Neuro-genomics - class 5
Clustering analysis and functional analysis

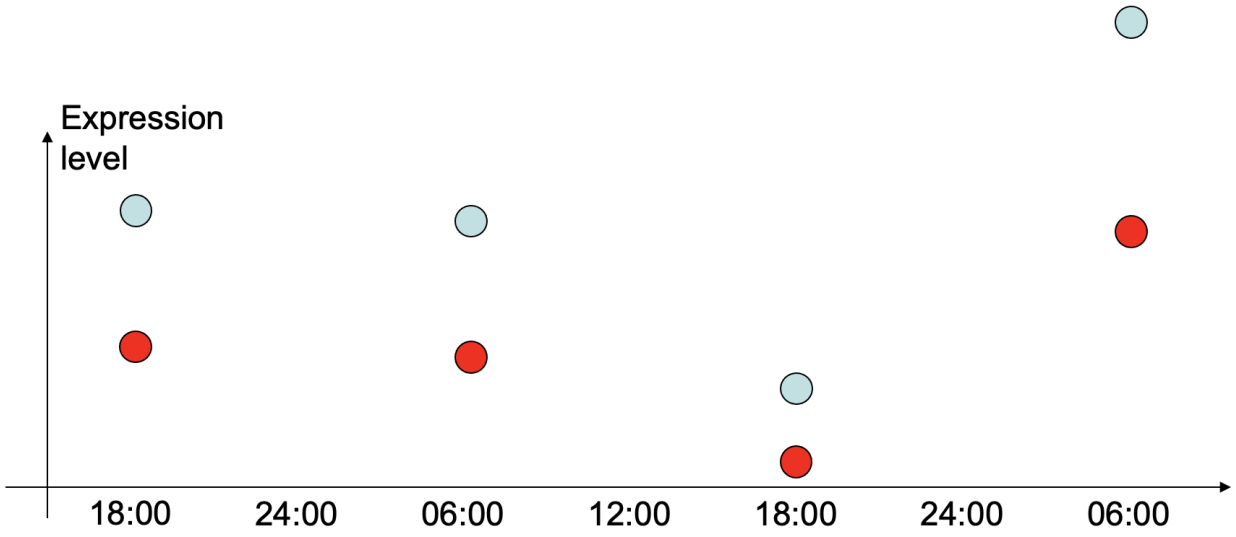
Introduction

Last week we introduced the concept of clustering analysis. The idea is simple: say that we measure thousands of genes using several samples of the same tissue type -- clustering analysis allows us to cluster, or group, genes that 'behave similarly'.

A concrete example that we presented last week is a coral colony, which we sampled every few hours for several days using RNAseq. Another example can be lung tissue, which we sample from patients with COVID-19, in different time points after the exposure to the virus. If we compare early time points to late time points, or infected patients to uninfected patients, we will likely get hundreds of genes that are differently expressed. Clustering allows us to detect groups of genes, which is a first step towards understanding the molecular rule of this group of genes.

Today we will discuss clustering approaches, and then go a step further to functional analysis of groups of genes. Simply put, molecular functional analysis allows us to estimate the function of a group of genes.

First, a reminder from last week. What does it mean that genes 'behave similarly'? We introduced Pearson's definition, i.e. genes are similar if they go up and down together, regardless of their absolute expression values.



Important note: below we discuss clustering of genes according to their expression patterns. Similar analysis can be performed on the level of cells (and not genes) as we will discuss later on in the course.

Distance metric between expression profiles
All clustering approaches need a distance metric between the expression profile of each gene. In many cases, Pearson's correlation will serve as a good distance metric.

