Connecting Nutrition, Health and Environment

Wiktor Jurkowski, Perla Troncoso Rey, Earlham Institute

25 - 26 January 2017

Contents

List of Figures

List of Tables

1 Introduction

In this practice, we will look at how to explore gene expression data (which could be extracted from microarray or RNA sequencing data). In the first part of this practical session we will see general techniques to explore the patterns or structure of the data using Principal Component Analysis, PCA, and Hierarchical Clustering. We will then look into compiling an interaction network using an online resource and how to visualise the network using Cytoscape [].

In the second part, we will look at ways to rank genes (and/or metabolites) using approaches based on logistic regression. We will then perform pathway analysis using those rankings.

We will use a publicly available data from a study on fatty liver disease of obese and lean human subjects [].

Part I

Part I

1.1 Exploring the expression data

One could obtain gene expression from microarrays or RNA sequencing data. It is not within the scope of this session to look at the details of obtaining quantifying the expression of genes from microarrays or RNA sequencing data but instead we will start our analysis assuming that expression data has been quantified. The gene expression data is normally stored in tabular file, representing a matrix where the columns are the samples or experiments, and the rows represent the genes.

Example of expression data is shown in Figure 1. The table shows the expression of six genes in four different experiments or samples. This is, gene A has expression of 0.1 for sample 1, 0.8 for sample 2, 0.3 for sample 3, and so forth.

Expression data can be obtained using different algorithms. One of the most well know are TopHat and Cufflinks protocol for the analysis of RNA sequencing data, which includes quantification of gene expression. Table 1 shows an example of the output provided by cufflinks with the estimated gene-level expression values. Cufflinks uses the notation "XLOC_*numeric_sequence*" to identify a gene.

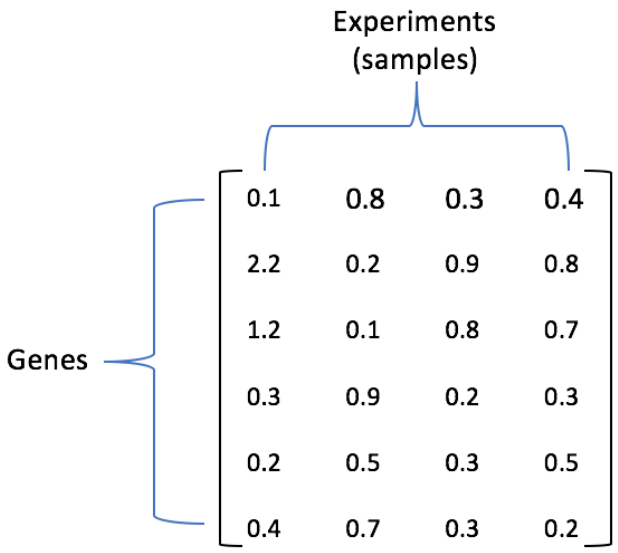


Figure 1: Example of expression data with samples across the columns and individual genes down the rows.

1.1.1 Principal Component Analysis

Principal Component Analysis, commonly known as PCA, is a mathematical technique that is used to explore data, specially high-dimensional data, to extract the most important trends in the data.

When thinking of gene expression data, high dimensionality comes from the large number of dimensions of the data. This is, the result of each experiment can be thought as a kind of space, where each each feature is a coordinate in the space. There are typically thousands of genes (dimensions) and the structure of pattern in the data extends to all the dimensions.

How PCA works

The mean represents the average of the values in the data:

$$\bar{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

The variance provides the the spread of the data:

$$\mathrm{Var}(\mathbf{X}) = \sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{\mathbf{X}})^2 \tag{2}$$

Table 1: Example of the output provided by cufflinks for the quantification of gene expression from RNA sequencing data.

