

# Myself

---

**Name:** Paolo  
**Surname:** Simonetto  
**Born on:** November 23, 1982  
**Home town:** Bassano del Grappa – Italy  
**Languages:** Italian (properly), English (improperly)

## Undergraduate studies:

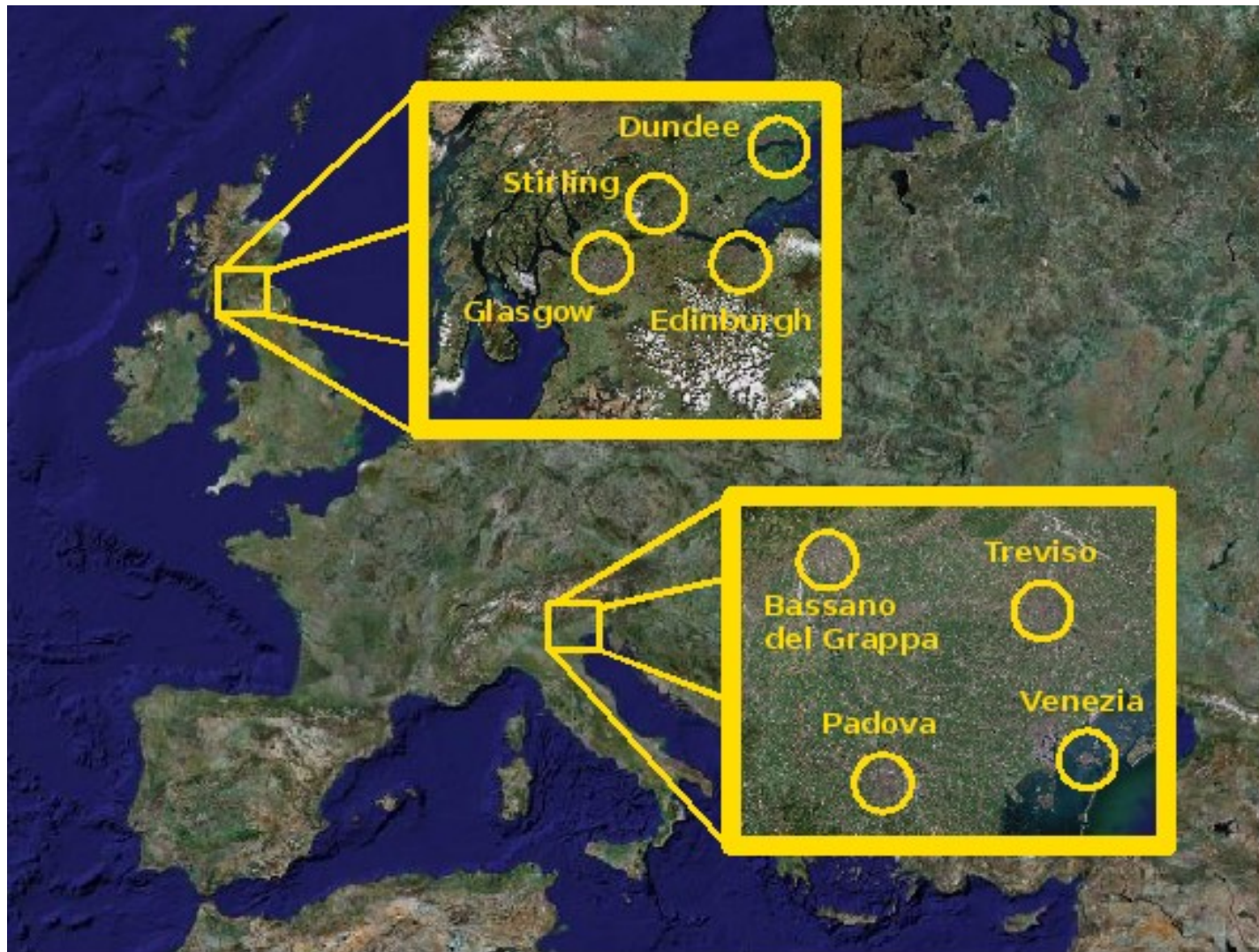
Informatics engineering at Padova University

## International exchanges:

2005-2006 Erasmus at Glasgow University

2007 Master thesis at BRC, Glasgow University

# Some geography



---

# Framework for computation and evaluation of metabolic network modules, application to metabolomic data

Master thesis in Informatics Engineering  
March-October 2007

Candidate: Paolo Simonetto  
Supervisors: Dr. Fabien Jourdan  
Prof. David Gilbert  
Prof. Carlo Ferrari

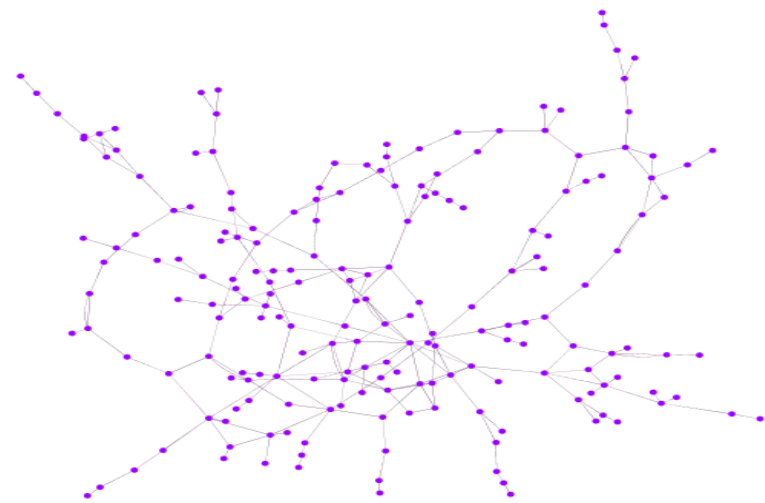
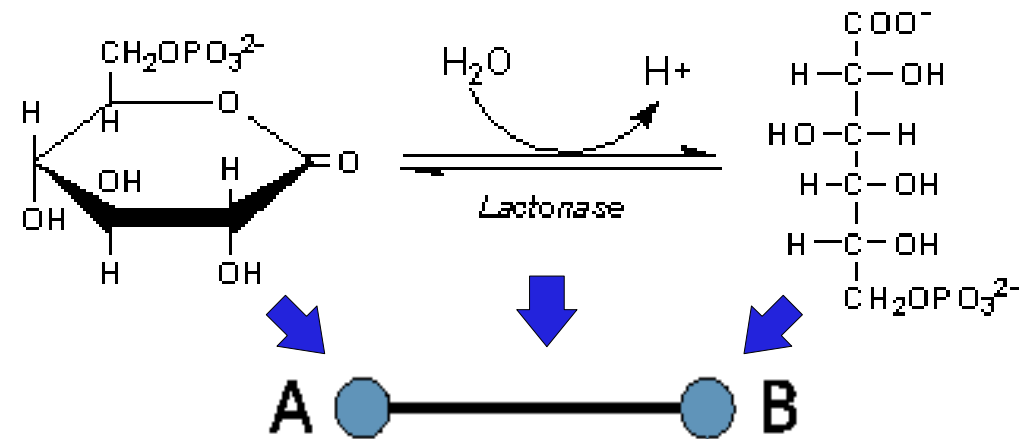
Bioinformatics Research Centre  
University of Glasgow  
Dip. di ingegneria dell'informazione  
Università di Padova

# Metabolism and metabolic networks

The **metabolism** describes the set of chemical reactions that occur in an organism.

The reactions involved can be represented as transformations of **metabolites**.

