KNOWLEDGE IN ACTION

Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt
Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek

2013•2014
FACULTY OF SCIENCES
Master of Statistics

Master's thesis

Integrated data anlaysis via clustering

Promotor :
Prof. dr. Ziv SHKEDY
Mevrouw Nolen Joy PERUALILA

Marijke Van Moerbeke

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics

Maastricht University

2013•2014

FACULTY OF SCIENCES

Master of Statistics

Master's thesis

Integrated data anlaysis via clustering

Promotor :
Prof. dr. Ziv SHKEDY
Mevrouw Nolen Joy PERUALILA

Marijke Van Moerbeke

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics

Acknowledgments

This thesis project is the conclusion of five years hard work. I would like to take the time to thank a number persons who without them I would not have made it as good as I did now. A big and massive thank you to my family. My sister Anne, my brother Quinten and my parents Martine and Paul. Thank you for your unlimited patience with me and taking so many things out of my hands such that I did not need to worry about them. I know I do not always say it out loud but I love you.

I want to thank professor Dr. Ziv Shkedy. You are an incredible teacher and mentor. Thank you for your inspiring ideas, your advice, your kindness, your humour, your curiosity for new discoveries and telling me when to stop working. I cannot wait to be part of your team in October.

A thank you to Nolen Perualila for your creative solutions, ideas and advice during this project. I hope to learn many more things from you during the coming years. Your kindness is inspiring.

Two persons deserve an explicit thank you. Kevin en Ewoud, thank you for being there for me, not only when I needed you but always. We shared many laughs, an occasional tear and so much friendship. Because of you I aim higher. I am grateful for you and overjoyed that you are joining me on the next step of this adventure. I am already decorating our office.

Lastly, there is someone I need to thank and I wish it was not yet time to do so since I had promised her a seat for my doctorate thesis defense. Martine Van Gastel, you were and still are an inspiration to me and it is unfair that death took you so soon. You introduced me to the delights of mathematics and believed in me long before I did. It was you who predicted that I would do better every year at university and told me that obtaining a doctorate position would be no problem. You transformed from teacher to friend and I wish I had had the chance to say goodbye. You are missed dearly.

Abstract

Discovering the exact activities of a compound is important in drug discovery. The compound may or may not have the desired effects and indeed, if a compound reacts to an off-target and this is not seen, severe side effects can be the result. A single source of information is limited by its specific point of view. The integration of multiple data sources can offer more insight into the mechanism of action and help to shine a light on the global picture of the working of compounds. Several integrative data clustering techniques were performed on two data sets which were accompanied by two data sources each. By comparing the clustering on the separate sources with the integrative analyses, the influence of each could be investigated. A best integrative method is not declared. Rather interest lies in clusters that are found to be stable over the different methods. These compounds are indicated to be similar on different aspects of the underlying biology. It can than be hypothesized that the data sources are related for those compounds. If compounds do not show a clear resemblance to one or possibly multiple groups, they can be clustered differently for each method. It was seen that the differential expression of the cluster is greatly influenced by the compounds joined to them.

Contents

1 Introduction 1

2 Data 2

3 Methodology 2

3.1 Agglomerative Hierarchical Clustering 2

3.1.1 Dissimilarity Matrices for Binary Data 3

3.1.2 The Ward link 3

3.1.3 The gap statistic 3

3.2 Clustering on a Single Source 4

3.3 Multi-Source Clustering 4

3.3.1 Aggregated Data Clustering 4

3.3.2 Aggregated Data Ensemble Clustering 4

3.3.3 Weighting on Membership 6

3.3.4 Complementary Ensemble Clustering 6

3.3.5 Weighted Clustering 7

3.3.6 Similarity Network Fusion 7

3.4 Comparison of Results 10

3.5 Selection of a Specific Cluster 11

3.6 Secondary Analyses 11

3.6.1 Differential Gene Expression 11

3.6.2 Pathway Analysis 11

4 Results 13

4.1 Sensitivity Analysis 13

4.1.1 ADEC & CEC 13

4.1.2 SNF 14

4.2 MCF7 Cmap Data 14

4.2.1 Clustering on a Single Source 14

4.2.2 Multi-Source Clustering 16

4.2.3 Differential Expression and Pathway Analysis 20

4.3 Inhouse1 Data 23

4.3.1 Clustering on a Single Source 23

4.3.2 Multi-Source Clustering 24
4.3.3 Differential Expression and Pathway Analysis 29

5 Software Development 33

6 Discussion 35

7 Conclusion 38

8 References 39

9 Appendix 40

9.1 Methodology 40
9.2 Sensitivity Analysis 44
9.2.1 ADEC 44
9.2.2 CEC 45
9.2.3 SNF 48
9.3 MCF7 Cmap Data 50
9.3.1 Clustering on Separate Sources 50
9.3.2 Integrated Clustering 53
9.3.3 Differential Expression and Pathway Analysis 57
9.4 Inhouse1 Data 68
9.4.1 Clustering on Separate Sources 68
9.4.2 Integrated Clustering 71
9.4.3 Differentially Expression and Pathway Analysis 75
9.5 Functions 89

1 Introduction

Discovering the exact activities of a compound is important in drug discovery. The compound may or may not have the desired effects and indeed, if a compound reacts to an off-target and this is not seen, severe side effects can be the result. It was only recently discovered that a drug for arthritis also cures a rare condition causing baldness. This only proves that predicting the exact behavior remains a challenge. Therefore, it is encouraged to rely on not one but several sources of information as a single data source is limited by its specific point of view. The integration of multiple data sources can offer more insight into the mechanism of action and help to shine a light on the global picture of the working of compounds.

If data from multiple sources are available, integrative data methods are necessary to analyze these simultaneously. The methods used in this project are several adaptations of a clustering procedure. Clustering is, in its essence, a technique to find groups of objects that with a high similarity between each other among all available objects. Many methods and algorithms are already established, differing in the applied similarity measures between the objects, the measuring of distances between groups of objects and the structure of joining objects together. Often clustering is performed on only one data source. The aim is now to focus on two data modalities and combine their information. It is interesting to look for compounds that always appear together and whose grouping is thus robust against the used method. This is an indication that those compounds show similarities on both data sources and thus are similar to each other on several aspects. It can than be suspected that the information on those compounds is related. A comparison with clustering on the single sources shows which data source had the upper hand in the grouping.

The used data sets are the MCF7 cell line (breast cancer) of the Cmap data and a company-provided data set referred to as Inhouse1. Two data modalities are given by each data set. The challenge is now to exploit all the provided information by several clustering techniques and compare the results.

In section two the data is introduced and the methods are presented in section three. Technical details are given if these are in order. Section four contains the results of the performed techniques and a comparison is made between them for each data set. Since for this project many functions were developed, section five provides basic details of the used software. The functions were bundled into the package *IntClust* of which the help files are provided in the Appendix. A discussion is held in section six and section seven provides a conclusion.

2 Data

The mechanism of a compound is related to its structure and bio-activity. Information can be given in the form of fingerprints, target predictions or bio-assays. Fingerprint data indicates whether or not a special structure is present in the molecules of the compounds. The absence or presence of one of these fingerprints can alter the behavior. Targets are predicted with the help of a binary target prediction scoring algorithm and are recorded in the target prediction data source. During the development, a goal to which the compound should react to is set but whether or not it will actually hit this target will not be seen until experiments are conducted. Frequently, it is the case that a compound will hit multiple targets. Bioassay scores measure the biological activity of drug and thereby its strength can be valued. Often this is determined by comparing the newly developed drug to a standard substance.

For this project, two data sets were provided: the MCF7 and Inhouse1 data sets. MCF7 is a connectivity map (Cmap) data set concerning a breast cancer cell line. Cmap data is publicly available form the Connectivity Map server and provides gene expression data for 2434 genes. The MCF7 cell line contains 56 compounds and information on 250 fingerprints and 477 predicted targets of those compounds. Inhouse data 1 involves 1056 genes and 94 compounds. Further, a fingerprint matrix with 324 fingerprints and a bio-assay activity matrix giving information of 13 bio-assay measuring points were provided. The names of the genes and compounds of the Inhouse1 data are masked.

3 Methodology

3.1 Agglomerative Hierarchical Clustering

Clustering algorithms are unsupervised data mining techniques and aim to find subgroups of objects with a high similarity among each other. The clusters can be formed by different methods but all rely on a measure of distance or dissimilarity between the objects. The used technique here is hierarchical clustering. In this procedure, the clusters are arranged into a natural hierarchy. At the lowest level, each object has its own cluster and at each higher level, the two clusters closest in terms of the used intergroup dissimilarity, are joined together. At the highest level, all objects form one cluster. The advantages of hierarchical clustering is that no prespecified number of clusters or starting points are necessary. It is only based on the dissimilarity between the objects. Hierarchical clustering is the clustering procedure that is performed in each of the methods below. The resulting clustering can be visualized with a dendrogram.

3.1.1 Dissimilarity Matrices for Binary Data

The method starts with the computation of the dissimilarity matrix between the objects. The fingerprint and target prediction data matrices are binary matrices. It was chosen to work with the *tanimoto coefficient* as a measure for the dissimilarity for these data matrices. The tanimoto coefficient is formulated as follows:

$$TC = \frac{N_C}{N_A + N_B - N_C}$$

in which A is one object and B is another object. The number N_A represents the number of features (fingerprints or targets) for object A and N_B is this number for object B. The amount of shared features is N_C. As explained in the paper by Li et al. (2011): the higher the Tanimoto coefficient, the more features are shared between the two objects and the more alike they are. Since clustering is based on dissimilarity, the resulting coefficients were first subtracted from 1 in order to work with dissimilarities. In literature, many different definitions of this coefficient exist. Therefore, these distances were computed with a self made function to make sure above definition was used. Since the bio-assay scores are of a continuous nature, the chosen distance there was the euclidean distance.

3.1.2 The Ward link

The dissimilarity matrix is the input for the hierarchical clustering method. It was chosen to work with the *Ward* link in the clustering process as a measure for the between group dissimilarity. The method of Ward is based on minimizing the total within variance of the clusters. At each step in the iteration, the sum of squared deviations from each object in the cluster to the mean of the cluster is calculated, referred to as SSE_k. The total sum of squares is then the sum of the SSE_k's:

$$SSE = \sum_{k=1}^{K} SSE_k$$

with K the total number of clusters. Then the merging of every pair of clusters is considered and the two whose joining results in the smallest increase in the SSE are effectively joined. Initially, all SSE are zero and in the end only one group remains. The initial cluster distances in Ward's minimum variance method are the squared euclidean distances between points.

3.1.3 The gap statistic

Since the aim is comparing the results of the methods and to observe how the clusters are influenced, it was decided to cut the dendrogram of each method into a fixed number of clusters. This way, it can be studied how differently the objects are grouped together. Although the optimal number of clusters is a rather subjective choice and any reasonable number would be appropriate, the number was based on the gap statistic. The gap statistic represents the difference between the within cluster dissimilarity that is truly observed from the data for a specific number of clusters k and the one that is obtained under the assumption that the data is uniformly distributed (no clustering). The optimal number of clusters is the k that causes the largest gap between these values. Tibshirani proposed to take the first k such that:

$$f(k) \geq f(k+1) - s_{k+1}$$

with s_{k+1} the standard deviation for the gap statistic for $k+1$ clusters. It was chosen to take a suitable (average or highest) k that was determined with the rule of Tibshirani for the original clustering on only one source of information to continue the project with in the integrative analyses.

Next, all clustering methods applied on the data sets are described.

3.2 Clustering on a Single Source

Dissimilarity matrices were computed from the provided data matrices and hierarchical clustering was performed on each of these separately. The results are visualized with a dendrogram. Two dendrograms are thus produced for each data set, each based on a single (different) source of information. For the MCF7 data, the clusterings are based on fingerprints and target predictions while for the Inhouse1 data, they were based on fingerprints and bio-assays scores. The results are later compared to those of the methods in which information from the data sources is combined.

3.3 Multi-Source Clustering

3.3.1 Aggregated Data Clustering

If the data sources are of the same type, these can be combined into one larger data matrix. For the MCF7 data set this is possible since the two provided data sources are binary. Then this technique, referred to as aggregated data clustering (ADC), proceeds with hierarchical clustering on this single (larger) data matrix. Clustering is thus performed on both data sources simultaneously after a simple combination of the data which results in having access to more variables at a time.

3.3.2 Aggregated Data Ensemble Clustering

Aggregated data ensemble clustering (ADEC) is described in Fodeh et al. (2013). This technique and adaptations of it were performed on the provided data sets. The original method, version a, is described next, followed by a description of the differences with the two adaptations, versions b and c. It was decided to work with different interpretations of the same basic idea to see what the influence would be of the changed parameters. The method can only be performed if the available data matrices are of the same type.

ADEC starts with the merging of the data matrices into one larger data matrix. Then, ensemble clustering is performed on the fused data. This comes down to repeatedly applying hierarchical clustering and consist of the following steps. Call m the number of features of the large data matrix A.

1. In every iteration, a random sample of features r of A is taken and form matrix A'. The number of features is randomly set between $\frac{m}{2}$ and $m-1$ each time.
2. Hierarchical clustering is performed on A'. The dendrogram is cut into a specific number of clusters k.
3. The incidence matrix C is computed. This binary matrix has as rows and columns the objects of the data set. A value of one indicates that these objects belong to the same cluster.
4. The co-association matrix S is iteratively computed as

$$S^{(t+1)} = S^{(t)} + C$$

   and indicates the number of times a pair of objects belong to the same cluster.
5. The steps in 1-4 are repeated t times.
6. Finally, hierarchical clustering is performed on the resulting co-association matrix S which yields the result of the ensemble clustering.

The procedure is visualized in Figure 21 in the Appendix and is, apart from fusion of data matrices, the same as performing ensemble clustering on a single data source. The method is said to be effective in the improvement of robustness and stability since it is capable of combining several clustering results into one single solution. Many methods and variations exists to this theme already with in this project version a described above.

In version b, all features will be used in every iteration. However, variation is inserted by not splitting the dendrogram a single time into one specific number but multiple times and for a range of values for k. The last version, version c, is a combination of version a and b. It takes a new random number of samples at every iteration and cuts the dendrogram several times for a range of values for k. For version a and c, the option is available to fix the number of features to sample in each iteration.

3.3.3 Weighting on Membership

The "weighting on membership" (WonM) method has similarities with version b of ADEC. Hierarchical clustering is performed on each data source separately. With k the number of clusters, the resulting dendrograms are cut into clusters for a range of values for k. Each time, an incidence matrix is put up. This is a matrix with as rows and columns the objects of the data set. Its values are zero and one with one indicating that a pair of object resides in the same cluster. The incidence matrices are first summed over all values of k per data source and then these of the different data sources are added as well. On the resulting consensus matrix, hierarchical clustering is performed once again to obtain the final clustering taking into account information of all data sources.

3.3.4 Complementary Ensemble Clustering

Complementary ensemble clustering is described in Fodeh et al. (2013) and shows similarities with ADEC. The main difference is the first step. Instead of merging the data matrices, ensemble clustering is performed on each data matrix separately. The technique is presented in Figure 22. As for the ADEC method, three versions were implemented. Version a is explained below. Suppose A_i with $i = 1, 2$ refers to the corresponding data matrix, k is the number of clusters and m_i is the total number of features.

- For each data source A_i:

  1. In every iteration, a random sample of features r of A_i is taken and form the matrix A_i'. The number of features is randomly set between $\frac{m_i}{2}$ and $m_i - 1$ each time.
  2. Hierarchical clustering is performed on A_i'. The dendrogram is cut into a specific number of clusters k.
  3. The incidence matrix C is computed. This binary matrix has as rows and columns the objects of the data set. A value of one indicates that these objects belong to the same cluster.
  4. The co-association matrix S_i is iteratively computed as

     $$S_i^{(t+1)} = S_i^{(t)} + C$$

     and indicates the number of times a pair of objects belong to the same cluster.
  5. The steps in 1-4 are repeated t times.

- The co-association matrices S_1 and S_2 are linearly combined into the final co-association matrix S.

For a weight α, S is determined as:

$$S = \alpha \cdot S_1 + (1 - \alpha) \cdot S_2.$$

- Finally, hierarchical clustering is performed on the resulting co-association matrix S which yields the result of the ensemble clustering.

Version b of CEC uses all features instead of a random sample but cuts the dendrogram multiple times in a varying number of clusters just as version b of ADEC. Version c is also here a combination of versions a and b in which in each iteration a random sample of features is taken and for each clustering result the dendrogram is split several times into a different number of clusters. For version a and c, the option is available to fix the number of features to sample in each iteration. An extra parameter of choice in each version is the weight α to be given to the co-association matrix. It is a challenge to determine the weight that provides the optimal clustering. One option is to give the higher weight to the ensemble clustering of the single data sources with the least distortion. Another is to try out several values of α and choose the value that results in the better clustering.

3.3.5 Weighted Clustering

In weighted clustering, a weighted dissimilarity matrix is computed combining the available sources. Suppose that W_1 and W_2 are the dissimilarity matrices of the two data sources. The weighted dissimilarity matrix W is formed as:

$$W = \delta \cdot W_1 + (1 - \delta) \cdot W_2$$

where δ is the weight. It was decided to let δ vary from one to zero. Hierarchical clustering is then performed on the weighted dissimilarity matrix. For the weight one and zero, the clustering is based on a single source of information. It is investigated how the original clustering evolves when the weight changes and what the influence is of the addition of extra information on the objects.

3.3.6 Similarity Network Fusion

Similarity network fusion (SNF) was recently introduced by Wang et al. (2014). The technique basically consists of two steps and results in a sharing of information over all available data sources. The ideas behind the steps are as follows:

1. The initial step. A similarity network is set up for each data matrix by the means of similarity measure. This results in a matrix with a similarity value for each pair of objects and can be seen as a distance

matrix. The connection to a network is made when the matrix is visualized as weighted graph with the objects as nodes and the pairwise similarities as weights on the edges.

2. The network-fusion step. Each network is iteratively updated with information of the other network which results in more alike networks every time. This eventually converges to a single network.

The result of the network-fusion step is a network in which weak similarities are smoothed out, reducing noise, and strong similarities are highlighted. Figure 23 in the Appendix is the figure depicted in the paper by Wang et al. (2014) and shows the steps of SNF clearly. A more detailed outline follows.

Consider the objects x_i and x_j. Suppose that the data is continuous in nature and the euclidean distance between x_i and x_j is measured by $\rho(x_i, x_j)$. The distances are scaled exponentially by the means of the kernel of the standard normal distribution:

$$W(i,j) = exp\left(-\frac{\rho^2(x_i,x_j)}{\mu \cdot \epsilon_{i,j}}\right).$$

The parameter μ is a hyperparameter and according to Wang recommended to be in the interval $[0.3; 0.8]$. Parameter $\epsilon_{i,j}$ is added to eliminate the scaling problem and computed as:

$$\epsilon_{i,j} = \frac{\bar{\rho}(x_i, N_i) + \bar{\rho}(x_j, N_j) + \rho(x_i, x_j)}{3}.$$

The first and second element refer to the mean value of the distances between x_i (x_j), and each of the neighbours N_i (N_j). A question to be resolved is how many neighbours (k) to consider in the vicinity of the points. It is advised by Wang et al. to determine the number of neighbours by dividing the total number of objects by the number of clusters. If the number of clusters is unknown or not set to a specific value, the total should be divided by 10. It is reported that the method is not influenced by the choice of this parameter. When the scaled similarity values are computed, these are stored in a similarity matrix W. As mentioned before, a similarity matrix of samples can be looked upon as a graph (network) $G = (V, E)$ with V representing the objects as vertices and E the weighted edges. The weights are the similarity values. This is done for each data source and concludes the initial step.

In order to fuse the networks together, two more matrices are set up. Matrix P (status matrix) is a normalized version of the similarity matrix W and its values are obtained as:

$$P(i,j) = \begin{cases} \frac{W(i,j)}{2 \cdot \sum_{k \neq i} W(i,k)} & , \text{if } j \neq i \\ \frac{1}{2} & , \text{if } j = i \end{cases}$$

The similarity of each object to all other objects is stored in P and thus P carries the full information. Since the goal is to update the network of one source with information of the other source and thereby highlighting the stronger similarities, the matrix S focuses on the local information on each object and is referred to as the kernel matrix. Consider the point x_i and let N_i be its k closest neighbours including x_i itself. The values of S are computed by:

$$S(i,j) = \begin{cases} \frac{W(i,j)}{\sum_{k \in N_i} W(i,k)} & \text{, if } j \in N_i \\ 0 & \text{, if otherwise} \end{cases}$$

This way, S only contains the normalized similarities of the points in the vicinity of x_i. The assumption is that objects in the neighbourhood of x_i bring more reliable information forward than objects further away. After obtaining the matrices P and S for each data source, an algorithm is carried out iteratively. The procedure starts from the matrices P and uses the matrices S to update the information and grasping the local structures of the data.

For two sources of information, the fusing algorithm starts with $W^{(1)}$, $W^{(2)}$ of which $P^{(1)}$, $P^{(2)}$, $S^{(1)}$ and $S^{(2)}$ are calculated. These are the matrices at time $t = 0$. The main step is update the matrices $P^{(1)}$ and $P^{(2)}$ with information from the other source. This is done as follows:

$$P^{(1)}_{t+1} = S^{(1)} \cdot P^{(2)}_t \cdot (S^{(1)})^T$$

$$P^{(2)}_{t+1} = S^{(2)} \cdot P^{(1)}_t \cdot (S^{(2)})^T.$$

After each iteration, normalization is performed on $P^{(1)}_{t+1}$ and $P^{(2)}_{t+1}$ to ensure that every object is the most similar to itself. The overall status after t steps is:

$$P^{(c)} = \frac{P^{(1)}_t + P^{(2)}_t}{2}.$$

Since the matrices $S^{(1)}$ and $S^{(2)}$ only have values for the neighbourhood of the object, information comes from local similarities and not from remote ones. If x_i and x_j share objects in their neighbourhoods for both data sources, it is very likely that these belong to the same cluster and this will show up in the overall status matrix $P^{(c)}$. It is matrix $P^{(c)}$ that is subjected to hierarchical clustering.

The SNF method is a network based integrative approach. It is capable of detecting common and complementary signals across the data sets and also reduces noise by integrating over several types of data. Therefore the method is said to have the ability to capture the reality behind the different types of data.

This is the method as it is described in Wang et al. (2014). The paper is accompanied by an R package called *SNFtool*.

The functions of the package needed to perform SNF were gathered into one function called *SNFa*. However, if a closer look is taken at these functions, these to do not seem to confirm the method outlined in the paper. The differences and peculiarities are listed in the Appendix.

The concern lies in the multiple steps for normalization and not performing the normalization after every iteration as was indicated in the paper. It was decided to rewrite the *SNFa* function and outline the procedure more as in the description. Function *SNFb* performs SNF as outlined above. Function *SNFc* is similar to *SNFb* except for the computation of the kernel matrix S. The difference is that in *SNFa*, two normalizations are performed before obtaining S, first to compute P of which the subsets are taken and then again over these subsets to form S. In *SNFb* the subsets are taken first from the similarity matrices W followed by a normalization over the neighbours only and in *SNFc* the distance matrices are first normalized to obtain P of which then a subset of k neighbours is taken to form S. A comparison of the results of the methods will be held in the results and discussion section. Different values for the number of neighbours were considered and a sensitivity analysis was conducted.. The parameter μ was decided to be 0.5 and the number of iterations 20.

3.4 Comparison of Results

A comparison of the results is not done on sight. Different methods cluster the compounds in a different order and this results in non-corresponding cluster numbers. Therefore, it was decided to take one method as reference and rearrange the cluster numbers of the other results to this reference. The re-appointing of the cluster numbers is based on finding the cluster that relatively has the most in common with one of the reference clusters and taking over this number. The function *MatrixFunction* was written for this purpose. It creates a matrix of which the columns are the compounds in the order of clustering by reference method. The rows are the different methods and the values of the cells are the rearranged cluster numbers. If each value is associated with a colour, a visualization of the matrix can be made.

A similarity measure is handy for the comparison of multiple methods. Given a method to be used as reference, it is observed which compounds are residing in the same cluster for a second method after applying *MatrixFunction*. The number of compounds that belong to the same cluster is summed and divided by the total number of objects in the data set.

3.5 Selection of a Specific Cluster

The clustering procedures and differential gene expression of a specific cluster is discussed in detail for each data set. The choice of the cluster was influenced by the stability of each cluster over the results of the different methods. If a grouping of compounds is found regularly, it is implied that these compounds show resemblance on both data sources. One specific cluster was selected and it was studied how it alters over the different methods and over the weights in the CEC and weighted clustering methods. The function *ClusterDistribution* is capable of following a selection of compounds. It will investigate over how many clusters the compounds are divided and give information on each of these clusters. For example, it is given which compounds of the original selection it contains and which ones are extra to the newly formed cluster. A plot was created to see how exactly the compounds are divided for each method or weight. Since it is likely that some compounds of the selection are appointed to a different cluster, it was decided to focus on the maximum number of compounds that remain together. This can imply that the cluster number changes.

3.6 Secondary Analyses

3.6.1 Differential Gene Expression

After clustering the next step was to find genes of interest. The gene expressions of the objects in each cluster were compared to those of all other clusters combined. Interesting genes are those that behave differently in the selected compounds. The gene may have a higher variation or can be up-regulated or down-regulated. Several methods exist to find such differentially expressed genes such as a two sample t-test or a permutation test. It was opted to work with a method called Linear Models for Microarrays (limma). The resulting p-values were adapted for FDR and the significance level was chosen to be 0.05.

Limma is in fact a regular two sample t-test with an adjusted denominator. This adjustment is made to avoid that genes that have a small fold change and a small variance will be considered significant by the procedure. The denominator is estimated with an empirical bayesian approach. More information can be found in Smyth, G. K. (2004).

3.6.2 Pathway Analysis

The final step in the analysis is to allocate the genes to a gene set or pathway. If a gene set is enriched, i.e. the probability to see this many significant genes of this gene set by chance is low in the selected compounds, one may be fairly sure that the compounds share an activity on this pathway. The selected database for pathway analysis is the Gene Ontology (GO) database. This is an hierarchical database which starts from more general terms (annotations) to very specific ones. It can be seen as a large tree with parent and child nodes. It is more than likely that a node has multiple parents or children.

The pathway analysis performed was functional class scoring, also known as *MLP*. The search for enriched gene sets starts from the p-values of all genes. For a predefined gene set, it calculates a score that summarizes the significance of all the genes included in that specific set. This score is the mean of the negative logarithm of the p-values:

$$MLP = mean(-log(p - values)).$$

It is then determined how likely it is to see the MLP value by chance. This is done by a comparison with the empirical null distribution. To attain this distribution, the labels are permuted across the samples and the MLP is recalculated. This is repeated a few times for all gene sets. The null distribution of MLP and the observed value of MLP are compared and if there is a small probability to find the observed MLP (a small p-value for this distribution), the score is deemed significant.

The MLP method to perform pathway analysis is based on resampling of the data. Therefore it is recommended to perform the pathway analysis multiple times to observe how much the results are influenced by a different sample. Here, the pathway analysis is performed 10 times in a loop and the intersection of the results of these 10 loops is considered as the result.

4 Results

This section shows the results of the clustering techniques, the differential expression and the pathway analyses that were conducted on the two data sets. The used software is R version 3.1.0. Many of the applied functions in this project were developed outside of the already existing functions available in R and bundled into a new package called *IntClust*. More information can be obtained in section five and the Appendix.

4.1 Sensitivity Analysis

A few methods had multiple parameters that needed specification. For ADEC and CEC, this concerns the number of iterations and whether or not to resample a specific number of features at each iteration. For SNF, a free parameter is the number of neighbours. The influence of these is presented first before comparing the results of the data sets.

4.1.1 ADEC & CEC

The methods ADEC and CEC are both ensemble clustering methods and three versions were implemented for each. For versions a and c, the option is available to specify the number of iterations and features to sample. In CEC, also a weight can be specified for the linear combination of the data sources but is set to 0.5 here to investigate the influence of the other parameters by an equal contribution of both data sources. Since the data matrices of the Inhouse1 data set are of different types, the ADEC procedure can only be conducted on the MCF7 data.

For ADEC, Figures 24 and 25 show a comparison for different numbers of iterations for version a and c respectively. The first row of the figures, representing methods ADECa10R and ADECc10R respectively, are the reference to which the other results are rearranged. The first five are indicated with an “R” since for these the number of features was fixed to 544. This number is halfway between the half of the total number of variables and the total minus one. The latter five were produced with a different number of features in each iteration. Next, it is investigated what the influence is of the resampling of the features. This is considered for 25 iterations and shown in Figures 26 and 27.
The same was conducted for versions a and c of CEC on both data sets. If the number of features is specified, it must be given for each data source. For the MCF7 data, 187 features were to be sampled from the fingerprints and 357 from the target predictions. For the Inhouse1, 243 features were sampled from the fingerprints and 9 of the bio-assays. The comparison plots in Figures 29 to 32 present the results of versions a and c over several numbers of iterations. A word should be said on the light blue colour arising

in the figures for some results. This is an indication that one of the clusters of a method did not find a did not find a suitable match to any of the reference clusters. It is possible that one cluster was fused to another cluster or was completely divided among others by this method. This implies that one of the original clusters dissapears as a separate unit. However, the tree of the method was still cut into a specific number of clusters and the light blue colour shows the “extra” cluster that does not match an original cluster. The results for different resamplings for a constant number of 25 iterations are presented in Figures 33 to 36.

It is observed that overall the results are similar over the number of iterations, the specification of the number of features and the resampling. This holds especially when excluding the results with only 10 iterations. Figure 28 shows the results for version a, b and c for 25 iterations for ADEC and are practically identical. The results of 25 iterations for CEC are shown in Figures 37 and 38. For MCF7, the results of version a and c share a high similarity while the result of version b differs somewhat. For the Inhouse1 data the results of version a and c agree on some clusters but not on all. However, smaller groups of compounds are always found together although the group might have changed cluster over the versions.

4.1.2 SNF

The varying parameter in SNF is the number of neighbours. For version a in Figure 39 of the MCF7 data parts show similarities although more differences arise when the number of neighbours increases. A possible reason is that when the number of neighbours increases, information from more distant compounds is involved. This alters the similarities. In the results for version b and c, presented in Figures 41 and 43, a similar situation arises. Figure 40 depicts the result of version a for the Inhouse1 data. It is seen that large parts of the clusters remain independent of the used number of neighbours. This is especially true for version b and c in Figures 42 and 44.

Since SNF is based on updating a global network with local information and MCF7 has 56 compounds in total, it was decided to continue with 15 neighbours only in each method. This to ensure that the information comes from objects in the vicinity. The same reasoning lead to a choice of 20 neighbours for the Inhouse1 data. The resemblance between the methods is depicted in Figures 45 and 46 and is striking. The results differ in only one compound each for the MCF7 data and the differences for the Inhouse1 data are minor.

4.2 MCF7 Cmap Data

4.2.1 Clustering on a Single Source

The MCF7 data set was provided with fingerprint and target prediction data on 56 compounds. Both matrices are binary, thus the tanimoto coefficient was used to compute the dissimilarity matrices. In a first stage, clustering was performed on each data matrix separately. Figure 47 represents the resulting clustering based on fingerprints and Figure 48 the result based on target predictions. The rule of Tibshirani determined that the optimal number of clusters for fingerprints and target predictions was six and seven respectively. The number of clusters to continue with was decided to be seven.