tracking_id	sample1	sample2	sample3	sample4	sample5	sample6	sample7	sample8
XLOC_000001	35.1077	50.9662	78.7724	35.4736	69.6067	63.9241	57.7967	61.4227
XLOC_000002	49.7359	64.6178	46.8884	74.617	66.0371	42.9654	645.65	64.8351
XLOC_000003	0	0	0.937767	0	0	0	0	0
XLOC_000004	89.7196	85.5504	185.678	74.617	142.783	168.718	172.63	167.206
XLOC_000005	12.6778	39.1347	158.483	22.0181	28.5566	45.0613	15.9701	50.0481
XLOC_000006	10.7273	9.1011	10.3154	13.4555	7.13915	6.28762	7.60483	12.512
XLOC_000007	0	0	0.937767	0	0	0	0	0
XLOC_000008	55.5871	37.3145	86.2746	66.0544	66.9295	53.4448	54.7548	75.0722
XLOC_000009	37.0581	16.382	24.3819	24.4646	38.3729	15.7191	24.3355	50.0481
XLOC_000010	812.352	483.269	696.761	748.616	1094.97	521.873	675.309	741.622
XLOC_000011	0	0	0	0	0	1.04657	0.760483	1.13746

For example, 3 show two distribution with the same mean but different variance.

(i.e. the data points are at the same location but with a different strength. So the third statistic we'll need is the covariance)

The covariance represents the degree of co-dependence of two variables, i.e., it measures the co-dependency of two variables, given by:

$$\mathrm{Cov}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{\mathbf{X}})(y_i - \bar{\mathbf{Y}}) \tag{3}$$

Increases with increasing co-dependency and variance. Just as the variance measures the degree to which a set of data varies, the co-variance is a measure of the way two sets of data vary together.

$$\mathrm{Cov}(\mathbf{X}, \mathbf{X}) = \mathrm{Var}(\mathbf{X}) \tag{4}$$

The covariance also increases in magnitude as the variance of each of the two datasets increases.

Coordinate transformations

In a two dimensional space described by coordinates, a point in space is described by X and Y such that $\mathbf{v} = [x_1, y_1]$. For example, the vector $v_1 = [1\ 2]^T$ represents the point:

An alternative coordinate systems described by the coordinates X′ and Y′, has a different column vector describing the same point $\mathbf{v}' = [x_1', y_1']$.

The two coordinate systems are $T\mathbf{v} = \mathbf{v}'$, related to the orthogonal transform matrix T (an orthogonal matrix is the kind of matrix which performs rotated-axis coordinate transforms).

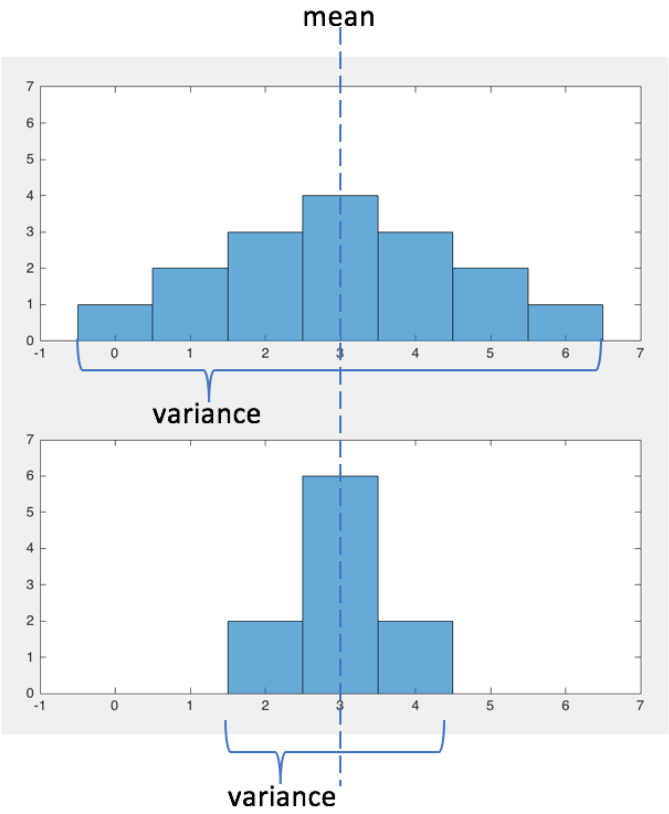


Figure 2: Two distributions with the same mean but different variance.

We can make a new coordinate system by using a transformation matrix T, which relates the two coordinates vectors by matrix multiplication. There are many types of transformations but we are particularly interested in transformations which rotate the coordinate axis. These are performed by matrices which have the property called orthogonality.