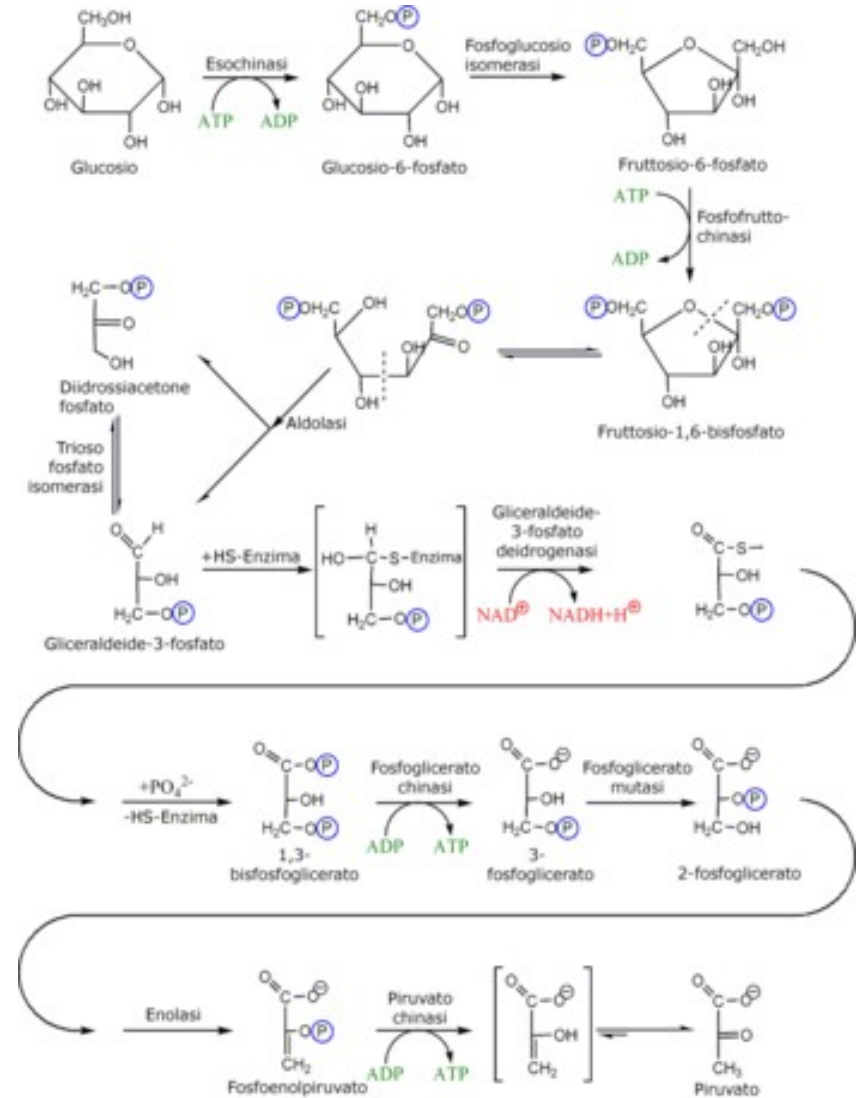
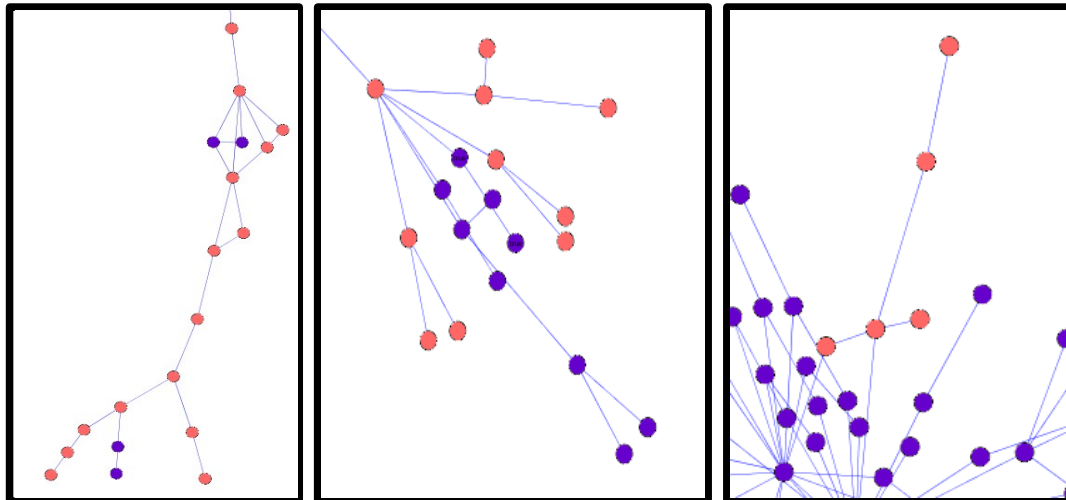
The interactions between the reactions define a complex structure called **metabolic network**.



# Functional modules (pathways)

In the metabolic networks it is possible to identify functional modules, also called **pathways**.

Pathways collect all the reactions that co-operate in order to realize the related function.

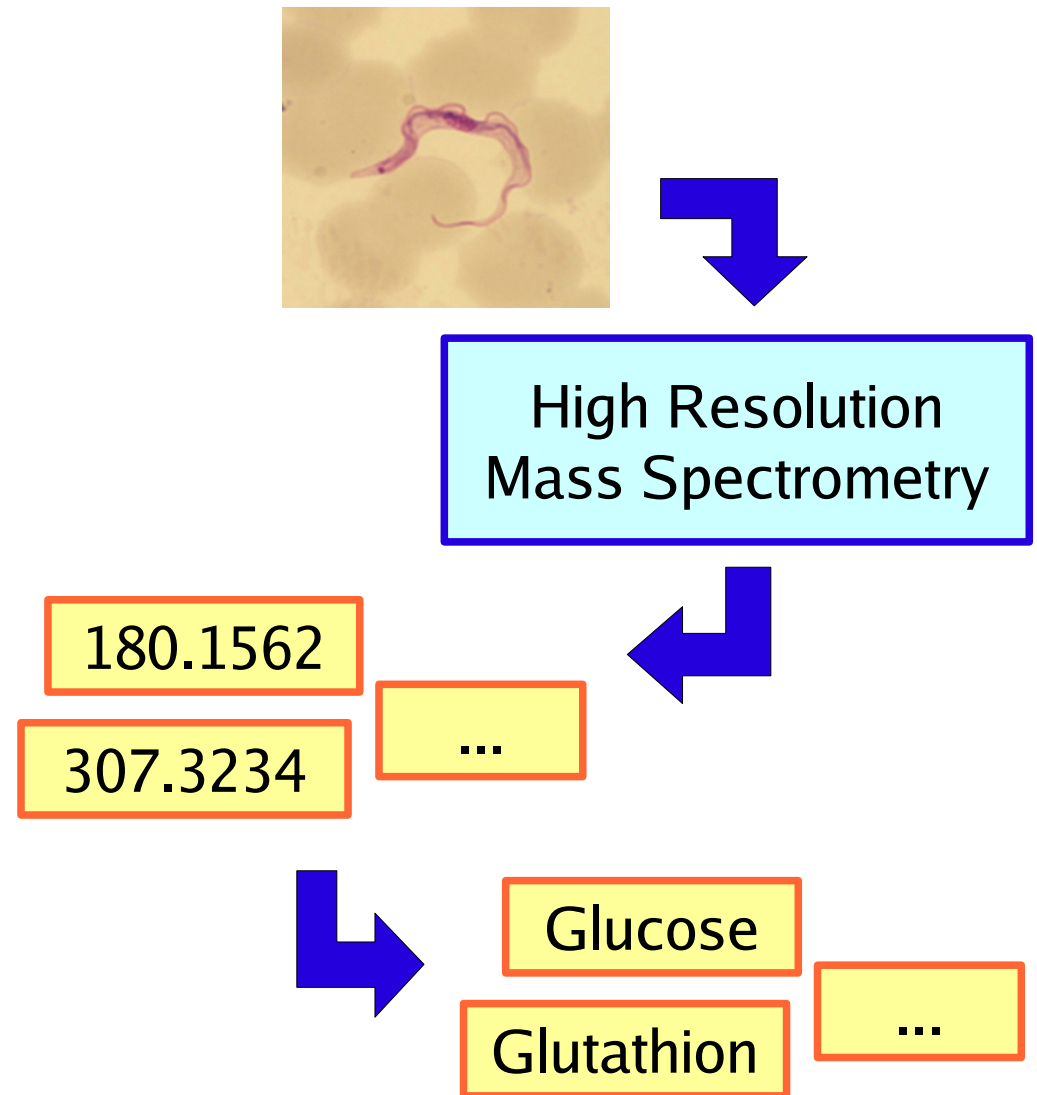


# Metabolomics

**Metabolomics** is the study of the chemical fingerprints that the cellular processes leave behind.

It describes techniques for detecting and analysing the metabolites present in an organism.

By measuring the mass of the metabolites with high resolution methods, we can deduce their chemical formula.



