Measure (TOM)?**
  - TOM is a pairwise similarity measure between network nodes (genes)
  - TOM(i,j) is **high if genes i,j have many shared neighbors** because overlap of their network neighbors is large
- So, a high TOM(i,j) implies that genes have similar expression patterns

- **How to calculate TOM similarity between two nodes:**
  - 1. Count number of shared neighbors: "agreement" of the set of neighboring nodes
  - 2. Normalize to [0,1]

    TOM(i,j) = 0 means: no overlap of network neighbors

    TOM(i,j) = 1 means: identical set of network neighbors

$$TOM_{ij} = \frac{\sum_u a_{iu} a_{uj} + a_{ij}}{min(k_i, k_j) + 1 - a_{ij}}$$

$$DistTOM_{ij} = 1 - TOM_{ij}$$

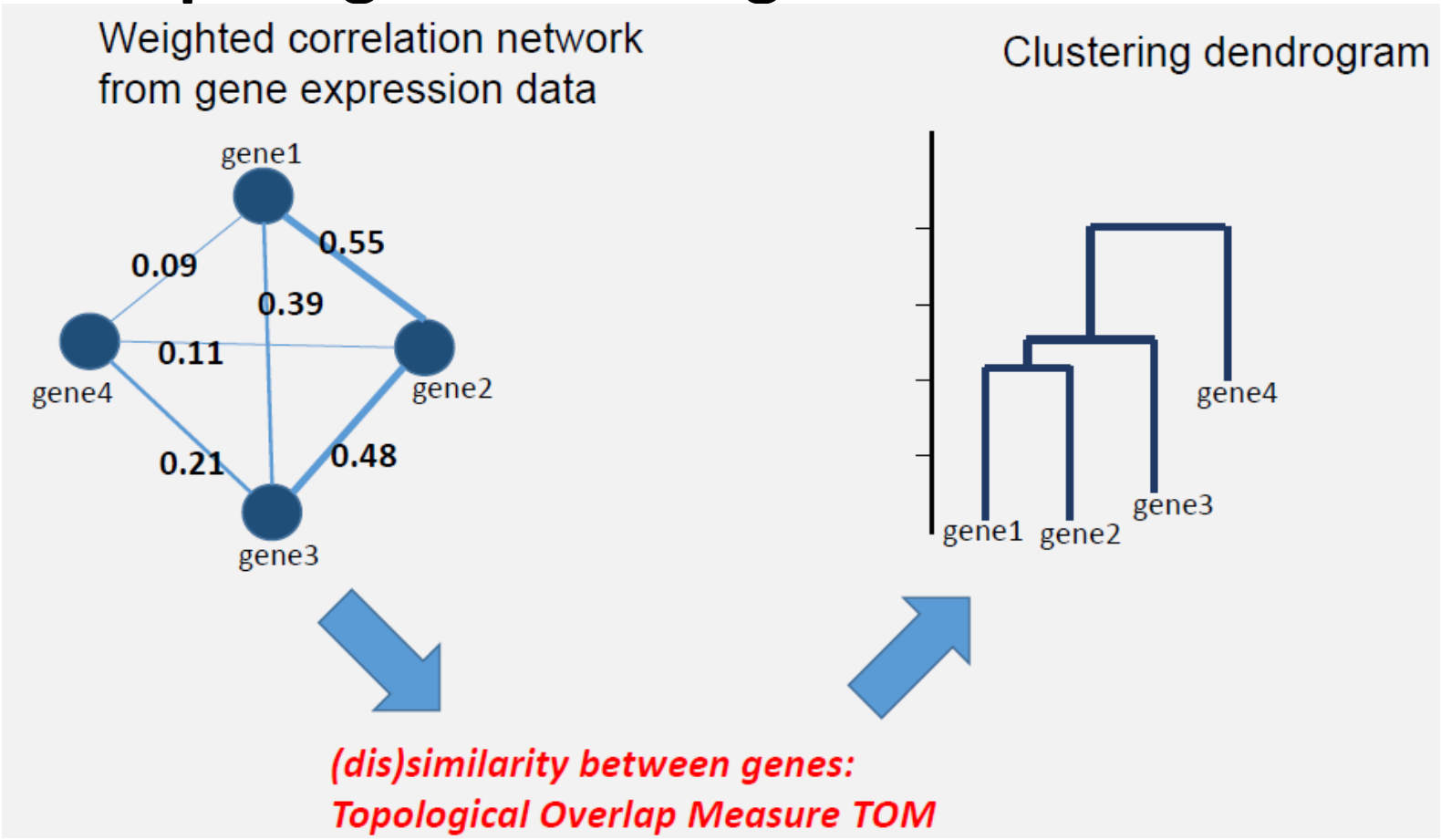
- ***NOTE that: Generalized to the case of weighted networks in Zhang and Horvath (2005), first WGCNA paper***
  - All nodes are neighbors; counting them is not informative.
  - Compute agreement of the set of neighboring nodes based on edge strengths.

- But, we need a **dissimilarity measure** for clustering!!

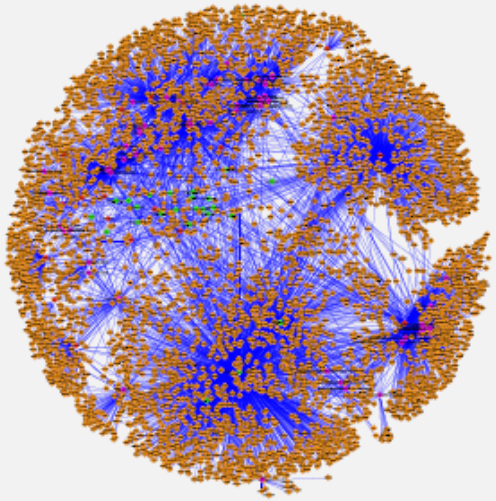
- TOM as a **similarity measure** can be transformed into a **dissimilarity measure:** **distTOM= 1-TOM.**