Eigenvalues and Eigenvectors

When a transformation matrix maps a vector to a multiple of itself, then the vector is called an Eigenvector. The amount by which the vector is multiplied (stretched) is the associated Eigenvalue:

$$Tx = \lambda x \tag{5}$$

λ are the Eigenvalues and x are the Eigenvectors. A matrix formed from the Eigenvectors placed in the columns is orthogonal.

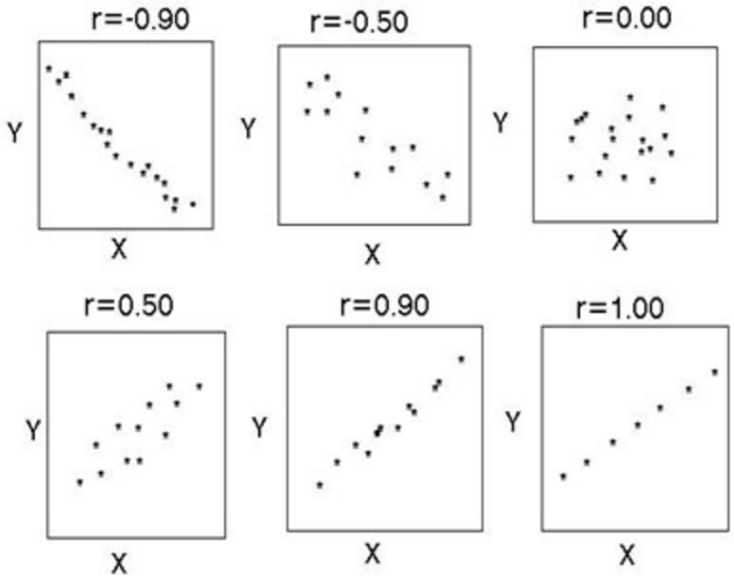


Figure 3: Examples of correlation

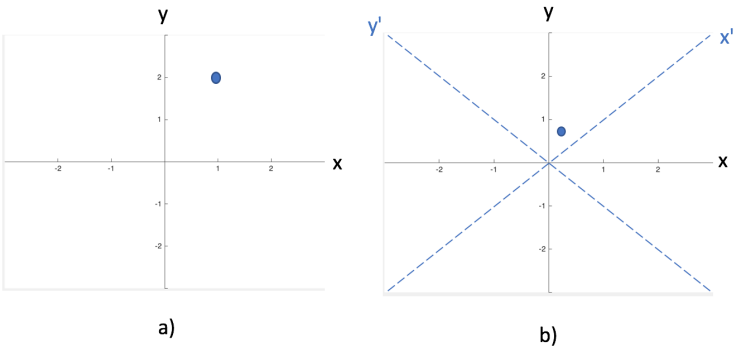


Figure 4: Example of a coordinate transform

In general terms, PCA uses covariants to encode the structure in the data and then eigenvectors to devise a new set of coordinates that best reveals the structure by finding the appropriate set of directions. One result from linear algebra is that if the eigenvectors are placed next to each other then the result is an orthogonal matrix that performs a coordinate transformation. This is central for PCA.

Example:

The matrix: $\begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix}$ has eigenvalues 4 and -1. and the eigenvectors $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 3 \\ -2 \end{pmatrix}$

such that

$$\begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 4 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 3 \\ -2 \end{pmatrix} = -1 \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

In summary, PCA benefits are:

- A powerful tool to visualise high dimensional data
- Shows quantified difference among observations
- Used to assess data quality and discover relationships between data points

Some tools to perform PCA include:

- MATLAB
- R

Example 1: PCA for the expression of two genes

We will perform PCA on the example data which represents several measurements of the expression of two genes, x and y.