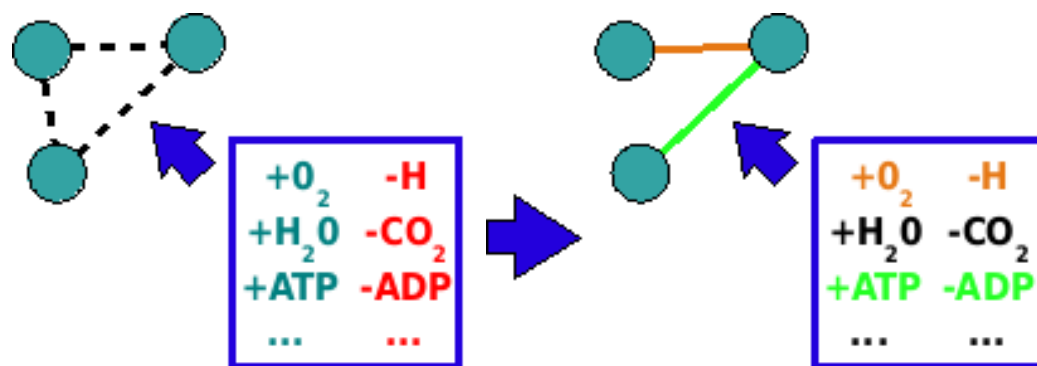
# Metabolomic networks

We now have:

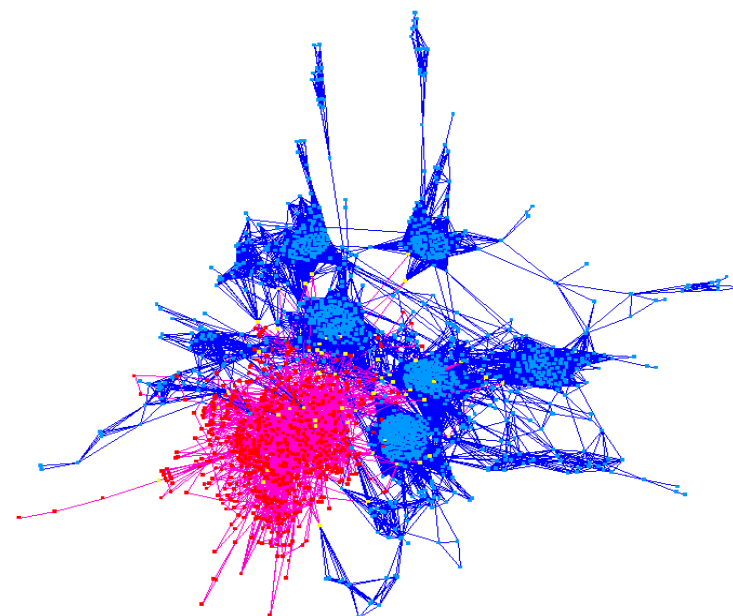
- the nodes of the graph
- ✗ the edges of the graph

We can detect the edges:

- through the data collected in databases like KEGG.
- with techniques “*ab initio*”.



We then obtain a  
**metabolomic network.**



Nodes present: 2085  
Edges present: 42397

# Graph clustering for detecting pathways

We only have topological information about the metabolomic network.

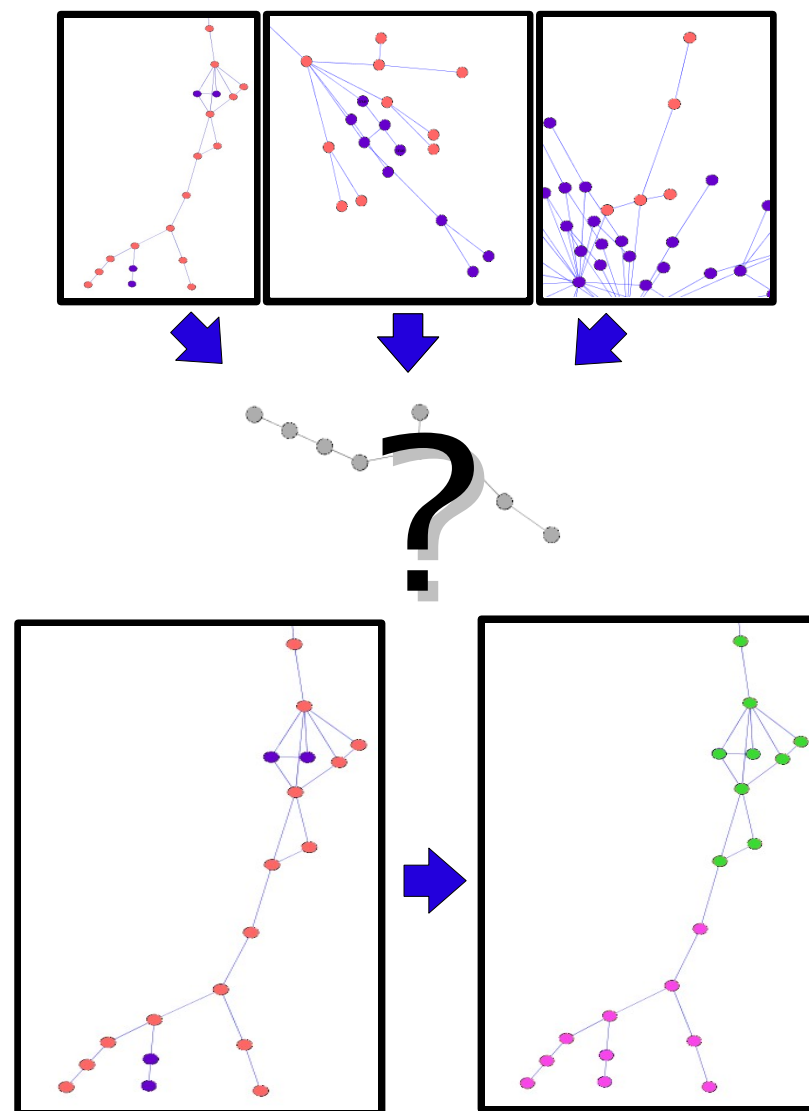
As it is not possible to define a structure generally valid, we will refer just to the connectivity of the network.

Being the density of a graph:

$$d(G) = \frac{|E|}{|V|(|V|-1)/2}, \quad G=(V, E)$$

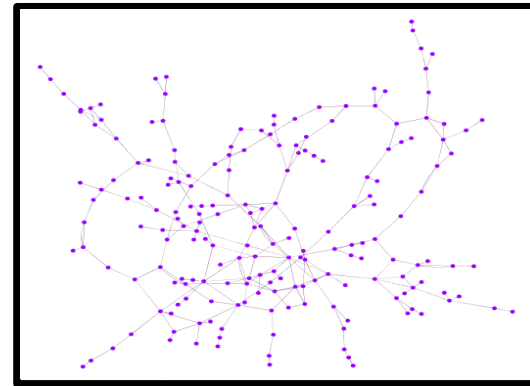
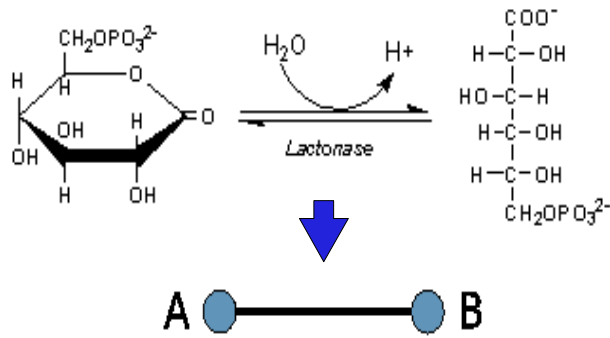
we will cluster the nodes in order to obtain dense subsets.

Clusters = pathways?