Step 2: Perform hierarchical clustering of genes

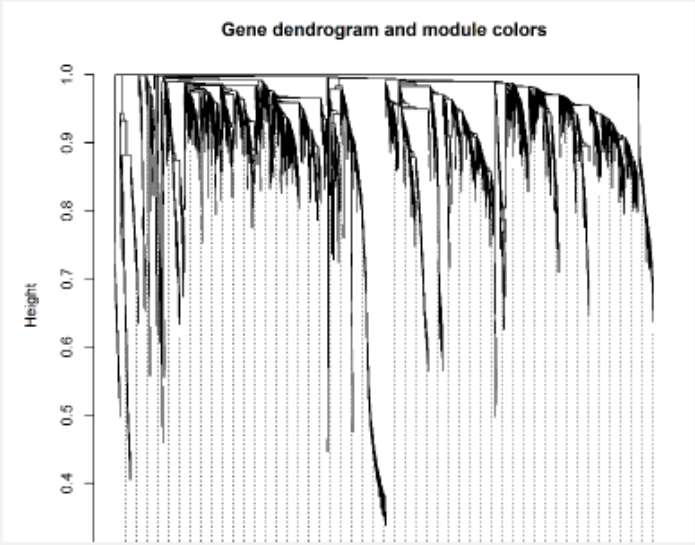
- **Compute gene dendrogram:**



Weighted correlation network
from gene expression data



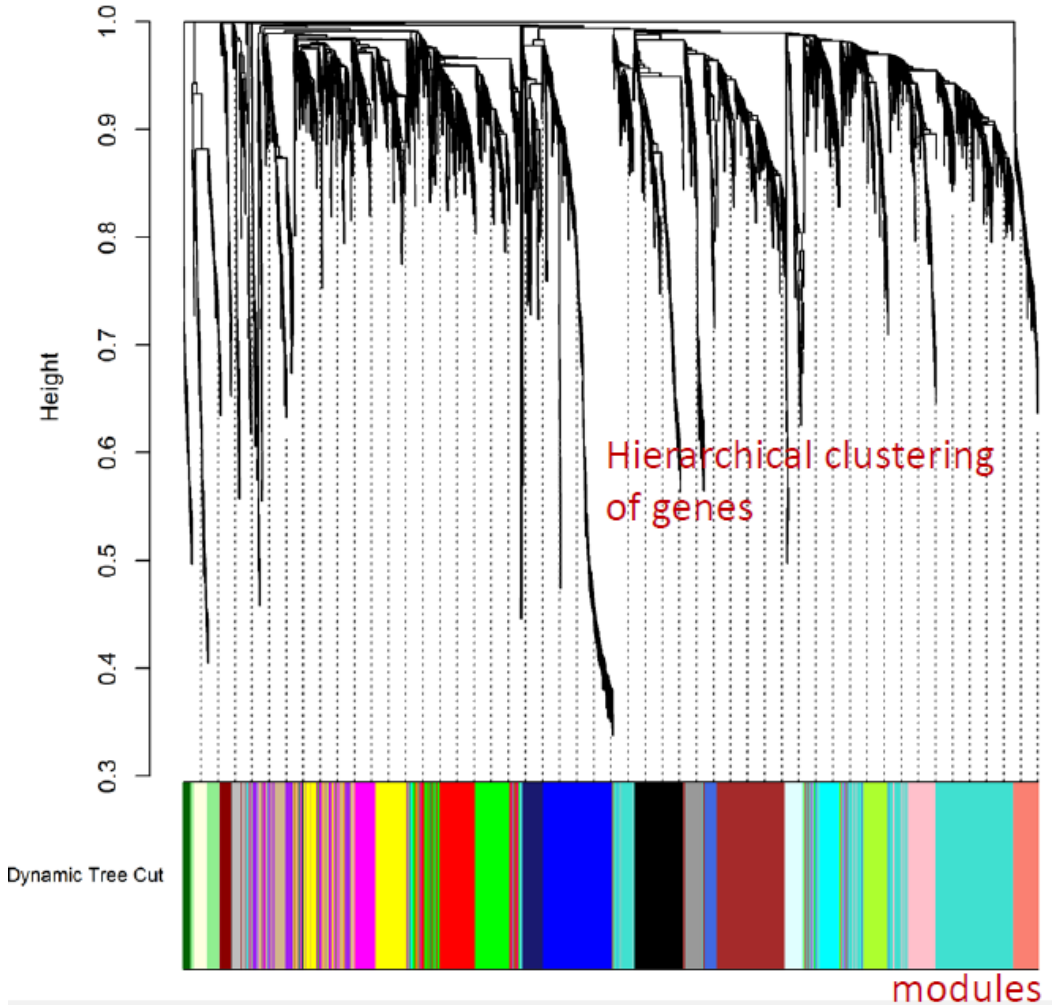
Clustering dendrogram



(dis)similarity between genes:
Topological Overlap Measure TOM

Step 3: Divide clustered genes into modules

- **Gene dendrogram and detected modules**

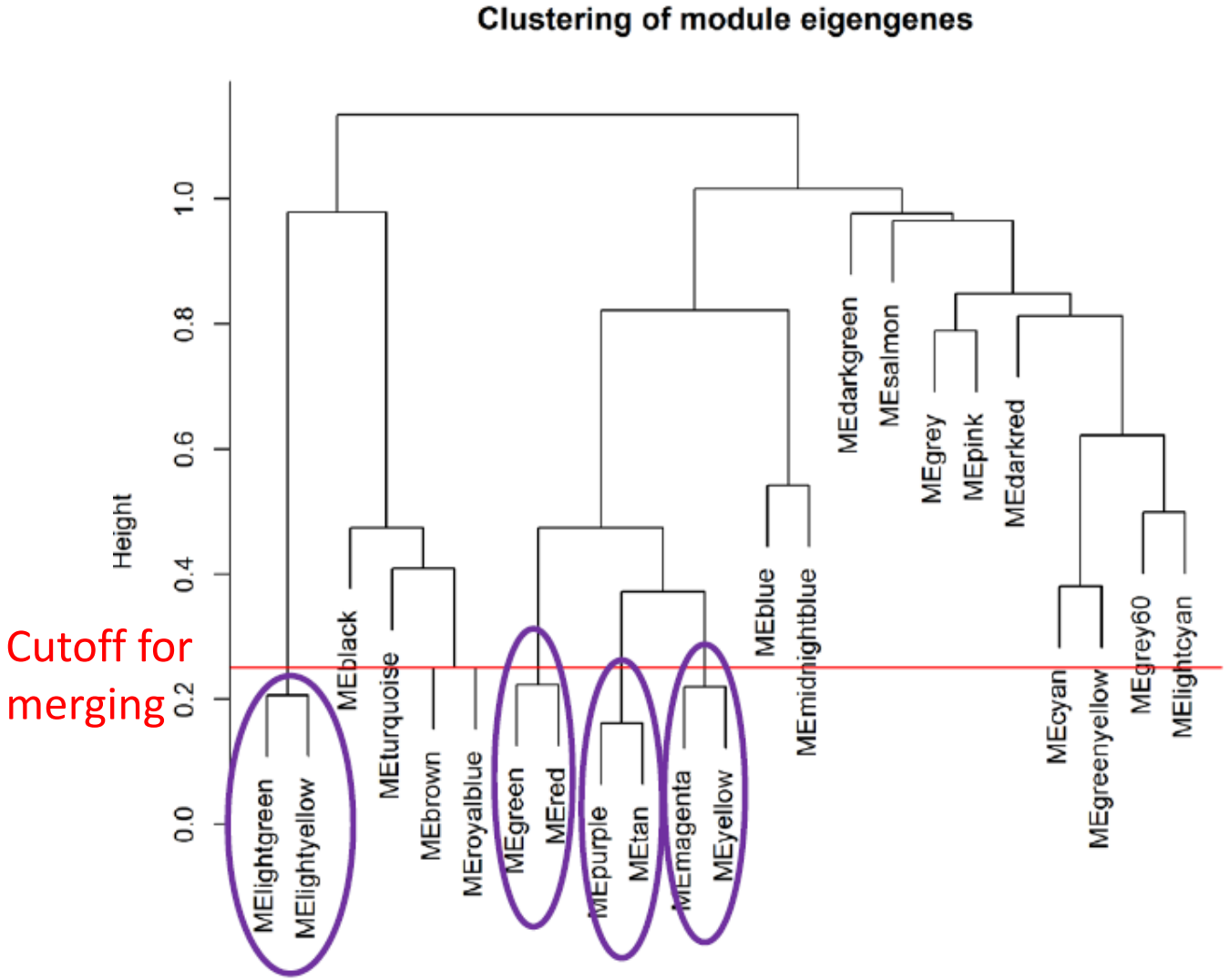


Dynamic tree cut algorithm groups genes into modules:
corFnc="pearson"; power=6; min.modulesize=30

Step 4 (Optional): Merge very similar modules

- **A module eigengene is** a 1-dimensional data vector, summarizing the expression data of the genes that form a module
- **How it is computed:** the 1st principal component of the expression data
- **What it is used for:** to represent the module in mathematical operations:
  - modules can be correlated with one another
  - modules could be **clustered together** (we can combine them)
  - modules can be correlated with external traits