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.1	0.9

Create a matrix of points in 2-d space (gene expression data)

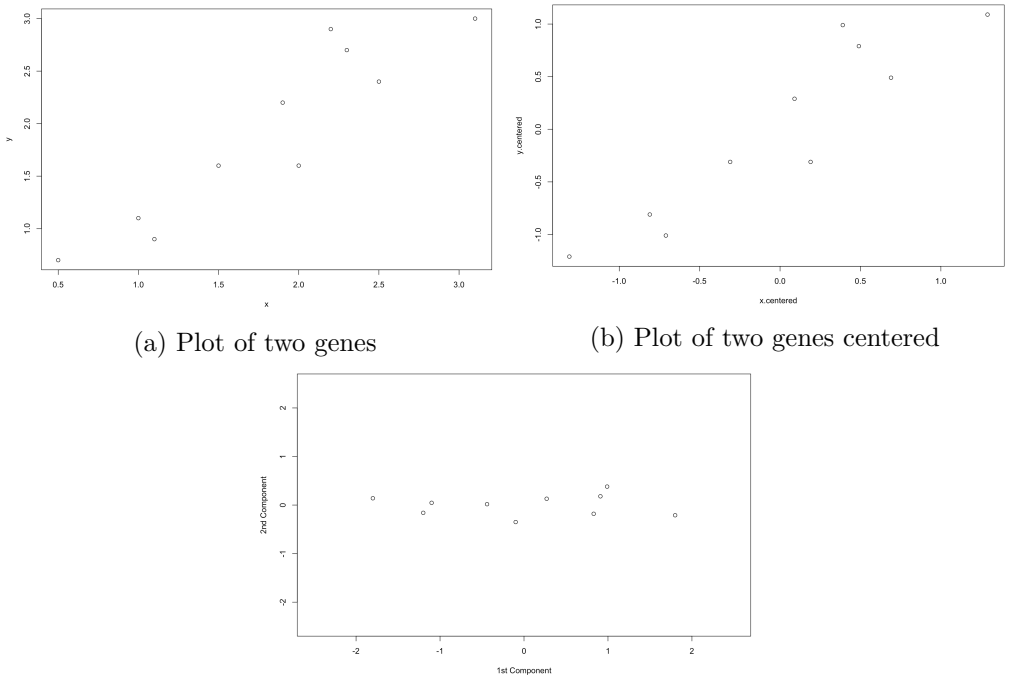
```
#n number of samples
name_gene_1 <- c('sample_1', 'sample_2', ..., 'sample_n')
#m number of genes
name_gene_2 <- c(gene_1, gene_2, ..., gene_m)
samples <- c(sample1, ..., sampleN)

ExpData <- data.frame(gene1 = name_gene_1, ..., gene_m = gene_name_m)
```

We plot these two genes in figure 5a

Trends are already apparent because data is simple but this is not usually the case.

Next: Analysis statistical

PCA: It is best to first have centered the data with mean zero. (1. Calculate the mean of each of the two variables, 2. Substract the means, Data is centered)



(a) Plot of two genes

(b) Plot of two genes centered

(c) Plot of two genes in the new coordinates

Figure 5: Plot two genes expression in (a) and centered in (b) and in the new coordiantes in (c)

Now, let us calculate the covariance matrix. Covariance matrix for two variables:

$$\begin{bmatrix} \text{Cov(x,x)} & \text{Cov(x,y)} \\ \text{Cov(y,x)} & \text{Cov(y,y)} \end{bmatrix}$$

Covariance matrix for our data:

$$\begin{bmatrix} 0.016 & 0.615 \\ 0.615 & 0.716 \end{bmatrix}$$

Calculate the Eigenvalues of this matrix: 1.284 and 0.0490. Eigenvalues gives the relative variance of the data in the direction defined by the Eigenvectors. From the values we can inferred that most variation is in one direction.

```
eigen(covariance_matrix)
```

The corresponding eigenvector are then placed in a matrix in descending order of eigenvalue:

$$\begin{bmatrix} 0.6778734 & -0.7351787 \\ 0.7351787 & 0.6778734 \end{bmatrix}$$

The transform of this will perform the coordinate transformation:

$$W^T = \begin{bmatrix} 0.6778734 & 0.7351787 \\ -0.7351787 & 0.6778734 \end{bmatrix}$$