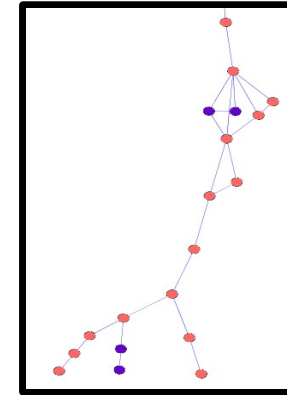


# Summarising

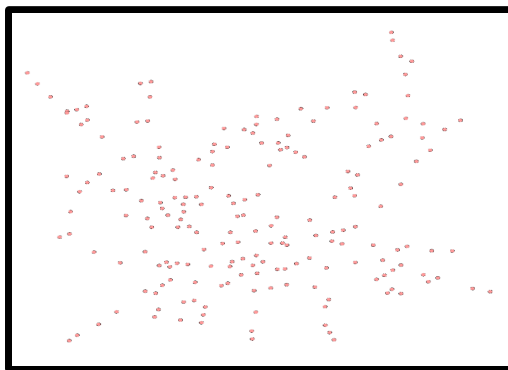
## Network representation of the metabolism



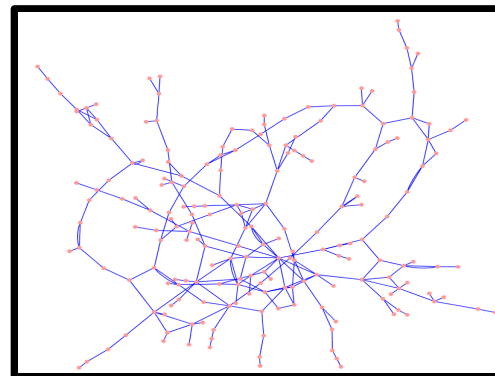
## Pathways



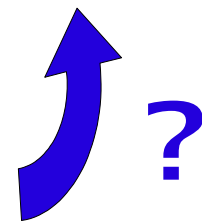
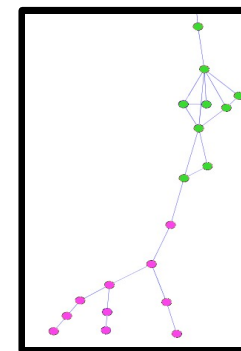
## Metabolites identification



## Metabolomic net construction



## Clustering

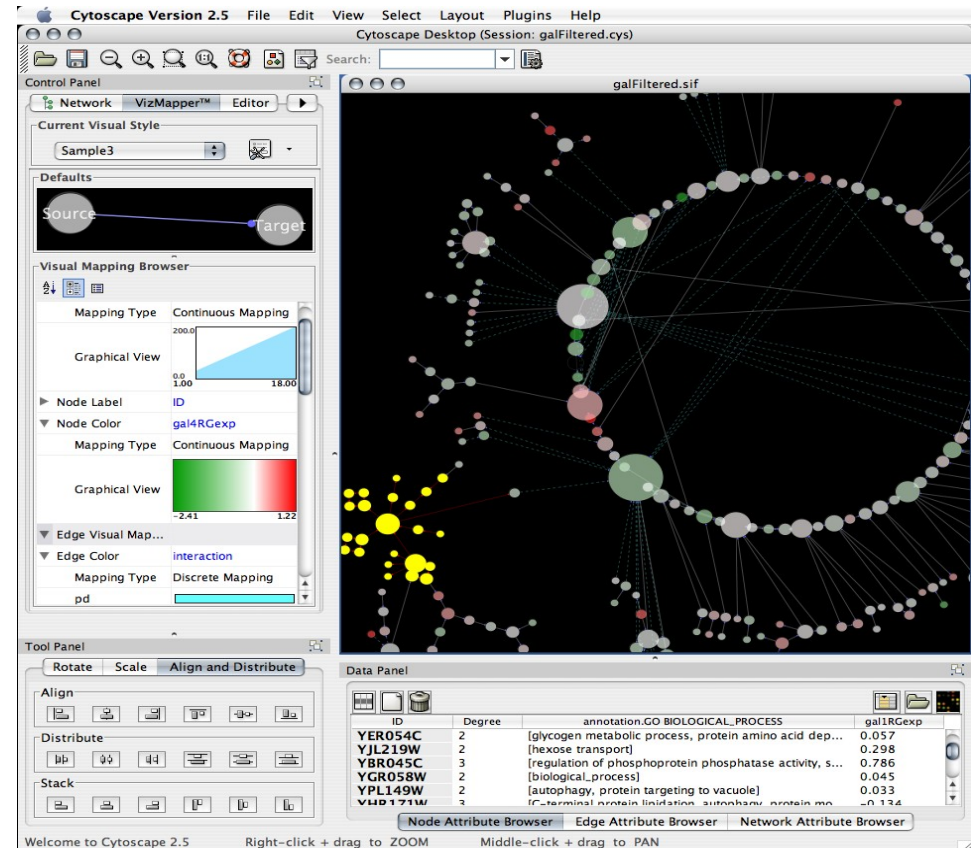


# Cytoscape

**Cytoscape** is a software for visualising and analysing biologic networks, in particular protein-protein interaction networks.

It allows to navigate multiple networks, composed of 100.000+ nodes/edges.

- opensource
- cross platform (Java)
- extendible (plug-in)



[www.cytoscape.org](http://www.cytoscape.org)

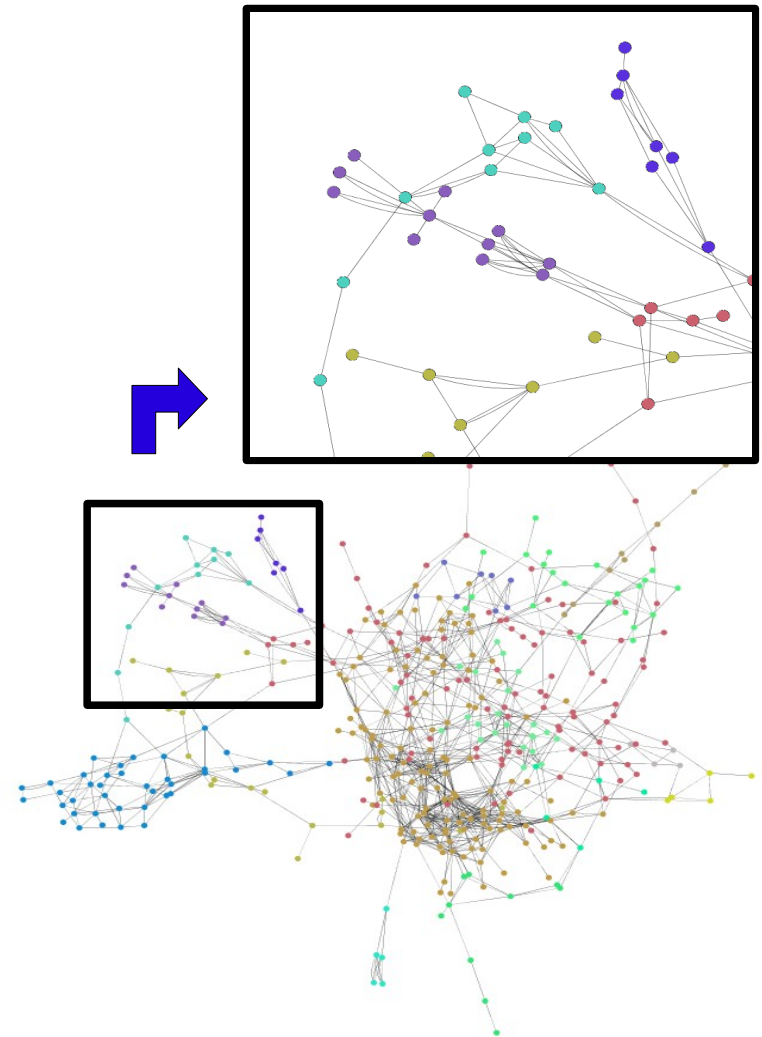
# ClustPlugin

**ClustPlugin** is a Cytoscape plugin that provides a framework for clustering, in particular on metabolomic networks.

It implements several clustering algorithms, metrics and quality measures.

It allows the visualisation of the clusters detected and their comparison in different networks.

It grants the possibility to easily develop new features.



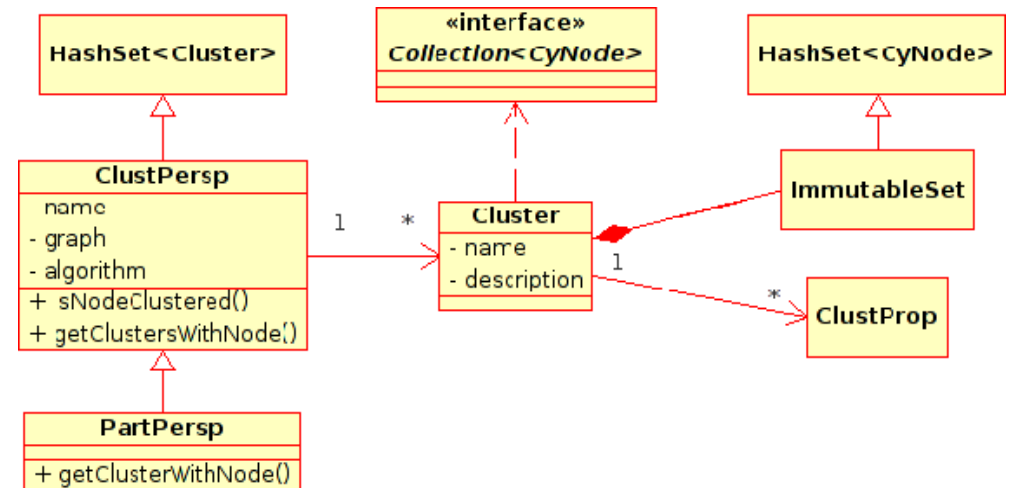
# Clustering data structures

ClustPlugin defines classes for storing clustering data.