A comparison is made between clustering on fingerprints and on target predictions just to see how far these are apart. The dendrogram of the fingerprint clustering was cut into seven clusters and each one was given a specific colour as depicted in Figure 47. The tree of the target predictions was also cut into seven clusters and a colour was given to each compound. However, each compound was coloured into the same colour it had in the clustering based on fingerprints. This way, it can be visualized how compounds have changed in the clustering under the influence of the other source of information. A comparison can be presented as well by an adaptation of the heatmap in Figure 49. Here, the ordering of the target predictions clustering is plotted against the ordering by the fingerprint clustering. This implies that the colours are determined by the clusters based on fingerprints. Vertically, the clusters based on fingerprints are seen while the horizontal direction shows the clusters based on target predictions. A final way to visualize the clustering based on the two sources of information is presented in Figure 1. The values on the right side of the plot are the similarity values computed with the first row as the reference.

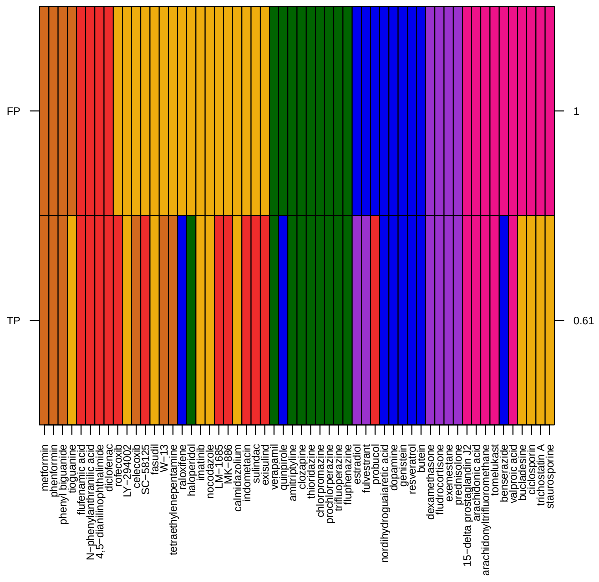


Figure 1: *A Cluster Comparison between Fingerprints and Target Predictions for MCF7*

Again the compounds, represented by the columns, are in the order of clustering on fingerprints. Each colour represents a cluster and the rows show the result for each method. Here, the clustering on target predictions is compared to the clustering on fingerprints.

From Figure 1, it can be seen that some clusters are well preserved over the methods as for example the brown, red, green and purple clusters. Others are split up over different clusters as for example the compounds belonging to the yellow cluster in the fingerprint clustering. This shows that however the compounds are the same for both data sources, the clustering depends on the used data source. Some compounds are alike in fingerprints but differ on their predicted targets. Others however, are similar whether the information is based on fingerprints or on target predictions. The goal is now to see how the clustering is influenced when both sources of information are used simultaneously. It is expected that the compounds belonging to the brown, red, green and purple cluster will group together again since these clusters share many compounds on fingerprints and target predictions.

4.2.2 Multi-Source Clustering

The reference method for a comparison of the integrative results for the MCF7 data is the result of ADC. A dendrogram is depicted in Figure 2 and the clusters of the other methods will be arranged to this ordering of compounds.

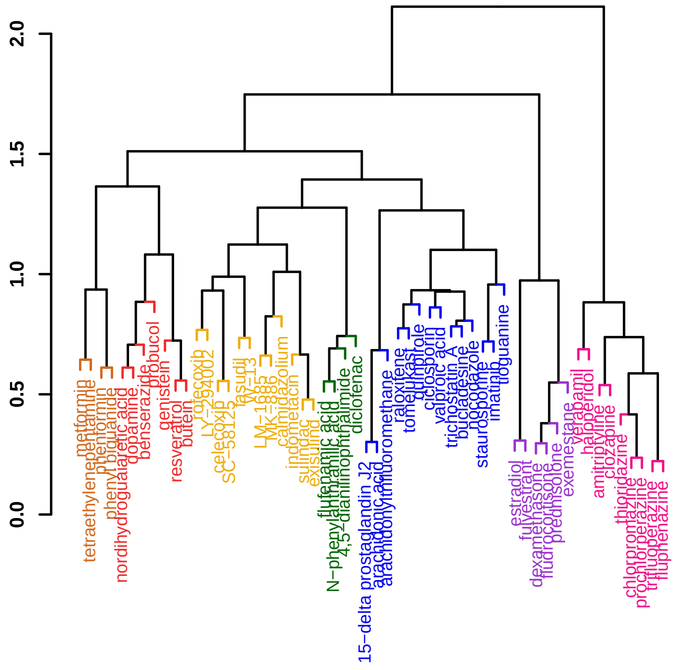


Figure 2: *The Dendogram based on ADC for MCF7*

The methods CEC and weighted clustering involve a weighting of data sources. A comparison over these weights is presented first. For CEC, influence of the weights is seen in Figures 50, 51 and 52. The results of weighting the dissimilarity matrices is depicted in Figure 53. By also showing the results of the fingerprints and the target predictions, the figures show that at a weight of 0.5 and higher, the target prediction information is reflected in the clustering of the compounds while for lower weights more resemblance is seen with the fingerprint clustering. For version b of CEC, the line can be drawn very clearly. In the comparison over all methods, the result of weight of 0.5 will be used. This implies that both sources contribute an equal amount of importance.

Figure 3 presents a comparison of all integrated data methods versus the reference method ADC.

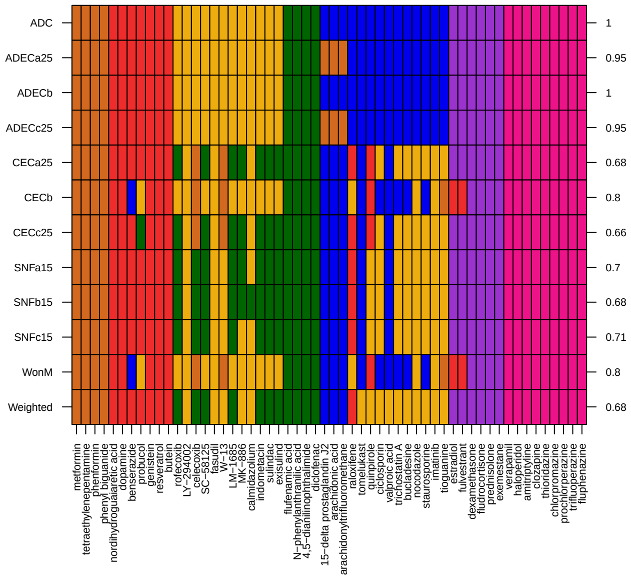


Figure 3: *Comparison of All Methods versus ADC for MCF7*

The methods ADECa, ADECb and ADECc agree highly with ADC and with each other. ADECb is identical to ADC and only three compounds are shifted for ADECa and ADECc. In the results of CEC, CECb differs from CECa and CECc while these two show a high similarity. SNFa, SNFb and SNFc only differ in one compound each and show similarities with CECa and CECc. The same compounds are shifted. A detailed comparison is shown in Figure 54. Only the result of CECb deviates. Many compounds of the purple cluster are moved to the blue cluster although that part of the cluster remains together. The result of the weighted clustering compares to the results of CEC and SNF as shown in Figure 54. Further is the result of WonM identical to that of CECb. A comparison with the result on fingerprints and target predictions is presented in Figure 4.

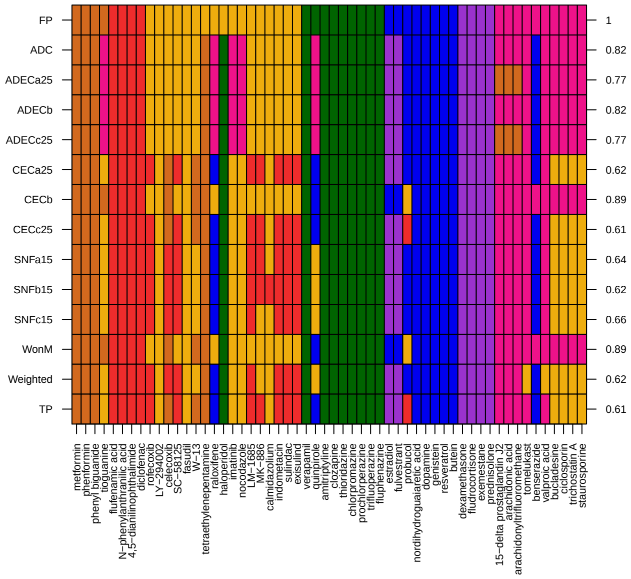


Figure 4: *Comparison of All Methods versus F for MCF7*

One compound of cluster four is moved for every other method. This implies that the information on fingerprints highlights a characteristic of that compound that is not seen for the target prediction information. Note that the fingerprint clustering resembles ADC highly, just as the clustering methods ADEC, CECb and WonM. A clear influence of the target prediction information is seen in CECa, CECc, SNF and weighted clustering. Cluster three shows a high similarity amongst these. Except for four compounds, the clustering of WonM and CECc is identical to the fingerprints clustering.

By studying Figure 4 it is seen that cluster four only differs in one compound over the methods and this only for the fingerprints. This is found to be the most stable cluster for the MCF7 data. The compounds of cluster four are summarized in Table 1.

	Compounds		**Compounds**		**Compounds**
1	verapamil	4	clozapine	7	prochlorperazine
2	quinpirole	5	thioridazine	8	trifluoperazine
3	amitriptyline	6	chlorpromazine	9	fluphenazine

Table 1: *The Selected Cluster for the FP of MCF7*

This selection of compounds was tracked over the different methods and weights in the CEC and weighted clustering procedures. For each, two plots will be given that focus on the alteration of the selection. The first plot will show how the compounds get divided over different clusters for the different methods or weights. The algorithm takes into account which compounds can be found in which cluster. It can be seen as if every dot represents the compounds of the selection in that cluster. If a compound shifts cluster in the next method, the dots are connected. The second figure will only keep track of the maximum number of compounds that can still be found together. This however implies that a change of cluster is possible. With the use of the same colour coding for the clusters as in the figures above, it can be seen to which cluster the compounds have shifted. The number under each dot also reveals the cluster number.

The evolution of the selected compounds over the methods is presented in Figures 5 and 6.

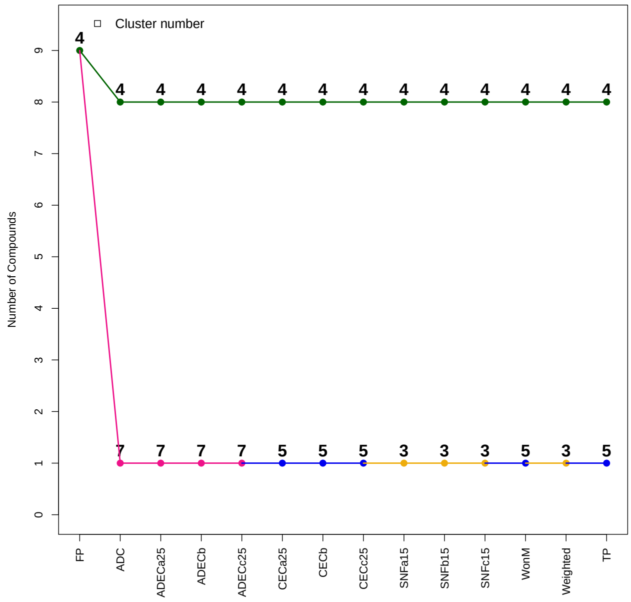


Figure 5: *The Evolution of Cluster 4 of FP over All Methods for MCF7*

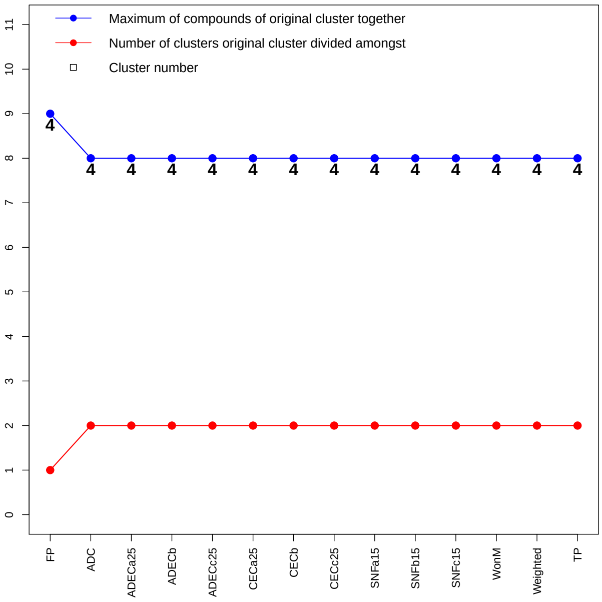


Figure 6: *The Evolution of Cluster 4 of FP over All Methods for MCF7 - Maximum Number of Compounds*

The figures show that over the different methods, eight of the nine selected compounds are always found to together. Figure 6 indicates that these eight compounds do not shift cluster and remain in cluster four. Only the clustering based on solely the fingerprint information appoints the compound quinpirole to this cluster. This implies that the dissapearing compound is alike to the other compounds in terms of fingerprints but shows different characteristics when it comes to target predictions. All other methods replace it by haloperidol and group quinpirole to several other clusters as seen in Figure 5. The cluster was also followed over the weights in the weighted clustering and Figures 7 and 8 show the influence of the weights on the selection.

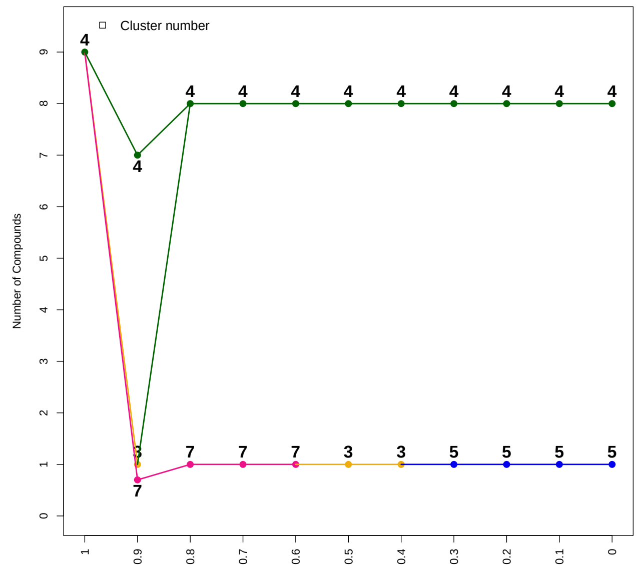


Figure 7: *The Evolution of Cluster 4 of FP over Weighted for MCF7*

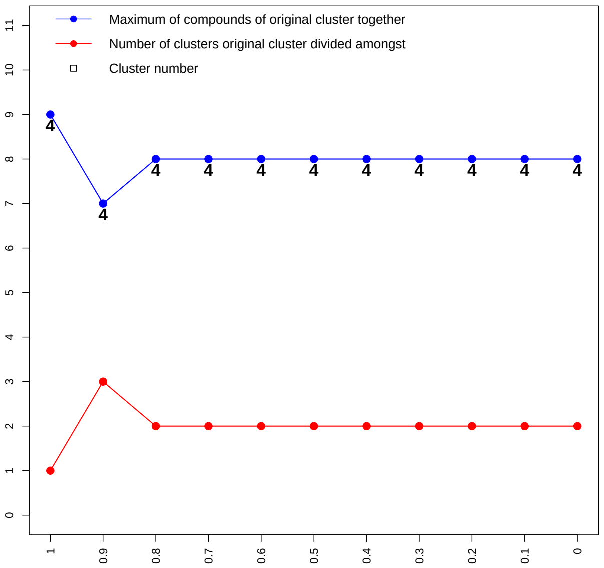


Figure 8: *The Evolution of Cluster 4 of FP over Weighted for MCF7 - Maximum Number of Compounds*

Over the weights of the weighted clustering, first two compounds are removed from the cluster for a weight of 0.9 but one rejoins the cluster for all other weights. The dissapearing compound is also here quinpirole and haloperidol is joined to the cluster for a weight of 0.7 and lower as seen in Figure 53. Similar patterns are seen over the weights in the CEC methods. The plots are presented in Figures 55 to 60 and the clustering on fingerprints was used as a reference. For CECa and CECc quinpirole is not even joined to the cluster and haloperidol joins the cluster at weight 0.7 while for CECb, it is only at the weight of 0.5 that haloperidol replaces quinpirole. For weights 0.1 and 0.0 in CECc, the compounds are fused to those of cluster three. Overall, once the information of the target predictions is involved, eight compounds are found similar on both sources of information with haloperidol as the ninth compound. This cluster will also be discussed in more detail in terms of differentially expressed genes and pathways.

4.2.3 Differential Expression and Pathway Analysis

After clustering, it was investigated whether interesting genes and pathways could be found for the clusters and more importantly, if these results depend highly on the used methods. Finding genes of interest and related pathways was done with the help of the *DiffGenes* and *Pathways* functions. These functions give respectively per method an overview which genes and pathways are differentially expressed per cluster. The result of the fingerprints is used as a reference and determines the order of the compounds and the colours. A global overview of the gene expression and pathways per method can be found in Table 8. Per cluster it is summarized how many genes and pathways were found to be significant for a significance level of 0.05.

Further, it is also indicated how many of these were shared over all methods per cluster. However, it is seen that no single gene was found to be differentially expressed for any cluster. Therefore, the row “Shared” indicates between brackets the number of genes that would be shared if the top 10 differentially expressed genes were studied. Per cluster it is also indicated how many compounds were shared.

Table 9 shows genes of interest for every cluster. A gene is not necessarily found to be differentially expressed by each method. This might be due to the dissapearing or addition of only a few compounds to a cluster. The interesting genes are plotted in Figures 61 to 65. The compounds that are plotted to the left are those that are shared for that cluster over the methods for which genes were found. The dots in the plots indicate those compounds that are joined to the cluster for some of the methods. Finally, Table 10 summarizes the top three of shared pathways over the methods per cluster. This procedure was repeated for weighted clustering as well. A summary can be found in Table 11, Table 12, Figures 66 to 70 and Table 13. The focus here will be put on the previously selected cluster.

The compounds of the selection can be found in Table 1. It was decided to focus only on the cluster that contains the maximum number of compounds of the selection for the differential expression over the methods and weights in CEC and weighted clustering. The found differentially expressed genes and pathways are summarized in Table 14. Between brackets are the number of genes and pathways that are shared among those that actually find differentially expressed genes and pathways. For the CECc method, also the exclusion of weight 1.0 and 0.9 is taken into account. Putting the table aside the Figures 5, 7, 50 to 52 and 55 to 59, the following is seen. In comparison over all methods, it is only the clustering on fingerprints that has found no genes. As soon as quinpirole is removed and replaced by haloperidol, five genes show a differential expression. The result over the weights of the CECb method show an identical pattern. For CECa, first two compounds dissapear and no genes of interest were found. At a weight of 0.7 one of these is rejoined to the cluster with the extra addition of haloperidol. Five genes were now discovered. In the method CECc, it is seen that for weights one and 0.9 only five compounds of the selection are found together. These five reveal nine genes that are differentially expressed. At a weight of 0.7, again five genes were found. For the weighted clustering results, it is seen that if quinpirole is no longer present for weight 0.9 but is not replaced by haloperidol either, one gene is differentially expressed for the remaining eight compounds. All other weights join haloperidol to the cluster and the same five genes as for the other methods were found.

The finding of differentially expressed genes is greatly influenced by the removal of the compound quinpirole. With the remaining eight compounds, one gene (PI3) of interest is found. If the compound haloperidol is joined to these eight, five differential expressed genes are found. These are presented in Table 2.

	Genes		Genes
1	PI3	4	SRSF7
2	IDI1	5	PNO1
3	MSMO1		

Table 2: *The Interesting Genes for the Selected Cluster - MCF7*

The gene profiles of the interesting genes are plotted below in Figure 9. The compounds plotted to the left are the selected compounds without quinpirole but with haloperidol. The values of quinpirole and haloperidol are indicated with a dot.

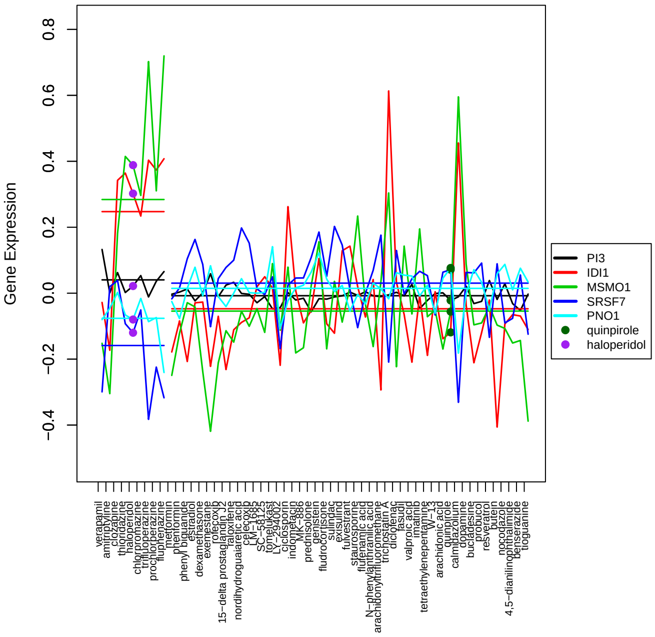


Figure 9: *The Interesting Genes for the Selected Cluster - MCF7*

The gene expression of the five genes is seen to be indeed different from the majority of the other compounds. Some genes are upregulated while others are down regulated. The CECc method reveals that when only five of the nine compounds are considered, nine genes are differentially expressed. The compounds and the genes are listed in Tables 3 and 4.

	Compounds		Compounds
1	fluphenazine	4	chlorpromazine
2	trifluoperazine	5	thioridazine
3	prochlorperazine		

Table 3: *The Compounds of the Selected Cluster for CECc in weight 1.0 and 0.9 - MCF7*

	Genes		Genes		Genes
1	MSMO1	4	HMGCS1	7	SRSF7
2	SQLE	5	INSIG1	8	HMGCR
3	IDI1	6	CCNG2	9	PPIF

Table 4: *The Interesting Genes for CECc for weights 1.0 and 0.9 - MCF7*

It is observed that three of the nine genes are also found for many of the other methods. Figure 10 shows the gene profiles of these nine genes making a distinction between the five compounds still found together and the others.

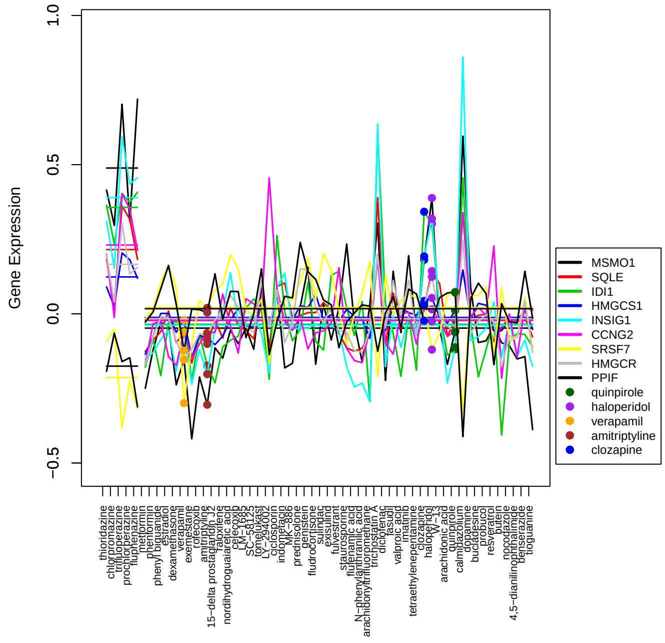


Figure 10: *The Interesting Genes for CECc for weights 1.0 and 0.9 - MCF7*

The number of found pathways lies around 53 for many methods. The top three for the selection with haloperidol but without quinpirole is shown in Table 5.

	Pathways
1	cholesterol biosynthetic process
2	sterol biosynthetic process
3	cholesterol metabolic process

Table 5: *The Shared Top 3 Pathways for the Selected Cluster - MCF7*

The first two pathways are still found when taking quinpirole into account. The third pathway is replaced by regulation of ATPase activity.

4.3 Inhouse1 Data

4.3.1 Clustering on a Single Source

The Inhouse1 data set included information on fingerprints and bio-assays for 94 compounds. In computing the dissimilarity matrices, the tanimoto coefficient was applied to the fingerprint matrix and the euclidean distance to the bio-assays. The clustering results on the data matrices separately can be found in Figures 71 and 72. The optimal number of clusters were respectively 12 and eight determined by the rule of Tibshirani.

The analysis of this data set continues with the average of 10 clusters.

As for the MCF7 data set, a comparison is made of the clustering on the two sources separately. Figure 71 indicates the colouring of the several clusters and Figures 72 and 73 show how the grouping of the compounds has changed. This is even more clearer depicted in Figure 11

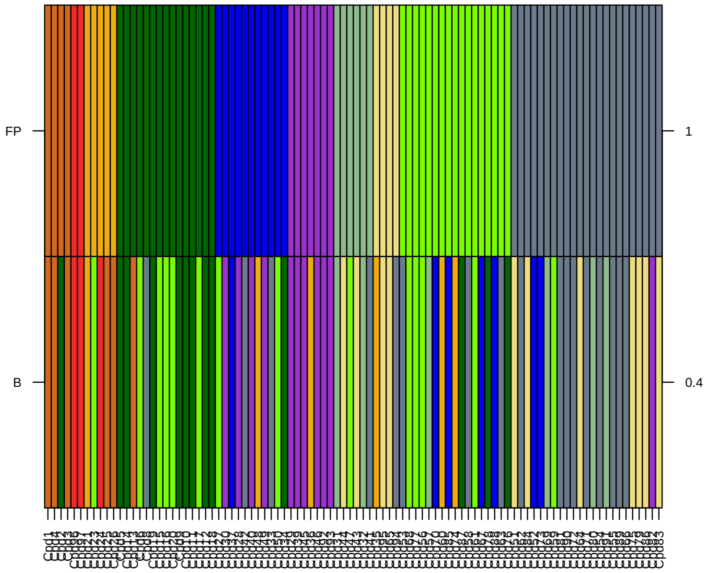


Figure 11: *A Cluster Comparison between Fingerprints and Target Predictions for Inhouse1*

It is observed that for this data set, the clustering of the compounds differs reasonably between the sources of information since many compounds have changed clusters. However, smaller groups of compounds do seem to appear together as for example the brown, red and purple cluster and parts of the light green and grey cluster. This implies that smaller groups of compounds have similar characteristics on both fingerprints and bio-assays. It is expected that these little groups will be preserved among the other techniques in which the information of the data matrices is combined.

4.3.2 Multi-Source Clustering

The reference method for integrative results of the Inhouse1 data is WonM as aggregated data clustering could not be performed. A dendrogram of the result is shown in Figure 12.

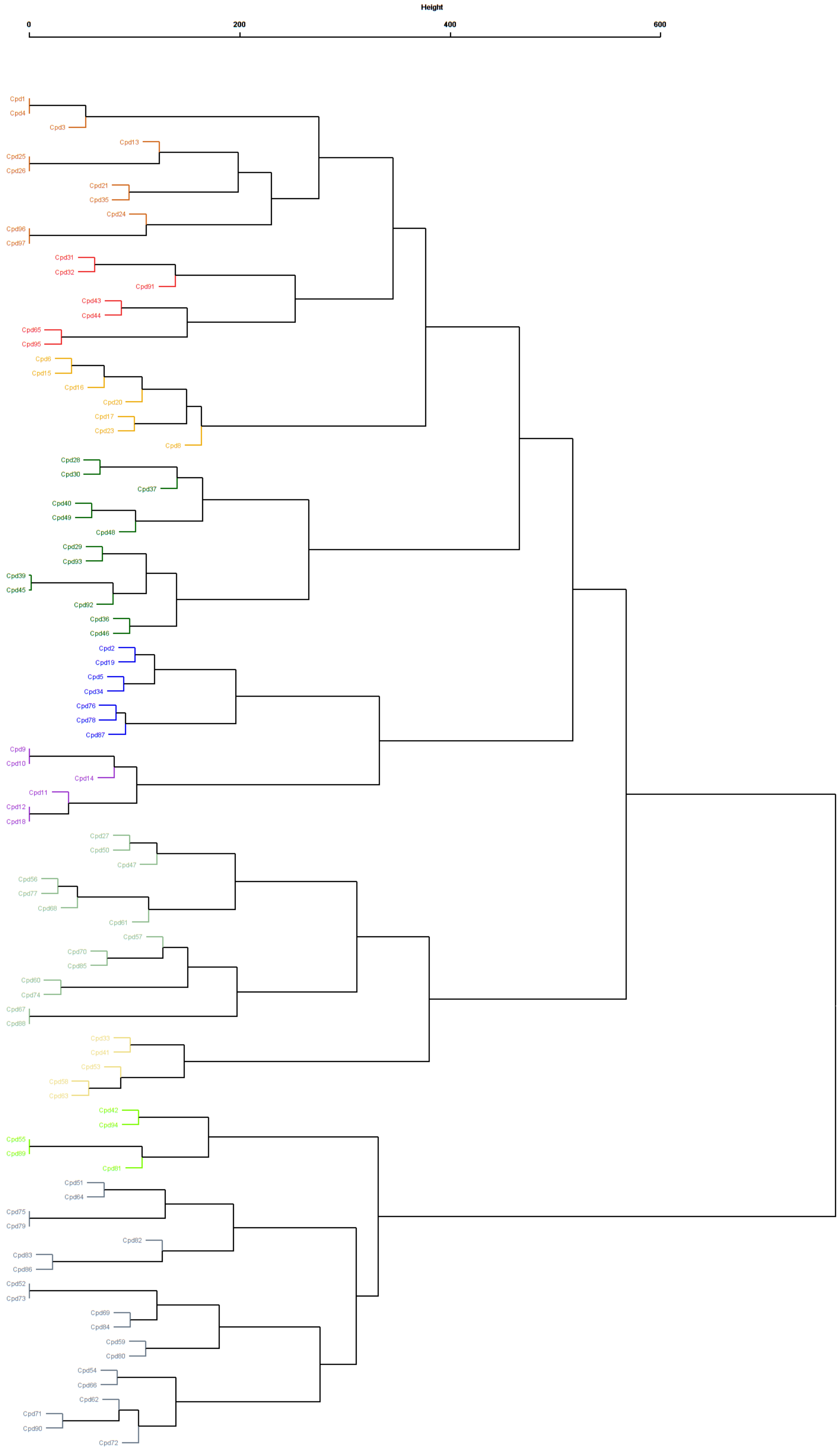


Figure 12: *The Dendogram based on WonM for Inhouse1*

The influence of the weights for each version of CEC is studied in Figures 74, 75 and 76. In comparison with the results of fingerprints and bio-assays, the influence of the bio-assays increases as more weight is appointed to this data source. Smaller groups of compounds, in groups of two and three, shift clusters as the weight decreases. Also here, the line for version b can be drawn clearly. More colours arise in these results, this is due to the fusion or complete segregation of multiple clusters compared to the reference method. For example, clusters one, two and three in version b were fused together and cluster seven and eight as well. This implies that three clusters dissapear as a separate unit and three colours are necessary to indicate those that are not matched appropriately. Figure 77 shows the results of the weighted clustering versus the fingerprint clustering. As soon as information from the bio-assays is involved, the clustering changes greatly. This implies that the data matrices bring forth different aspects of information on the compounds and the biological information of the bio-assays has great influence. The clustering results share a higher similarity with the clustering on bio-assays than with the fingerprint clustering as is seen in Figure 78. The results for a weight of 0.5 will be used is the comparison over all methods.