Given two genes, one with expression values marked as x and the second with expression values marked as y, the Pearson's correlation is defined below:

$$PE(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$$

Another popular metric is Euclidean distance. Definitions from here:

> *Minkowski:* Distances measures known as Manhattan (MAN), Supreme (SUP) and, Euclidean (EUC) are particular cases of the more general Minkowski family of metric distances [23], defined in Equation (10). Such distances are obtained with different configurations of λ, in Equation (10). For the three particular cases of the Minkowski distance we consider in this work, i.e., MAN, SUP and, EUC, we have $\lambda = 1$, $\lambda = \infty$ and, $\lambda = 2$, respectively.

$$Minkowshi\,(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{n}\left|x_i - y_i\right|^{\lambda}\right)^{1/\lambda} \quad (10)$$

The distance metric is usually represented via a distant matrix, in which the rows and the columns are the names of the genes (in the same order), and the values of the matrix are the distance metric. This matrix is expected to be symmetrical with respect to the diagonal. If the distance matrix is constructed using Pearson's correlation, and say that Pearson's correlation between the expression profiles of gene i and gene j is marked R, then the distance $d_{i,j}$ will be 1-R.

Hierarchical clustering algorithm
Hierarchical clustering starts with a representation of the data (the distances between the gene's profiles) as a tree structure. The data points (i.e. the gene's profiles) are represented as the leaves. A particular tree representation is a dendrogram.

Here we will describe a popular algorithm for hierarchical clustering, called UPGMA. This algorithm recursively merge similar gene profiles, and compute the new distances between the merged cluster and all other gene profiles.

The algorithm is as follows (adapted from here):

Start point:

Overall N genes, each with an expression profile, which are represented by a distant matrix d which is N by N.

Iteration:

Combine two gene profiles (or two clusters) to form a new cluster.

1. Find gene/cluster i and gene/cluster j that have the smallest distance between them. At the beginning, this is just the smallest element in d.

At later steps (i.e. after we start the clustering), use a matrix D instead of d, which is:

$$D_{i,j} = \frac{1}{n_i + n_j} \sum_{p \in C_i} \sum_{q \in C_j} d(p, q)$$

n_i and n_j are the number of genes in cluster i and cluster j, respectively. The double summation just means that we are running over all members of cluster i (say 2 genes) and all members of cluster j (say 3 genes), and sum all the distances d between them. We then divide by the total number of genes (in our simple example this is 5). Note that when we start the clustering D=d.

2. Create a new cluster – (ij), which at the beginning has 2 members.

At later steps the number of members will be $n(ij) = n_i + n_j$.

n_i and n_j are the number of genes in cluster i and cluster j, respectively.

3. Connect gene/cluster i and gene/cluster j to a new node, which corresponds to the new cluster (ij). Compute the distance from the new cluster to all other genes/clusters (except for i and j, which are no longer relevant) as an average of the distances from its components. At the beginning, this is just:

$D(ij),k = ½*d_{i,k} + ½*d_{j,k}$

At later steps, in which the number of members in i and j are not necessarily 1, use the weighted average to calculate this distance:

$$D_{(ij),k} = (\frac{n_i}{n_i + n_j}) D_{i,k} + (\frac{n_j}{n_i + n_j}) D_{j,k}$$

n_i and n_j are the number of genes in cluster i and cluster j, respectively.

4. Delete the columns and rows in D that correspond to clusters i and j, and add a column and row for cluster (ij), with D(ij),k computed as above.

5. Return to step 1 until there is only one cluster left.

The advantage of using hierarchical clustering is that it's simple and intuitive. The disadvantage is that this clustering always ends with one cluster, and therefore we need to stop the procedure at some point to get several clusters and not just one. The dendrogram representation can help

us to decide how many clusters to create. In addition, we can always decide in advance that we want X clusters, and just stop the procedure described above after X clusters are obtained.

Note however that the problem of creating X clusters with hierarchical clustering is more profound. Overall, we would like the clusters to be homogenous (i.e. the genes in the clusters should have similar profiles) but we would also want high separation between the clusters. Hierarchical clustering is very good at connecting two (or more) genes that are similar to one another, but not as good at creating separation between the X clusters. Therefore, additional methods for clustering are needed.

K-means clustering

K-means clustering starts with a decision of the user -- we need to decide what is the number of clusters, marked by K. Obviously, in most cases we don't know in advance what is the correct number of clusters. We can test several options for K, and then decide which K is optimal using silhouette analysis.

However, once we have K, the algorithm is simple. We start by a random partitioning of the genes to K clusters. For each one of these K clusters, we detect a center point. As the name implies, the center point is located in a position that minimizes the distances (as defined above) from all other genes in that cluster. For each center point, we record the sum of distances from all the genes in that cluster, this sum is termed 'cost'.