**Clustering perspective** represent the global clustering view:

- collects clusters
- provides the inverse map node clusters

**Partitioning perspective** extends ClustPersp to support partitioning.



**Cluster** class:

- collect nodes
- use the standard Java collection framework
- allow the definition of **cluster properties**

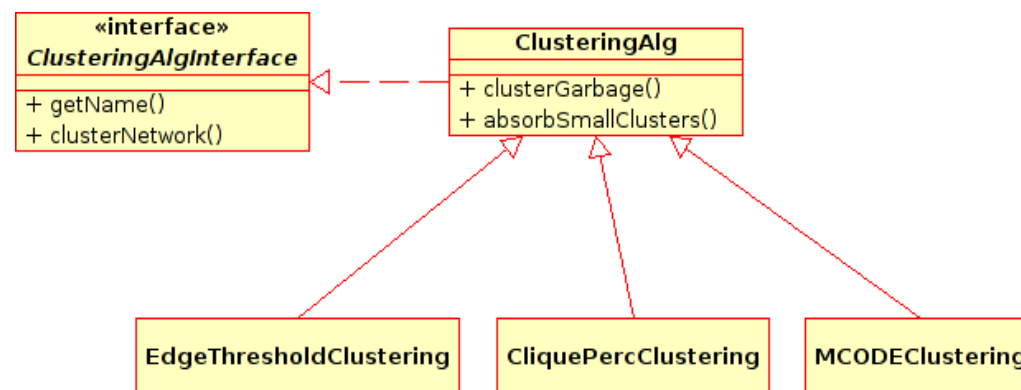
# Algorithm integration

ClustPlugin allow easily to integrate new clustering algorithms.

The **clustering algorithm interface** defines how a new algorithm must be.

The **clustering algorithm class** is an abstract class that defines some general post-processing methods.

The implemented algorithms are all concrete extensions of this abstract class.



ClustPlugin also provides a mechanism for handling the parameters of the algorithms, once they are opportunely listed in the algorithm class.

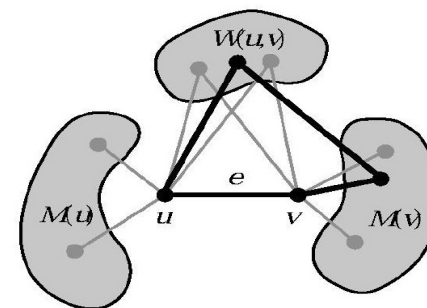
# Implemented algorithms – Strength-MQ (1)

## Strength-MQ (Edge threshold)

Networks with “small world” structure.

Auber, Chiricota, Jourdan, Melançon. Multiscale visualization of small world networks.  
INFOVIS. IEEE Computer Society, 2003.

The **Strength metric** evaluates the hardness of an edge based on the connectivity of the neighbourhood of its nodes.



$$\frac{1}{n} \sum_{i=1}^n d(C_i) - \frac{1}{n(n-1)/2} \sum_{i < j} s(C_i, C_j)$$

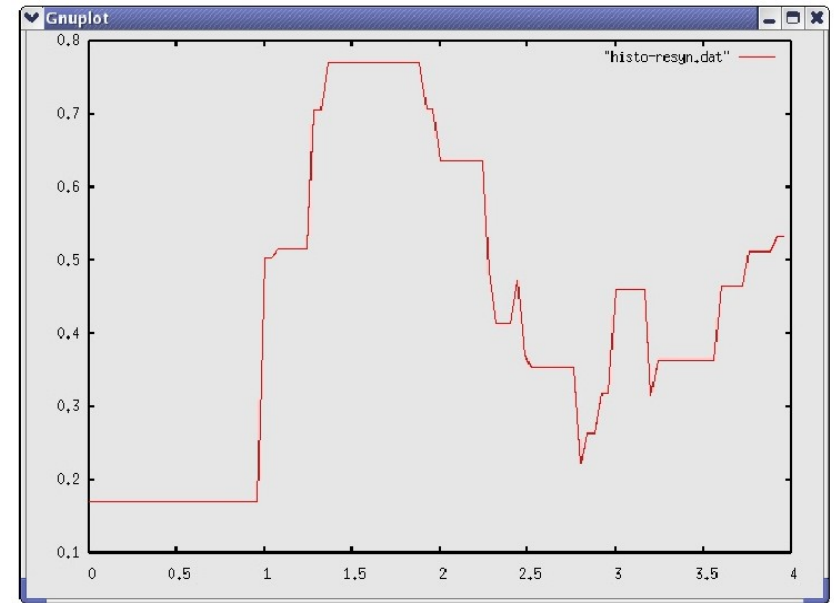
The **MQ measure** evaluates the quality of the clustering comparing the density of the clusters to their interconnection ratio.

The implementation of the algorithm as **Edge threshold** allows to use different weighting metrics and evaluation measures.

# Implemented algorithms – Strength-MQ (2)

The algorithm:

1. calculate the strength metric for each edge.
2. define a set of thresholds to test.
3. for each threshold in the set:
  - 3a. cuts the edges with metric lower than the threshold value.
  - 3b. clusters together the resulting connected components.
  - 3c. evaluate the clustering calculating the MQ measure.
4. apply the best clustering obtained.



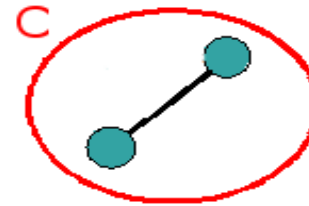
The selection of the thresholds to test can be improved once the relation between the threshold value and MQ will be more clear.

# Implemented algorithms – Strength-MQ (3)

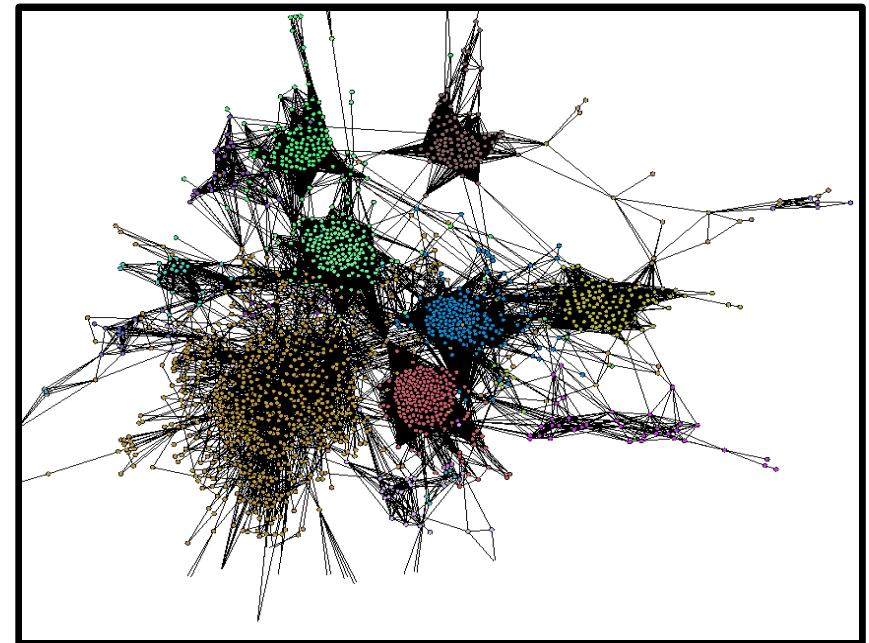
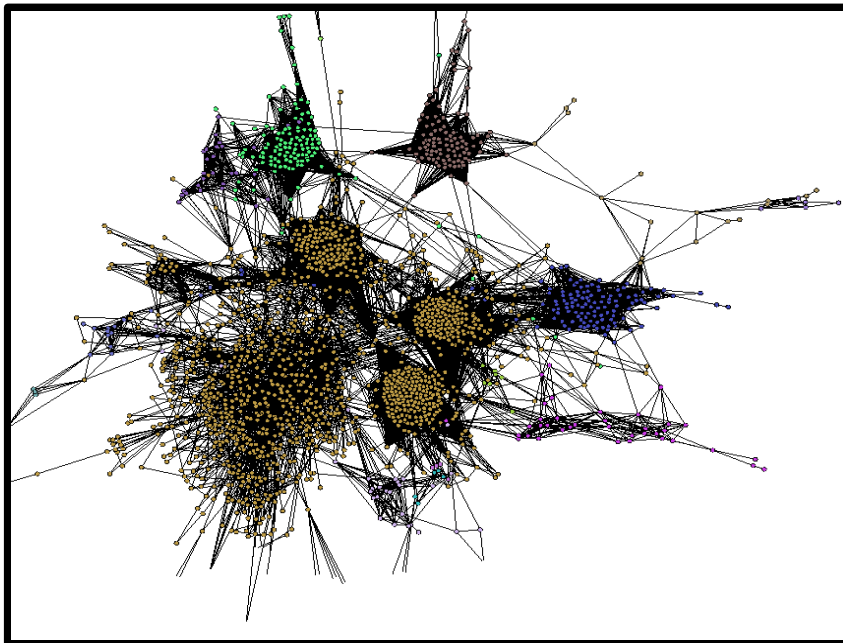
The original MQ formula resulted to be too affected by small clusters.