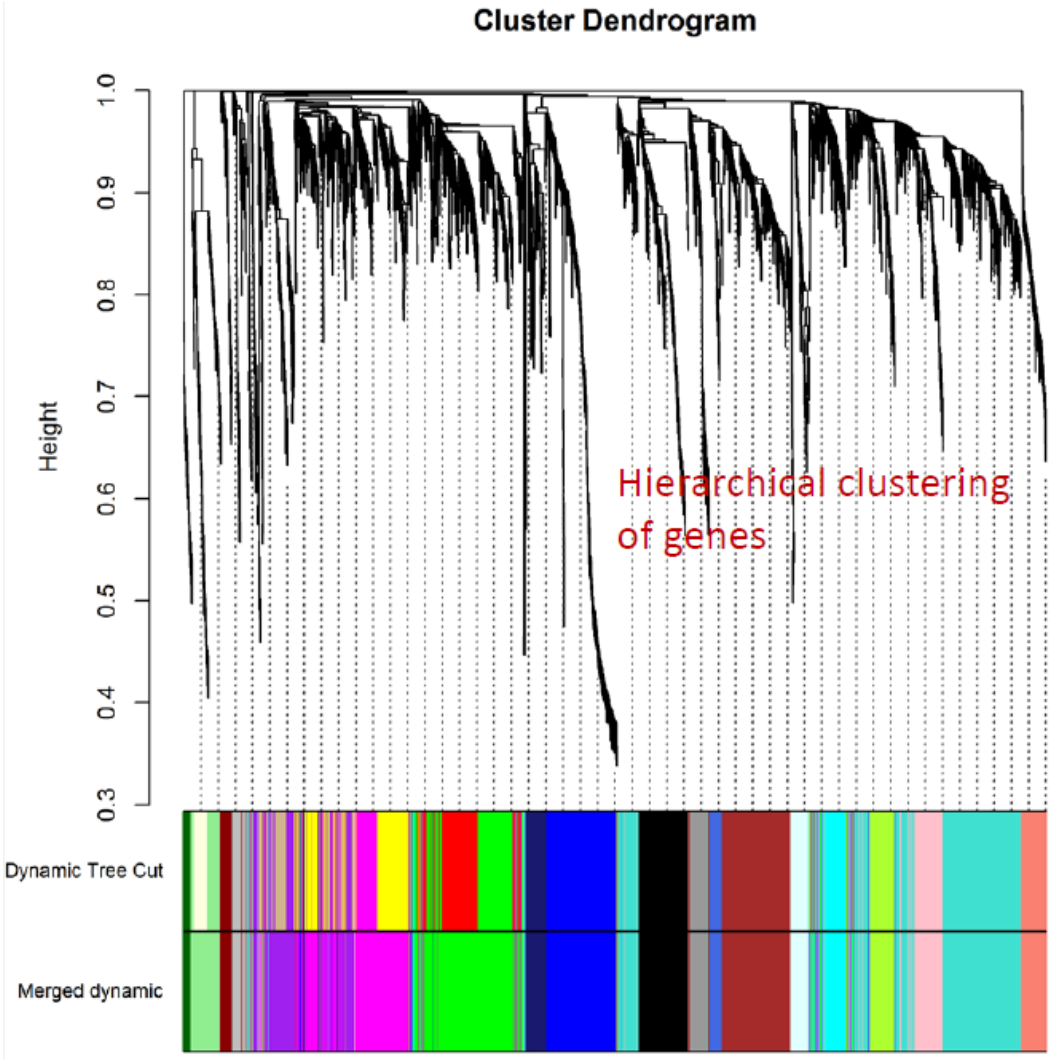
Clustering of module eigengenes



Dissimilarity measure: *1 –cor(Meigengenes)*

Merge modules whose dissimilarity is below the merging cutoff

Gene dendrogram and detected modules, before and after merging



Modules, before merging

Modules, after merging

corFnc="pearson"; power=6; min.modulesize=30

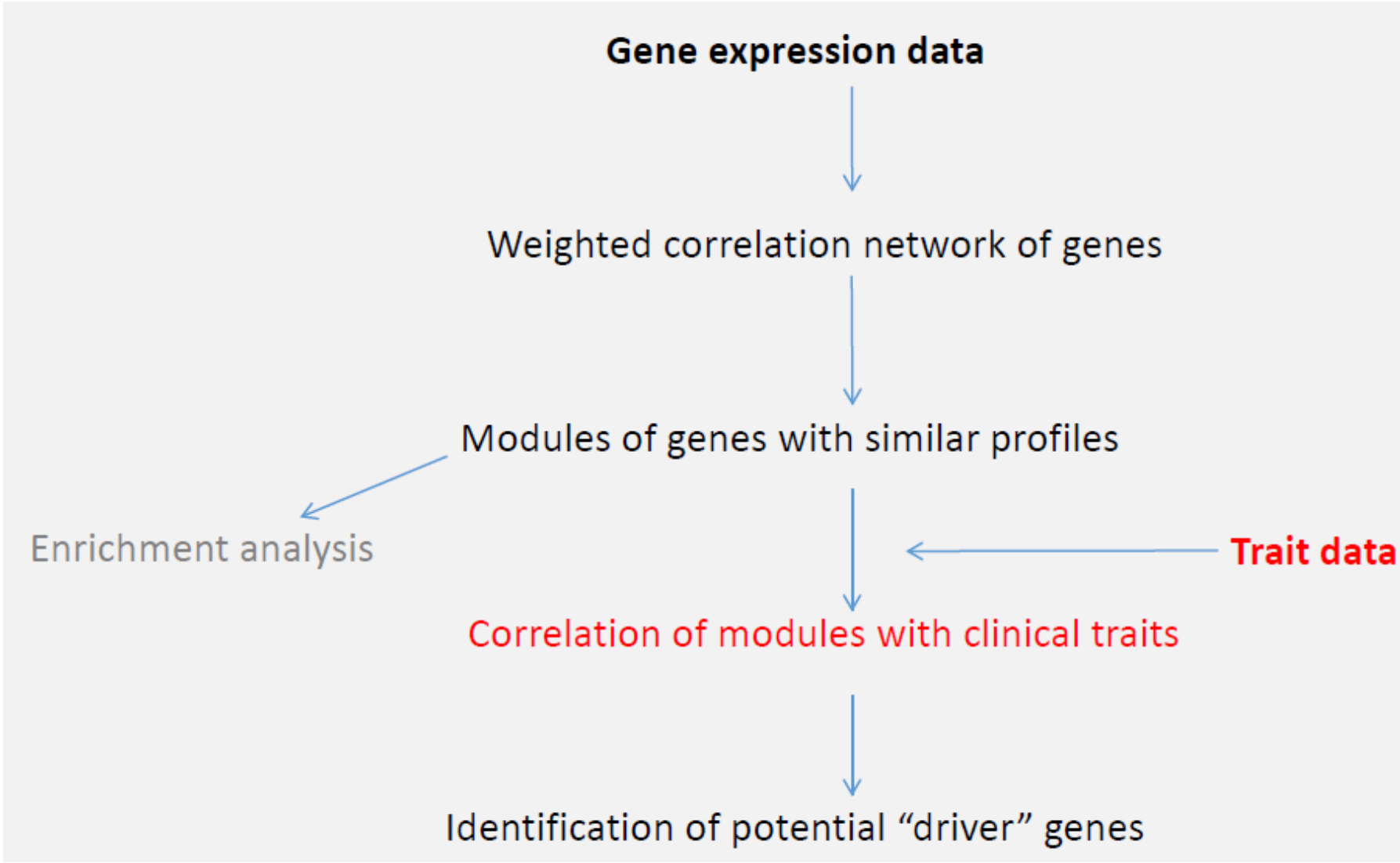
Module Detection: Decisions to make How to choose optimal parameters?

Dynamic Tree cut procedure

- **Minimal module size:** typically 20 or 30
- **CutHeight:** try different heights such as 0.95 or 0.9995, etc.