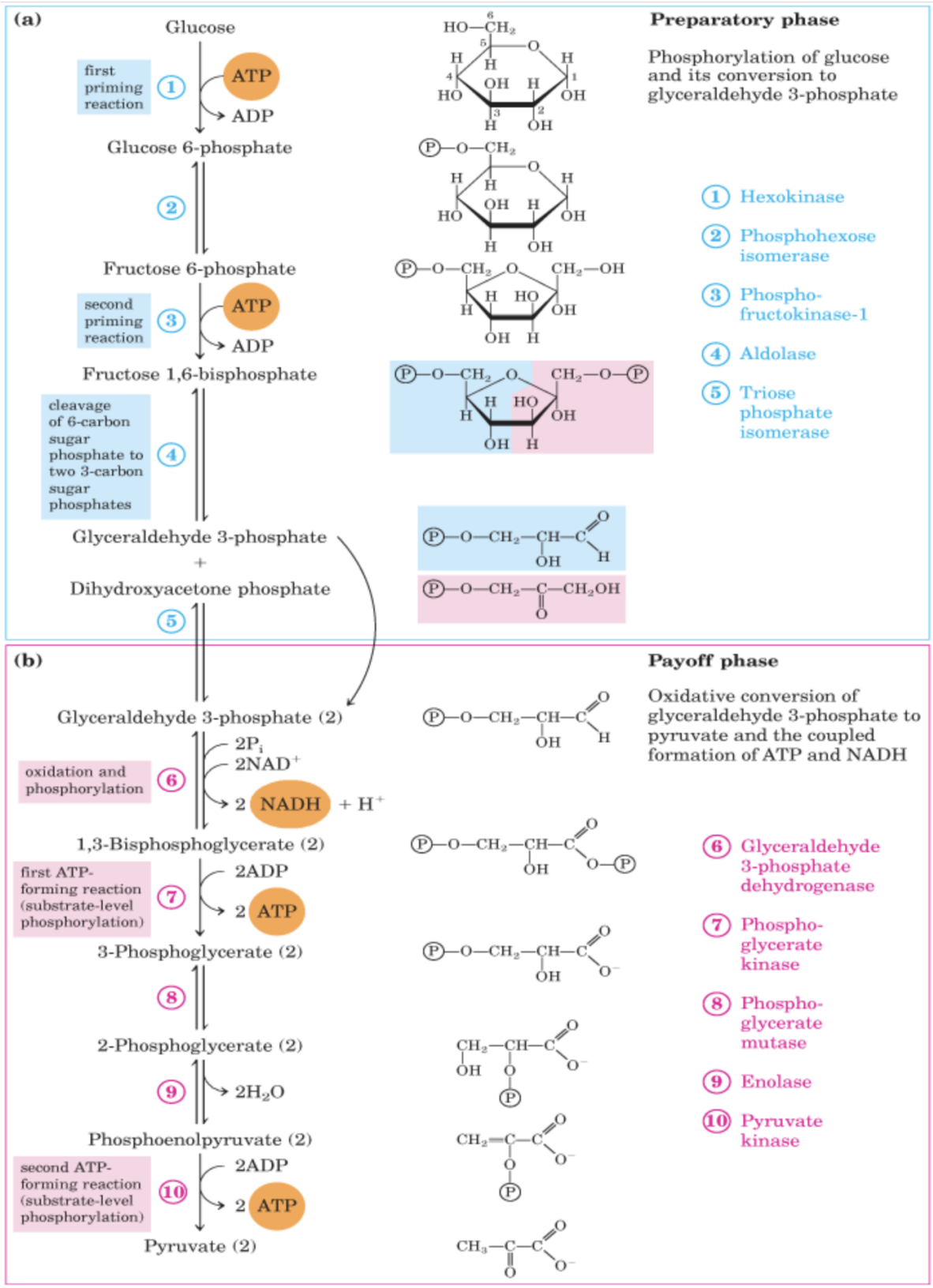
Next, we calculate the change in cost if we change the cluster of a gene. Say gene X was in cluster A, what will be the change in cost of clusters A and B if gene X will now be in cluster B? In each step we change the cluster annotation of the gene that creates the maximal reduction in cost. We continue these changes until no further reduction in cost is obtained.