We replaced the normal average of the density with an average weighted on the number of nodes.

$$\frac{1}{n} \sum_{i=1}^n d(C_i) \rightarrow \frac{1}{n(n-1)/2} \sum_{i < j} s(C_i, C_j)$$



$$C = K_2$$
$$d(C) = d(K_2) = 1$$

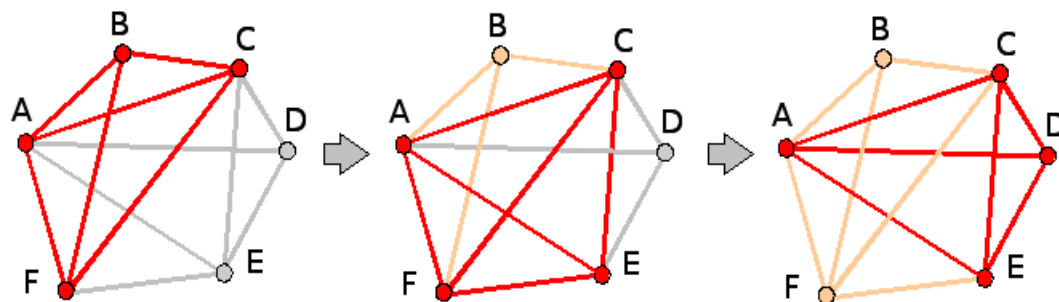


# Implemented algorithms – Clique percolation (1)

## Clique percolation Social networks.

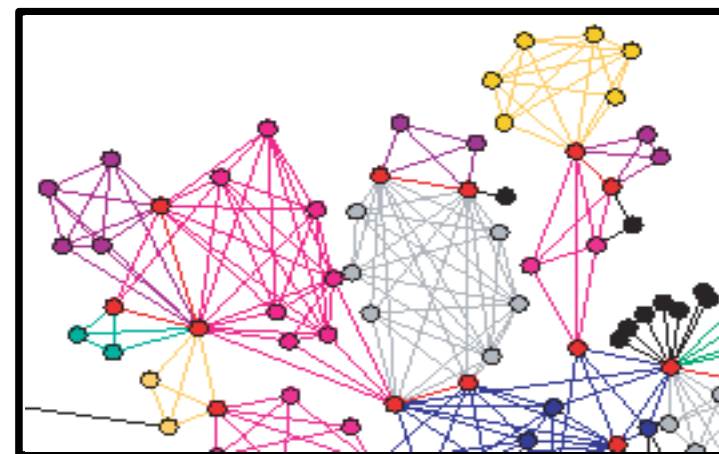
Palla, Barabassi, Vicsek. Quantify social groups evolution.  
Nature, April 2007.

Given two  $k$ -clique, they are defined **adjacent** if they share  $k-1$  nodes, and **reachable** if they are in a chain of adjacent  $k$ -cliques.



A  **$k$ -clique percolation cluster** collects the nodes in all the  $k$ -cliques reachable each other.

We don't obtain a proper clustering, as clusters can overlap (red nodes) and nodes can be unclustered (black ones).

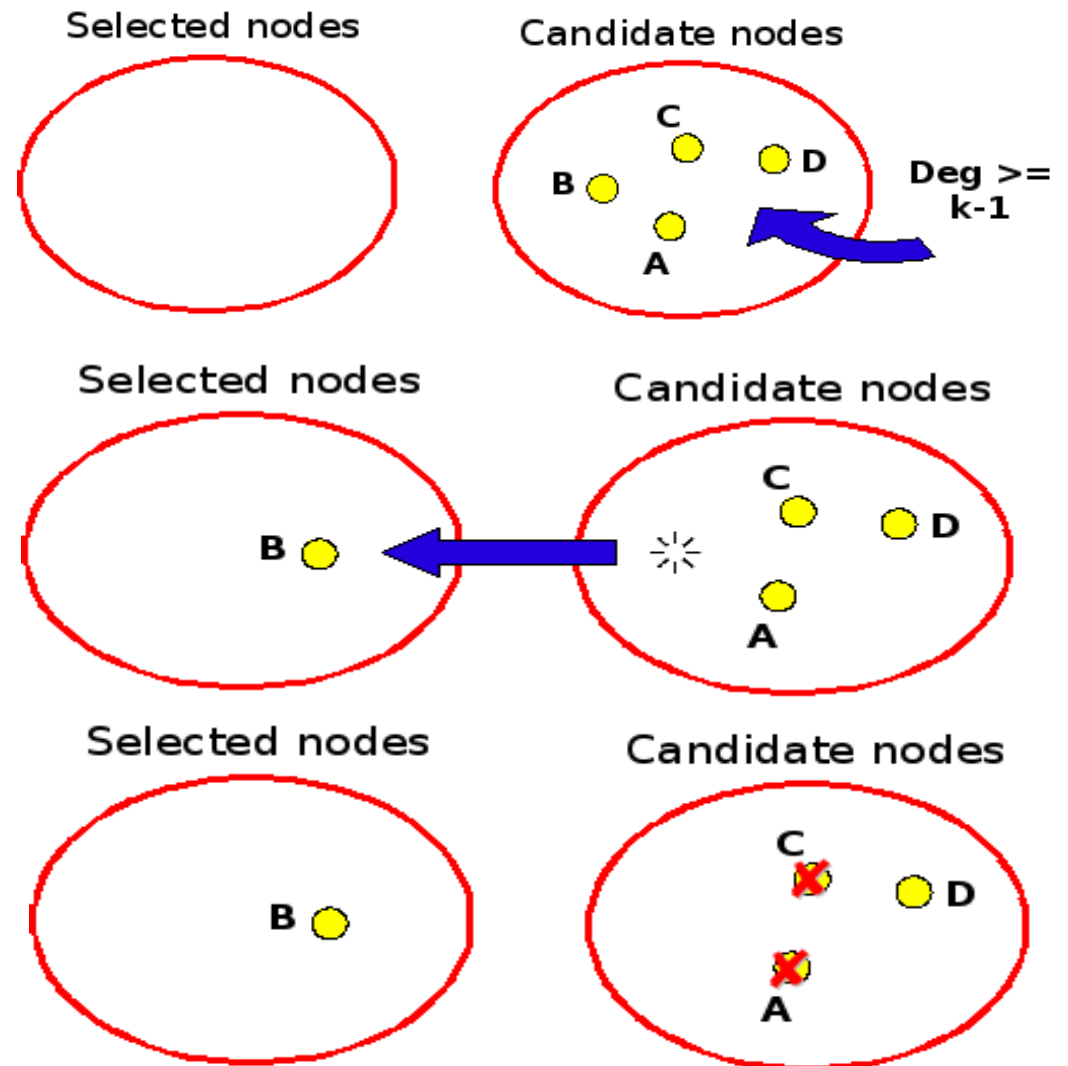


# Implemented algorithms – Clique percolation (2)

The algorithm:

1. for each  $k$ :
  - 1a. detects all the maximal  $k$ -subcliques of the graph.
2. calculate their overlaps.
3. merge in the same cluster the maximal sub-cliques that share at least  $k-1$  nodes.

The determination of the maximal sub-clique of a graph is a NP-complete problem!