Module Merging procedure

- **Cutoff for module eigengene dendrogram:** typically between 0.15 and 0.25
  - *check if clusters look ok on dendrogram*
- **Merge once or several times?** Usually once, but merge step can be repeated:
  - *if some modules are very similar*
  - *if we want larger modules*


Gene expression data
Weighted correlation network of genes
Modules of genes with similar profiles
Enrichment analysis
Trait data
Correlation of modules with clinical traits
Identification of potential “driver” genes

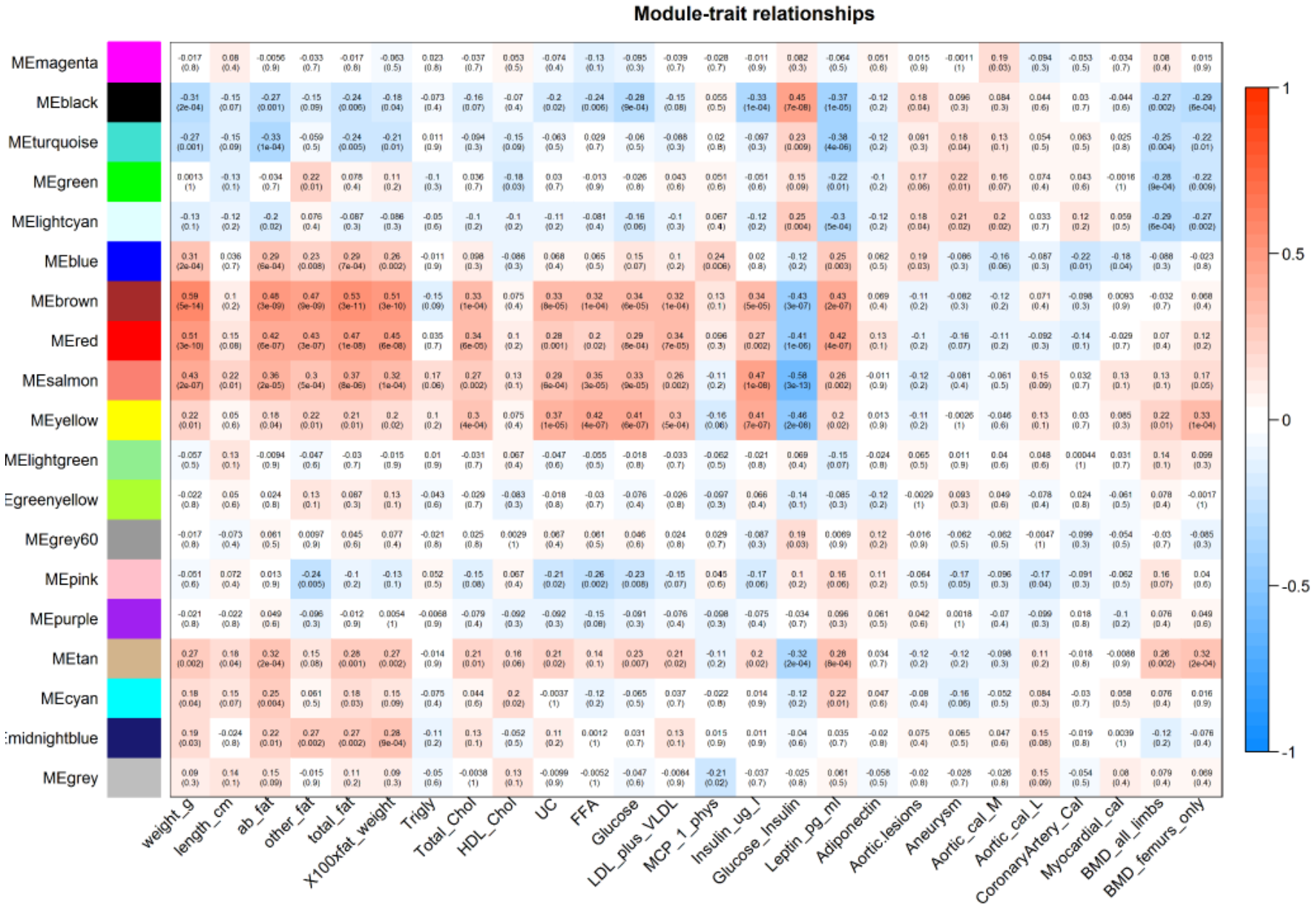
Correlate modules to external traits

Examples: trait variables from obtained from some samples/individuals (preferably the same samples that we generated the expression data set):

- weight_g
- length_cm
- ab_fat
- other_fat
- total_fat
- Trigly
- Total_Chol
- HDL_Chol
- UC
- FFA
- Glucose
- LDL_plus_VLDL
- MCP_1_phys
- Insulin_ug_l
- Glucose_Insulin
- Leptin_pg_ml
- Adiponectin
- Aortic lesions
- Aneurysm
- Aortic_cal_M
- Aortic_cal_L
- CoronaryArtery_Cal
- Myocardial_cal
- BMD_all_limbs
- BMD_femurs_only

How to compute correlations: each module eigengene to each trait variable: cor(MEs, traitDat)

Example


Module-trait relationships
MEmagenta
MEblack
MEturquoise
MEgreen
MElightcyan
MEblue
MEbrown
MEred
MEsalmon
MEyellow
MElightgreen
Egreenyellow
MEgrey60
MEpink
MEpurple
MEtan
MEcyan
midnightblue
MEgrey
weight_g
length_cm
ab_fat
other_fat
total_fat
X100xfat_weight
Trigly
Total_Chol
HDL_Chol
UC
FFA
Glucose
LDL_plus_VLDL
MCP_1_phys
Insulin_ug_l
Glucose_Insulin
Leptin_pg_ml
Adiponectin
Aortic.lesions
Aneurysm
Aortic_cal_M
Aortic_cal_L
CoronaryArtery_Cal
Myocardial_cal
BMD_all_limbs
BMD_femurs_only
1
0.5
0
-0.5
-1

Module preservation analysis in WGCNA

Is my network module preserved and reproducible?*

*Langfelder et al PloS Comp Biol. 7(1): e1001057.

Network-based module preservation statistics

- Input: module assignment in reference data.
- Adjacency matrices in **reference** A^{ref} and **test** data A^{test}
- Network preservation statistics assess preservation of
  - 1. network density: Does the module remain densely connected in the test network?
  - 2. connectivity: Is hub gene status preserved between reference and test networks?
  - 3. separability of modules: Does the module remain distinct in the test data?

Several connectivity preservation statistics

For general networks, i.e. input adjacency matrices

- *cor.kIM*=cor(kIM^{ref},kIM^{test})
    - *correlation of intramodular connectivity across module nodes*
- *cor.ADJ*=cor(A^{ref},A^{test})
    - *correlation of adjacency across module nodes*

For correlation networks, i.e. input sets are variable measurements

- cor.Cor=cor(cor^{ref},cor^{test})
- cor.kME=cor(kME^{ref},kME^{test})

One can derive relationships among these statistics in case of weighted correlation network

Choosing thresholds for preservation statistics based on permutation test

- For correlation networks, we study **4 density** and **3 connectivity** preservation statistics that take on values <= 1
- Challenge: Thresholds could depend on many factors (number of genes, number of samples, biology, expression platform, etc.)
- Solution: Permutation test. Repeatedly permute the gene labels in the test network to estimate the mean and standard deviation under the null hypothesis of no preservation.
- Next we calculate a Z statistic: Z=(observed-mean)/sd
- We have had **4 density** and **3 connectivity** preservation statistics:

$$Z_{density} = \text{median}(Z_{meanCor}, Z_{meanAdj}, Z_{propVarExpl}, Z_{meanKME}).$$

$$Z_{connectivity} = \text{median}(Z_{cor.kIM}, Z_{cor.kME}, Z_{cor.cor}).$$

Permutation test for estimating Z scores

- For each preservation measure we report the observed value and the permutation Z score to measure significance (Z=(observed-mean)/sd).
- Each Z score provides answer to “Is the module significantly better than a random sample of genes?”
- Summarize the individual Z scores into a composite measure called **Z.summary**

$$Z_{summary} = \frac{Z_{density} + Z_{connectivity}}{2}.$$

- **Zsummary < 2 indicates no preservation,**
- **2<Zsummary<10 weak to moderate evidence of preservation,**
- **Zsummary>10 strong evidence**