Note that this algorithm is not guaranteed to result with 'best' clustering, as local minimum of cost is possible. Therefore, it is recommended to run the algorithm several times with different initial partitioning of genes into clusters. Also, note that this algorithm is good for creating homogenous clusters (i.e. the genes in the clusters will have similar profiles) but doesn't guarantee high separation between the clusters.

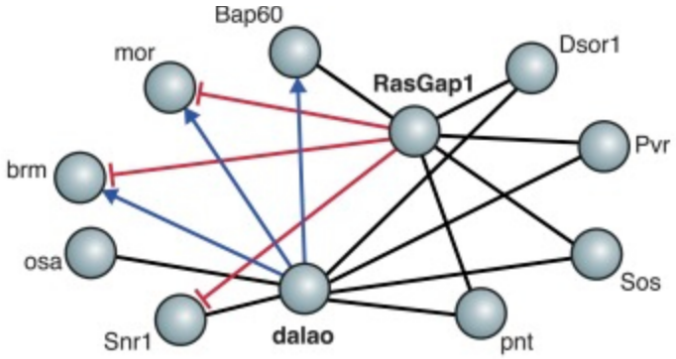
Functional analysis of genes

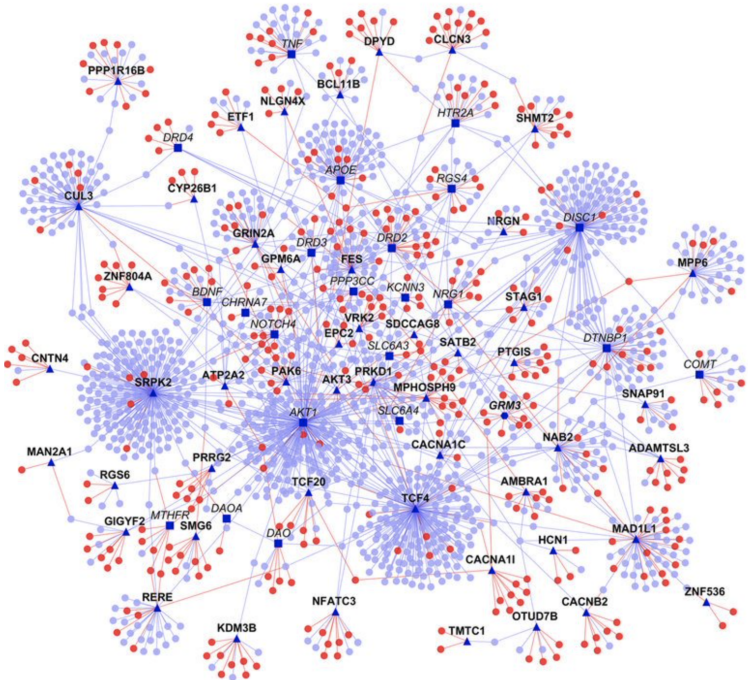
Once we have clusters of genes, our next goal is to detect the function of the genes in each cluster. For example, consider a cluster of genes that was detected as having a circadian pattern with a peak at 4pm in a coral colony (an example from the previous class). It will be interesting if we discover that these genes share a similar function -- for example, if most of them are chaperons that correct the 3D folding of proteins. Note that functional analysis is useful in many cases, and not only after clustering. For example, say that we have a tissue from a healthy patient and a tissue for a patient with a disease. We can detect differentially expressed genes, as discussed in the previous classes, and then apply functional analysis on the list of differentially expressed genes.

Functional analysis starts with knowledge regarding the genes, and especially about the protein level of genes. The first piece of knowledge is pathways, i.e., which proteins work together to produce a specific product. A classical example of a pathway is glycolysis - the breakdown of sugars to produce energy. As you can see in the figure (from Wikipedia), 10 proteins share this pathway. Overall, hundreds of pathways are known, and these can be easily accessed via KEGG for example.