This is an orthogonal matrix which performs a rotated-axis coordinate transformation. We can transform our data matrix so that the data is represented in the new coordinates:

$$D_{PCA} = W^T D$$

$$D_{PCA} = \begin{bmatrix} 0.6778734 & 0.7351787 \\ -0.7351787 & 0.6778734 \end{bmatrix} \begin{bmatrix} 0.69 & -1.31 & 0.39 & 0.09 & 1.29 & 0.49 & 0.19 & -0.81 & -0.31 & -0.71 \\ 0.49 & -1.21 & 0.99 & 0.29 & 1.09 & 0.79 & -0.31 & -0.81 & -0.31 & -1.01 \end{bmatrix}$$

$$= \begin{bmatrix} 0.83 & -1.8 & 0.99 & 0.27 & 1.8 & 0.91 & -0.099 & -1.1 & -0.44 & -1.2 \\ -0.18 & 0.14 & 0.38 & 0.13 & -0.21 & 0.18 & -0.35 & 0.046 & 0.018 & -0.16 \end{bmatrix}$$

The data are plotted in the new coordinate axis in figure 6, where each coordinate is called principal component. The first coordinate aligns with the direction in the expression space where has the most variation. Subsequent coordinates would align with directions with descending degrees of variation. This is why we are careful to order according to the size of the eigenvalues. Thus, PCA is capturing as much variation in the first component as possible, then the same for the second coordinate, and so on. In the case of our data, all the meaningful variation sees to have been captured with the first coordinate, or the first principal component. Specially compared to the second component which would seem to be random scatter. So we have reduced the dimensionality of our data from two to one. In cases when dealing with thousands of genes, PCA might be able to capture most of the variation of the data in with only two or three principal components. Thus making it easier to visualise it, which is one of the main motivations of performing PCA.

Example 2: PCA using example data

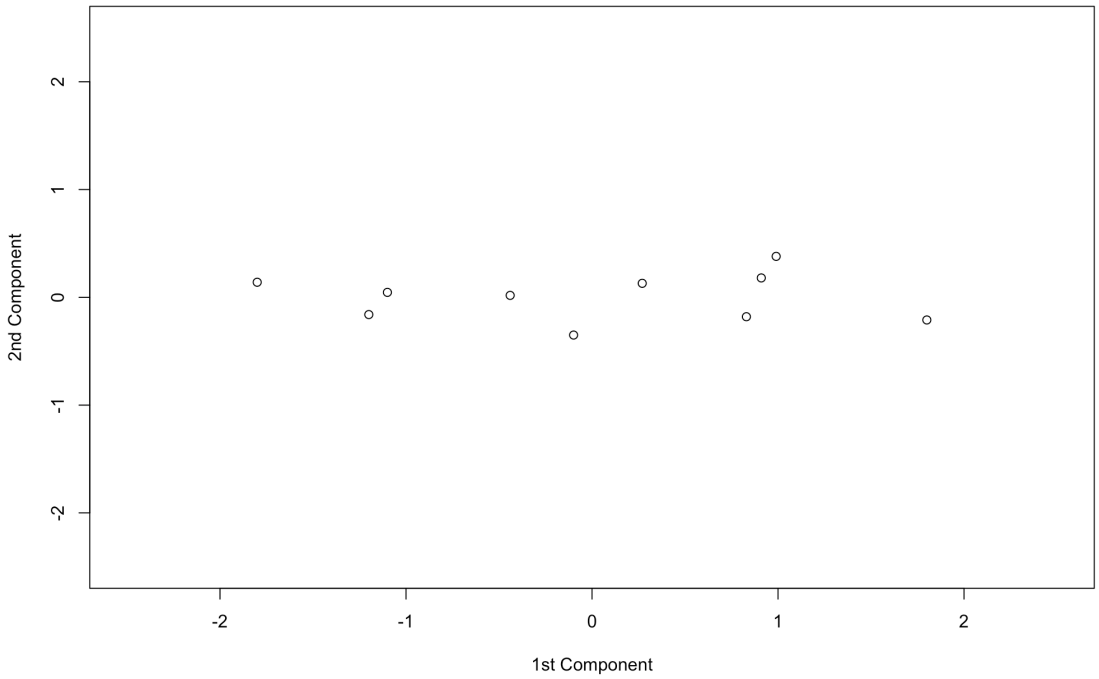


Figure 6: Plot of two genes in the new coordinates

Create a matrix with the gene expression from Table 1 by entering the first three genes for samples 1, 2, 3 and 4.

```
samples <- c('sample_1', 'sample_2', ..., 'sample_n')
genes <- c(gene_1, gene_2, ..., gene_m) #numbers
s1 <- c(gene1_sample1, ..., gene_m_sample_1)
...
sn <- c(gene1_sample_n, gene_m_sample_n)

ExpData <- data.frame(s1, ..., sn)
colnames(ExpData) <- samples
rownames(ExpData) <- genes
```