Summary preservation

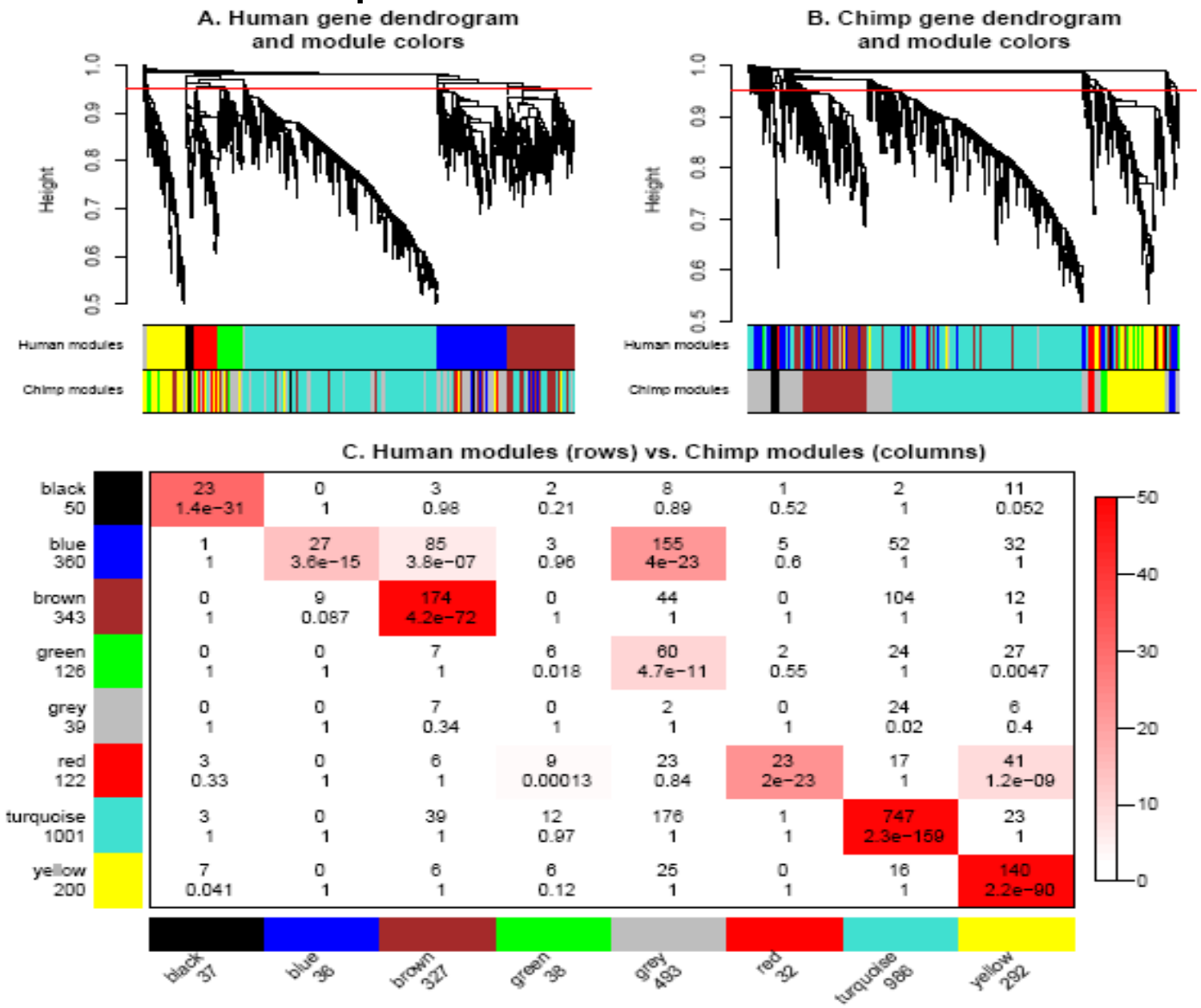
- Standard cross-tabulation based statistics are intuitive
  - Disadvantages: i) only applicable for modules defined via a module detection procedure, ii) ill suited for ruling out module preservation
- Network based preservation statistics measure different aspects of module preservation
  - Density-, connectivity-, separability preservation
- Two types of composite statistics: Zsummary and medianRank.
- Composite statistic Zsummary based on a permutation test
  - Advantages: thresholds can be defined, R function also calculates corresponding permutation test p-values
  - Example: Zsummary<2 indicates that the module is *not* preserved
  - Disadvantages: i) Zsummary is computationally intensive since it is based on a permutation test, ii) often depends on module size
- Composite statistic medianRank
  - Advantages: i) fast computation (no need for permutations), ii) no dependence on module size.
  - Disadvantage: only applicable for ranking modules (i.e. relative preservation)

Application:
Studying the preservation of human brain co-expression modules in chimpanzee brain expression data.

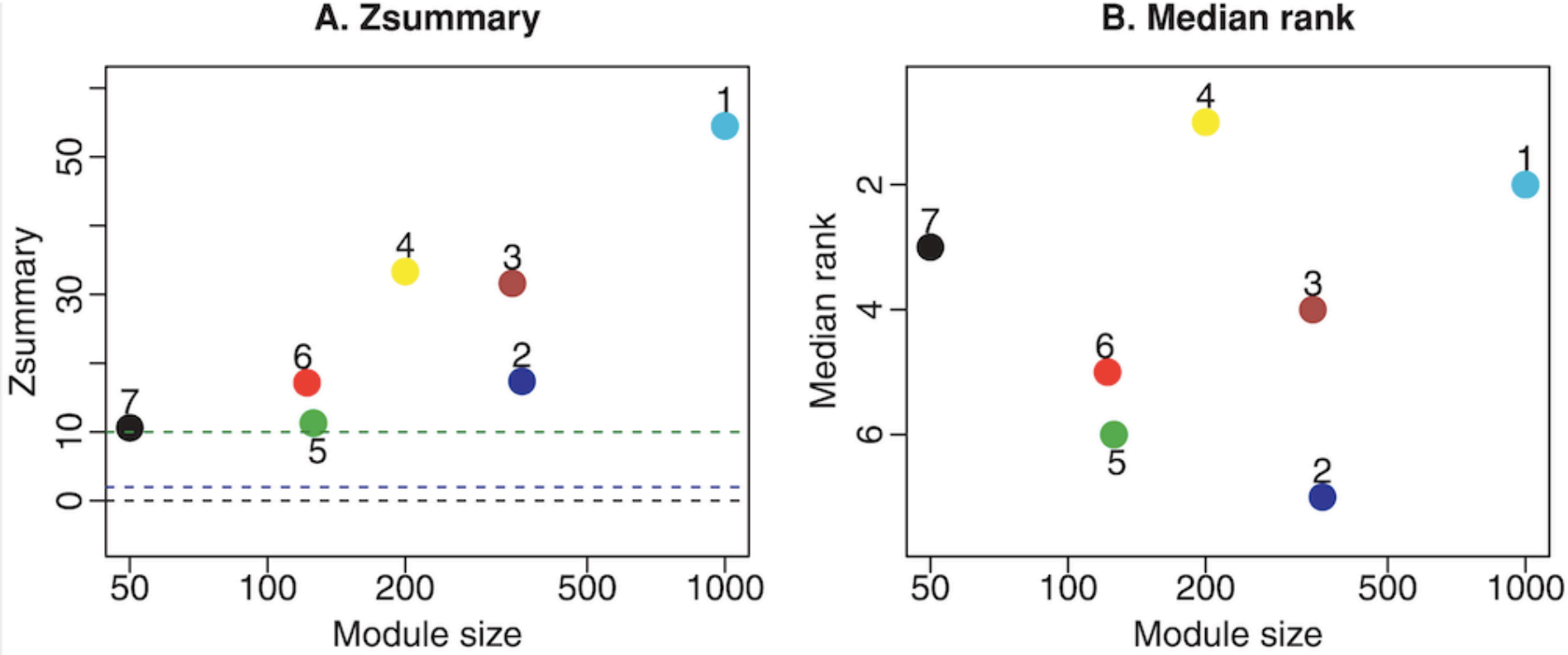
Modules defined as clusters
(branches of a cluster tree)

Data from Oldam et al 2006

Preservation of modules between human and chimpanzee brain networks



2 composite preservation statistics



- Zsummary is above the threshold of 10 (green dashed line), i.e. all modules are preserved.
- Zsummary often shows a dependence on module size which may or may not be attractive
- In contrast, the median rank statistic is not dependent on module size.
- It indicates that the yellow module is the most preserved one