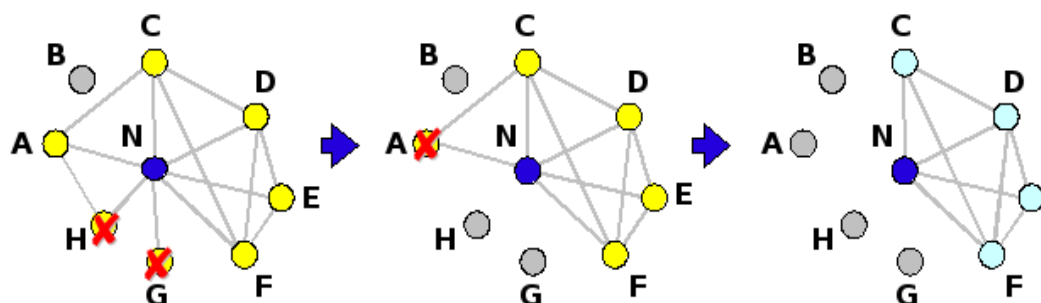
# Implemented algorithms – MCODE (1)

## MCODE

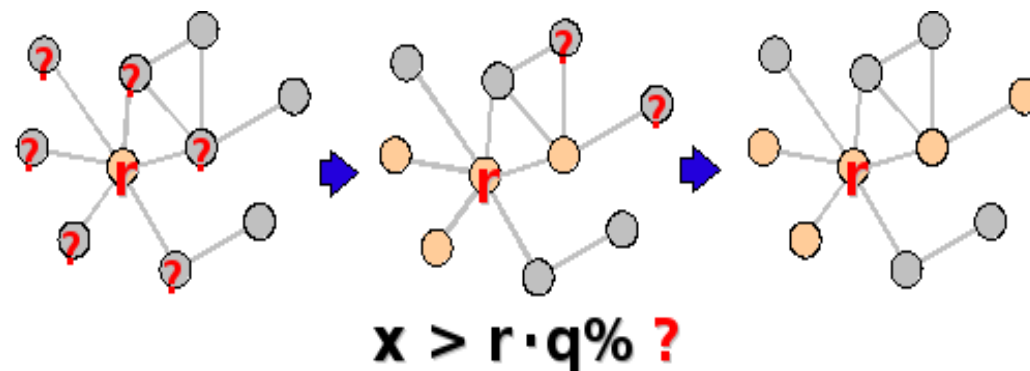
### Protein-protein interaction networks.

Bader, Hogue. An automated method for finding molecular complexes in large protein interaction networks. 2003.

At first, nodes are weighted with the product of the level of their maximal k-core and its own density.



Clusters are then assembled starting by high value root nodes and checking their neighbourhood.

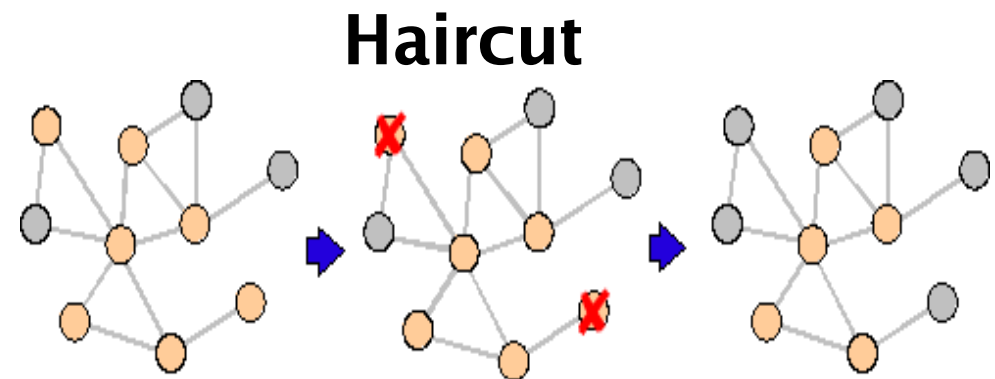
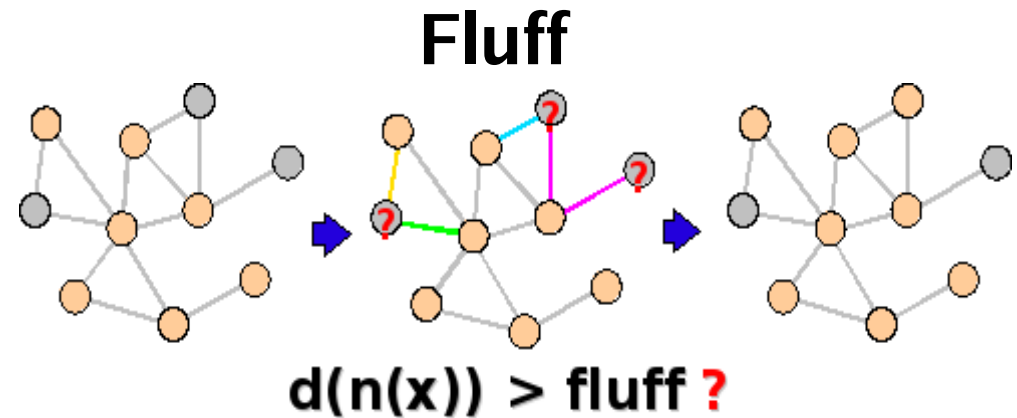


# Implemented algorithms – MCODE (2)

At last, the clustering can be refined by an optional post-processing phase:

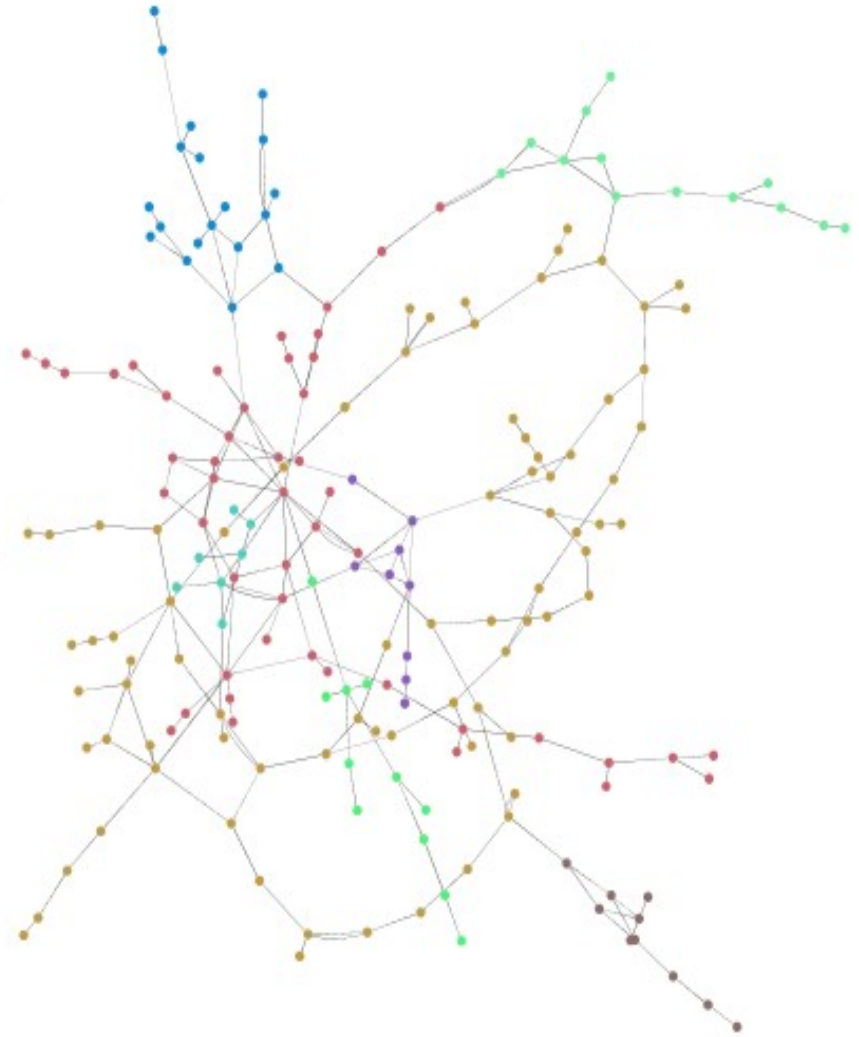
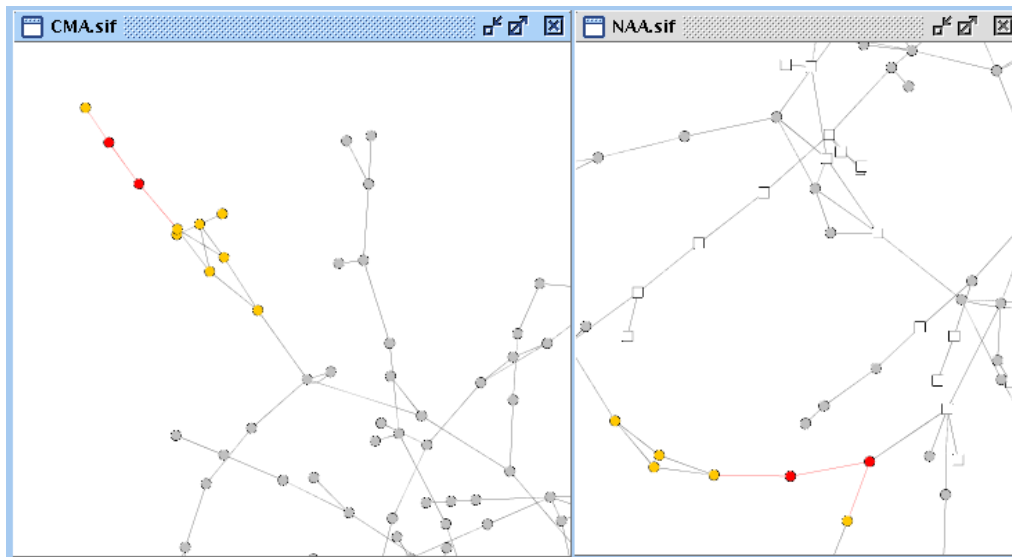
- **fluff**: includes all the neighbourhood of the cluster nodes if its density is greater than the fluff parameter
- **haircut**: delete all the pendent nodes from the clusters.