A comparison over all integrated methods is made in Figure 13 with the result of the WonM as reference.

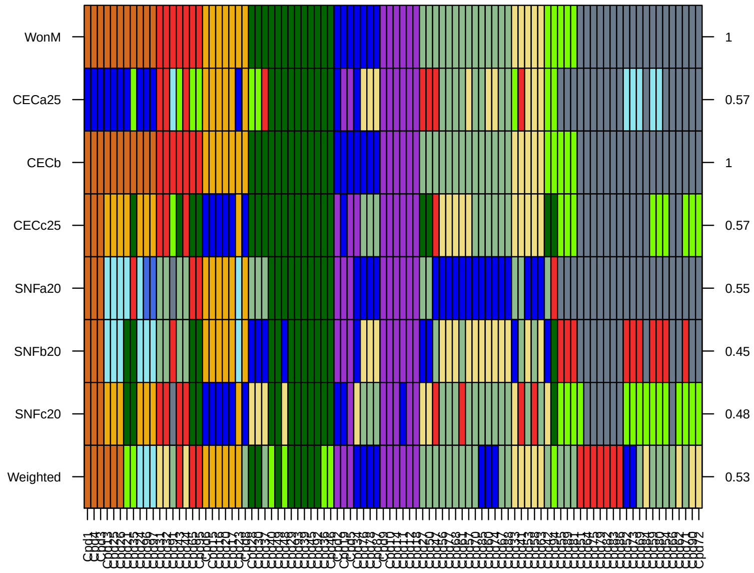


Figure 13: *Comparison of All Methods versus WonM for Inhouse1*

The similarity between the methods is lower compared to the MCF7 data. The method of CECb gives also here a clustering identical to WonM while CECa and CECc agree rather well on which smaller groups of compounds are to be put together. The same holds for the results of SNF. Although compounds are often given a different cluster colour for CEC and SNF, there is a similarity in which compounds form that cluster. Their resemblance is presented in Figure 79. Although the weighted clustering for the MCF7 data showed similarities with CEC and SNF, this is less the case for the Inhouse1 data. The weighted clustering has similarities with every other method.

By studying the results of the separate clusterings, it is seen that both sources of information contribute their share in each integrated method. This is seen in Figure 14.

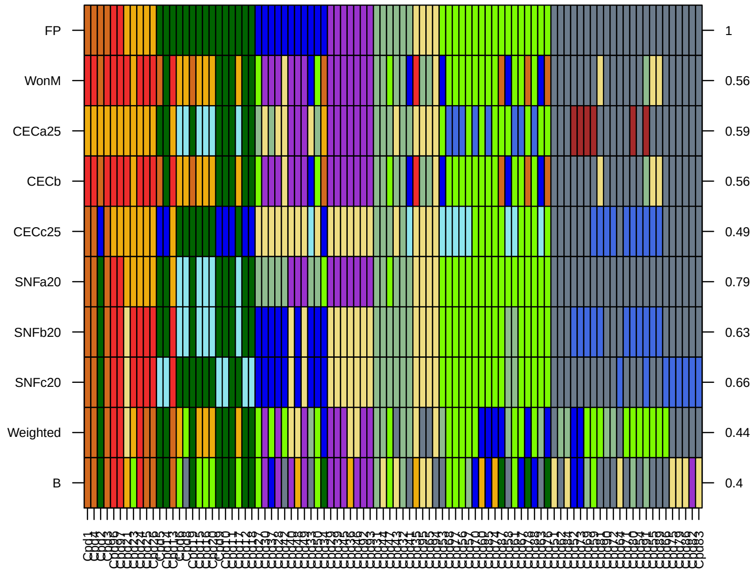


Figure 14: *Comparison of All Methods versus F for Inhouse1*

It is SNFa that shows the highest similarity with the fingerprint clustering and the weighted clustering with the bio-assays. WonM shows both similarities and differences with the results of the fingerprints and bio-assays. The interest lies particularly in those clusters formed on fingerprints that also arise in clustering on bio-assays. This implies that the fingerprint structure is related to the biology. An example is seen for cluster six (purple) and cluster three (yellow) which do not undergo a lot of shifting over the different methods unless in their whole. However, this is also seen for parts of the other clusters as clusters eight, nine and 10. For the latter clusters, large parts remain together except for weighted clustering and clustering on bio-assays.

The selection of the most stable cluster is harder for the Inhouse1 data than it was for the MCF7 data. Fusions and complete segregations have occurred in some methods and therefore not every reference cluster has found a suitable match. Visually, it can be seen in Figure 14 that the compounds of cluster six are, expect for the weighted clustering and the clustering on bio-assays, always found together although it is not always the same cluster. This is an indication that for these compounds the information on fingerprints and bio-assays is related. Therefore, this selection will be followed over the several methods and weights of CEC and weighted clustering. The selected compounds are presented in Table 6.

	Compounds		**Compounds**
1	Cpd29	5	Cpd46
2	Cpd39	6	Cpd92
3	Cpd45	7	Cpd93
4	Cpd36		

Table 6: *The Selected Cluster for the FP of Inhouse1*

The compounds were followed over the different methods and its alterations were recorded in Figures 15 and 16.

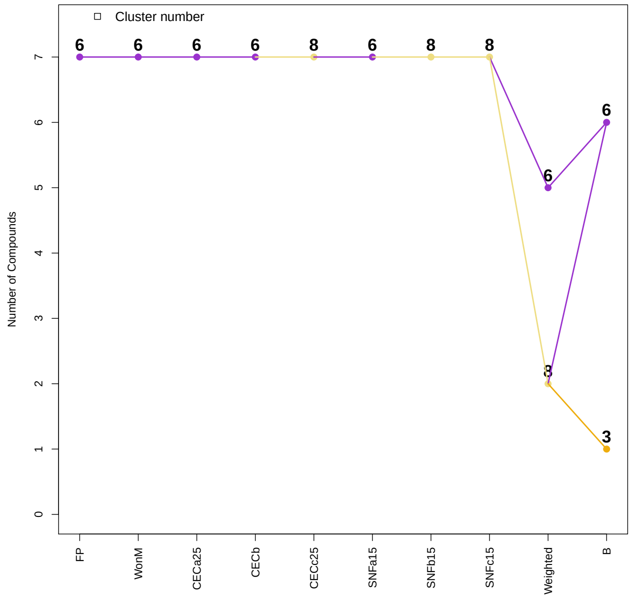


Figure 15: *The Evolution of Cluster 4 of FP over All Methods for Inhouse1*

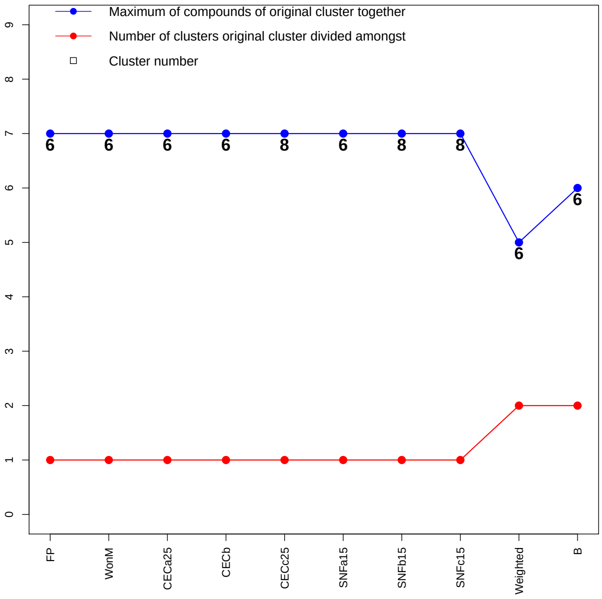


Figure 16: *The Evolution of Cluster 4 of FP over All Methods for Inhouse1 - Maximum Number of Compounds*

The seven compounds remain together in the same cluster, whether it is cluster six or eight, except for the weighted clustering and the clustering on bio-assays. For these, still respectively five and six compounds are grouped together. The two compounds that dissapear are Cpd36 and Cpd46 although Cpd46 rejoins the cluster for the bio-assays. The alteration of cluster number is due to the fusion with cluster eight and this is seen for CECa, SNFb and SNFc. Further, the compounds that shift cluster for the weighted clustering are joined to those of cluster eight. This cluster has four compounds which are Cpd35, Cpd95, Cpd65 and Cpd94. This might be an indication that the compounds found in cluster eight show similarities to the selected compounds for the fingerprints and bio-assays. The evolution of the cluster over the weights is tracked in Figures 17 and 18.

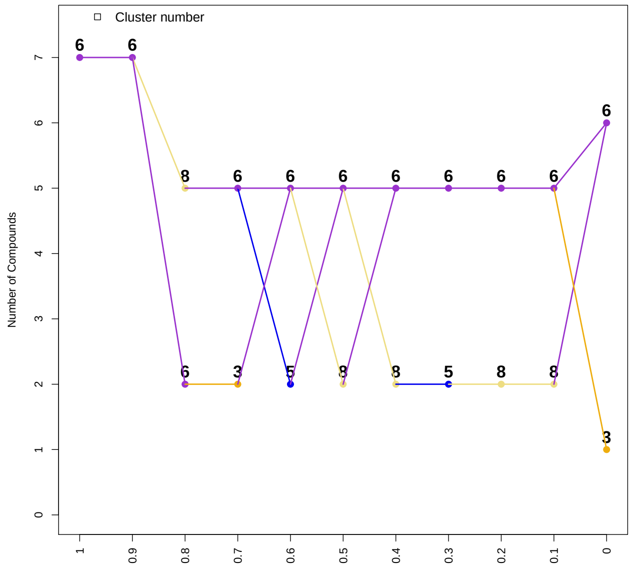


Figure 17: *The Evolution of Cluster 6 of FP over Weighted for Inhouse1*

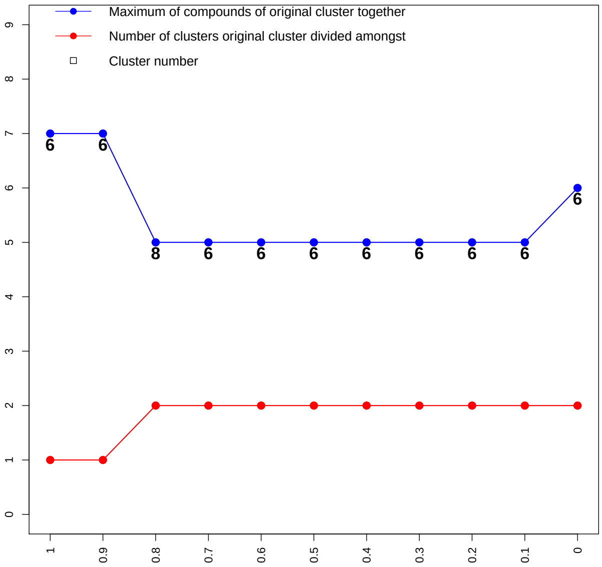


Figure 18: *The Evolution of Cluster 6 of FP over Weighted for Inhouse1 - Maximum Number of Compounds*

The compounds get maximally divided over two clusters and at least five compounds were always found grouped together. It is seen that for the weights 0.7 to 0.4 the methods show a disagreement on which compounds should be appointed to a different cluster. Also here, cluster eight is mentioned a number of times and it are either compounds Cpd36 and Cpd46 or compounds Cpd29 and Cpd93 that shift cluster. The selection was also followed for the weights over the three versions of CEC and the results are presented in Figures 80 to 85. It is observed that for all three methods the complete selection is found eventually. For CECa, often six compounds are together although it are not always the same six compounds. Only for weights 0.6 and 0.5 all seven compounds form a cluster. When the weight decreases further, the compounds are divided over cluster six and eight. Over the weights in CECb, it is seen that one compound dissapears under the influence of the target predictions after the half way point of weight 0.5. This is Cpd36 that also shifts cluster for the bio-assays. The method CECc shows a similar pattern as CECa and finds the whole selection for weights between 0.8 and 0.3 which form a part of cluster six or eight as can be seen in Figure 76. It are always the same five compounds that can be found together. For the differential expression, the focus lies on the cluster that contains the maximum number of compounds of the selection.

4.3.3 Differential Expression and Pathway Analysis

After obtaining the clustering results, differential expression of genes was investigated. Since for the Inhouse1 data set the names of the genes were masked a pathway analysis could not be carried out. The result of the fingerprint clustering will be used as the reference method. Table 15 shows how many genes were found to be differentially expressed for each cluster in each method for a significance level of 0.05 and how many are shared for this significance level and the top 10 genes respectively. Further, the same information on the compounds is contained in this table. It is noted that some methods did not have a specific cluster number. Either due to fusion or complete segregation of the cluster. If this was the case, a “-” indicates that the cluster was missing and not will not be taken into account when looking for shared genes over the methods. For some clusters, many genes were found to be differentially expressed but not for every method. The reason is that for some methods certain compounds are not present in the cluster of interest.

Interesting genes are summarized for each cluster in Table 16. It is possible that one cluster has merged with another cluster. A consequence is that a gene can be found to be differentially expressed for a specific cluster in one method and for a different cluster in another method probably because the cluster contains mainly the same compounds. If this is the case, the cluster is indicated with an underscore. Figures 86 to 92 show the interesting genes per cluster. Since for cluster one many significant genes were found for the fingerprints but none of these were shared for the other methods, a separate plot in Figure 87 shows the top five genes of the fingerprint clustering. The differentially expressed genes and genes of interest were also summarized for the weighted clustering in Tables 17 and 18 and Figures 93 to 100. The previously selected compounds will be discussed in detail next.

The selected cluster can be found in Table 6. The gene expression was investigated for the cluster that contained the maximum number of compounds of the selection. The results are summarized in Table 19. Between brackets are the number of genes that are shared among those that find differentially expressed genes and the number shared when excluding the methods that only find one gene of interest. With the help of Figures 14, 15, 17, 74 to 76 and 80 to 84, the following can be said.

Following the cluster over all methods, it can be seen that the compounds stay together expect for the weighted clustering and the bio-assays. The compounds that are joined to the selection influence the differential expression of the genes. The selection on its own, as it can be found for the fingerprints, finds one gene of interest. The same is seen for a weight of one to 0.6 in the clustering for CECb and a weight of 0.3 in the results of CECc. This one gene is Gene494 and is discovered by every method and weight that finds differentially expressed genes.

The methods SNFa, CECa for weights 0.6 and 0.5 and weighted clustering for a weight of 0.9 all find 50 differentially expressed genes. For these clustering results, three extra compounds are joined to the selection. These compounds are Cpd40, Cpd49 and Cpd48. For methods SNFb and SNFc, cluster six of the fingerprints is fused to cluster eight together with extra compounds Cpd40 and Cpd49. Here, eight genes of interest were found. The results for weights 0.5 and 0.7 for the weighted clustering are identical to those of CECc for weights 0.1 and zero. Compounds Cpd30, Cpd28 and Cpd49 are merged to the selection while Cpd36 and Cpd46 dissapear. The limma method finds 71 genes to be differentially expressed for this new cluster. The CECa methods removes Cpd45 from the cluster and joins Cpd40, Cpd48 and Cpd49 to the selection for weight one to weight 0.7. A total of 80 genes were discovered as significant. For a weight of 0.4 in the CECb method, Cpd45 dissapears from the cluster while Cpd40, Cpd49 and Cpd80 are appointed to it. This results in 65 differentially expressed genes. Finally, for weight 0.9 in the CECc method, 29 genes were discovered after the dissapearing of compounds Cpd92 and Cpd93 and the addition of five extra compounds: Cpd28, Cpd42, Cpd40, Cpd48 and Cpd49. The other resulting clusterings of the selected compounds found no genes of interest under the influence of the merged compounds. The compounds and the top five of the significant genes for each method that finds differentially expressed genes are summarized in Table 20.

The compounds that seem to be the most common are presented in Table 7. These consists of the selection and three compounds joined extra to them under the influence of the bio-assay information.

	Compounds		**Compounds**		**Compounds**
1	Cpd29	5	Cpd46	9	Cpd48
2	Cpd39	6	Cpd92	10	Cpd49
3	Cpd45	7	Cpd93		
4	Cpd36	8	Cpd40		

Table 7: *The Most Common Compounds for The Selected Cluster - Inhouse1*

All differentially expressed genes were compared over the methods. When all methods and weights were considered, it was found that three genes were discovered for each method. The gene profiles of Gene494, Gene954 and Gene307 for the most common compounds are shown in Figure 19. When excluding the methods SNFb and SNFc, 22 genes were in shared. The gene profiles of the top five genes is plotted in Figure 20.

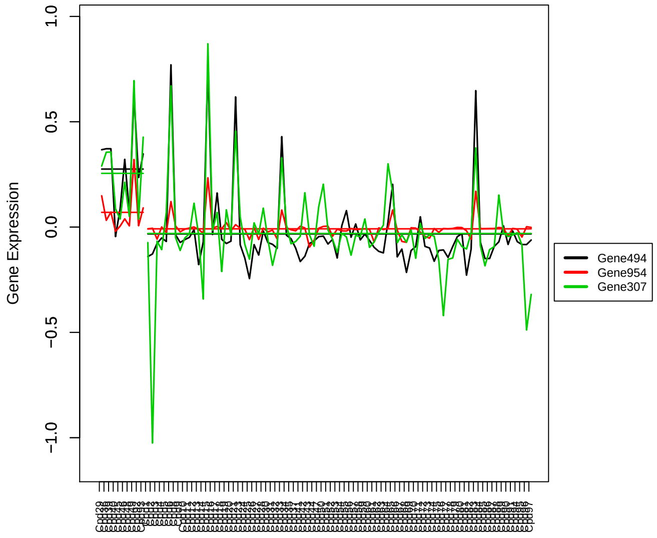


Figure 19: *The Gene Profiles for the DE Genes Found by Each Method - Inhouse1*

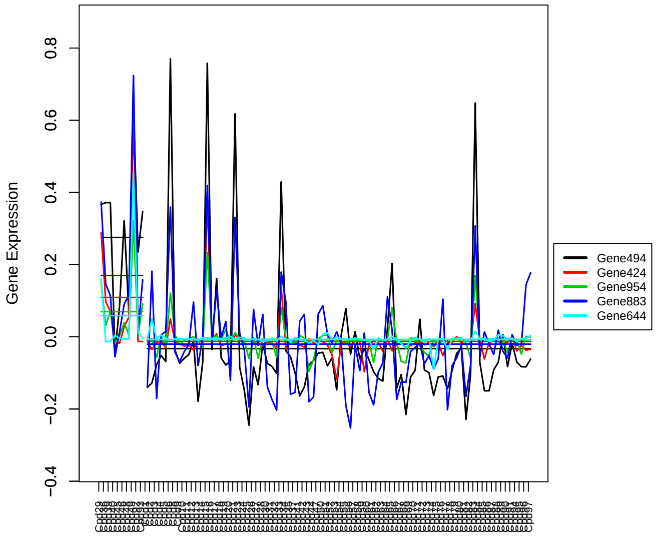


Figure 20: *The Gene Profiles for the DE Genes Found by Each Method Excluding SNFb and SNFc*

The figures show a difference between the gene expression of the selected compounds and the others. The three compounds joined the selection contribute to the discovery of interesting genes. The compounds Cpd40, Cpd48 and Cpd49 are often found in the same cluster, especially Cpd40 and Cpd49. It is under the influence of the bio-assays that these compounds are merged together.

5 Software Development

During this project a number of functions were written to facilitate the analysis and help in the comparison over the results. The functions concern execution of the methods, differential expression, pathway analysis, comparison functions and visualization functions. All were written in R version 3.1.0. and bundled into a package called *IntClust*. A help file for each function is available in the Appendix.

For clustering on a single source the function *Cluster* can be used and every multi-source method has it own function. The functions carry the same name as the reference used for the methods in this project. The user only needs to provide the data sets and specify available options. The result is a list which contains elements belonging to a specific method. The resulting clustering of the procedure can always be found under the element “Clust”. Examples are given for the function *Cluster* and *SNFa* on the MCF7 data.

```
MC7_F = Cluster(fingerprintMat,distmeasure="tanimoto",clust="agnes",
                linkage="ward",gap=FALSE,maxK=55)


MC7_SNFa15=SNFa(list(fingerprintMat,targetMat),distmeasure=c("tanimoto",
                "tanimoto"),NN=15,alpha=0.5,T=20,clust="agnes",linkage="ward")
```

After obtaining the different clustering results, a comparison can be made. However, this is not done easily. It has to be taken into account that different methods, have a different ordering of compounds and therefore a different numbering of clusters. Therefore the *MatrixFunction* function was written which takes one method as a reference and rearranges the cluster numbers of the other methods to this reference by looking for the reference cluster they have the most in common with relatively. The result of this function is a matrix of which the columns are the compounds in the order of clustering by the reference method and the rows represent the methods compared to it. Each cell contains the number of the cluster the compound is in for that method compared to the reference.

```
L=list(MC7_F,MC7_ADC,MC7_ADECa25,MC7_ADECb,MC7_ADECc25,MC7_CECa25,MC7_CECb,MC7_CECc25,
       MC7_SNFa15,MC7_SNFb15,MC7_SNFc15,MC7_WonM,MC7_Weighted,MC7_T)


names=c("FP","ADC","ADECa25","ADECb","ADECc25","CECa25","CECb","CECc25","SNFa15","SNFb15",
        "SNFc15","WonM","Weighted","TP")


MCF7_Matrix=MatrixFunction(L,nclusters=7,fusionsLog=TRUE,WeightClust=TRUE,names=names)
```

The *MatrixFunction* is employed by many other functions. In the visualization function *ComparePlot*, basically every cell of the matrix is given a colour corresponding to its value. A comparison over the methods can than be made visually.

```
ComparePlot(L,nclusters=7,cols=my_palette1a,fusionsLog=TRUE,WeightClust=TRUE,names=names,
            reverse=FALSE,margins=c(9.1,4.1,4.1,4.1))
```

The function *DiffGenes* and*PathwaysIter* also make use of *MatrixFunction*. Before applying the limma method or pathway analysis, the clusters are rearranged to the reference such that a comparison of the differential expression can be made. The function *Genset.intersect* takes the intersection over the number of loops specified in *PathwaysIter*.

```
MC7_DiffGenes_0.05=DiffGenes(L,geneMat,nclusters=7,"limma",0.05,top=NULL,fusionsLog=TRUE,
                             WeightClust=TRUE,names=names)
MC7_Paths=PathwaysIter(L,GeneExpr=geneMat,nclusters=7,method=c("limma", "MLP"),
                       ENTREZID=GeneInfo[,1], geneSetSource = "GOBP",GENESET=GS,
                       top=NULL,sign=0.05,niter=10,fusionsLog=TRUE,
                       WeightClust=TRUE)
MC7_Paths_intersection=Geneset.intersect(MC7_Paths_All10F,0.05,seperatetables=FALSE,
                                         separatepvals=FALSE)
```

The function *Shared* takes the output of the *DiffGenes* and*Geneset.intersect* and sets up a table that contains how many genes and pathways were found to be significant for each method. Further, it indicates how many of these were shared. It provides the same information over the compounds and a list is returned containing which compounds, genes and pathways are shared among the methods.

```
MC7_Shared=Shared(DataLimma=MC7_DiffGenes_0.05,DataMLP=MC7_Paths_intersection)
```

Disclaimer

All described functions were developed and tested on the MCF7 and Inhouse1 data sets. It is possible however, to encounter a situation for which the functions will not perform properly. This is especially true for the *MatrixFunction* whose algorithm is based on the perfect-matching algorithm.

6 Discussion

Clustering was first performed on each data source separately. The downside of the technique is that only one side of the reality is investigated. Since this gives a limited point of view, a interest has been taken in combining information from multiple sources. This is why integrative data analysis is important. Involving multiple sources of data reveals which compounds are similar to one another on different aspects. It also shows if and how sources are related for these compounds. When a similar grouping of compounds is found for each data source separately, the compounds are likely to be found together in the integrative data analyses as well. It will help create a global view of the mechanism of action of the compounds and reveal the underlying biology. Several integrative data methods were applied to the data sets.

In the ADC, ADECa, ADECb and ADECc methods, the available data matrices are combined into one larger matrix. Clustering and respectively ensemble clustering is then performed on more variables simultaneously. These techniques can only be applied if the data sources are all of the same type. In the three versions of the ADEC method, the number of iterations could be specified as well as a specific number of features to draw at each iteration. The sensitivity analysis revealed that the number of iterations should not be too few. Further, whether the number of features is fixed or not, compounds that are similar for many variables and distinguish themselves will form a group together. For others, the final cluster will depend on which and how many times features are involved. ADECa, ADECb and ADECc showed similar clusterings of compounds.

WonM sets up an incidence matrix for each number of clusters the dendrograms of the single data matrices are cut into. These are summed and the method thus tracks how many times compounds are found in the same cluster. The higher the number of clusters the dendrogram is split into, the smaller the groups of compounds. If compounds are still found in the same group for a high number of clusters, the closer these are together. This is a similarity measure and therefore the final clustering on the sum of the incidence matrices is performed on a similarity base.

The other ensemble clustering technique is CEC of which three versions were implemented as well. The sensitivity analysis showed the same conclusion as for ADEC. As long as the number of iterations is not too low, enough information surfaces to result in an adequate clustering. The fixation of the number of sampled features seems to have no great influence. An extra parameter is the weight to be given to each data source in the linear combination. This is a question also to be answered in the weighted clustering technique. Determination of an optimal weight remains a challenge. It can be chosen to give a higher weight to the data source with the most stable clusters or to try out a few values and discover the influence of the weight. However, the weight can also be optimized with the help of the EM-algorithm as described by Xu et al. (2012).

The SNF technique was recently developed by Wang et al. (2014) and relies on updating a global network with local information. The parameters of choice were the hyperparameter μ, the number of iterations and number of neighbours. Wang et al. recommend boundaries for each parameter and state that between these limits, the method is insensitive to the choice. The hyperparameter μ was therefore set to 0.5, the number of iterations to 20 and the sensitivity analysis showed that respectively 15 and 20 neighbours for the data sets were appropriate. Between SNFa, SNFb and SNFc, a high agreement was met although they are different interpretations of the SNF method. There was a concern about the implementation of the functions in the SNFtool package. Its results however, agree with the results of the methods as described in the methodology. In all methods the actual grouping of the compounds was performed by agglomerative hierarchical clustering. If distance matrices were calculated from the data, these were obtained by applying the tanimoto coefficient for binary data sets and the euclidean distance for continuous values. It was necessary for some methods and for the later comparison of the results to determine an appropriate number of clusters for each data set. With the help of the rule of Tibshirani and the clustering on the singe sources of information, a suitable number of clusters was determined. The choice fell on seven clusters for the MCF7 data and 10 for the Inhouse1 data.

In a comparison over all methods and both data sets, general remarks can be made. By involving the results of the separate clustering, the influence of each data modality can be seen. Over the weights in CEC and weighted clustering, some compounds change cluster as soon as the other source of information is involved. Others remain together for the different weights. This depends on how much of the information is shared between the data matrices. The higher the weight for a source of information, the higher its influence. The results of CECa and CECc agree rather well. Especially for the MCF7 data. For the Inhouse1 data, they do not show a similarity as high as for the MCF7 data. They do not always agree on the cluster number but rather on which compounds to put together. Looking at the clustering from a more global point of view shows that smaller groups of compounds are often found in both methods. The same conclusion holds for the results of SNF. A high similarity is seen over SNFa, SNFb and SNFc for MCF7 while the methods agree on smaller groupings in the clustering for Inhouse1. The result of CECb and WonM is identical for both data sets. Seeing a high degree of resemblance is not unexpected. Both are based on splitting the dendrogram into a specific number of clusters several times. Further, the contribution of both information sources is identical here since the result of CECb shown is the one for a weight of 0.5. CECb deviates from the results of CECa and CECc due to the resampling in the latter methods.

The ADC and ADEC methods could only be applied on the MCF7 data sets. The methods show a high degree of resemblance with each other and with the clustering on fingerprints. It seems that in the aggregated data techniques, the information of the fingerprints dominates the grouping and only the most prominent information of the target predictions comes through. Overall, the results of the MCF7 data show a high similarity. Some clusters are preserved over the methods, others undergo only minor changes under the influence of the target predictions and often compounds are shifted in small groups. For some compounds, it will be the fingerprints that give more information on their resemblance while for others it will be the target predictions. Depending on the method of combination, compounds that show no remarkable similarity to each other on either data source, will be dominated by one or the other. The fingerprints were actually used to calculate the target predictions and therefore the sources are automatically related. Therefore these bring forth similar information on the compounds and such a high similarity is seen.

The data sources for the Inhouse1 data are not known beforehand to be related but it is the goal of the integrated analysis to discover a relation between the fingerprints and the bio-assay scores of the compounds. The results are therefore a lot less similar. The fingerprint and bio-assay informations resulted in many smaller groups of compounds with similar values. This implies that if one compound shifts cluster, its neighbouring compounds will as well. This indicates that these compounds are all similar but the combination of the sources determines to which cluster they are appointed. The difference between these might only be minor.

The goal of the different integrated data analyses is to reveal clusters of compounds that are stable over the several methods. This implies that independent of the used technique, these compounds were found to be similar on different aspects and that a connection can be seen between the structure of the compounds and the predicted targets or the bio-assays. A stable cluster was selected for each data source and discussed in detail in the results section. Under the influence of the data source, compounds dissapear and get joined to it. This influences the differential expression of the cluster. For MCF7, the compound quinpirole leaves the cluster as soon as target predictions are involved. Haloperidol is merged to the cluster and the differential expression changes from zero genes found to five genes. In the Inhouse1 data, the selection of compounds is mostly retained with the addition of three extra compounds. The gene expression changed from one to multiple genes.

7 Conclusion

Understanding the mechanism of a drug is a crucial step in the development of new compounds. It is important to know what the drug will react to and what its actions to this trigger will be. This all to avoid side effects and to see whether or not the drug accomplishes what is has been created for to do. Unraveling this is not always easy but a great help in this process is the comparison with other compounds that are similar but not exactly the same or completely different compounds. A deduction of what some compounds share but others do not, is often a good step in the right direction. The approach in this project was to rely on clustering techniques to found groups of similar objects.

Clustering on a single source of information gives a limited point of view. Therefore a interest has been taken in combining information from multiple sources. This is why integrative data analysis is important. The combination of multiple aspects of the compounds helps to discover the underlying biology of their actions. By comparing the clustering on the separate sources with the integrative analyses, the influence of each source can be seen. If parameters such as the number of iterations, number of features to sample, number of neighbours or weights need to be specified, it is recommended to try different values to establish a lower limit. The rule of thumb is “not too few”. A best integrative method is not declared. Rather interest lies in cluster that are found to be stable over the different methods. These compounds are indicated to be similar on different aspects of the underlying biology. It can than be hypothesized that the data sources are related for those compounds. If compounds do not show a clear resemblance to one or possibly multiple groups, they can be clustered differently for each method. If the available data of the compounds is known to be related, it can be expected to see a number of similarities over the method. If, however, this is unknown it is interesting to look for stable clusters. It was seen that the differential expression of the cluster is greatly influenced by the compounds joined to them.