Implementation and R software tutorials, WGCNA R library

- General information on weighted correlation networks
- Google search
  - "WGCNA"
  - "weighted gene co-expression network"
- R function `modulePreservation` is part of WGCNA package
- Tutorials: preservation between human and chimp brains

www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/ModulePreservation

Pseudocode for modulePreservation()

- setLabels = c("Control", "Disease");
- multiExpr = list(Control = list(data = dataExp_Control), Disease = list(data = dataExp_Disease));
- multiLabel = list(Control = moduleLabelsControl, Disease=moduleLabelsDisease);
- mp = modulePreservation(multiExpr, multiLabel, referenceNetworks = c(1:2), nPermutations = 100, randomSeed = 1, quickCor = 0, verbose =3);
- save(multiExpr, multiLabel, mp, file =“modulepreservationLabel_Alldata.RData")
- ref = 1; test = 2
- Zsummary1=mp$preservation$Z[[ref]][[test]][, 2]
- names(Zsummary1)=rownames(mp$preservation$Z[[ref]][[test]])
- low.preserved1=Zsummary1[which(Zsummary1<2)]
- ref=2; test=1
- Zsummary2=mp$preservation$Z[[ref]][[test]][, 2]
- names(Zsummary2)=rownames(mp$preservation$Z[[ref]][[test]])
- low.preserved2=Zsummary2[which(Zsummary2<2)]

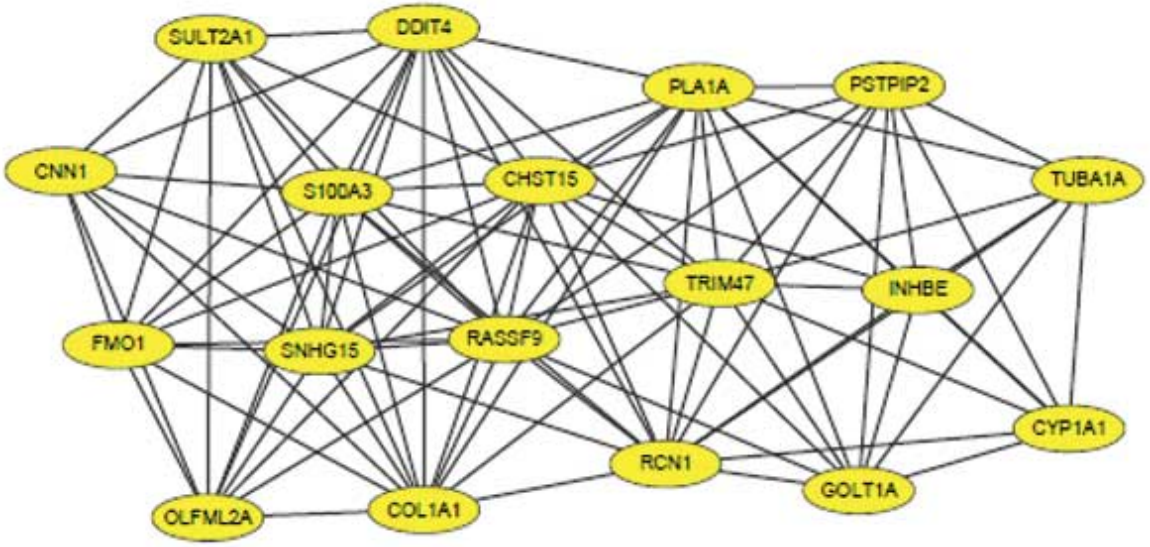
Introduction to Bayesian Networks

Content

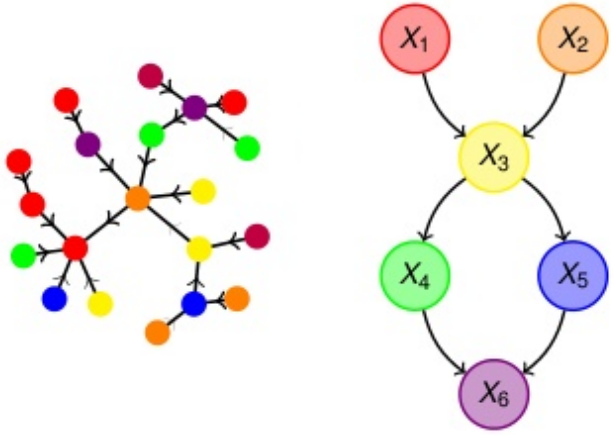
- Definition of Bayesian networks (BN)
- An example, Rimbanet…
- Mandatory and optional input files
- Comparison of BN tools

Gene-gene interaction networks

Co-expression networks

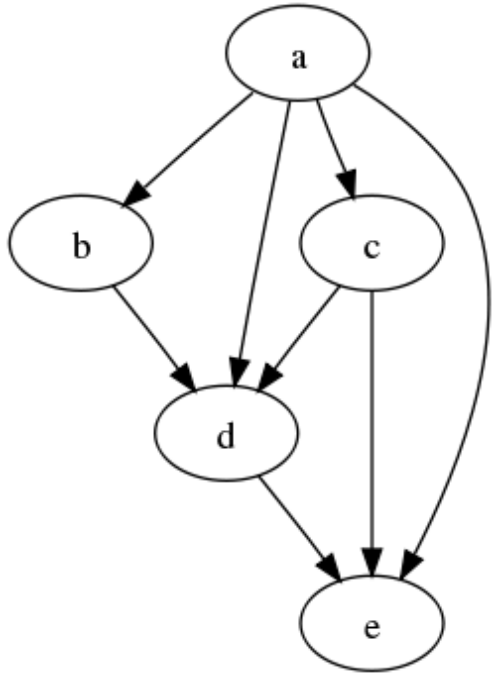


Bayesian networks



What is a DAG

- A directed acyclic graph (DAG) is a finite directed graph with no directed cycles.
- There is no way to start at any vertex *v* and follow a consistently-directed sequence of edges that eventually loops back to *v* again.



Bayes' theorem

- Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

- Bayes' theorem is stated as:

$$P(A,B) = P(B \mid A)P(A) = P(A \mid B)P(B)$$

$$\Rightarrow P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} = \frac{P(B \mid A)P(A)}{\sum_i P(B \mid A_i)P(A_i)}$$

Bayes' theorem - an example

- Suppose there is a test to detect a disease in a human cohort is 99% sensitive and 95% specific.

- Suppose that:

Ratio of the people carrying this disease in the population: P(C)=2%

Ratio of the people not carrying this disease (Healthy): P(H)=98%

- **Q: If a randomly selected individual's test result is positive, what is the probability that he/she carries the disease?**

Bayes' theorem - an example

- Suppose a test to detect a disease in human is 99% sensitive and 95% specific, means:

For the patients, the test gives a positive result with 99% (TP) and gives a negative result with 1% (FN): P(+ | C)=0.99 and P(-| C)=0.01

For the healthy samples, test gives a negative result with 95% (TN) and a positive result with 5% (FP): P(-| H)=0.95 and P(+| H)=0.05

- **Q: If a randomly selected individual's test result is positive, what is the probability that he/she carries the disease?** Hence, **P(C | +) = ?**

$$P(C|+) = \frac{P(+|C)P(C)}{P(+)} = \frac{P(+|C)P(C)}{\sum_i P(+|Option_i)P(Option_i)}$$

$$= \frac{P(+|C)P(C)}{P(+|C)P(C) + P(+|H)P(H)} = \frac{0.99 \times 0.02}{0.99 \times 0.02 + 0.05 \times 0.98} \approx 29\%$$

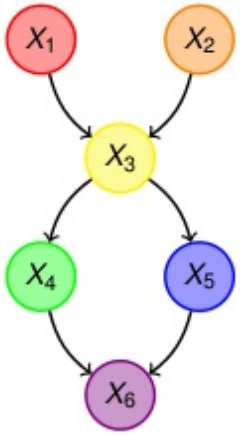
Bayesian networks (BNs)

- BNs are directed acyclic graphs (DAGs) in which the edges of the graph are defined by conditional probabilities that characterize the distribution of the states of each node given the state of its parents.