(a)
Glucose
first priming reaction
1
ATP
ADP
Glucose 6-phosphate
2
Fructose 6-phosphate
second priming reaction
3
ATP
ADP
Fructose 1,6-bisphosphate
cleavage of 6-carbon sugar phosphate to two 3-carbon sugar phosphates
4
Glyceraldehyde 3-phosphate
+
Dihydroxyacetone phosphate
5
Preparatory phase
Phosphorylation of glucose and its conversion to glyceraldehyde 3-phosphate
1 Hexokinase
2 Phosphohexose isomerase
3 Phospho-fructokinase-1
4 Aldolase
5 Triose phosphate isomerase
(b)
Glyceraldehyde 3-phosphate (2)
oxidation and phosphorylation
6
2 Pi
2NAD+
2 NADH + H+
1,3-Bisphosphoglycerate (2)
first ATP-forming reaction (substrate-level phosphorylation)
7
2ADP
2 ATP
3-Phosphoglycerate (2)
8
2-Phosphoglycerate (2)
9
2H2O
Phosphoenolpyruvate (2)
second ATP-forming reaction (substrate-level phosphorylation)
10
2ADP
2 ATP
Pyruvate (2)
Payoff phase
Oxidative conversion of glyceraldehyde 3-phosphate to pyruvate and the coupled formation of ATP and NADH
6 Glyceraldehyde 3-phosphate dehydrogenase
7 Phospho-glycerate kinase
8 Phospho-glycerate mutase
9 Enolase
10 Pyruvate kinase

Another source of information regarding genes and proteins are protein-protein interactions or PPi in short. These datasets are a result of large scale experiments aimed to detect which proteins interact. We assume that proteins interact if they are working together, i.e. share a function. Therefore, PPi can be useful in trying to detect the function of a gene (examples below from Wikipedia).

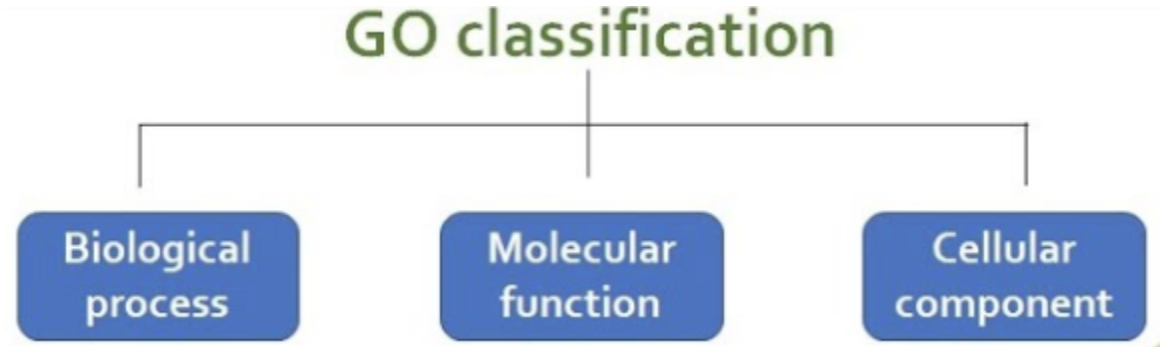




Co-localization is another approach to detect proteins that might work together. The idea is to detect proteins that share the same subcellular location (for example, the nucleus or the cell membrane). An extension of this idea is co-expression. With co-expression we examine the RNA level and not the protein-level, and ask if several genes have similar expression patterns across cell types and tissues. This might indicate that these genes have a similar function.

A main resource for functional knowledge about genes is the Gene Ontology (GO), which gives information about genes in three different levels: 1) Molecular Function. For example, is the gene a transcription factor or a kinase?. 2) Biological Process. For example, does this gene take

part in the glycolysis or in the circadian clock?. 3) Cellular Component. For example, does the protein of this gene is usually in the cell membrane or in the nucleus?.



Statistical significance of functional analysis

One of the main uses of GO, as explained here, is to perform enrichment analysis on gene sets. For example, given a set of genes that are up-regulated under certain conditions, or a cluster of genes, an enrichment analysis will find which GO terms are over-represented (or under-represented) using annotations for that gene set. Functional enrichment analysis can be performed via the GO website. Other tools are useful as well, for example DAVID and GeneMANIA.