8 References

FODEH, J. S., BRANDT, C., LUONG, B. T., HADDAD, A., SCHULTZ, M., MURPHY, T., KRAUTHAMMER, M. (2013). Complementary Ensemble Clustering of Biomedical Data. *J Biomed Inform.* 46(3) pp.436-443.

HENS, N. (2014). *Data Mining: course text*, lecture notes on unsupervised learning at Hasselt University, Diepenbeek during the second term of the academic year 2014.

LI, Y., KU, T., ZHENG, S., WANG, J., LI, Y., HAO, P., LI, X. (2011). Association of Feature Gene Expressions with Structural Fingerprints of Chemical Compounds. *Journal of Bioinformatics and Computational Biology.* 9(4) pp. 503-519.

SMYTH, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology.* 3(1).

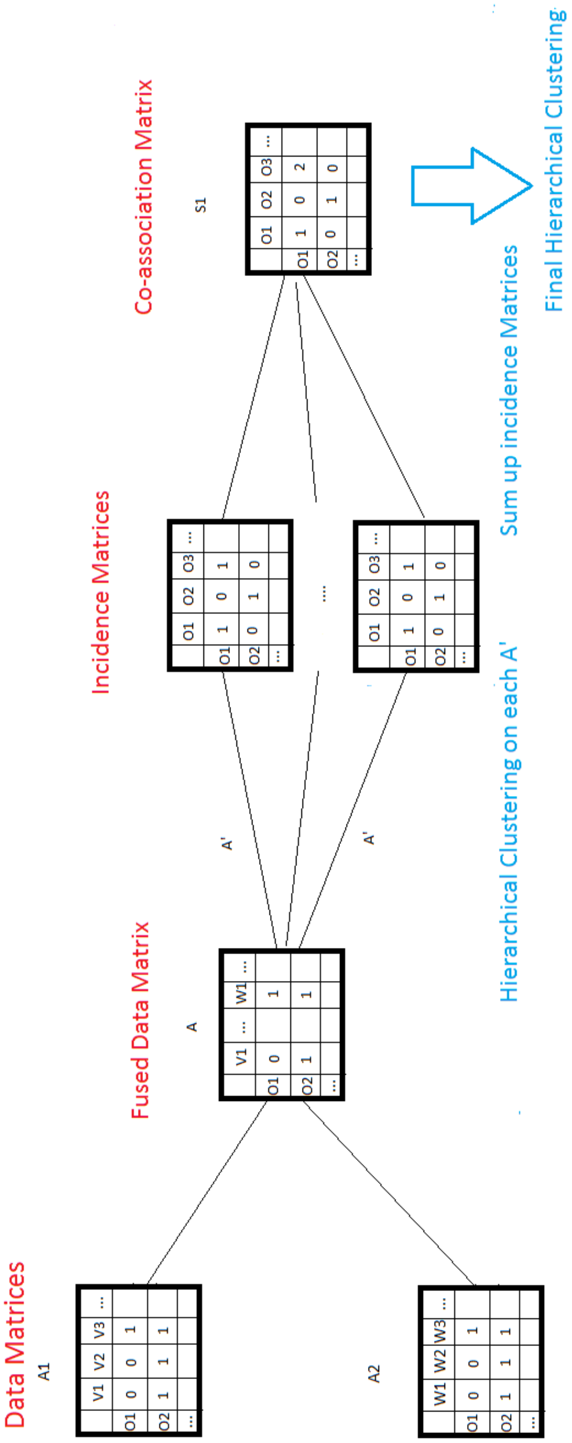
WANG, B., MEZLINI, M. A., DEMIR, F., FIUME, M., TU, Z., BRUDNO, M. HAIBE-KAINS, B., GOLDENBERG, A. (2014). Similarity Network Fusion for aggregating data types on a genomic scale. *Nature.* 11(3) pp. 333-337.

XU et al. (2012). Quantitatively integrating molecular structure and bioactivity profile evidence into drug-target relationship analysis. *BMC Bioinformatics.* 13(75).

9 Appendix

9.1 Methodology

ADEC


Data Matrices
A1
A2
Fused Data Matrix
A
A'
A'
Incidence Matrices
...
Co-association Matrix
S1
Hierarchical Clustering on each A'
Sum up incidence Matrices
Final Hierarchical Clustering

	V1	V2	V3	...
O1	0	0	1	
O2	1	1	1	
...				

	W1	W2	W3	...
O1	0	0	1	
O2	1	1	1	
...				

	V1	...	W1	...
O1	0		1	
O2	1		1	
...				

	O1	O2	O3	...
O1	1	0	1	
O2	0	1	0	
...				

	O1	O2	O3	...
O1	1	0	1	
O2	0	1	0	
...				

	O1	O2	O3	...
O1	1	0	2	
O2	0	1	0	
...				

Figure 21: *Aggregated Data Ensemble Clustering*

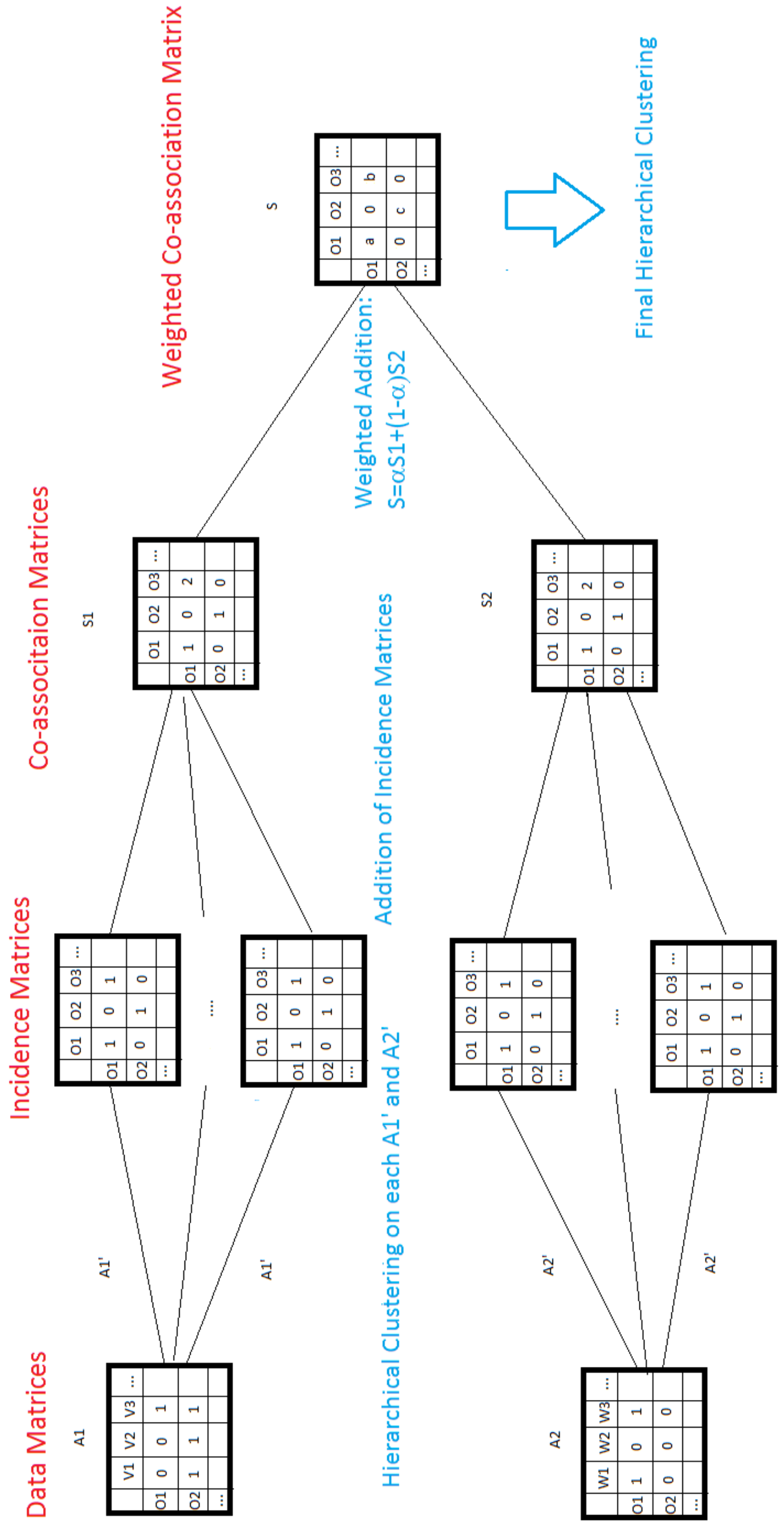
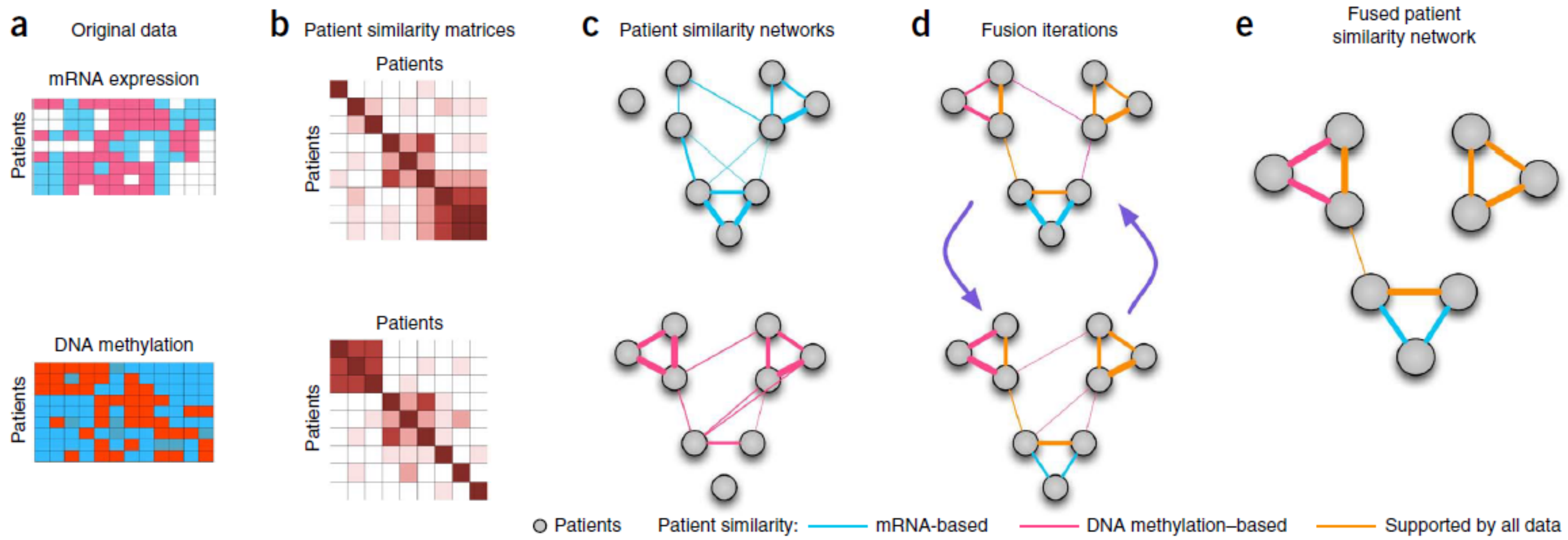


Figure 22: *Complementary Ensemble Clustering*



Illustrative example of SNF steps. (**a**) Example representation of mRNA expression and DNA methylation data sets for the same cohort of patients. (**b**) Patient-by-patient similarity matrices for each data type. (**c**) Patient-by-patient similarity networks, equivalent to the patient-by-patient data. Patients are represented by nodes and patients’ pairwise similarities are represented by edges. (**d**) Network fusion by SNF iteratively updates each of the networks with information from the other networks, making them more similar with each step. (**e**) The iterative network fusion results in convergence to the final fused network. Edge color indicates which data type has contributed to the given similarity.

Figure 23: *Similarity Network Fusion*

An overview of the differences and peculiarities of the functions in the *SNFtool* is presented below. The *affinityMatrix* function is used to compute the similarity function W after receiving the distance matrix:

- affinityMatrix function:
    - The input distance matrix and its transpose are added and divided by 2 in the same step.
    - In computing the parameter $\epsilon_{i,j}$, the necessary means are taken beforehand and only the average of the 3 elements in the formula should be taken. The first element however, is summed with itself, divided by 2 and then multiplied by 2 again. This is unnecessary.
    - The values of W are scaled by the density of the normal distribution and not the kernel. The values therefore deviate with a factor $\frac{1}{(\epsilon_{i,j} \cdot \mu \cdot \sqrt{2 \cdot \pi}}$.
    - The resulting matrix W is added to its transpose and the sum is divided by 2.
- SNF function:
    - The normalization in computing the P matrix is different from the outline in the methodology. The similarity matrices are divided by $\sum_{k=1}^{k=N} W(i,k)$ with N the total number of objects instead of excluding the object i and multiplying by 2.
    - The matrix P and its transpose are added and the sum is divided by 2.
    - The internal function *.dominateset* is used to determine the neighbourhoods of the objects and calculate the matrix S. The subsets of neighbours are determined from the normalized similarity matrices P and the resulting matrix is again normalized by dividing each value by the sum of its row. Thus 2 normalizations are performed here to compute the kernel matrix S.
    - In every iteration, the value 1 is added to each diagonal element. This is probably to ensure that each object is the most similar to itself but when looked at the values without this addition, it is was not discovered that this was not automatically the case.
    - In every iteration, the matrix $P^{(t)}$ was added to its transpose and the sum was divided by 2.
    - After every iteration, normalization was not performed.
    - When the overall status matrix $P^{(c)}$ was obtained, another normalization was performed.
    - After this normalization, the resulting matrix was added to its transpose and a value of 1 was added to each diagonal element as well. The sum was divided by 2 and this final matrix is returned to the user for clustering.

If the distance measure is symmetric, taking the average of a matrix and its transpose will not make a difference. However, this might be important if the distance matrix is not symmetric.

9.2 Sensitivity Analysis

9.2.1 ADEC

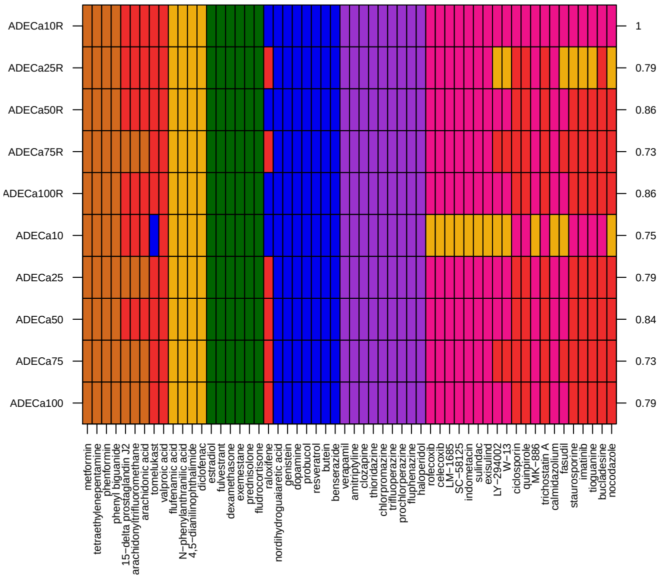


Figure 24: *Comparison for ADECa over Different Iteration Numbers for MCF7*

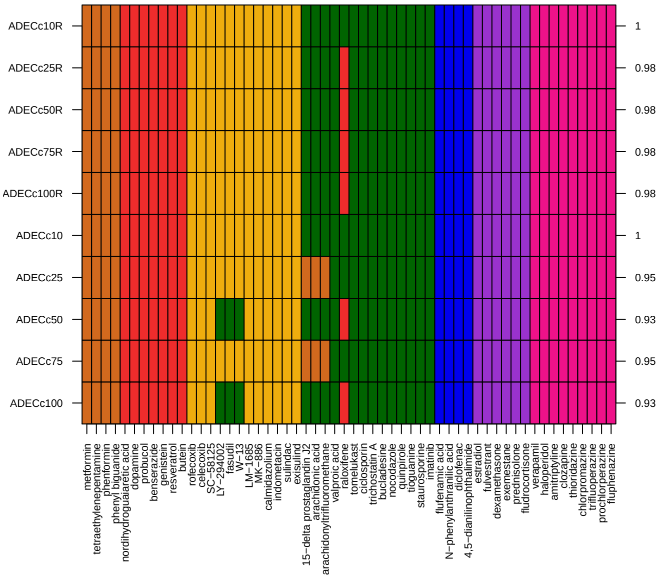


Figure 25: *Comparison for ADECc over Different Iteration Numbers for MCF7*

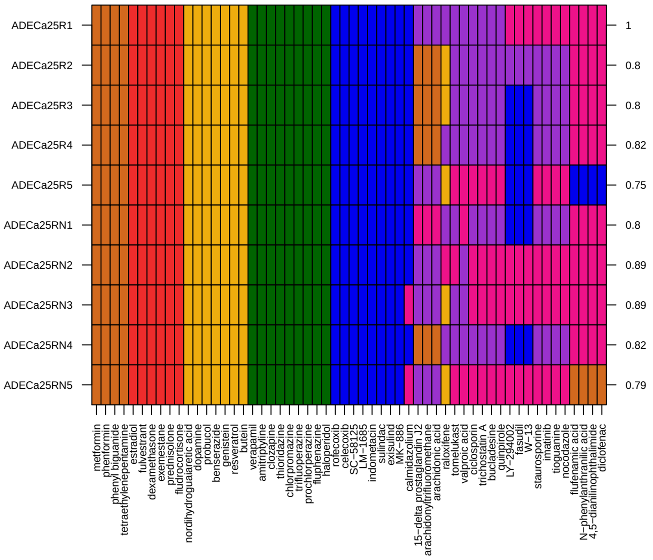


Figure 26: *Comparison for ADECa over Different Random Samples for 25 Iterations for MCF7*

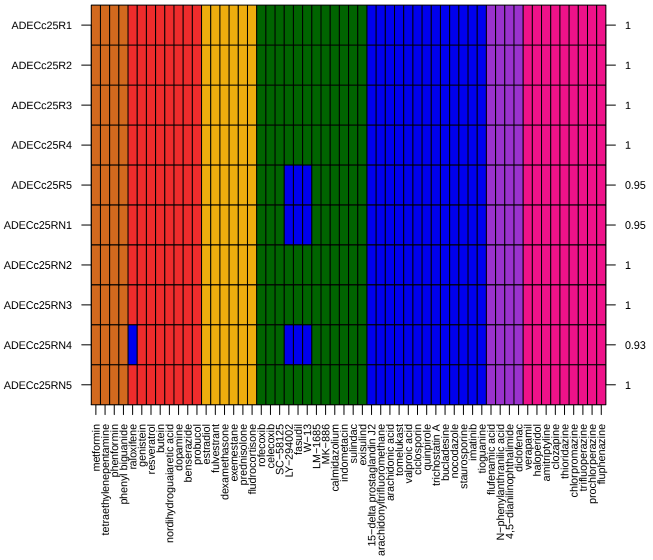


Figure 27: *Comparison for ADECc over Different Random Samples for 25 Iterations for MCF7*

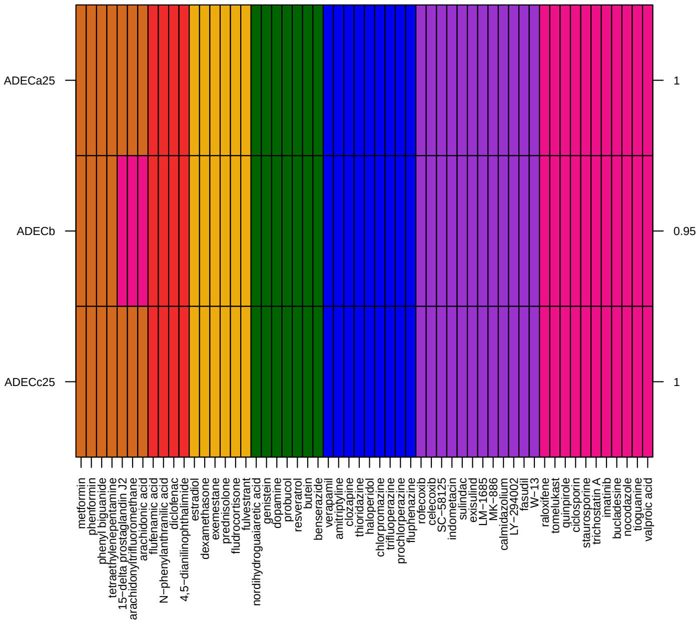
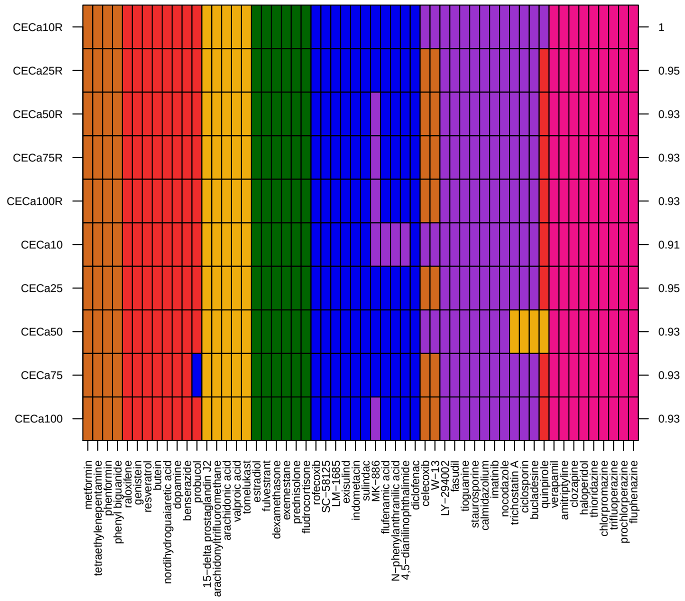


Figure 28: *Comparison for ADEC over Versions a, b and c for 25 Iterations for MCF7*

9.2.2 CEC



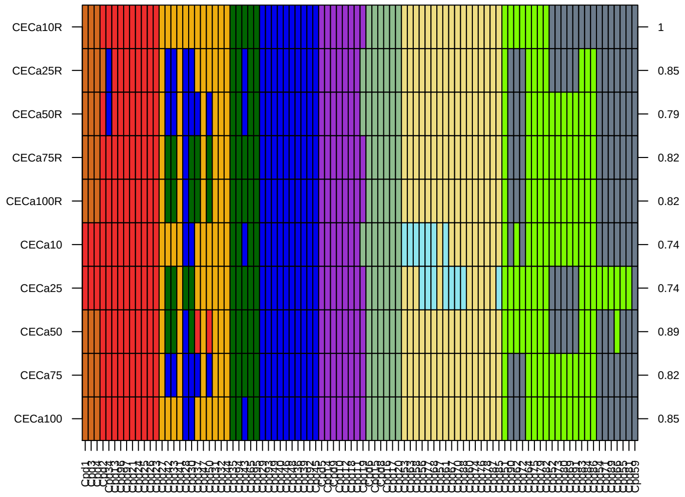


Figure 29: *Comparison for CECa over Different Iteration Numbers for MCF7*

Figure 30: *Comparison for CECa over Different Iteration Numbers for Inhouse1*

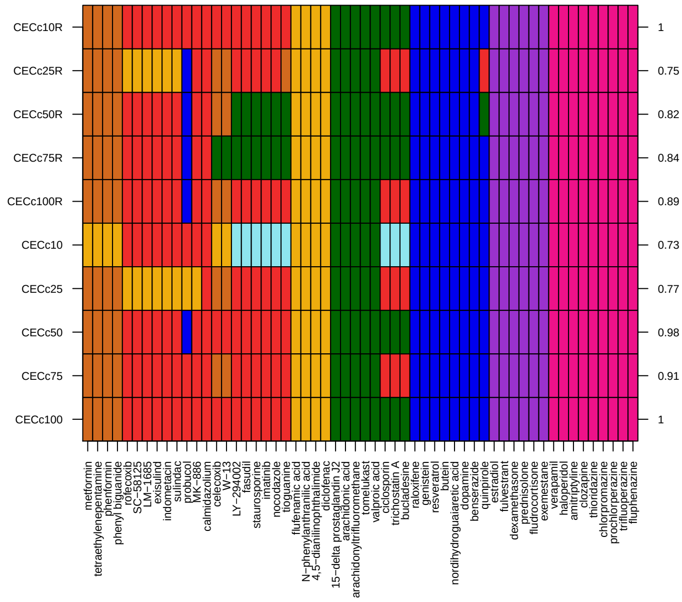


Figure 31: *Comparison for CECc over Different Iteration Numbers for MCF7*

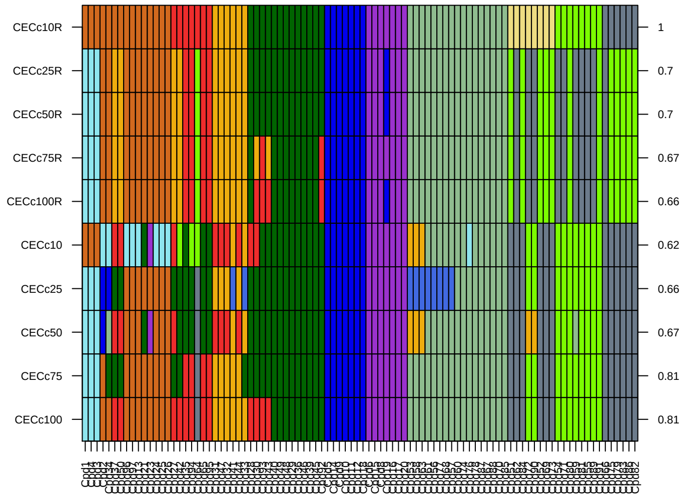


Figure 32: *Comparison for CECc over Different Iteration Numbers for Inhouse1*

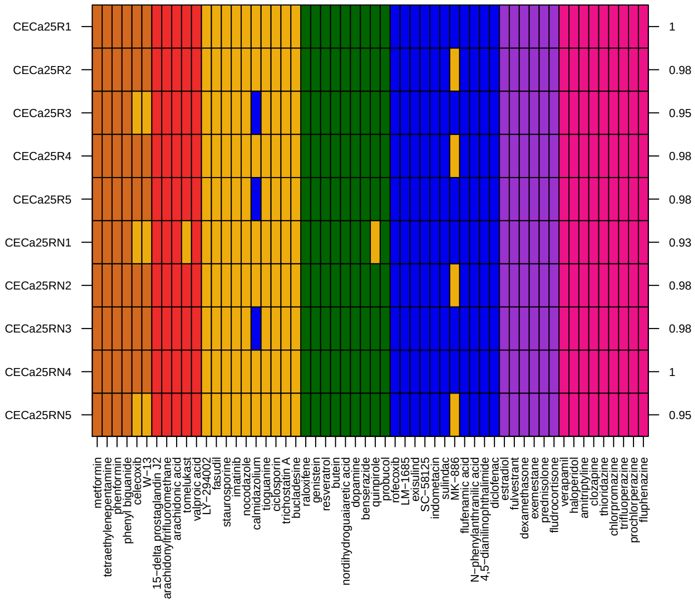


Figure 33: *Comparison for CECa over Different Random Samples for 25 Iterations for MCF7*

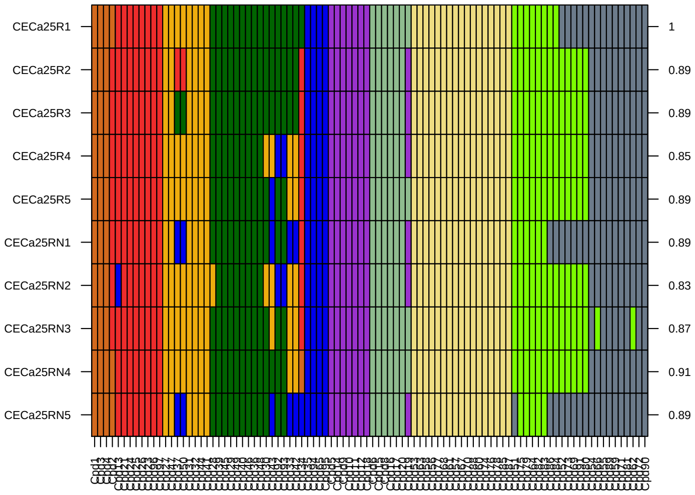


Figure 34: *Comparison for CECa over Different Random Samples for 25 Iterations for Inhouse1*

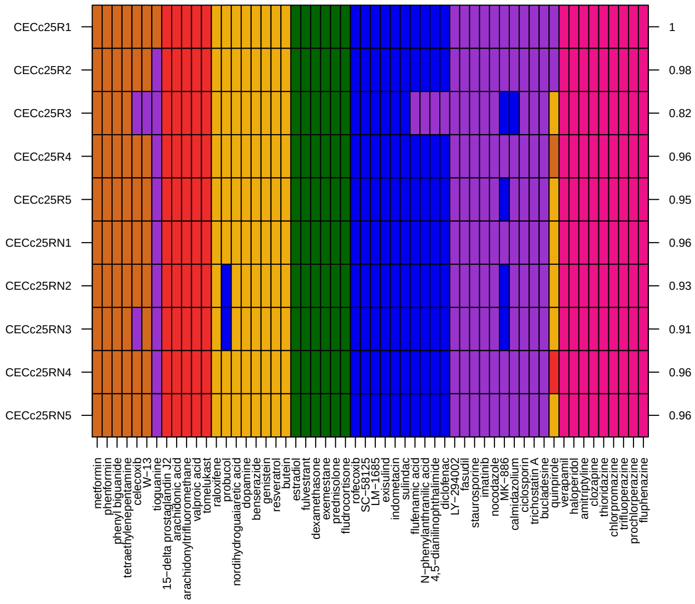


Figure 35: *Comparison for CECc over Different Random Samples for 25 Iterations for MCF7*

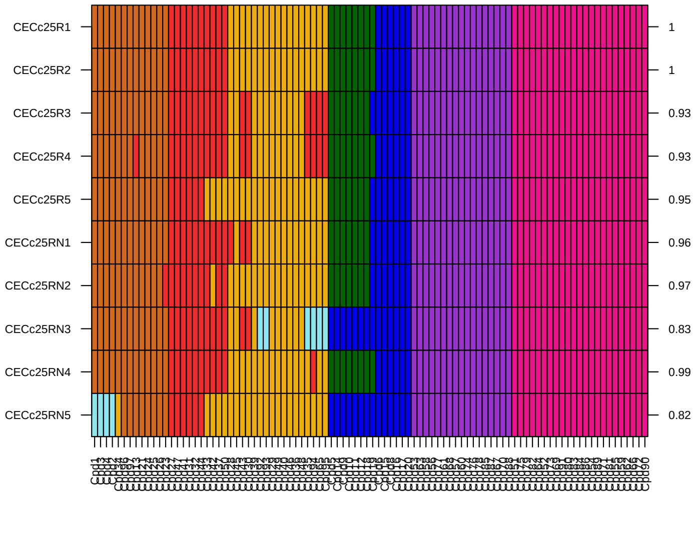


Figure 36: *Comparison for CECc over Different Random Samples for 25 Iterations for Inhouse1*