  $P(M \mid D) \approx P(D \mid M) \times P(M)$, where M: network model, D: observation data

- We can define a partitioned joint probability distribution over all nodes:

$P(X) = \prod_i P(X^i \mid \mathrm{Pa}(X^i))$ where $\mathrm{Pa}(X^i)$ is parent set of X^i.



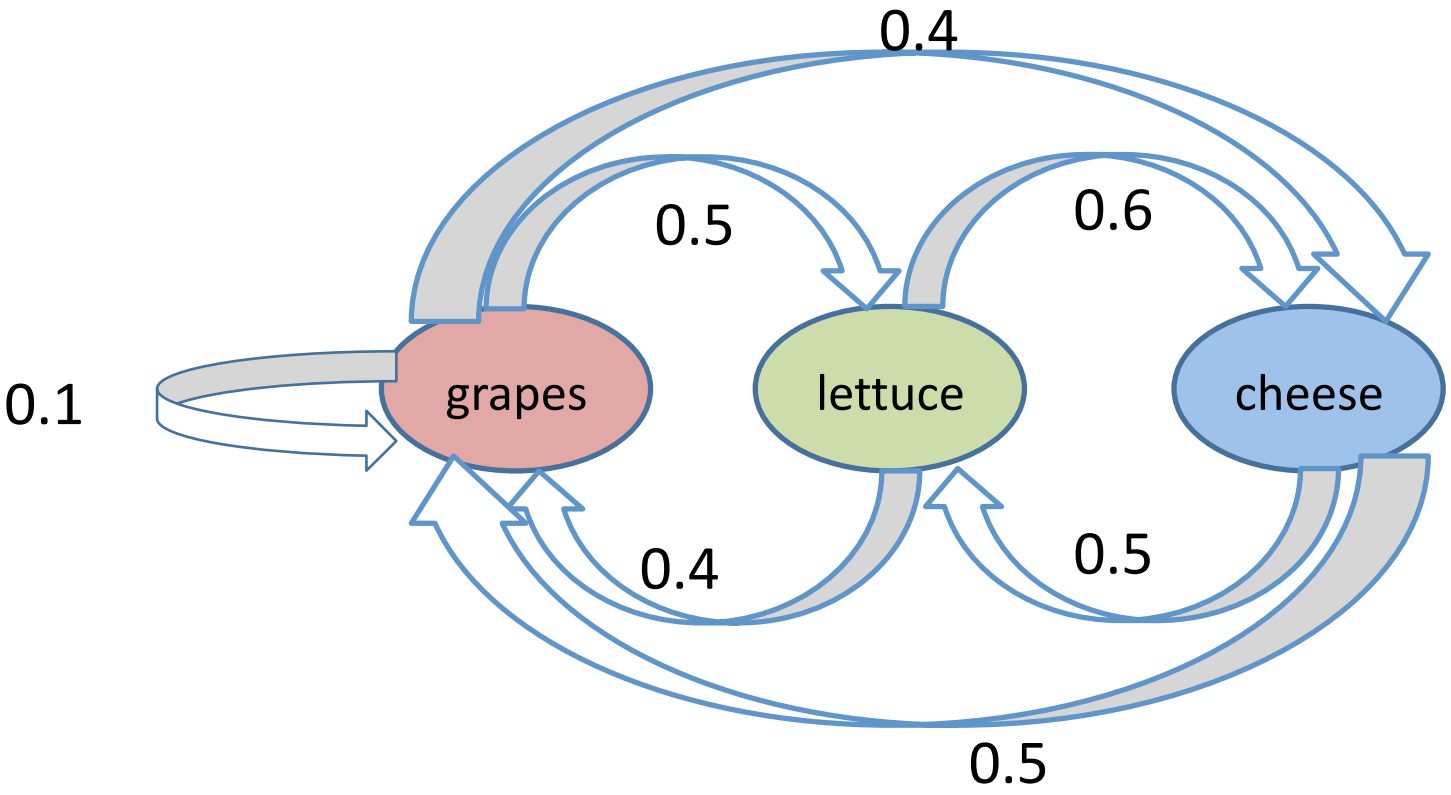
$$P(X_1, \cdots, X_6) = P(X_1)P(X_2)P(X_3|X_1, X_2)$$
$$P(X_4|X_3)P(X_5|X_3)P(X_6|X_4, X_5)$$

Application of Bayes' theorem in gene regulatory network inference

- The numbers of the possible network structures (M) ↑ with the # of nodes ↑.
- Search of all possible structures to find the best supported one by the data is not feasible.
- Solution: We can **use Markov chain Monte Carlo (MCMC)** simulation to identify a particular amount (e.g. 1,000) of plausible networks.
- A consensus network is obtained by combing these plausible networks.

Markov chain

- State of the system at time “t+1” is predicted by solely using the state at time “t”, i.e. previous states (t-1, t-2,..) will not be used for predicting t+1.



Markov chain Monte Carlo (MCMC)

- MCMC methods are a class of sampling algorithms.
- It is used for sampling from a probability distribution based on a **Markov chain** model that has the desired distribution as its equilibrium distrib.
- The state of the chain after a number of steps is used as a sample of the desired distribution.
- The quality of the sample ↑ as the number of steps ↑.
- MCMC methods mainly used for numerical approximations of the integrals

MCMC in the BN problem

- MCMC algorithm to identify 1000 of networks:
  - Start with a null network (for each 1000 cases) including prior edges.
  - Make random changes on each network such as:
    - Flip,
    - Add, and
    - Delete individual edges;
  - Accept the random changes made above that lead to an improvement in the fit of the network to the data.
  - The fitness is assessed by **Bayesian information criterion (BIC)**, which also penalizes the network if the #of the parameters (complexity)↑.
  - **Note:** $BIC = -2\ln P(D \mid M) + k\ln(n)$ where n is # of data points in D (sample #); k = # of the parameters to be estimated(parent # of node i).

- Create a consensus network by combining above 1000 networks and check for the DAG feature of the final network.

An example Bayesian Network method:

Rimbanet

Preprocessing steps and preparing the input files

- **For RNAseq data, normalize gene expression: log2(expr+1)**
- **Remove lowly expressed genes (e.g. genes with expr<1 for >50% of the samples)**
- **Remove outlier samples**
- **Find the top 5,000 highly variable genes**
- **Discretize your gene expression data e.g. using k-means clustering for k=3.**
- **Prepare prior files (e.g. cis-genes, causal relationships, partial corr, TF-TG), for your expression data.**
- **Ready to call Rimbanet!**

Mandatory Arguments

1. <discretized_data_path>: full path to tab delimited discretized data (discretize by preferred method to three states; 0,1,2). (k-means is used) Format: first column is gene names, no header

2. <continuous>: Default is TRUE (to generate initial BIF [Bayesian interchange format] xml file and to update probs).

3. <continuous_data_path>: If using continuous data (flag -C is TRUE), then use this flag to give the path to the continuous data. Format: first column is gene names, no column headers, tab delimited.

4. <path_to_BN_extra>: full path to Run BN directory includes perl codes and c executable files.

Mandatory Arguments - 2

5. <output_path>: output directory for BN. Driver will create this directory if it is not yet made. THIS DIRECTORY MUST BE EMPTY!

6. <cis_eQTL>: full path to file that contains a list of cis eQTLs (one-column gene names) with NO HEADERS OR ROW NAMES [**see <high_quality_cis_eqtl_file> in optional arguments**]. The cis-genes should exist in your continuous and discretized expr. data.

NOTE: IF YOU WILL NOT USE EQTL PRIORS, PASS IN PATH TO "empty_cis_file" in BN.

Optional Arguments

1. <high_quality_cis_eqtl_file>: Full path to file of high quality cis eqtls (e.g. FDR<0.01). Format: same as <cis_eQTL>.