PCA Plot of gene expression data

Each dot is a gene expression from a sample in each category (class) from a patient, and is coloured by its sub-type

The three axis are the three principal components and the numbers represent the percentage of variance that is captured by each component

The first component captures the most variance, whereas the second and third capture only a small percentage

Dots of the same subtype tend to cluster together which means that samples of the same subtype have similar transcritption profiles.

The distance on the dots on each axis should not be treated equally (as each component captures a different percentage of variance). difference on the first component should be taken into more consideration.

Random data

Simulated gene expression data by random numbers

This is how a random dataset would look like in a PCA plot:

- dots of different classes mix all together

- the first three components capture almost equal and small variance

- from the plot once would conclude that the different subtypes are not distinct from each other or that subtype has no influence on tumor cell transcriptome

DATA PREPARATION

- Use microarray gene expression data as an example

- Gene expression data is usually stored in a tab delimited text file. The extension of such files could be .csv, .soft, .xls(x), etc. Use Excell of Sublimetext to open and preview the file.

- Gene expression values must be normalised before PCA plotting.

PCA

- We need to transpose the matrix because the function requires the rows of the input matrix to be observations and the columns variable, which means rows to be the gene expression profiles and columns to be the genes.

- There are three outputs to the function:

1. The first output is the coefficient matrix (not used here!)

2. The second output is the scores, which are the transformed coordinates by PCA.

3. The third output, pca variance, stores how much variance each component captures.

Looking into detail on the outputs:

- The first several components capture most variance of the data.

- The score matrix has the same arrangement as expression matrix, which are rows as gene expression profiles and columns are genes. We pick the first three columns, namely the first three components. The first component will be the x-axis, the second to be the y-axis and the third component to be the z-axis.

Running PCA

PCA Plot using MultiPEN

Parameters:

gene expression data

groups

In summary, PCA is a method of revealing underling trends in large amounts of data. PCA reduces high dimensional data to just a few principal components which hopefully capture most of the variation of the data and allows inferring meaningful structure.

A new coordinate system is constructed by rotating the axes (each representing a gene). The first new coordinate, or first principal component, is the direction in which the data varies most, then the second component, and so on. PCA allows to select a few new variables which contain most of the variation of the data which can also be visualised.

1.1.2 Hierarchical Clustering

Hierarchical Clustering is another way to visualise high dimensional data. It clusters observations by distance and builds a hierarchical structure It gives more detailed information of the differences among clusters, for example, what genes contributes the most to the differences between two clusters.

Figure: Example of clustergram The clustergram is made of a heat map in the middle and dendograms in the left and top, with row and column labels on the right and bottom (depending on the number of genes and samples), and a scale bar. Each column is a sample expression profile, and each row represents a gene. The colours suggests relative expression values, where red indicates high expression values and blue indicates low expression values. Ideally, samples of the same type will cluster together

Figure: Example of clustergram of simulated random numbers No distinct clusters are observed. High and low expression values (red and blue colours) are mixed all together and the sample of the same type are mixed.

Hierarchical clustering uses a distance metric (typically Euclidean but could be correlation, Hamming distance, etc.) between each pair of genes to create a hierarchical tree-like structure of the data. Then it uses a linkage function to calculate the distance between clusters. For more details please see [**?**].

EXAMPLE USING MULTIPEN

1.2 Interaction Network

1.2.1 Compile the network with Cytoscape

Part II

Part II

2 Ranking genes (Feature Selection)

Feature selection with MultiPEN from expression leves and network

3 Pathway Analysis

Objective

Steps

Input data A list of ranked genes. For this session we will use the rankings from feature selection in ExampleOutputsMultiPENRankings_lambda0.0001.txt.

To run the R script type in the terminal:

Rscript enrichmentGO.R ../ExampleOutputs/MultiPENRankings_lambda0.0001.txt output/

References

[1] MATLAB and MathWorks. Compute hierarchical clustering.