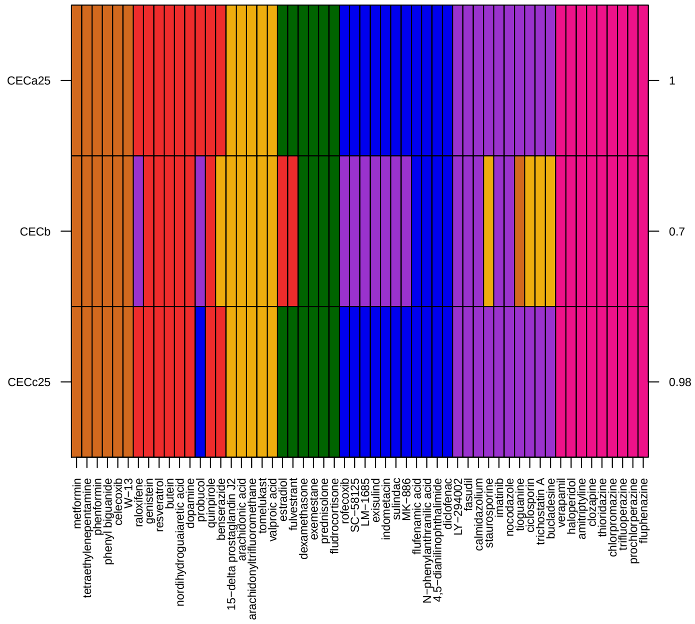


Figure 37: *Comparison for CEC over Versions a, b and c for 25 Iterations for MCF7 at weight* 0.5

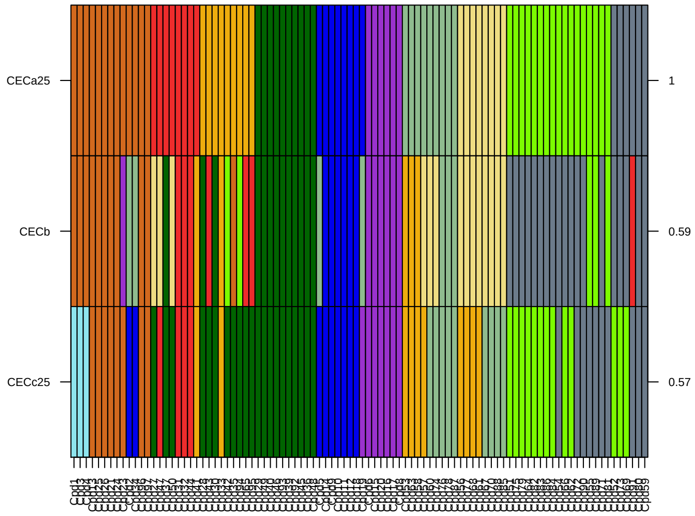


Figure 38: *Comparison for CEC over Versions a, b and c for 25 Iterations for Inhouse1 at weight* 0.5

9.2.3 SNF

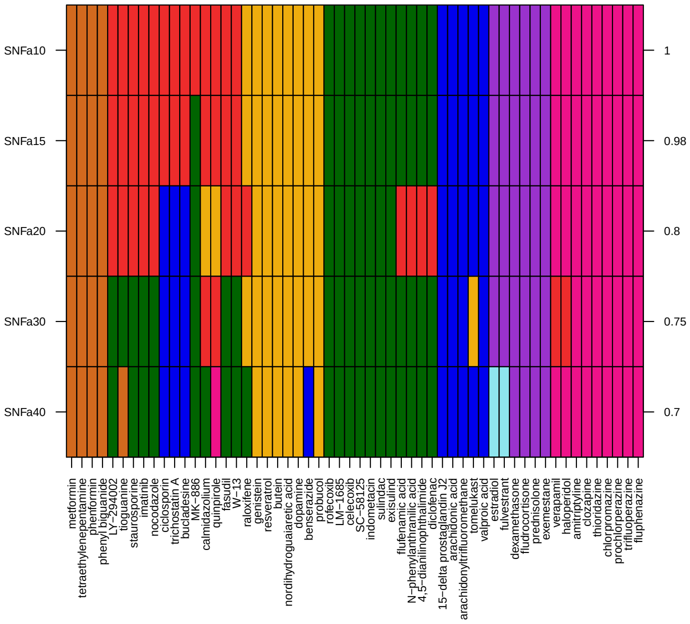


Figure 39: *Comparison for SNFa over the NN for MCF7*

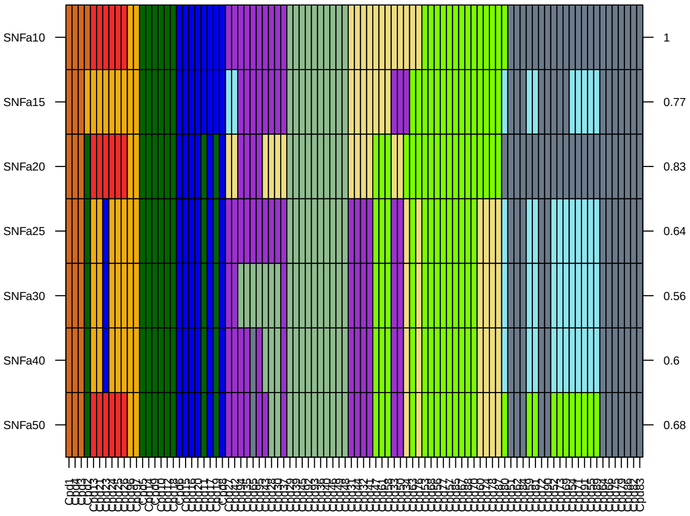


Figure 40: *Comparison for SNFa over the NN for Inhouse1*

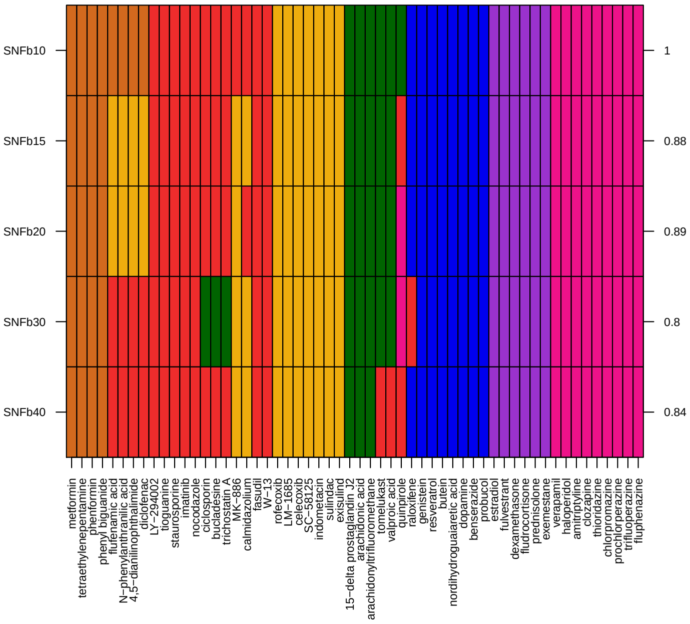


Figure 41: *Comparison for SNFb over the NN for MCF7*

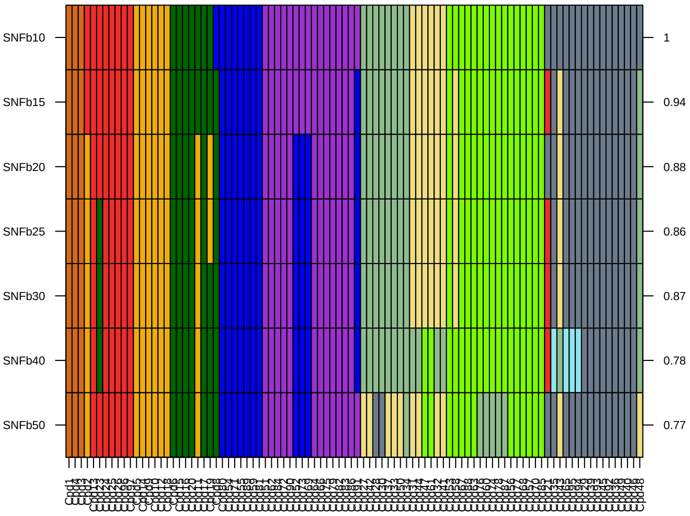


Figure 42: *Comparison for SNFb over the NN for Inhouse1*

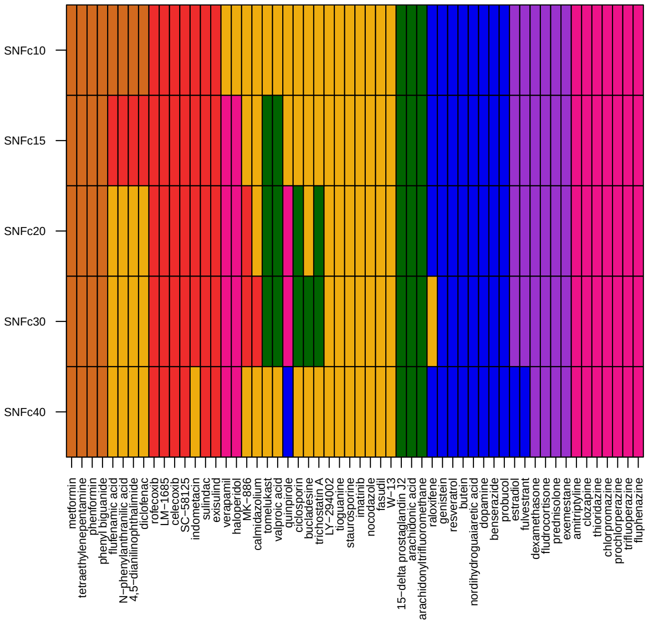


Figure 43: *Comparison for SNFc over the NN for MCF7*

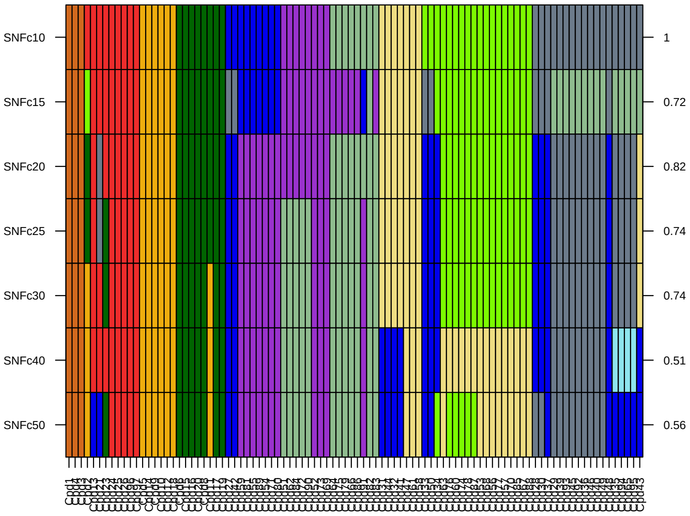


Figure 44: *Comparison for SNFc over the NN for Inhouse1*

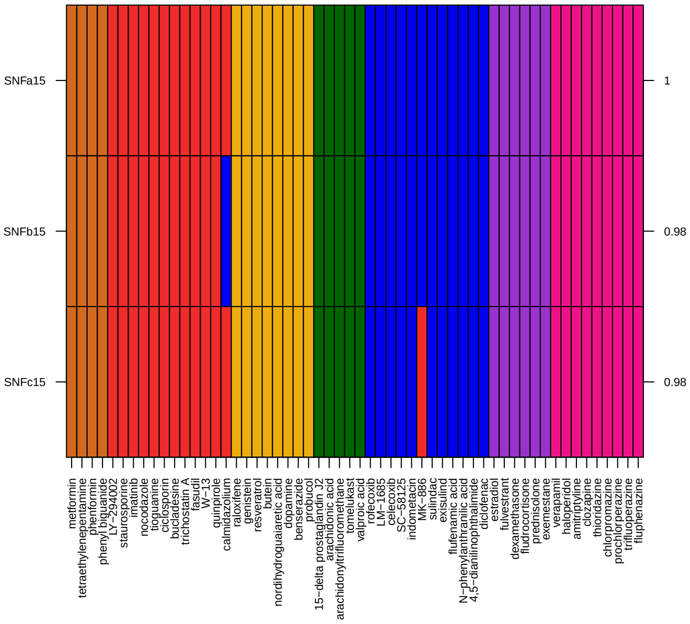


Figure 45: *Comparison for SNF over Versions a, b and c over 15 NN for MCF7*

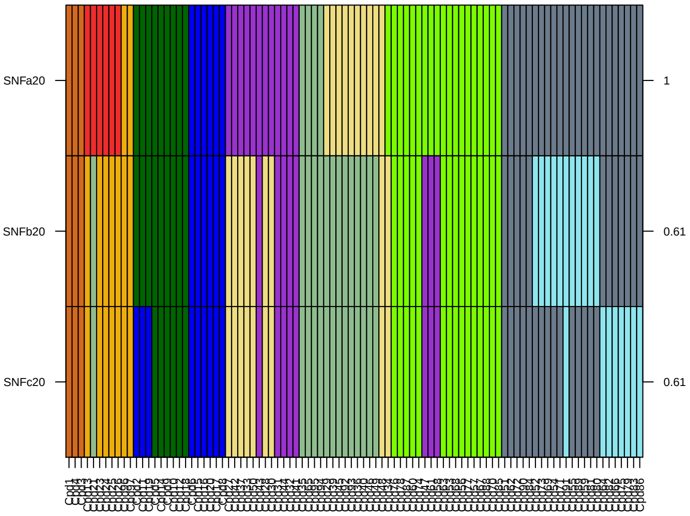


Figure 46: *Comparison for SNF over Versions a, b and c over 20 NN for Inhouse1*

9.3 MCF7 Cmap Data

9.3.1 Clustering on Separate Sources

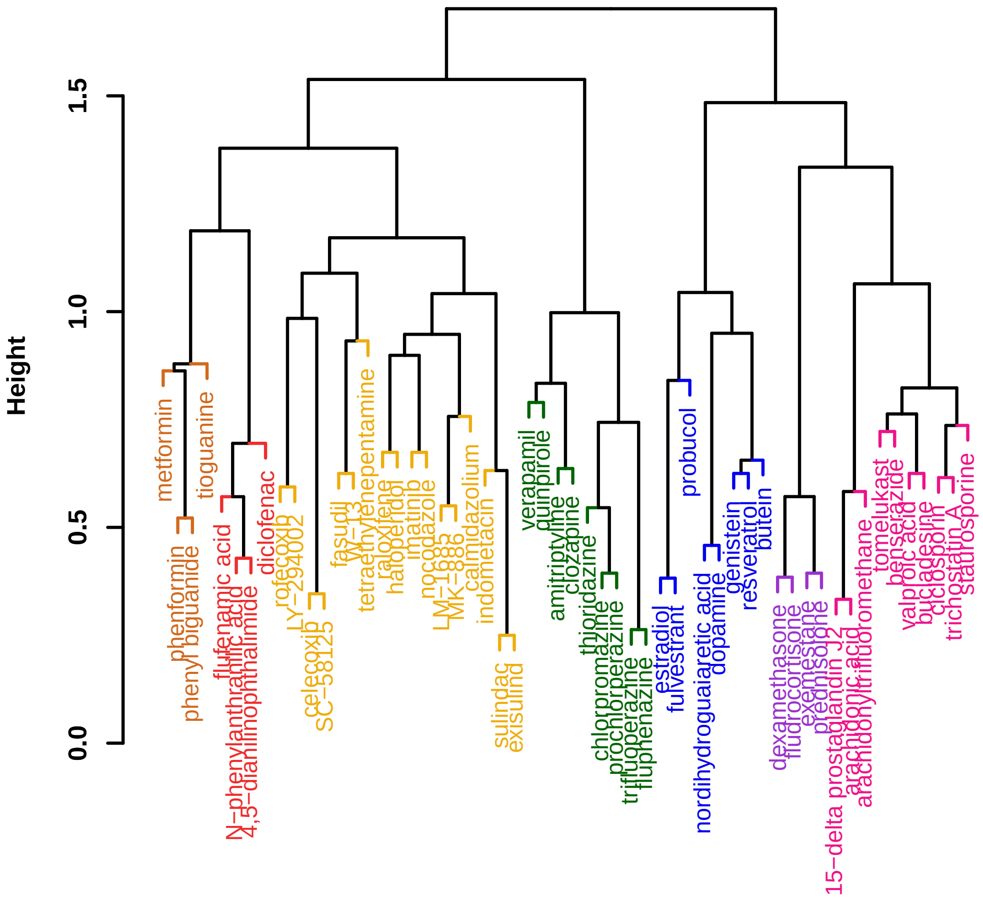


Figure 47: *The Dendogram based on Fingerprints for MCF7*

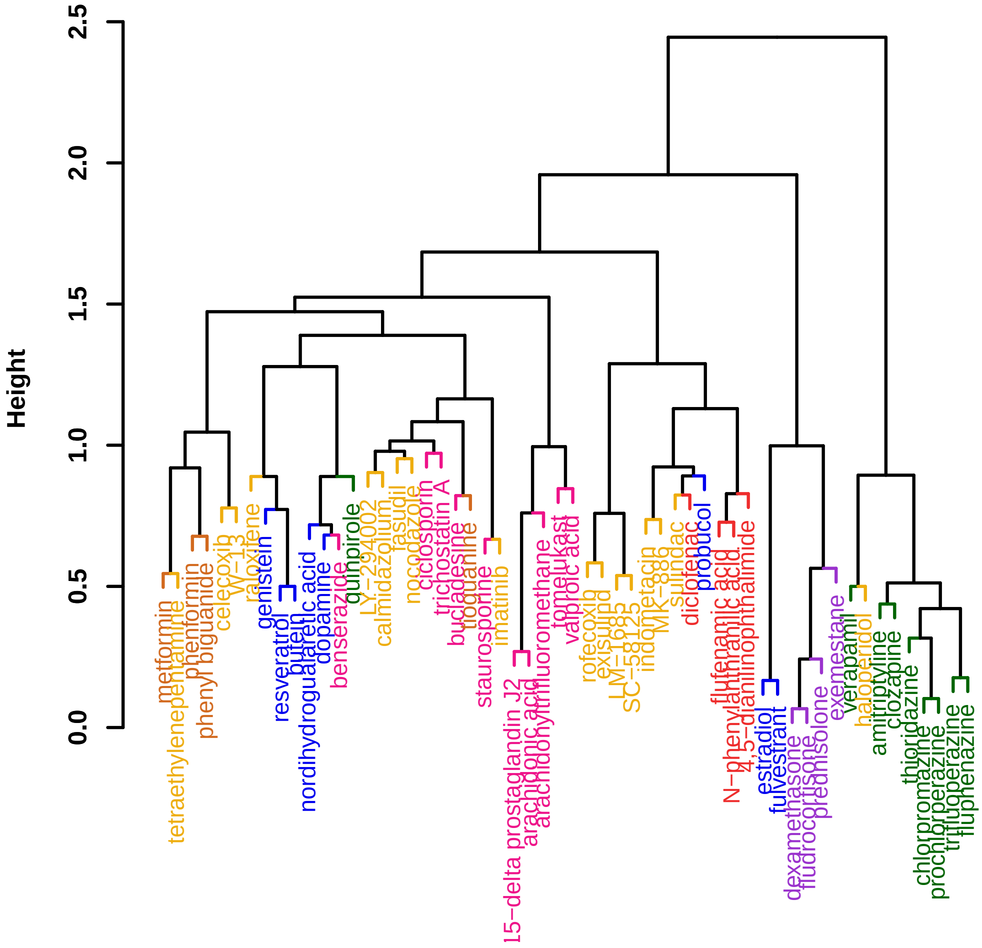


Figure 48: *The Dendogram based on Target Predictions for MCF7*

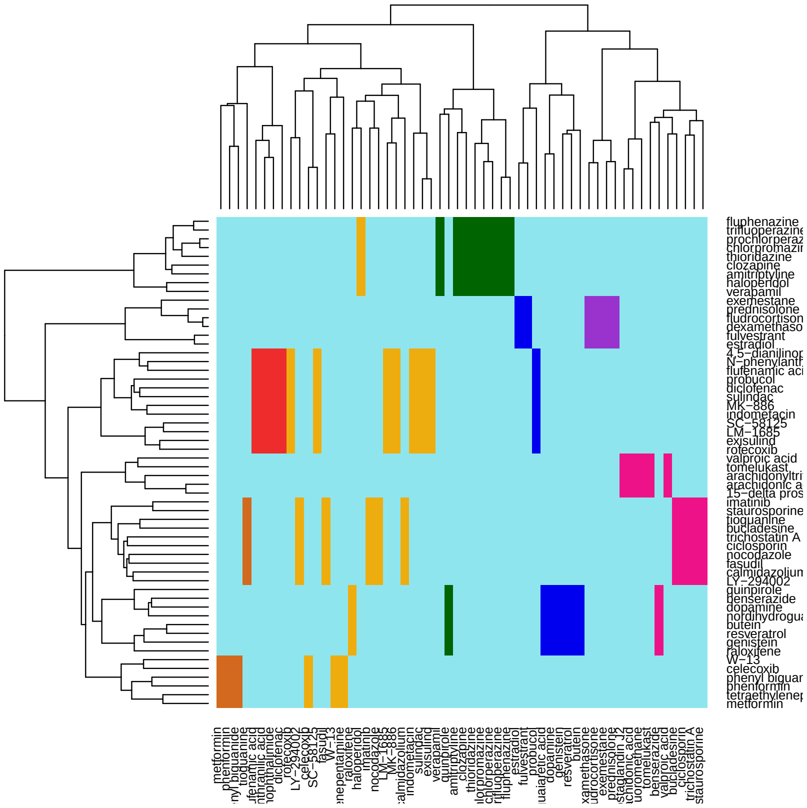


Figure 49: *Heatmap of MCF7: Fingerprints vs Target Predictions*

9.3.2 Integrated Clustering

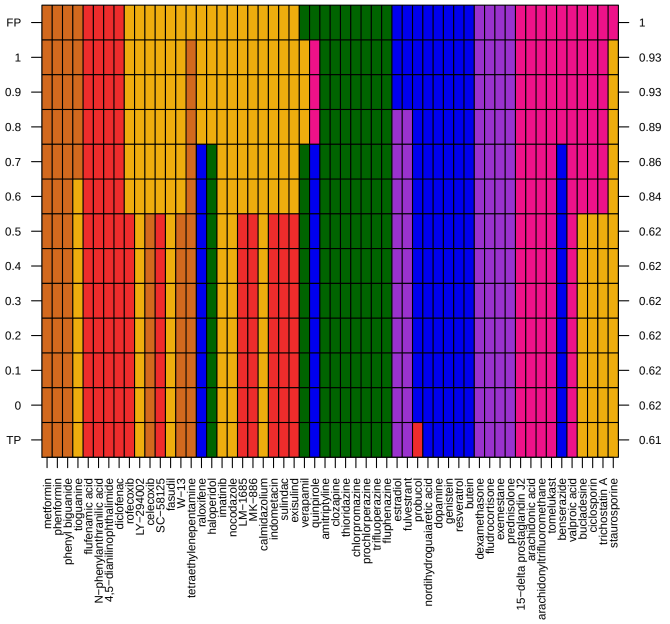


Figure 50: *Comparison for CECa over the Weights for 25 Iterations for MCF7*

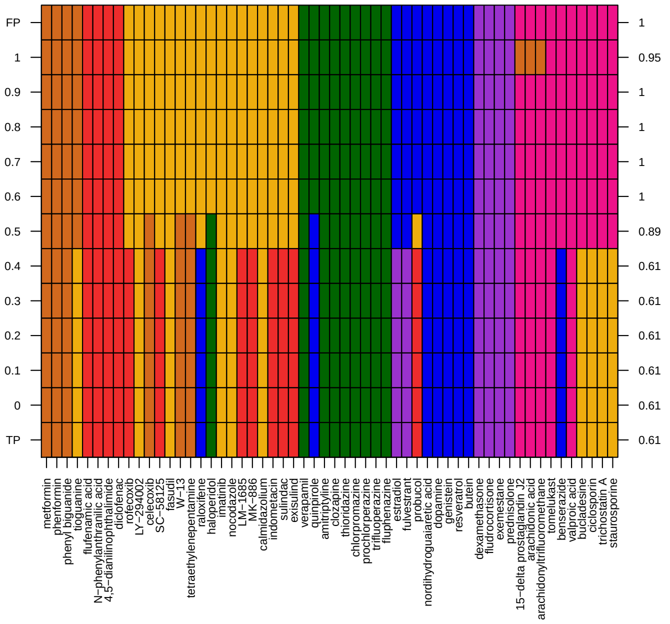


Figure 51: *Comparison for CECb over the Weights for MCF7*

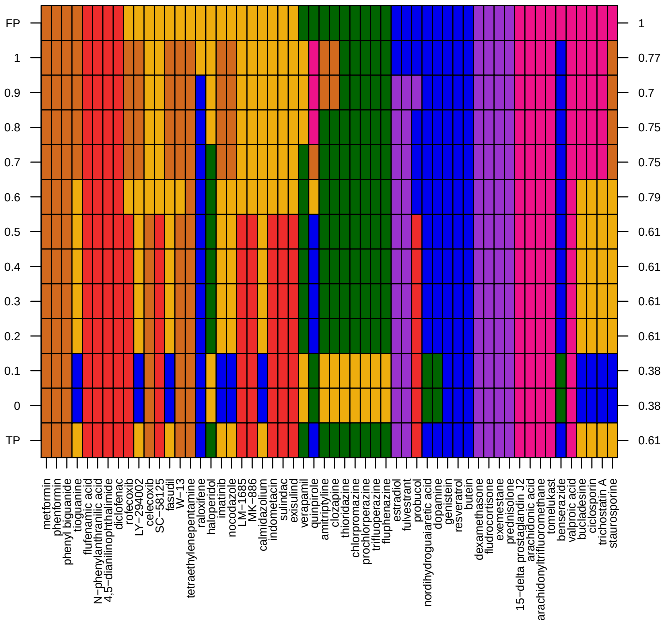


Figure 52: *Comparison for CECc over the Weights for 25 Iterations for MCF7*

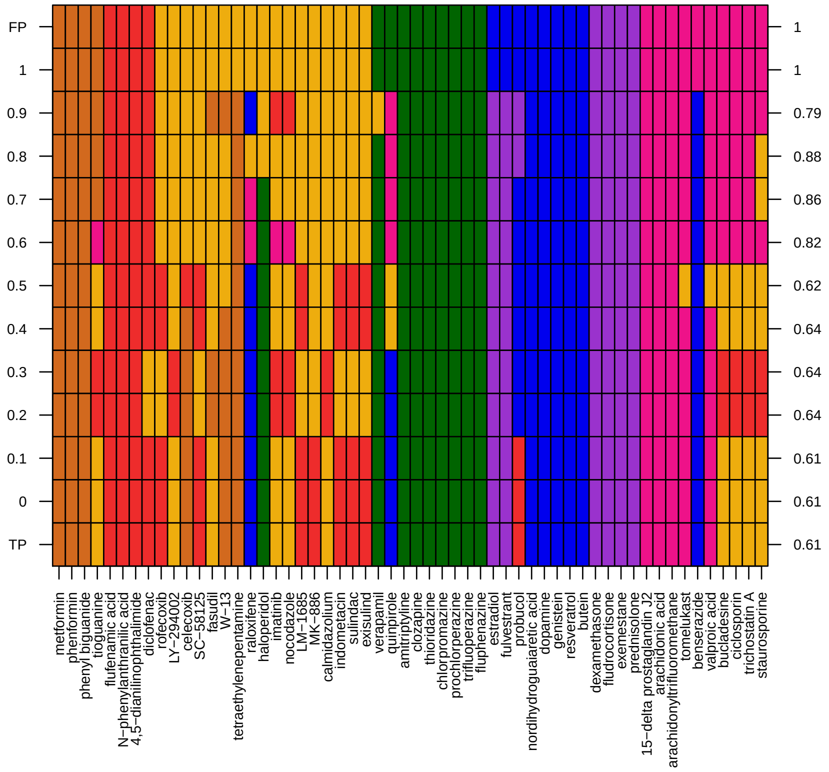


Figure 53: *Comparison for Weighted Clustering over the Weights for MCF7*

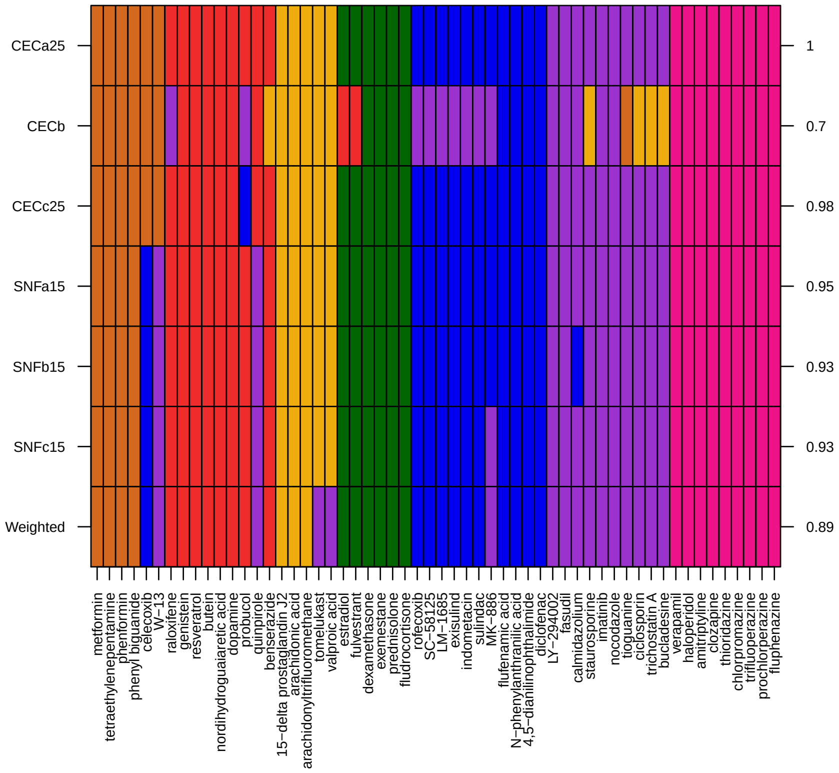


Figure 54: *Comparison of CEC, SNF and Weighted for MCF7*

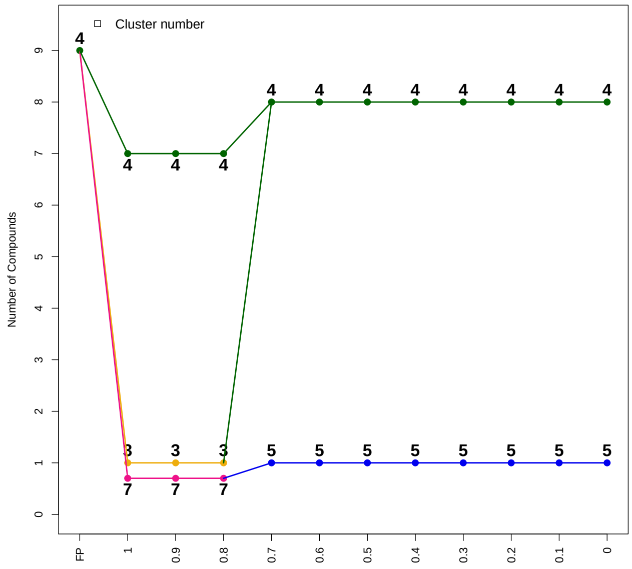


Figure 55: *The Evolution of Cluster 4 of FP over CECa for MCF7*

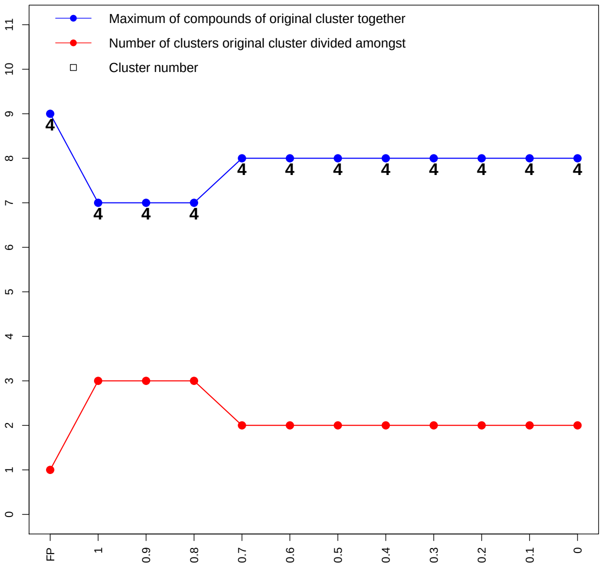


Figure 56: *The Evolution of Cluster 4 of FP over CECa for MCF7 - Maximum Number of Compounds*