2. <causal_list>: Full path to file with causal gene relationships used for priors. Default: NULL. Format: gene<tab>gene

3. <partial_correlation_list_file>: Full path to file with list of genes that have partial correlation. Default: NULL.

   Format: gene<tab>-><tab>gene<tab>0.5<tab>partial corr. Value

   NOTE: For partial cor: ppcor CRAN package can be used:

   pcor(t(expr))$estimate)

Optional Arguments – 2

4. <kegg_prior_file>: Full path to file with list of genes that have shared KEGG pathways (that might be extended to other pathways' shared genes?). Default: NULL. Format:gene<tab>gene

5. <TF_prior_file>: Full path to file with list of genes with TF binding relationships. Default: NULL. Format: gene<tab>gene

6. <ppi_prior_file>:full path to file with list of genes that have protein-protein interactions. Default: NULL.

   Format: gene<tab>gene<tab>value

How to pass these files as arguments to the rimbanet?

- -d: Discretized data file
- -b: Path for the rimbanet BN program path
- -e: cis eQTL data file
- -o: Output path
- -E: High quality cis eQTL
- -c: Causal gene list
- -r: partial correlation list
- -K: KEGG prior file
- -T: TF prior file
- -P: PPI prior file
- -C: Using continuous data to update prior is TRUE or FALSE?
- -w: Continuous data file to update prior (above flag should be TRUE)

BN driver shell script:

- #! /bin/bash
- # Use current working directory
- #$ -cwd
- #$ -o BN_output_GTEx_v7_whole_blood.joblog
- #$ -j y
- #$ -S /bin/bash
- # Notify at beginning and end of job?
- #$ -m n
- # Use your normal environment variables in the job
- #$ -V
- ## Submit for 14 days:
- #$ -l h_data=16G,h_rt=335:59:58,highp -now n

Example

```
qsub /u/YOURPATH/BN_driver_GTEx_v7_whole_blood.sh \

-d /u/YOURPATH/discretized_data_forBN_whole_blood.txt \

-C TRUE \

-w /u/YOURPATH/continous_data_forBN_whole_blood.txt \

-b /u/YOURPATH/template_BN_codes \

-o /u/YOURPATH/output_BN_whole_blood \

-e /u/YOURPATH/cis_eQTL_prior_forBN_whole_blood.txt \

-E /u/YOURPATH/HQ_cis_eQTL_prior_forBN_whole_blood.txt \

-T /u/YOURPATH/TFtoTG_forBN_whole_blood.txt
```

Comparison of BN tools

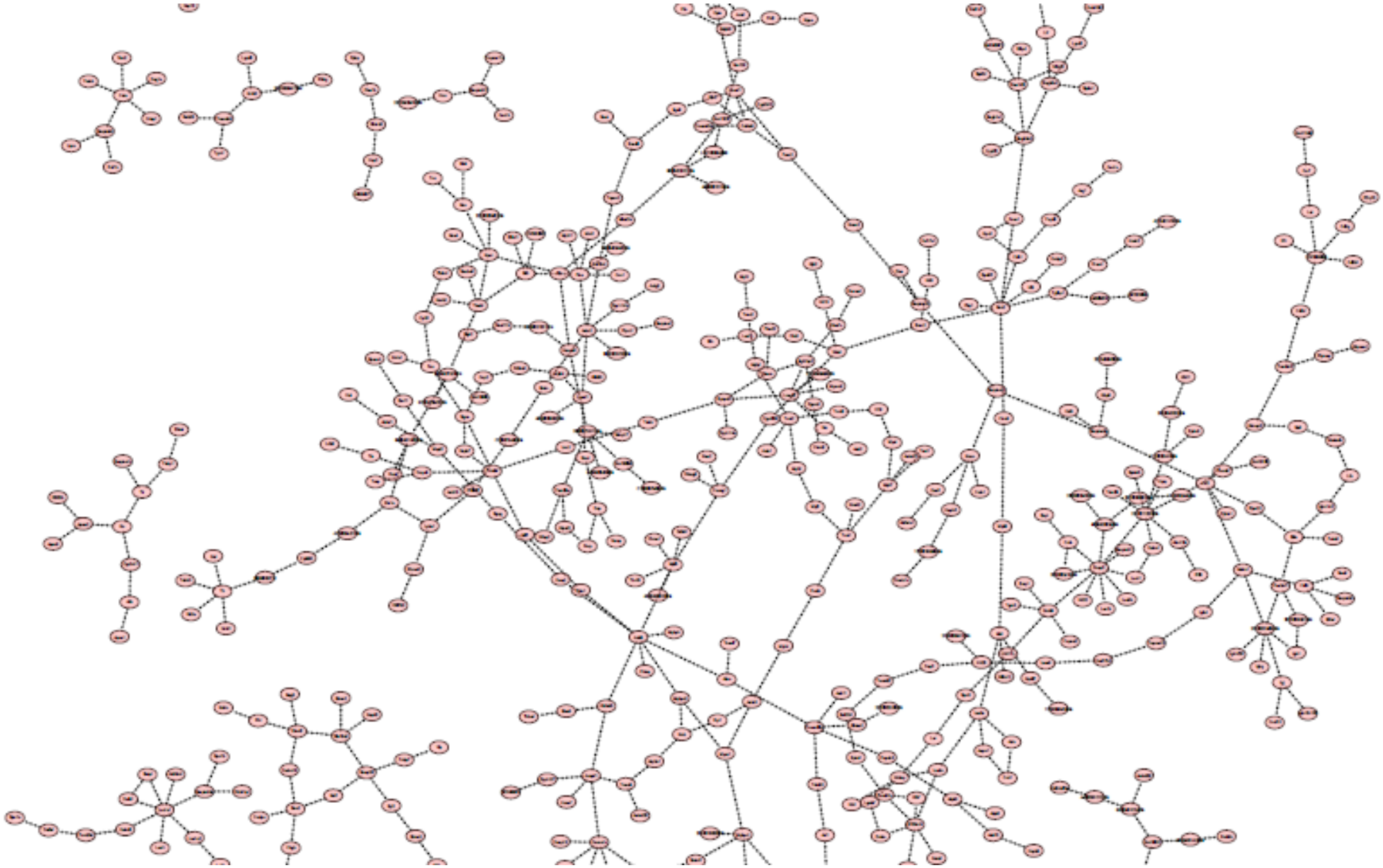
Software packages

Tool	Platform	Scoring method(s)	Data type	Multi processing?	Prior info support?
RIMBANet	Perl+C	BIC	Continuous + Discrete	No	Yes
Bnfinder	Python	BIC , BDE, MIT	Continuous + Discrete	Yes	Yes
sparsebn	R	BIC+CV	Continuous + Discrete	No	**No**
bnlearn	R	BIC, AIC, BDE	Continuous + Discrete	Yes	Yes
catnet	R	BIC, AIC	Continuous + Discrete	Yes	Yes

- BIC: Bayesian information criterion
- BDE: Bayesian Dirichlet equivalence
- MIT: $\chi 2$ distance based Mutual information test
- AIC: Akaike information criterion
- CV: cross validation

Example - 500 genes 100 samples from an aorta tissue expression dataset

Tool	Scoring method	Runtime (sec)	Edge #
RIMBANet	BIC	5,586 sec (~1.30 hour) on 1 node	502
Bnfinder	BIC	12,710 sec (~3.30 hours) on 8 nodes (Could not finish in 24h on 1 node)	479
sparsebn	BIC	2,409 sec on 1 node **Prior info integration is not available***	5,019
bnlearn	BIC	Could not finish in 24h (on 8 nodes)	--
catnet	BIC	Could not finish in 24h (on 8 nodes)	--



Summary

- Gene networks provide us to:
  - Identify biological mechanisms and molecular subnetworks underlying common human diseases.
  - Integrating diverse type of multi-dimensional biological datasets.
  - Predicting the key driver genes in disease-related subnets.

- After presenting our data-driven findings, experimentalists can conduct *in vivo* and/or *in vitro* experiments to test our candidate genes on a given tissue, for certain hallmarks of a disease/disorder.