If both are selected, first run fluff, then haircut.



# Results visualisation

- No node selected:  
clusters global perspective.
- Some nodes selected:  
highlighting of nodes and  
clusters for the comparison over  
different networks.



# Clustering comparison

## Evaluation of clustering algorithms, S. Brohée and J. Van Helden, 2006

Given two clusterings M and N over the same network:

$$M = \{M_1, M_2, \dots, M_m\} \quad N = \{N_1, N_2, \dots, N_n\}$$

we calculate the overlap  $T_{i,j} = |M_i \cap N_j|$  and then:

$$Q_{i,j} = \frac{T_{i,j}}{|M_i|} \quad P_{i,j} = \frac{T_{i,j}}{\sum_i |T_{i,j}|}$$

From here we can calculate the **sensitivity** and the **positive predicted value** as:

$$Sen = \frac{\sum_i |M_i| \max_j Q_{i,j}}{\sum_i |M_i|} \quad PPV = \frac{\sum_j |N_j| \max_i P_{i,j}}{\sum_j |N_j|}$$

and finally the **accuracy** as:

$$Acc = \sqrt{Sen \cdot PPV}$$

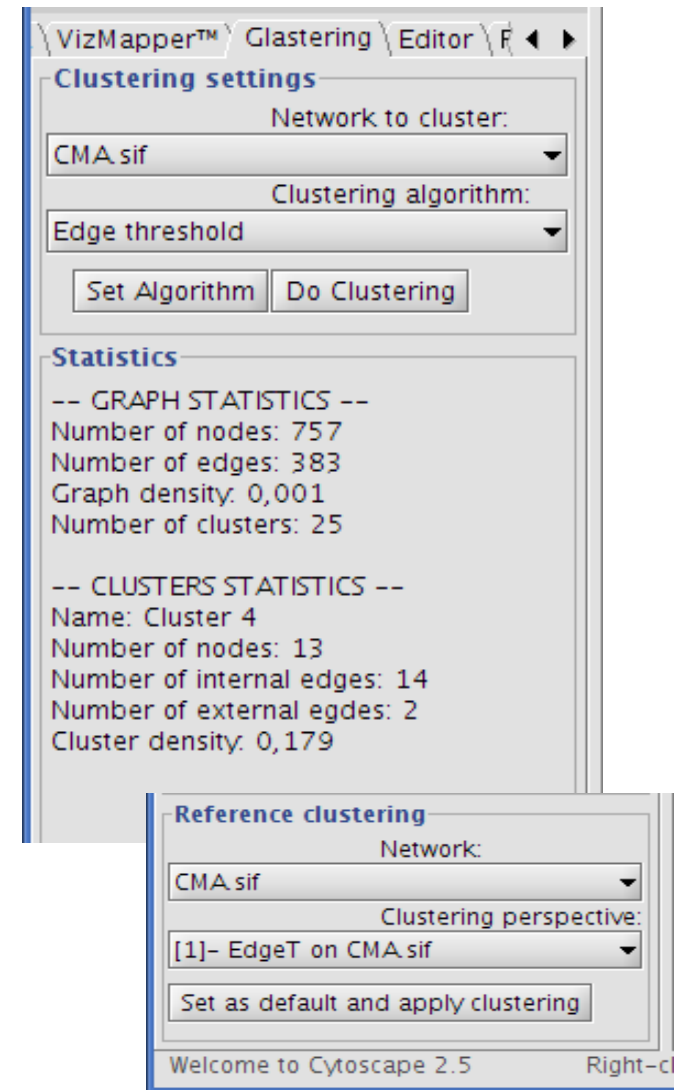
# Conclusions

The relation between dense sub-graphs and functional modules:

- looks like the only possible way, given these hypothesis.
- needs to be investigated more in depth.

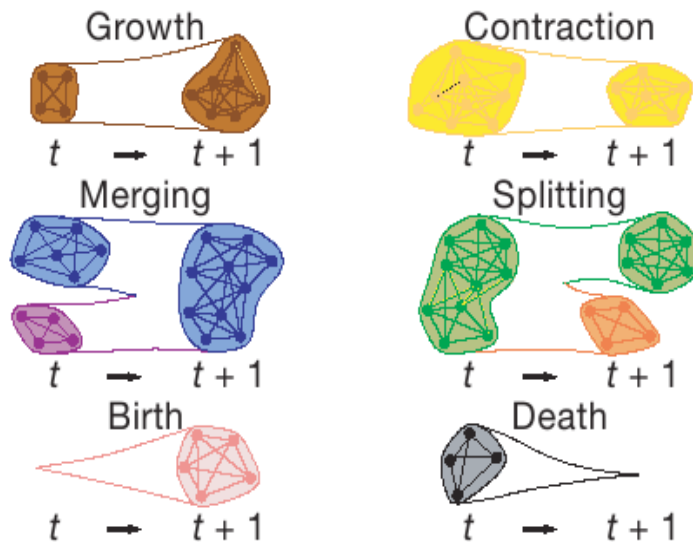
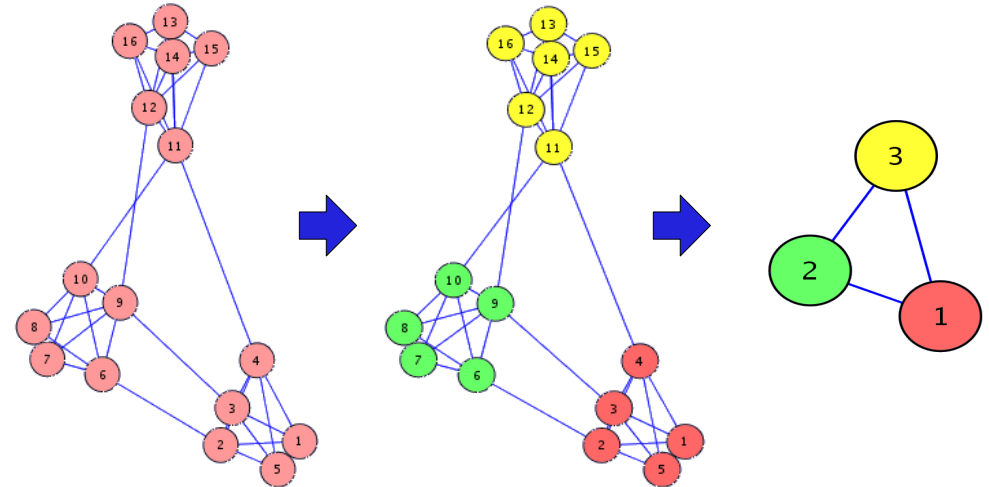
ClustPlugin is a good starting point for this kind of surveys:

- it implements tools usable for this aim.
- it provides the necessary base to develop new ones.



# Possible improvements

- New algorithms, metrics, statistical evaluations.
- Improvement of the visualisation features.
- Quotient graphs.



- Visualisation and comparison of different clusterings.
- Clusters evolution.
- Pattern matching of graph, identification of common sub-structures.



Thanks for your attention.