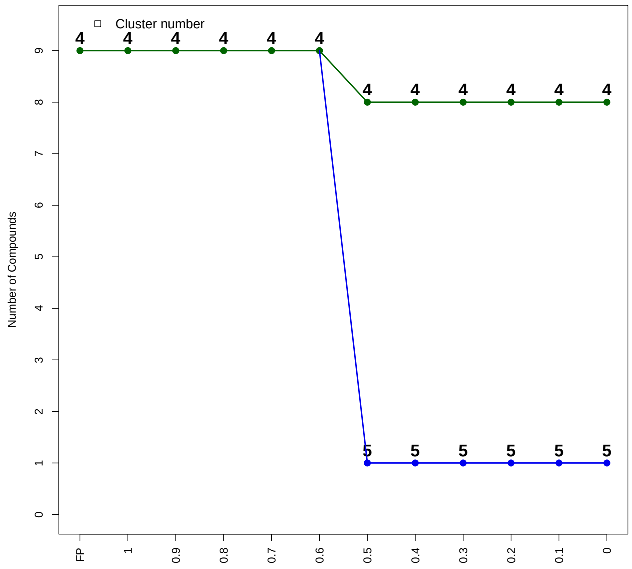


Figure 57: *The Evolution of Cluster 4 of FP over CECb for MCF7*

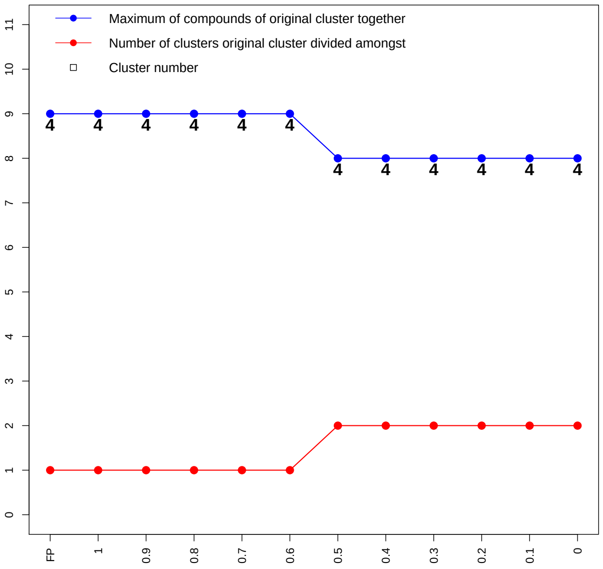


Figure 58: *The Evolution of Cluster 4 of FP over CECb for MCF7 - Maximum Number of Compounds*

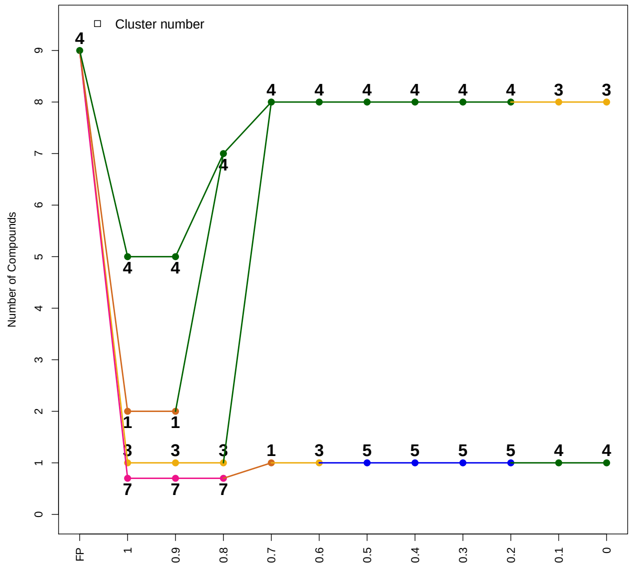


Figure 59: *The Evolution of Cluster 4 of FP over CECc for MCF7*

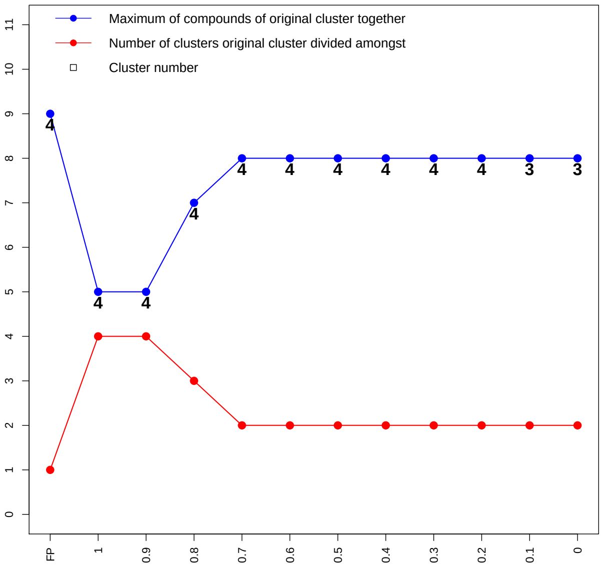


Figure 60: *The Evolution of Cluster 4 of FP over CECc for MCF7 - Maximum Number of Compounds*

9.3.3 Differential Expression and Pathway Analysis

	Cluster 1		*Cluster 2*		*Cluster 3*		*Cluster 4*		*Cluster 5*		*Cluster 6*		*Cluster 7*	
	G	**P**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**
Fingerprints	6	153	2	153	0	208	0	51	0	144	0	147	1	115
ADC	5	142	2	145	0	127	5	53	0	149	0	113	0	71
ADECa	0	180	2	149	0	130	5	53	0	149	0	113	0	92
ADECb	5	140	2	153	0	130	5	53	0	147	0	111	0	81
ADECc	0	180	2	153	0	127	5	54	0	148	0	112	0	96
CECa	2	152	0	153	1	192	5	53	0	75	0	111	1	95
CECb	1	143	2	152	0	338	5	53	0	138	0	139	1	114
CECc	2	155	0	188	1	206	5	53	0	90	0	105	1	96
SNFa	5	149	0	183	0	150	5	53	0	83	0	106	1	92
SNFb	5	147	0	187	0	117	5	54	0	84	0	104	1	88
SNFc	5	138	0	181	0	172	5	53	0	83	0	112	1	83
WonM	1	142	2	153	0	339	5	53	0	141	0	157	1	111
Weighted	5	141	0	180	0	167	5	53	0	82	0	112	2	237
Targets	2	156	0	185	1	208	5	53	0	91	0	104	1	92
Shared	0 (2)	26	0 (0)	15	0 (0)	0	0 (5)	37	0 (2)	36	0 (5)	49	0 (0)	0
	Comp		**Comp**		**Comp**		**Comp**		**Comp**		**Comp**		**Comp**	
Fingerprints	4		4		17		9		8		4		10	
ADC	4		4		12		9		7		6		14	
ADECa	7		4		12		9		7		6		11	
ADECb	4		4		12		9		7		6		14	
ADECc	7		4		12		9		7		6		11	
CECa	6		11		10		9		9		6		5	
CECb	7		4		14		9		8		4		10	
CECc	6		12		10		9		8		6		5	
SNFa	4		12		12		9		8		6		5	

SNFb	4	13	11	9	8	6	5
SNFc	4	11	13	9	8	6	5
WonM	7	4	14	9	8	4	10
Weighted	4	11	15	9	8	6	3
Targets	6	12	10	9	8	6	5
Shared	3	4	2	8	5	4	0

Table 8: *Number of significant genes & pathways at 0.05 significance level - All Methods vs FP - MCF7 Data*

	Genes	**Significant For**
Cluster 1	EMP1	ADC ADECb SNFa SNFb SNFc Weighted
	TGFB1I1	FP ADC ADECb SNFa SNFb SNFc Weighted
	GPNMB	FP ADC ADECb SNFa SNFb SNFc Weighted
	HLX	ADC ADECb CECa CECc SNFa SNFb SNFc WonM Weighted TP
	VIM	ADC ADECb SNFa SNFb SNFc Weighted
	CD1D	CECa CECc TP
Cluster 2	ACVR1	FP ADC ADECa ADECb ADECc CECb WonM
	TRIM22	FP ADC ADECa ADECb ADECc CECb WonM
Cluster 3	TUFT1	CECa CECc TP
Cluster 4	PI3	All except FP
	PNO1	All except FP
	SRSF7	All except FP
	MSMO1	All except FP
	IDI1	All except FP
Cluster 5	-	-
Cluster 6	-	-
Cluster 7	MAGT1	FP CECb WonM
	ASF1A	CECa CECc SNFa SNFb SNFc TP

Table 9: *Interesting for the Clusters over all Methods for MCF7*

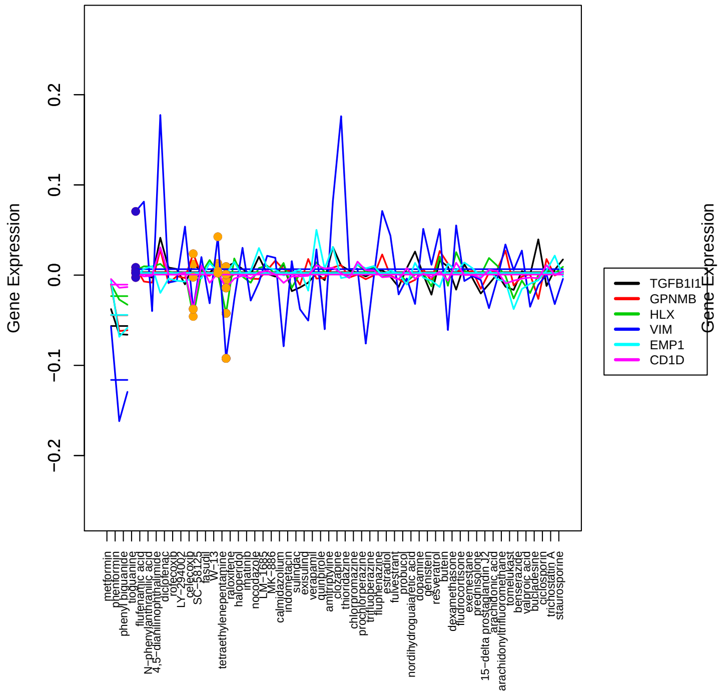


Figure 61: *Gene Profile of Interesting Genes for Cluster 1 of MCF7*

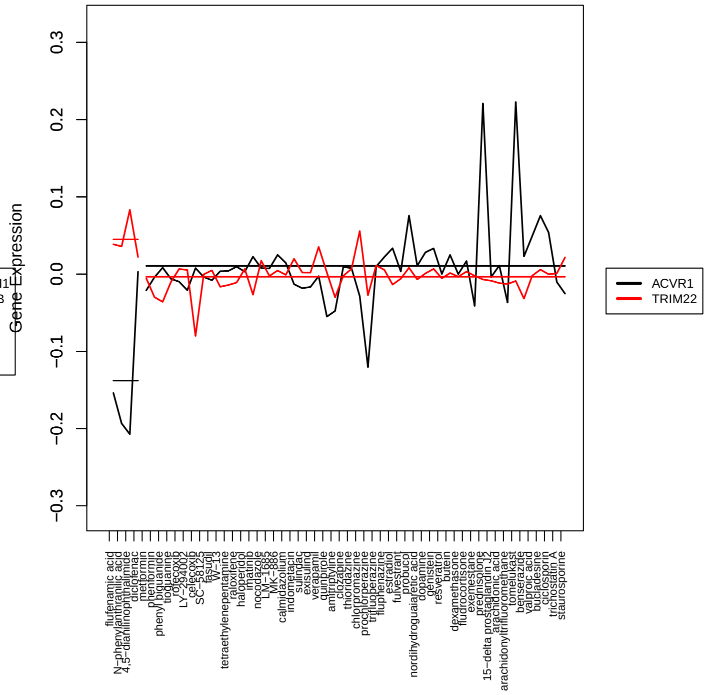


Figure 62: *Gene Profile of Interesting Genes for Cluster 2 of MCF7*

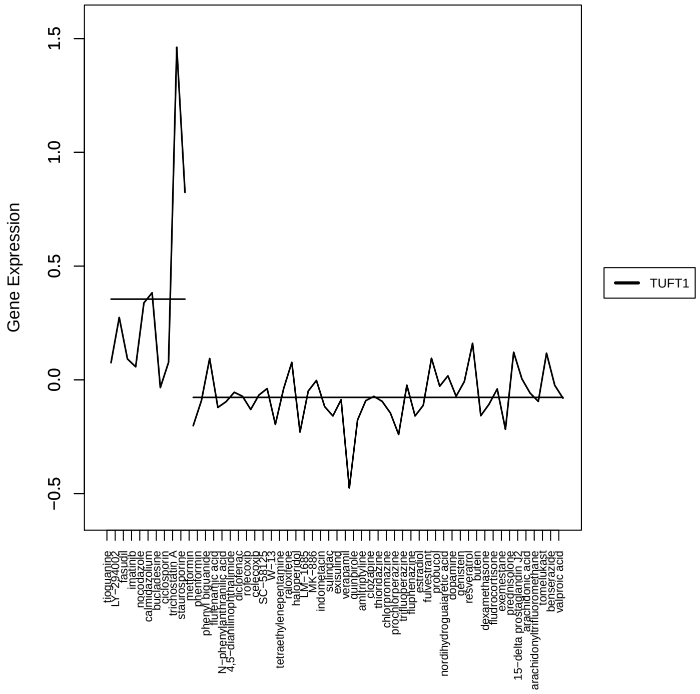


Figure 63: *Gene Profile of Interesting Genes for Cluster 3 of MCF7*

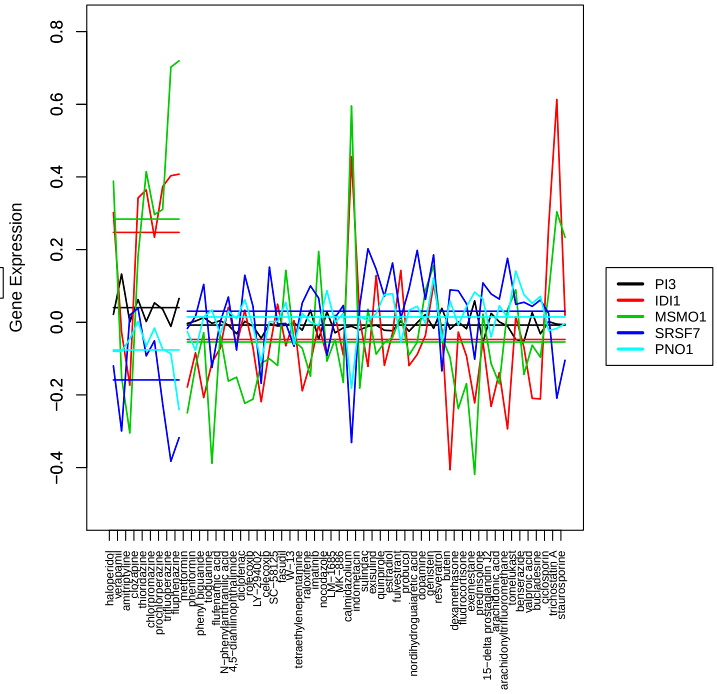


Figure 64: *Gene Profile of Interesting Genes for Cluster 4 of MCF7*

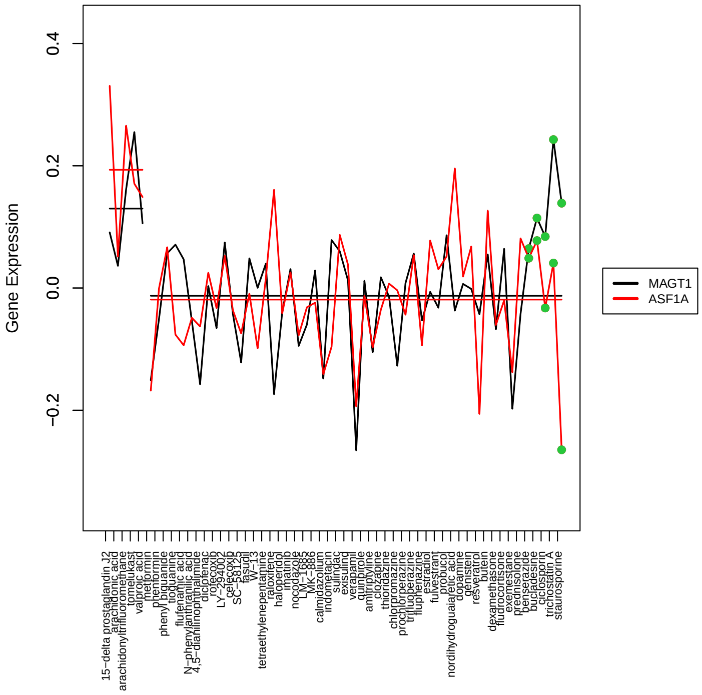


Figure 65: *Gene Profile of Interesting Genes for Cluster 7 of MCF7*

	Pathways	**Mean p-value**
	cellular amino acid metabolic process	0.0290
Cluster 1	peptide metabolic process	0.0010
	glutathione metabolic process	0.0013
	negative regulation of neuron projection development	0.0015
Cluster 2	ensheathment of neurons	0.0097
	axon ensheathment	0.0097
	-	-
Cluster 3	cholesterol biosynthetic process	0.0000
Cluster 4	sterol biosynthetic process	0.0000
	regulation of ATPase activity	0.0113
	regulation of mitochondrion organization	0.0044
Cluster 5	ER to Golgi vesicle-mediated transport	0.0001
	regulation of release of cytochrome c from mitochondria	0.0088
	pigmentation	0.0030
Cluster 6	inner ear morphogenesis	0.0014
	sensory perception of mechanical stimulus	0.0004
Cluster 7	-	-

Table 10: *P-values of the Shared Top 3 Pathways for the Clusters over all Methods for MCF7*

	Cluster 1		*Cluster 2*		*Cluster 3*		*Cluster 4*		*Cluster 5*		*Cluster 6*		*Cluster 7*	
Weight	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**
1.0	6	151	2	152	0	206	0	52	0	148	0	153	1	111
0.9	0	134	0	125	0	62	0	74	0	119	0	91	1	128
0.8	0	135	2	152	0	157	1	57	1	181	0	90	1	62
0.7	0	142	2	152	0	190	5	52	0	150	0	113	0	74
0.6	5	141	2	153	0	130	5	53	0	150	0	113	0	65
0.5	5	143	0	186	0	159	5	55	0	85	0	113	2	238
0.4	2	149	0	147	1	168	5	53	0	95	0	113	1	93
0.3	2	161	0	138	0	92	5	53	0	71	0	110	1	91
0.2	2	175	0	131	0	91	5	53	0	67	0	111	1	96
0.1	2	151	0	186	1	198	5	53	0	94	0	106	1	93
0.0	2	150	0	188	1	189	5	52	0	88	0	109	1	96
Shared	0 (2)	31	0 (0)	2	0 (0)	0	0 (2)	20	0 (2)	41	0 (4)	34	0 (0)	0
	Comp		**Comp**		**Comp**		**Comp**		**Comp**		**Comp**		**Comp**	
1.0	4		4		17		9		8		4		10	
0.9	7		6		12		7		7		7		10	
0.8	5		4		17		8		6		7		9	
0.7	5		4		15		9		7		6		10	
0.6	4		4		12		9		7		6		14	
0.5	4		11		15		9		8		6		3	
0.4	6		10		12		9		8		6		5	
0.3	7		12		8		9		9		6		5	
0.2	7		12		8		9		9		6		5	
0.1	6		12		10		9		8		6		5	
0.0	6		12		10		9		8		6		5	
Shared	3		3		0		7		5		4		3	

Table 11: *Number of significant genes & pathways at 0.05 significance level - Weighted Clustering - MCF7 Data*

	Genes	**Significant For Weight**
Cluster 1	EMP1	0.6 0.5
	TGFB1I1	1.0 0.6 0.5
	GPNMB	1.0 0.6 0.5
	HLX	0.6 0.5 0.4 0.3 0.2 0.1 0.0
	CD1D	0.4 0.3 0.2 0.1 0.0
	VIM	0.6 0.5
Cluster 2	ACVR1	1.0 0.8 0.7 0.6
	TRIM22	1.0 0.8 0.7 0.6
Cluster 3	TUFT1	0.4 0.1 0.0
Cluster 4	PI3	0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	PNO1	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	SRSF7	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	MSMO1	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	IDI1	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
Cluster 5	-	-
Cluster 6	-	-
Cluster 7	MAGT1	1.0 0.9 0.8
	ASF1A	0.4 0.3 0.2 0.1 0.0

Table 12: *Interesting Genes of the Weighted Clusters of MCF7*

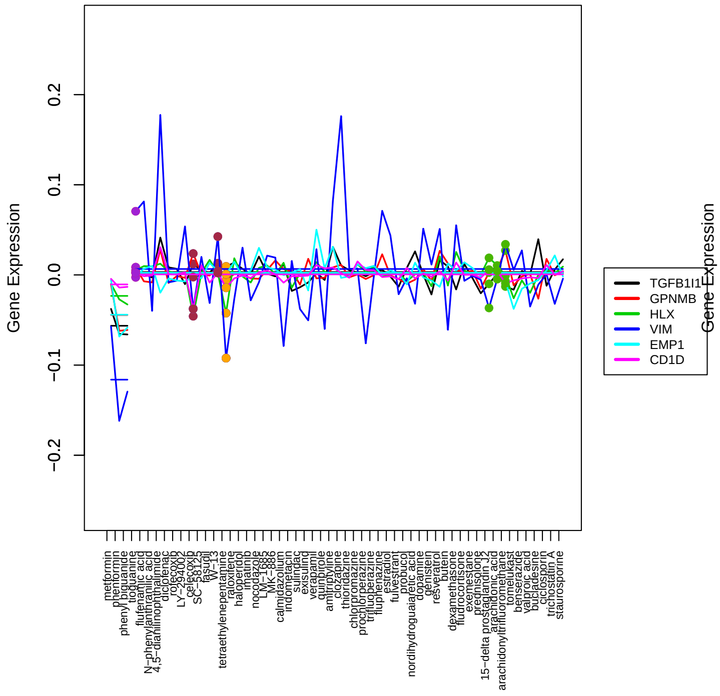


Figure 66: *Gene Profile of Interesting Genes for Cluster 1 of Weighted for MCF7*

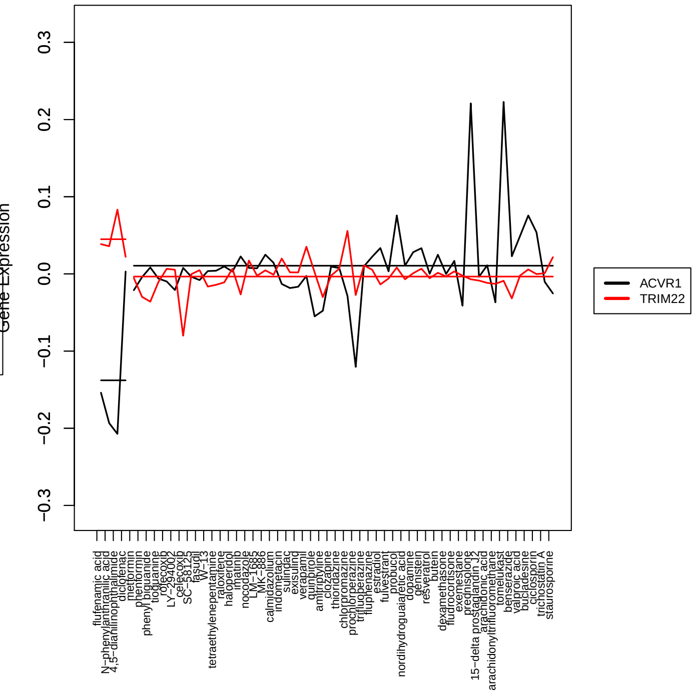


Figure 67: *Gene Profile of Interesting Genes for Cluster 2 of Weighted for MCF7*

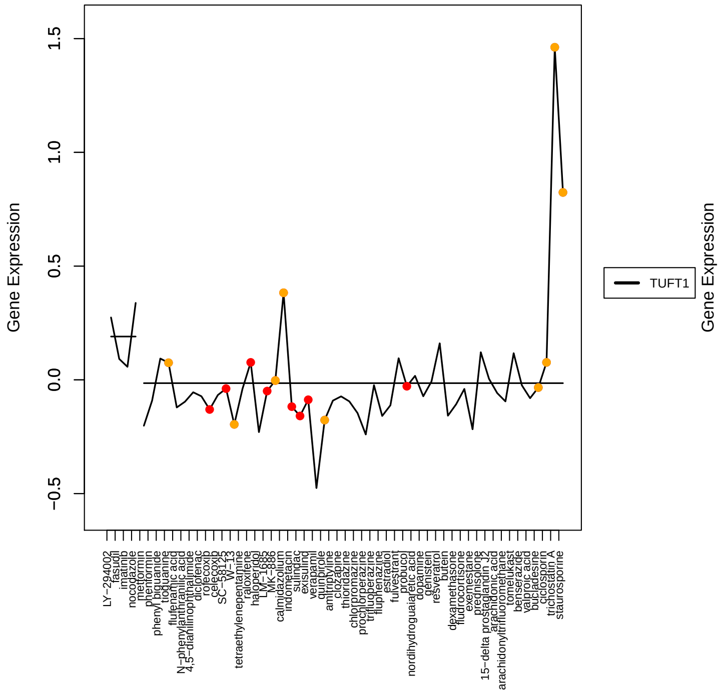


Figure 68: *Gene Profile of Interesting Genes for Cluster 3 of Weighted for MCF7*

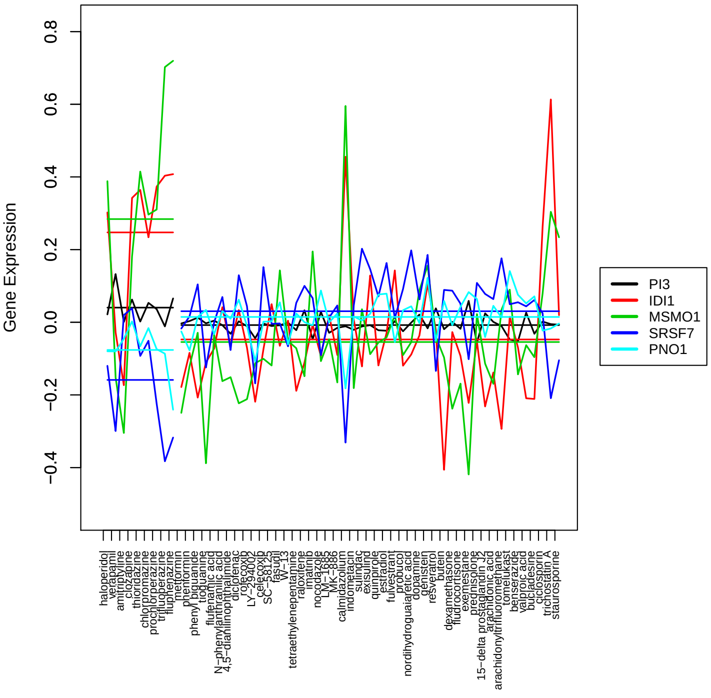


Figure 69: *Gene Profile of Interesting Genes for Cluster 4 of Weighted for MCF7*

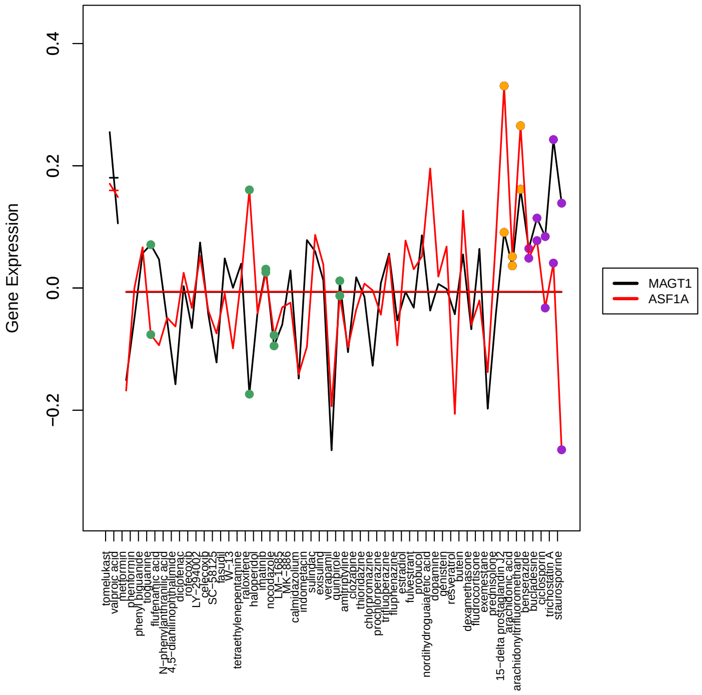


Figure 70: *Gene Profile of Interesting Genes for Cluster 7 of Weighted for MCF7*

	Pathways	**Mean p-value**
	cellular amino acid metabolic process	0.0005
Cluster 1	peptide metabolic process	0.0001
	glutathione metabolic process	0.0001
	skeletal muscle tissue development	0.0062
Cluster 2	myotube differentiation	0.0190
	-	-
Cluster 3	cholesterol biosynthetic process	0.0000
Cluster 4	sterol biosynthetic process	0.0000
	regulation of ATPase activity	0.0029
	regulation of mitochondrion organization	0.0052
Cluster 5	ER to Golgi vesicle-mediated transport	0.0004
	regulation of release of cytochrome c from mitochondria	0.0085
	pigmentation	0.0081
Cluster 6	inner ear morphogenesis	0.0029
	sensory perception of mechanical stimulus	0.0011
Cluster 7	-	-

Table 13: *P-values of the Shared Top 3 Pathways for the Weighted Clusters for MCF7*

	All Methods			*Weighted*		*CECa*		*CECb*		*CECc*	
Method	**G**	**P**	**Weight**	**G**	**P**	**G**	**P**	**G**	**P**	**G**	**P**
Fingerprints	0	48	**1.0**	0	50	0	74	0	50	9	84
ADC	5	52	**0.9**	0	73	0	74	0	52	9	79
ADECa	5	53	**0.8**	1	58	0	75	0	50	0	74
ADECb	5	53	**0.7**	5	53	5	52	0	50	5	53
ADECc	5	53	**0.6**	5	54	5	53	0	53	5	53
CECa	5	53	**0.5**	5	53	5	53	0	53	5	53
CECb	5	52	**0.4**	5	52	5	53	5	53	5	53
CECc	5	53	**0.3**	5	53	5	53	5	54	5	53
SNFa	5	53	**0.2**	5	55	5	53	5	53	5	53
SNFb	5	53	**0.1**	5	53	5	53	5	52	5	53
SNFc	5	53	**0.0**	5	53	5	53	5	53	5	53
WonM	5	53	**Shared**	0 (1)	19 (46)	0 (5)	20 (52)	0 (5)	36 (52)	0 (0/5)	17 (20/53)
Weighted	5	54									
Targets	5	52									
Shared	0 (5)	36 (51)									

Table 14: *Number of significant genes & pathways at 0.05 significance level - Cluster* 4 *of FP - MCF7 Data*

9.4 Inhouse1 Data

9.4.1 Clustering on Separate Sources

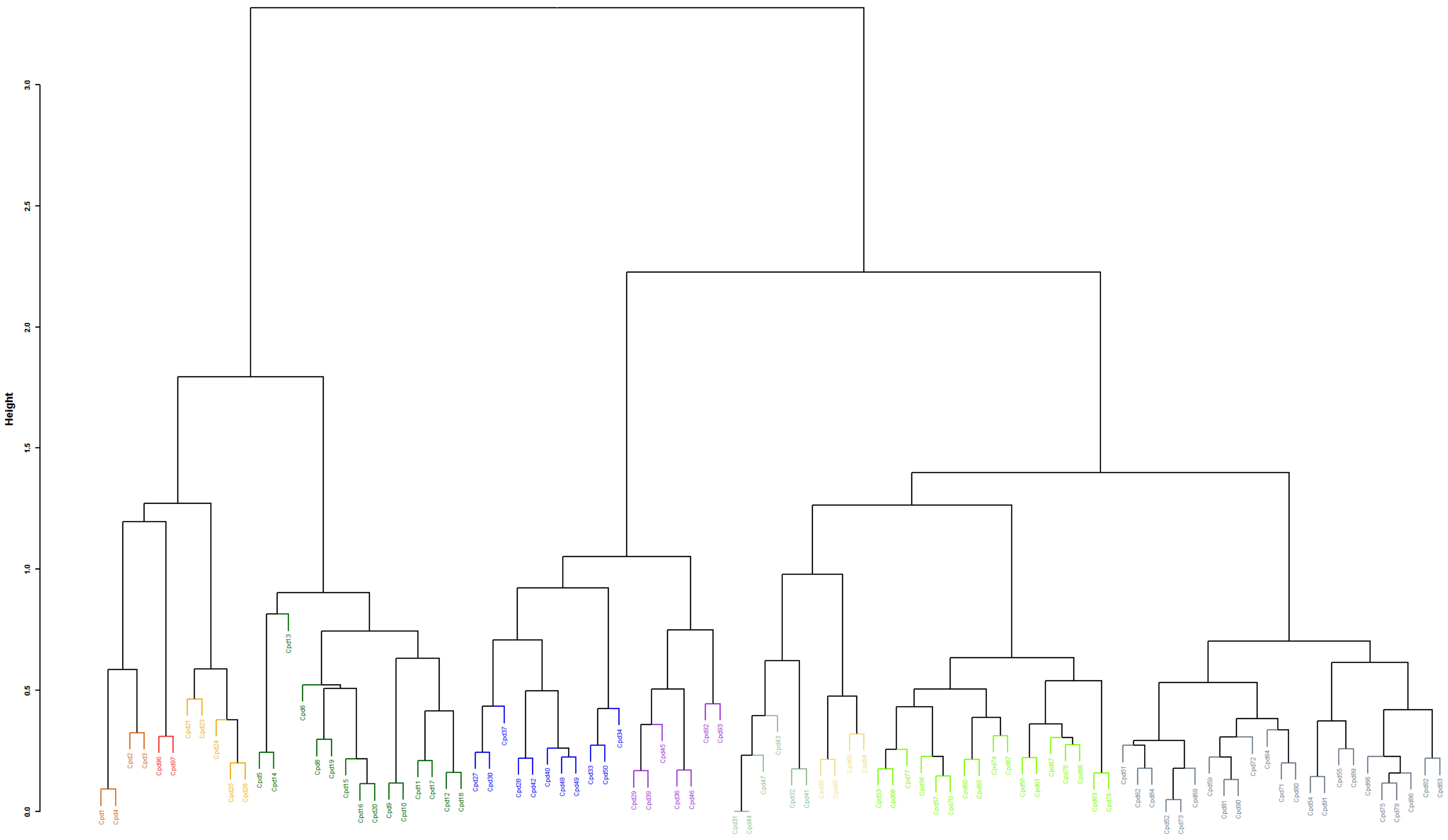


Figure 71: *The Dendogram based on Fingerprints for Inhouse1*

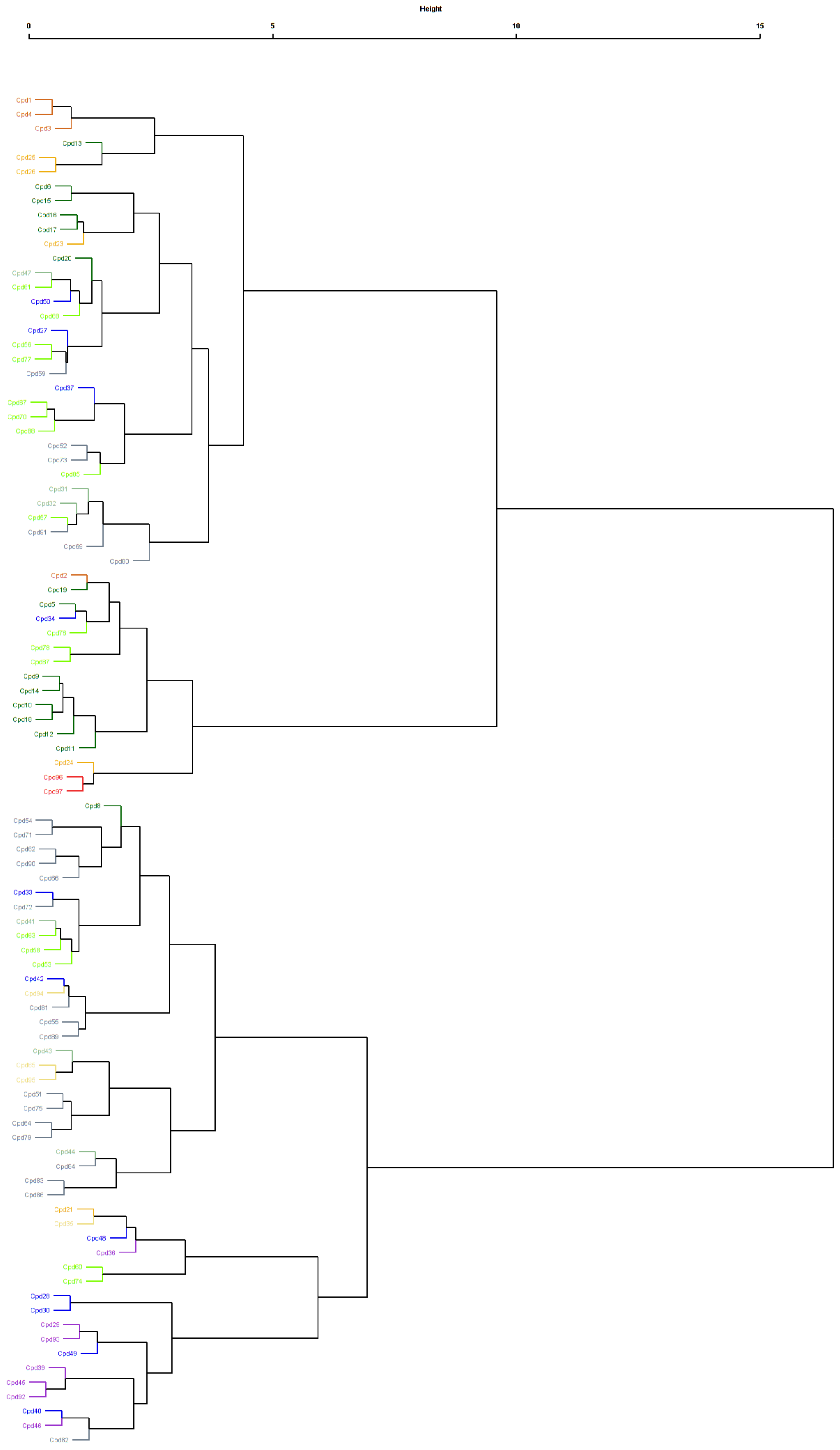


Figure 72: *The Dendogram based on Bio-assays for Inhouse1*

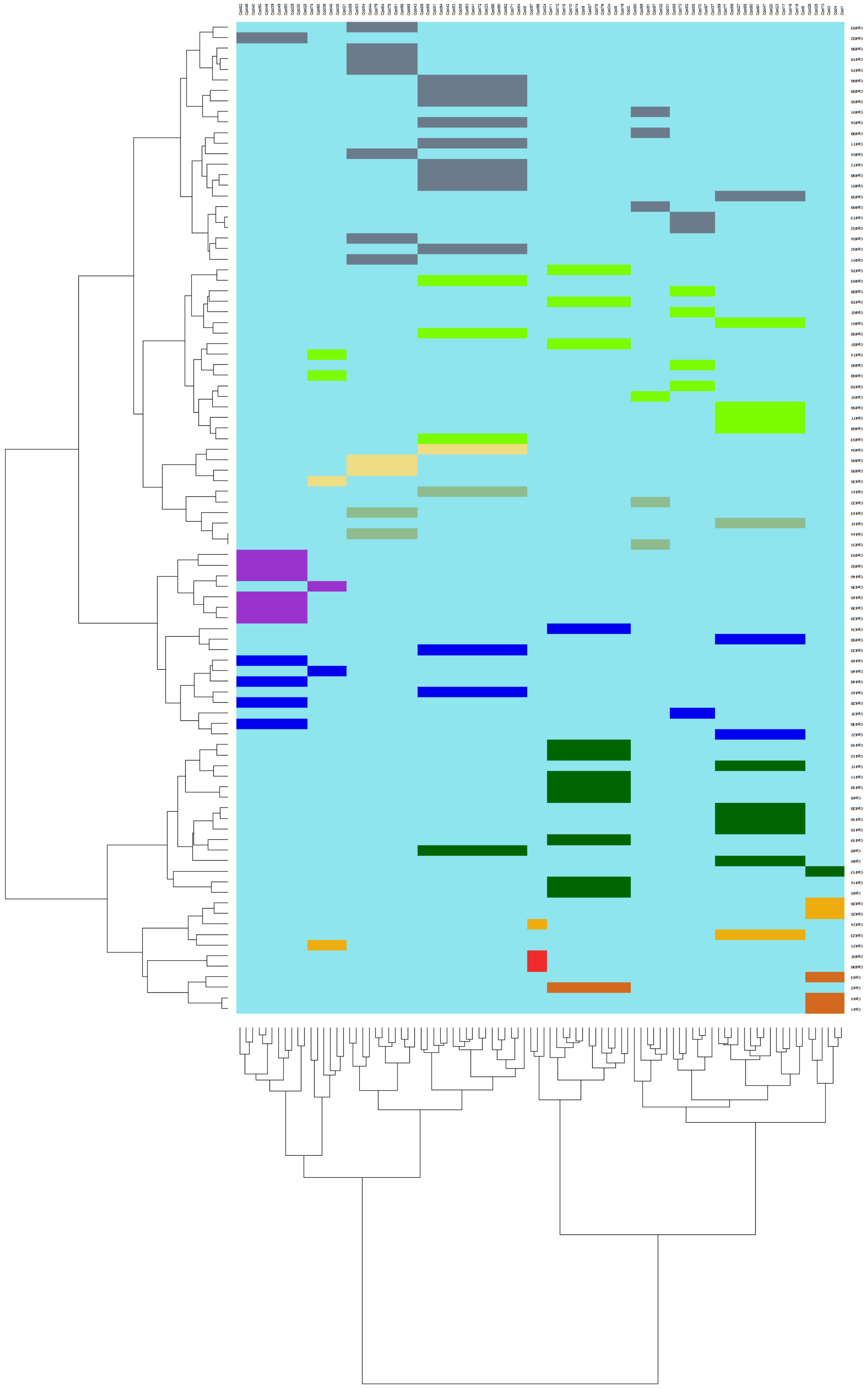


Figure 73: *Heatmap of Inhouse1: Fingerprints vs Bio-assays*

9.4.2 Integrated Clustering

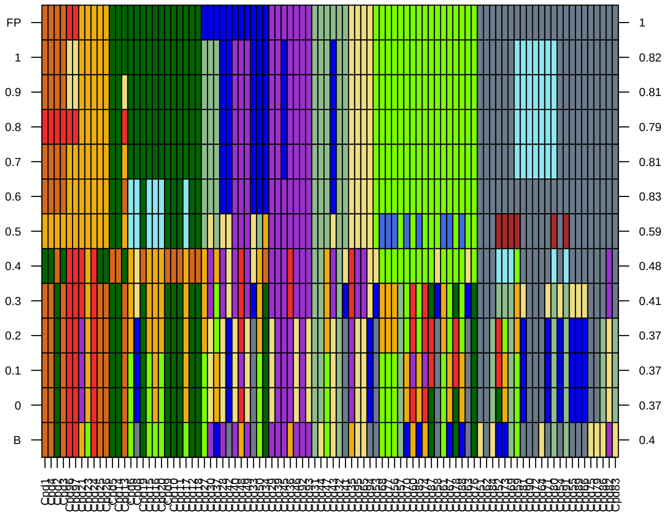


Figure 74: *Comparison for CECa over the Weights for 25 Iterations for Inhouse1*

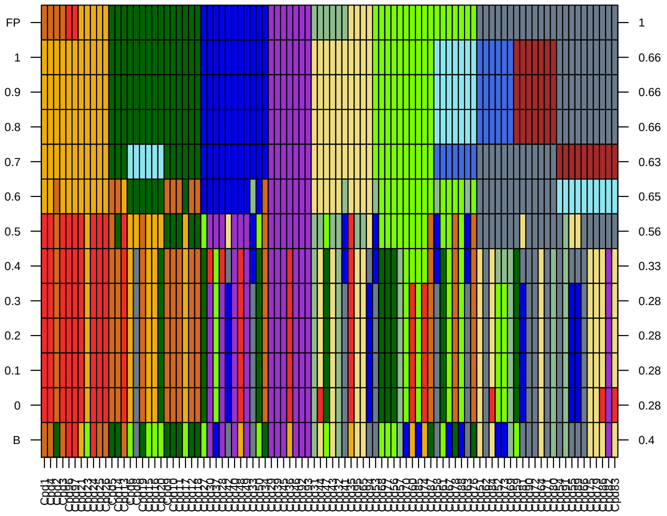


Figure 75: *Comparison for CECb over the Weights for 25 Iterations for Inhouse1*

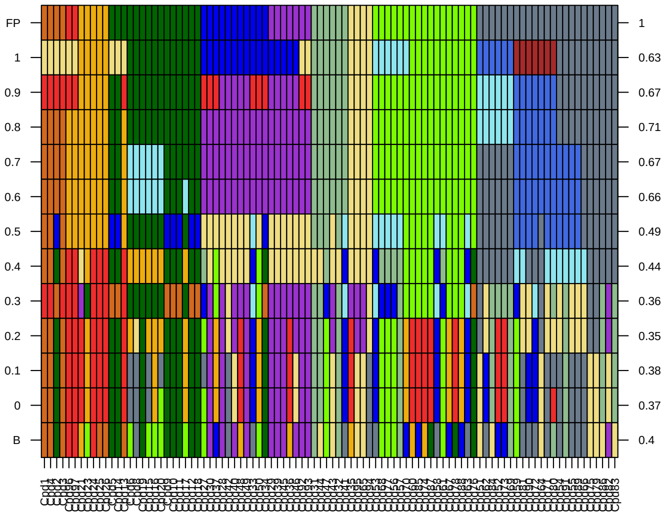


Figure 76: *Comparison for CECc over the Weights for 25 Iterations for Inhouse1*

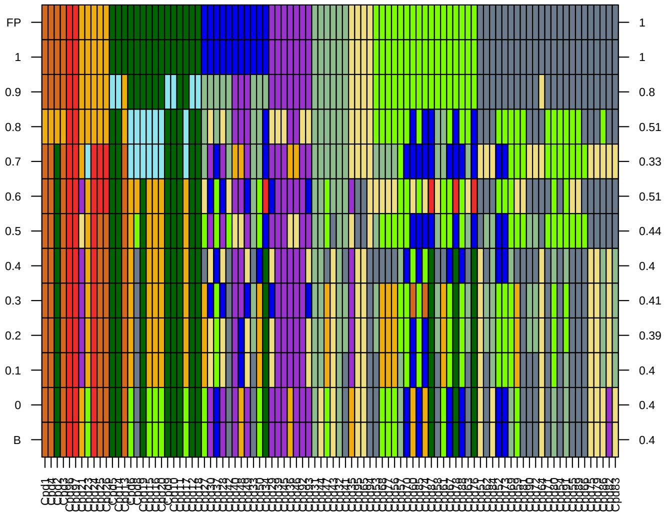


Figure 77: *Comparison for Weighted Clustering over the Weights for Inhouse1 vs FP*

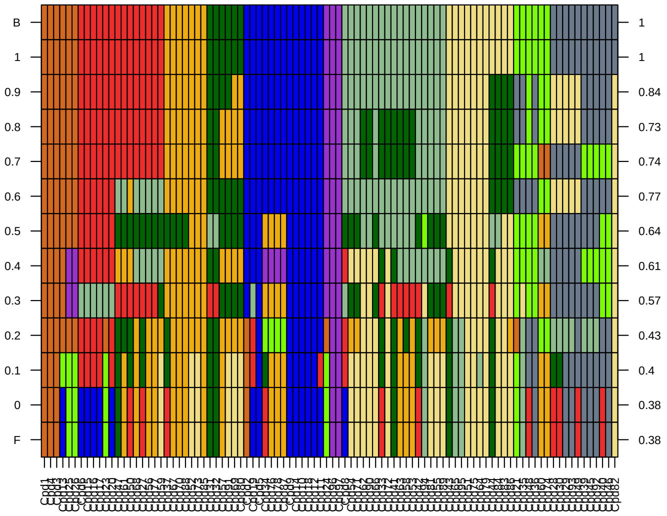


Figure 78: *Comparison for Weighted Clustering over the Weights for Inhouse1 vs B*

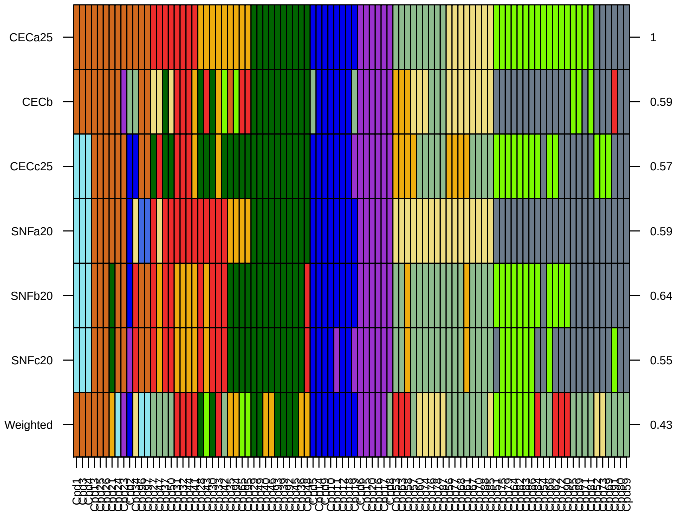


Figure 79: *Comparison of CEC, SNF and Weighted for Inhouse1*

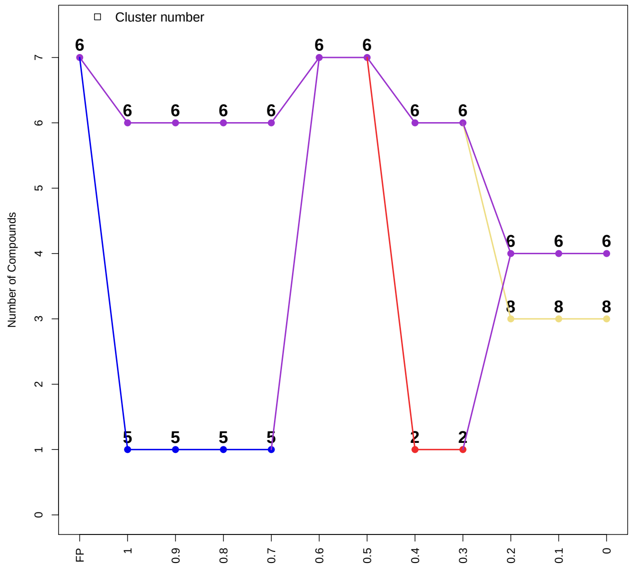


Figure 80: *The Evolution of Cluster 6 of FP over CECa for Inhouse1*

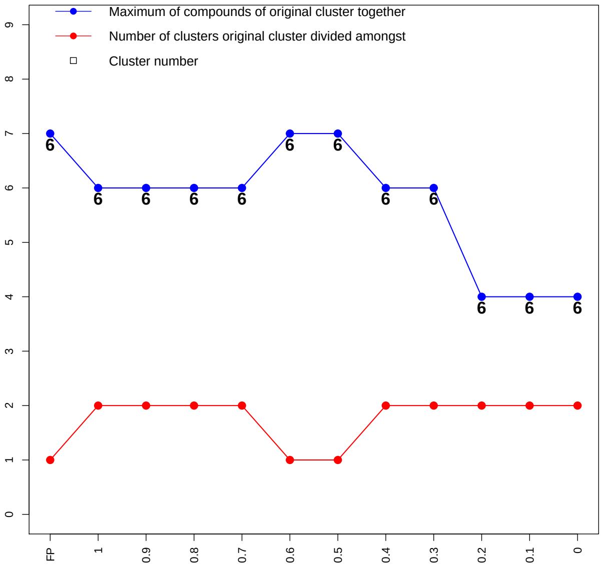


Figure 81: *The Evolution of Cluster 6 of FP over CECa for Inhouse1 - Maximum Number of Compounds*

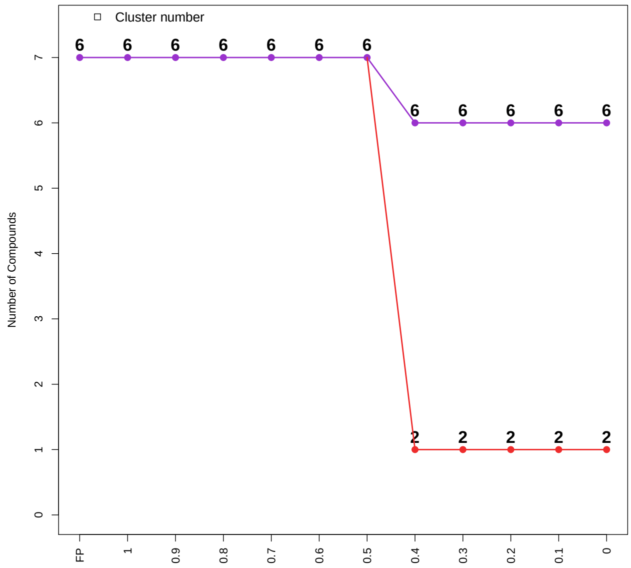


Figure 82: *The Evolution of Cluster 6 of FP over CECb for Inhouse1*

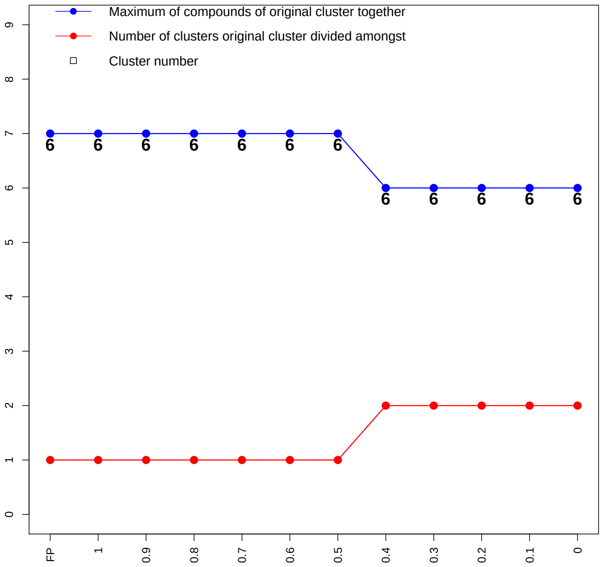


Figure 83: *The Evolution of Cluster 6 of FP over CECb for Inhouse1 - Maximum Number of Compounds*

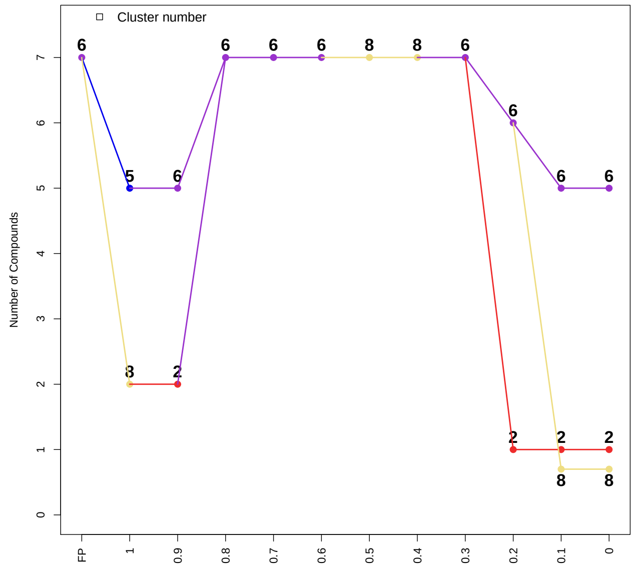


Figure 84: *The Evolution of Cluster 6 of FP over CECc for Inhouse1*

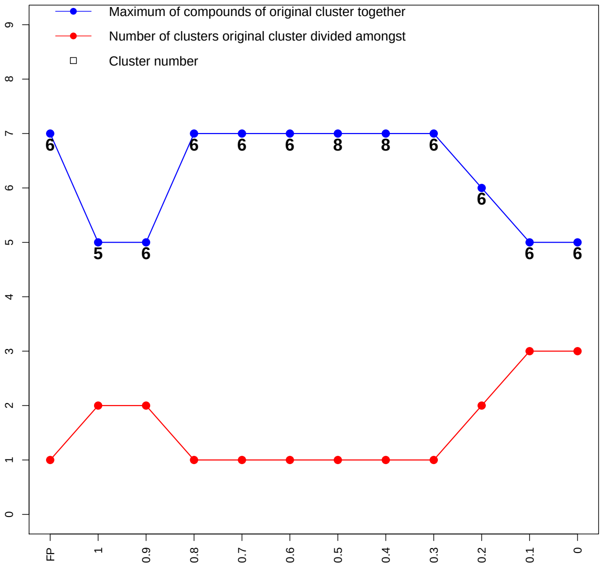


Figure 85: *The Evolution of Cluster 6 of FP over CECc for Inhouse1 - Maximum Number of Compounds*

9.4.3 Differentially Expression and Pathway Analysis

	Cluster 1	*Cluster 2*	*Cluster 3*	*Cluster 4*	*Cluster 5*	*Cluster 6*	*Cluster 7*	*Cluster 8*	*Cluster 9*	*Cluster 10*	*Cluster 11*	*Cluster 12*	*Cluster 13*
	G	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**
FP	135	197	0	0	0	1	0	5	0	0	-	-	-
WonM	257	0	57	0	0	0	0	0	0	0	-	-	-
CECa	-	-	154	0	-	50	0	0	0	0	84	0	0
CECb	257	0	57	0	0	0	0	0	0	0	-	-	-
CECc	0	-	0	50	96	-	1	0	0	0	0	0	-
SNFa	0	197	0	67	-	50	0	5	0	0	84	-	-
SNFb	0	0	-	67	1	-	0	8	0	0	84	0	-
SNFc	0	0	-	10	0	-	0	8	0	0	0	0	-
Weighted	0	128	85	67	0	71	0	0	0	0	-	-	-
B	0	128	0	149	0	0	0	0	0	0	-	-	-
Shared	0	0	0	0	0	0	0	0	0	0	0	0	0
	Comp	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**
FP	4	2	5	15	11	7	6	4	17	23	-	-	-
WonM	7	11	7	6	5	13	7	5	14	19	-	-	-
CECa	-	-	13	8	-	10	8	9	9	17	6	8	6
CECb	7	11	7	6	5	13	7	5	14	19	-	-	-
CECc	3	-	8	7	9	-	4	21	9	13	10	10	-
SNFa	3	2	6	9	-	10	12	4	19	23	6	-	-
SNFb	3	7	-	9	9	-	8	14	15	12	6	11	-
SNFc	3	7	-	9	9	-	8	14	15	15	6	8	-
Weighted	6	3	6	9	9	8	12	7	24	10	-	-	-
B	6	3	6	13	7	11	6	11	14	17	-	-	-
Comp Shared	0	2	0	0	0	5	2	0	0	0	0	0	6

Table 15: *Number of significant genes & pathways at 0.05 significance level - All Methods vs FP - - Inhouse1 Data*

	Genes	**Significant For**
Cluster 1	Gene1017	WonM CECb
	Gene78	WonM CECb B_4
	Gene1032	WonM CECb
	Gene105	WonM CECb
	Gene383	WonM CECb
Cluster 2	Gene275	FP SNFa Weighted B
	Gene172	FP SNFa Weighted B
	Gene669	FP SNFa Weighted B CECa$_3$
	Gene514	FP SNFa
	Gene887	FP SNFa
	Gene933	Weighted B CECa$_3$
	Gene695	Weighted B
Cluster 3	Gene395	WonM CECb Weighted CECc$_4$ SNFc$_4$ CECa$_{11}$ SNFa$_{11}$ SNFb$_{11}$
	Gene217	WonM CECb Weighted CECc$_4$ SNFc$_4$ CECa$_{11}$ SNFa$_{11}$ SNFb$_{11}$
	Gene725	WonM CECb CECc$_4$ CECa$_{11}$ SNFa$_{11}$ SNFb$_{11}$
	Gene975	WonM CECb Weighted CECc$_4$ SNFc$_4$ CECa$_{11}$ SNFa$_{11}$ SNFb$_{11}$
	Gene508	WonM CECb CECc$_4$ CECa$_{11}$ SNFa$_{11}$ SNFb$_{11}$
Cluster 4	Gene173	SNFa SNFb Weighted B CECc$_5$
	Gene120	SNFa SNFb Weighted B CECc$_5$
	Gene1009	SNFa SNFb Weighted B CECc$_5$
	Gene934	SNFa SNFb Weighted
	Gene411	SNFa SNFb Weighted
Cluster 5	-	-
Cluster 6	Gene494	FP CECa SNFa SNFb$_8$ SNFc$_8$
	Gene424	CECa SNFa
	Gene954	CECa SNFa
	Gene153	CECa SNFa SNFb$_8$ SNFc$_8$
	Gene140	CECa SNFa SNFb$_8$ SNFc$_8$
Cluster 7	-	-
Cluster 8	Gene768	FP SNFa
	Gene357	FP SNFa
	Gene586	FP SNFa

	Gene4	FP SNFa
	Gene642	FP SNFa
	Gene382	SNFb SNFc
	Gene540	SNFb SNFc
Cluster 9	-	-
Cluster 10	-	-
Cluster 11	-	-
Cluster 12	-	-
Cluster 13	-	-

Table 16: *Interesting for the Clusters over all Methods for Inhouse1*

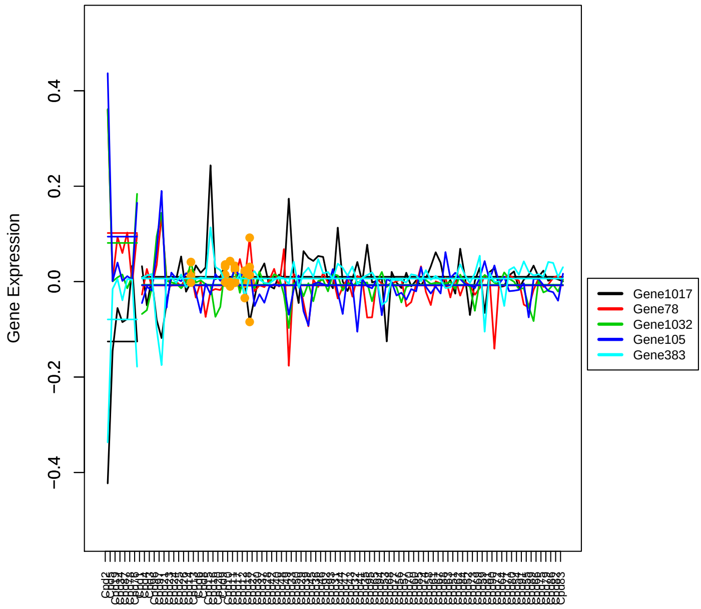


Figure 86: *Gene Profile of Interesting Genes for Cluster 1 of Inhouse1*

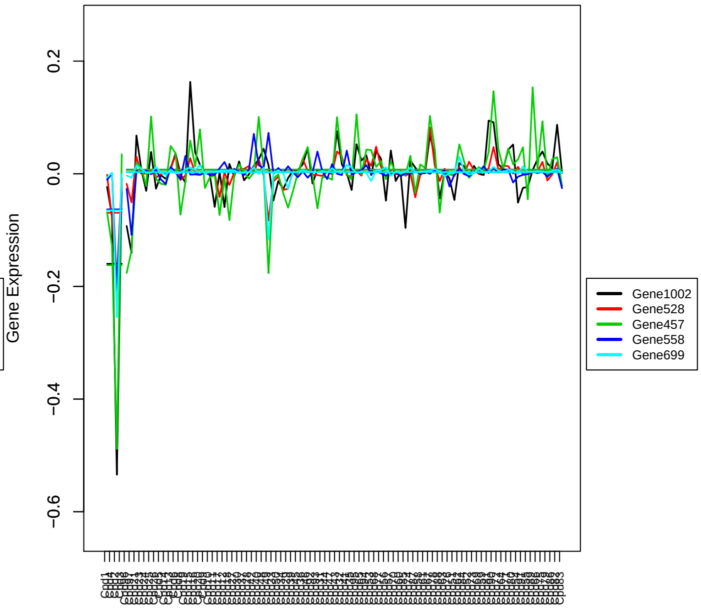


Figure 87: *Gene Profile of Interesting Genes for Cluster 1 of FP for Inhouse1*

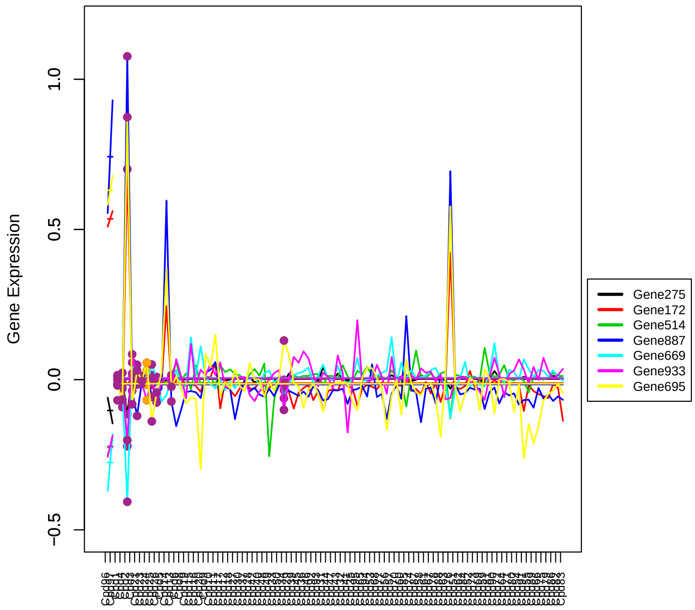


Figure 88: *Gene Profile of Interesting Genes for Cluster 2 of Inhouse1*

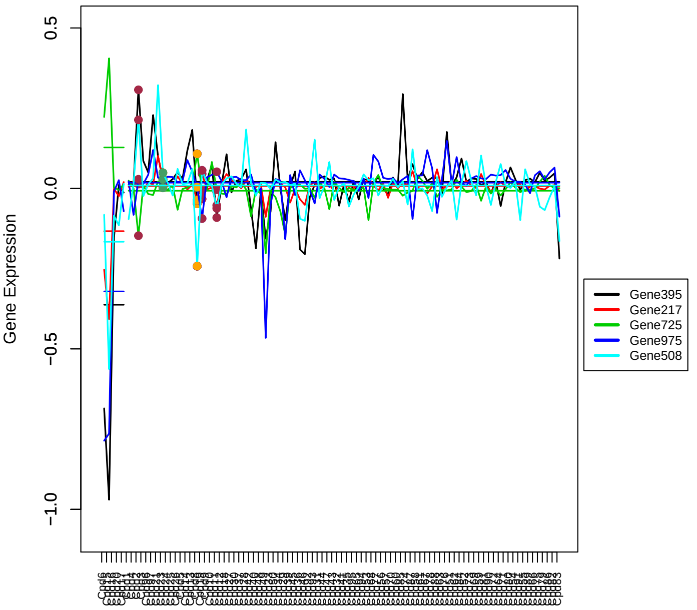


Figure 89: *Gene Profile of Interesting Genes for Cluster 3 of Inhouse1*

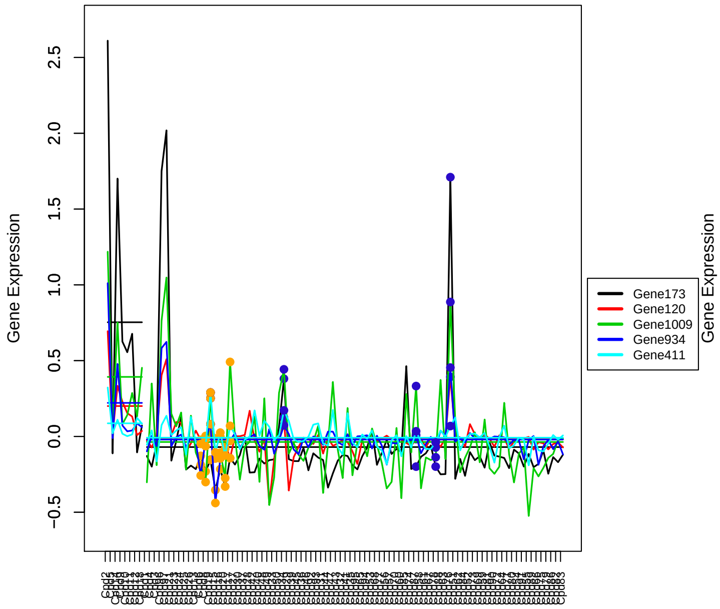


Figure 90: *Gene Profile of Interesting Genes for Cluster 4 of Inhouse1*

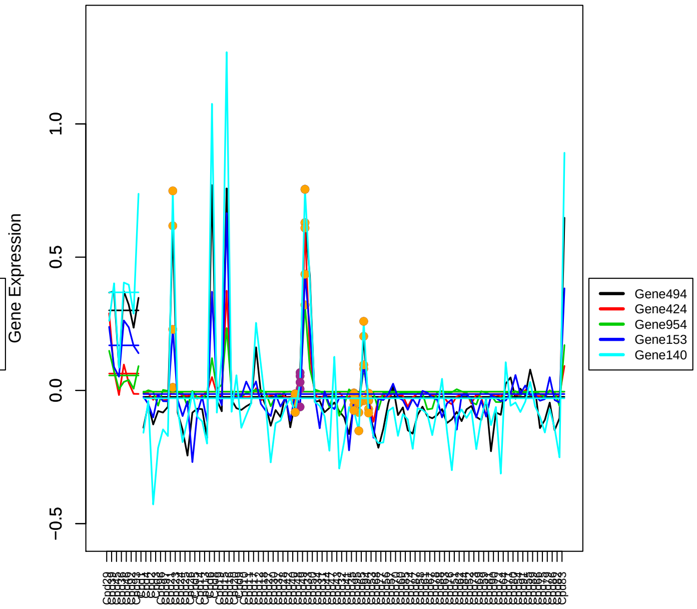


Figure 91: *Gene Profile of Interesting Genes for Cluster 6 of Inhouse1*

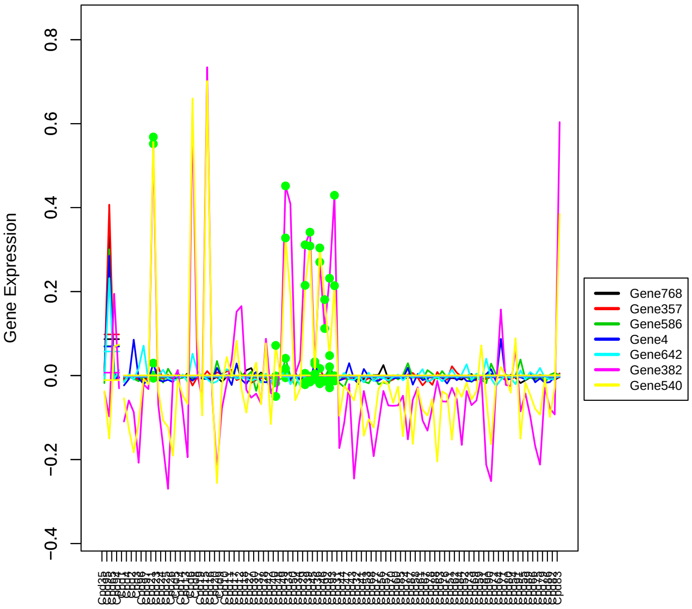


Figure 92: *Gene Profile of Interesting Genes for Cluster 8 of Inhouse1*

	Cluster 1	*Cluster 2*	*Cluster 3*	*Cluster 4*	*Cluster 5*	*Cluster 6*	*Cluster 7*	*Cluster 8*	*Cluster 9*	*Cluster 10*	*Cluster 11*
Weight	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**	**G**
1.0	135	197	0	0	0	1	0	5	0	0	-
0.9	135	197	0	30	-	50	0	4	0	0	0
0.8	-	197	0	0	0	65	0	0	0	0	50
0.7	0	5	7	85	0	71	0	0	0	-	26
0.6	0	28	57	67	137	0	0	8	0	0	-
0.5	0	128	85	67	0	71	0	0	0	0	-
0.4	0	128	85	149	1	0	0	0	1	0	-
0.3	0	128	0	149	137	0	0	0	0	0	-
0.2	0	128	0	149	0	0	0	0	0	0	-
0.1	0	128	0	149	0	0	0	0	0	0	-
0.0	0	128	0	149	0	0	0	0	0	0	-
Shared	0	4	0	0	0	0	0	0	0	0	0
	Comp	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**	**Comp**
1.0	4	2	5	15	11	7	6	4	17	23	-
0.9	4	2	6	8	-	10	14	5	17	22	6
0.8	-	2	10	7	6	5	13	11	22	11	7
0.7	4	5	5	8	13	8	17	15	11	-	8
0.6	4	9	7	9	5	9	6	15	14	16	-
0.5	6	3	6	9	9	8	12	7	24	10	-
0.4	6	3	6	13	8	9	10	13	2	24	-
0.3	8	3	14	13	5	9	14	8	11	9	-
0.2	6	3	14	13	3	8	14	13	11	9	-
0.1	6	3	14	13	3	8	8	13	9	17	-
0.0	6	3	6	13	7	11	6	11	14	17	-
Comp Shared	3	2	0	1	0	0	2	0	0	0	0

Table 17: *Number of significant genes - Weighted Clustering - Inhouse1 Data*

	Genes	**Significant For**
Cluster 1	Gene1002	1.0 0.9
	Gene528	1.0 0.9
	Gene457	1.0 0.9
	Gene558	1.0 0.9
	Gene699	1.0 0.9
Cluster 2	Gene275	All but 0.5
	Gene172	All
	Gene669	1.0 0.9 0.8 0.5 0.4 0.3 0.2 0.1 0.0
	Gene514	1.0 0.9 0.8
	Gene887	1.0 0.9 0.8
	Gene933	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	Gene418	0.7 0.6
	Gene1009	0.6
	Gene695	0.5 0.4 0.3 0.2 0.1 0.0
Cluster 3	Gene524	0.7 0.8_6
	Gene829	0.7 0.8_6
	Gene495	0.7 0.8_6
	Gene203	0.7 0.8_6
	Gene704	0.7 0.8_6
	Gene395	0.6 0.5 0.4 0.9_4 $0.8_1 1$ $0.7_1 1$
	Gene217	0.6 0.5 0.4 0.9_4 $0.8_1 1$ $0.7_1 1$
	Gene725	0.6 0.9_4 $0.8_1 1$ $0.7_1 1$
	Gene975	0.6 0.5 0.4 0.9_4 $0.8_1 1$ $0.7_1 1$
	Gene508	0.6 0.9_4 $0.8_1 1$
	Gene130	0.5 0.4
	Gene783	0.5 0.4
Cluster 4	Gene173	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	Gene120	0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
	Gene1009	0.6 0.5 0.4 0.3 0.2 0.1 0.0
	Gene934	0.6 0.5
	Gene411	0.6 0.5
	Gene78	0.4 0.3 0.2 0.1 0.0

	Gene309	0.4 0.3 0.2 0.1 0.0
Cluster 5	Gene1022	0.6 0.3 0.7_6 0.5_6
	Gene184	0.6 0.3 0.7_6 0.5_6
	Gene352	0.6 0.3 0.7_6 0.5_6
	Gene644	0.6 0.3 0.7_6 0.5_6
	Gene659	0.6 0.3
Cluster 6	Gene494	1.0 0.9
	Gene151	0.7 0.5
Cluster 7	-	-
Cluster 8	Gene768	1.0 0.9
	Gene357	1.0 0.9
	Gene586	1.0 0.9
	Gene4	1.0 0.9
Cluster 9	-	-
Cluster 10	-	-
Cluster 11	-	-

Table 18: *Interesting Genes of the Weighted Clusters of Inhouse1*

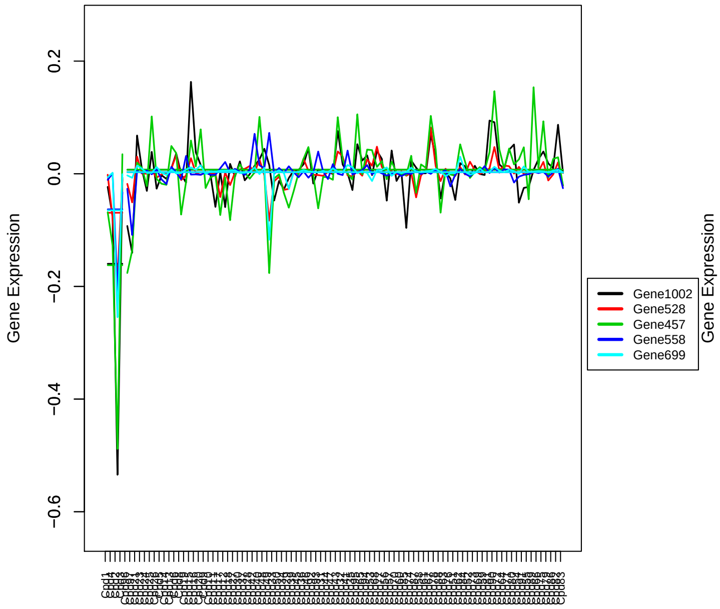


Figure 93: *Gene Profile of Interesting Genes for Cluster 1 of Weighted for Inhouse1*

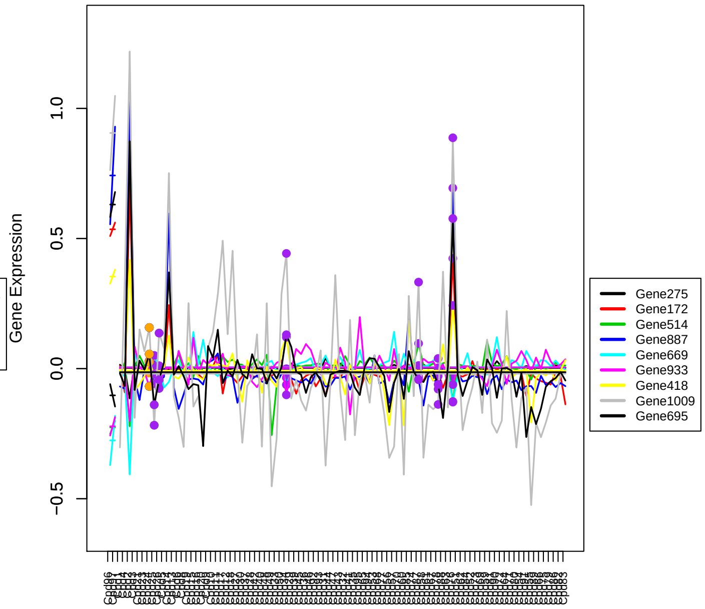


Figure 94: *Gene Profile of Interesting Genes for Cluster 2 of Weighted for Inhouse1*

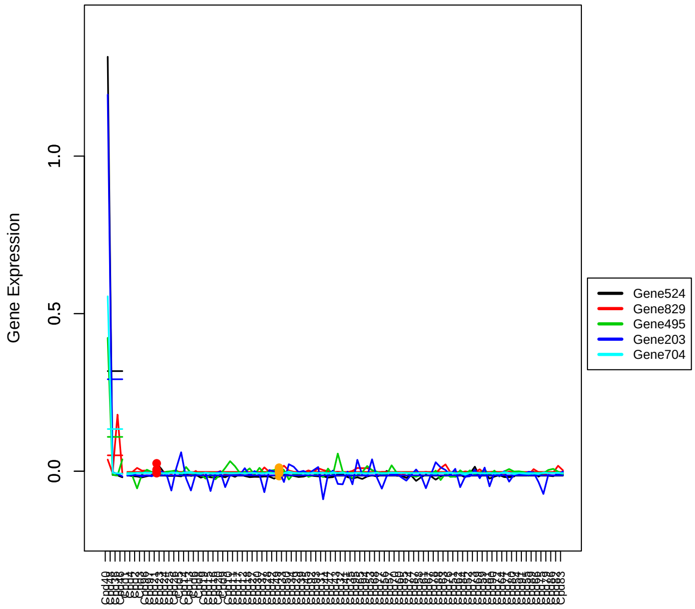


Figure 95: *Gene Profile of Interesting Genes for Cluster 3 of Weighted for Inhouse1 (1)*

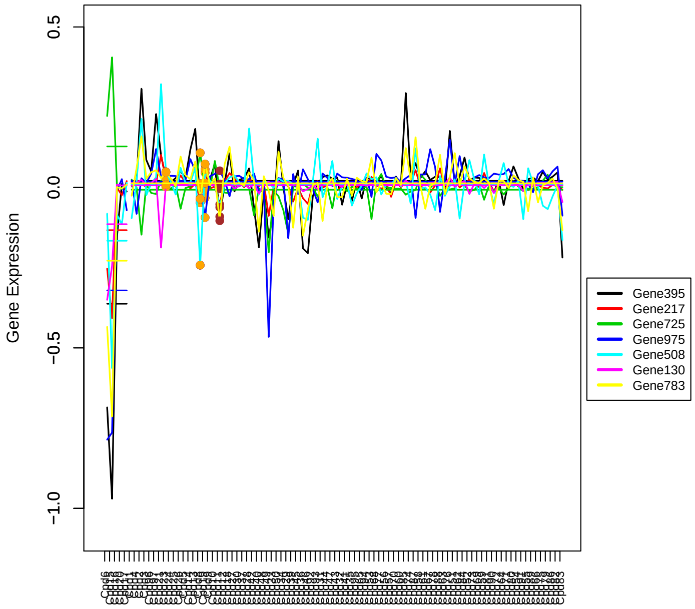


Figure 96: *Gene Profile of Interesting Genes for Cluster 3 of Weighted for Inhouse1 (2)*

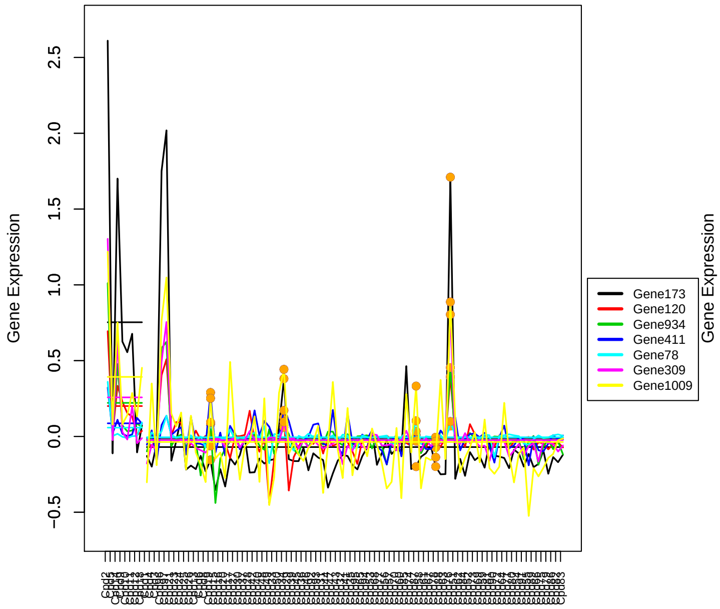


Figure 97: *Gene Profile of Interesting Genes for Cluster 4 of Weighted for Inhouse1*

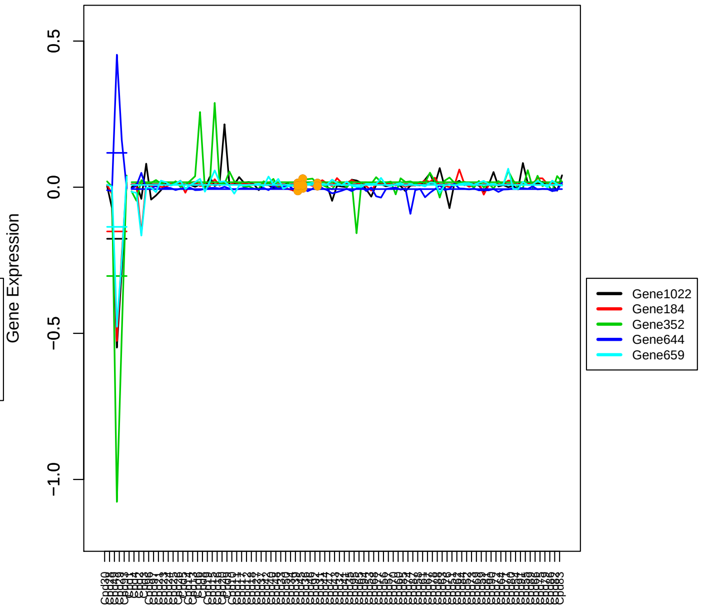


Figure 98: *Gene Profile of Interesting Genes for Cluster 5 of Weighted for Inhouse1*

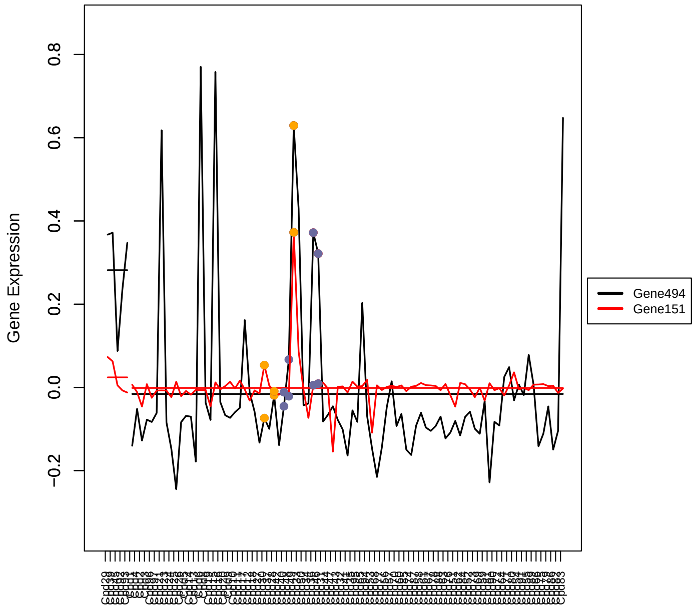


Figure 99: *Gene Profile of Interesting Genes for Cluster 6 of Weighted for Inhouse1*

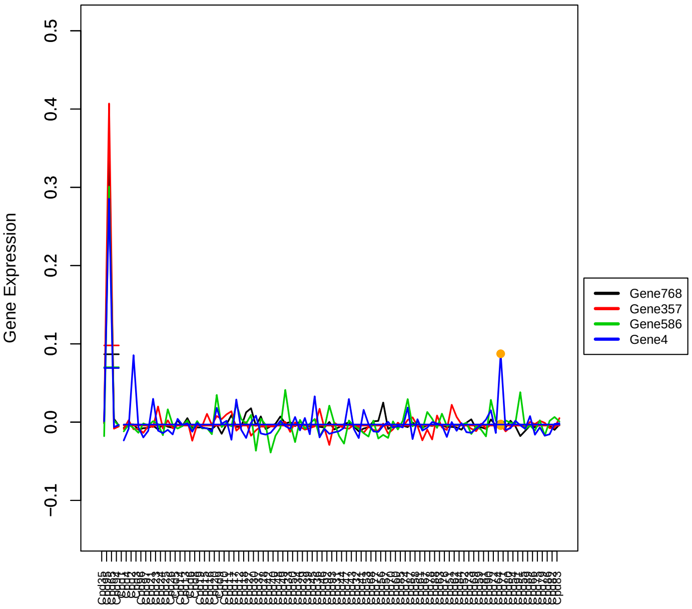


Figure 100: *Gene Profile of Interesting Genes for Cluster 8 of Weighted for Inhouse1*

	All Methods		*Weighted*	*CECa*	*CECb*	*CECc*
Method	**G**	**Weight**	**G**	**G**	**G**	**G**
Fingerprints	1	**1.0**	1	80	1	0
WonM	0	**0.9**	50	80	1	29
CECa	50	**0.8**	0	80	1	0
CECb	0	**0.7**	71	80	1	0
CECc	0	**0.6**	0	50	1	0
SNFa	50	**0.5**	71	50	0	0
SNFb	8	**0.4**	0	0	65	0
SNFc	8	**0.3**	0	0	0	1
Weighted	71	**0.2**	0	0	0	0
Bio-assays	0	**0.1**	0	0	0	71
Shared	0 (1/6)	**0.0**	0	0	0	71
		Shared	0 (1/38)	0 (50)	0 (1)	0 (1/23)

Table 19: *Number of significant genes at 0.05 significance level - Cluster* 6 *of FP - Inhouse1 Data*

	Methods						
	Fingerprints	**CECa**$_{0.5}$	**SNFb**	**Weighted**$_{0.5}$	**CECa**$_{1.0-0.7}$	**CECb**$_{0.4}$	**CECc**$_{0.9}$
	CECb$_{1.0-0.6}$	**CECa**$_{0.6}$	**SNFc**	**Weighted**$_{0.7}$			
	CECc$_{0.3}$	**SNFa**		**CECc**$_{0.1}$			
		Weighted$_{0.9}$		**CECc**$_{0.0}$			
	Compounds						
1	Cpd29	Cpd29	Cpd29	Cpd29	Cpd29	Cpd29	Cpd29
2	Cpd39	Cpd39	Cpd39	Cpd39	Cpd39	Cpd39	Cpd39
3	Cpd45	Cpd45	Cpd45	Cpd45	Cpd36	Cpd45	Cpd45
4	Cpd36	Cpd36	Cpd36	Cpd92	Cpd46	Cpd46	Cpd36
5	Cpd46	Cpd46	Cpd46	Cpd93	Cpd92	Cpd92	Cpd46
6	Cpd92	Cpd92	Cpd92	Cpd30	Cpd93	Cpd93	Cpd28
7	Cpd93	Cpd93	Cpd93	Cpd28	Cpd40	Cpd82	Cpd42
8		Cpd40	Cpd21	Cpd49	Cpd48	Cpd40	Cpd40
9		Cpd48	Cpd40		Cpd49	Cpd49	Cpd48
10		Cpd49	Cpd49			Cpd82	Cpd49
11			Cpd35				
12			Cpd95				
13			Cpd65				
14			Cpd94				
	Genes						
1	Gene494	Gene494	Gene494	Gene1022	Gene494	Gene644	Gene424
2	-	Gene424	Gene382	Gene184	Gene424	Gene1022	Gene1022
3	-	Gene954	Gene153	Gene644	Gene954	Gene184	Gene184
4	-	Gene153	Gene140	Gene352	Gene153	Gene444	Gene352
5	-	Gene140	Gene540	Gene151	Gene644	Gene494	Gene883

Table 20: *Top* 5 *of significant genes at 0.05 significance level - Cluster* 6 *of FP - Inhouse1 Data*

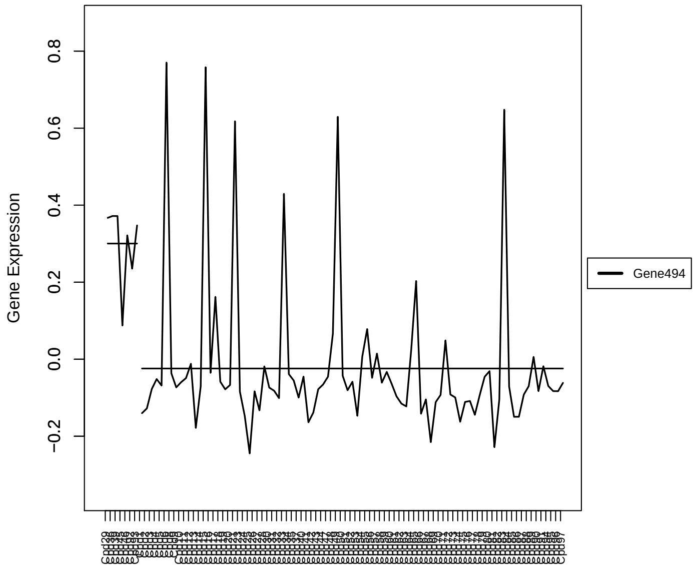


Figure 101: *Gene Profile of Interesting Gene for Cluster 6 of FP, CECb*$_{1.0-0.6}$ *and CECc*$_{0.3}$ *- Inhouse1*

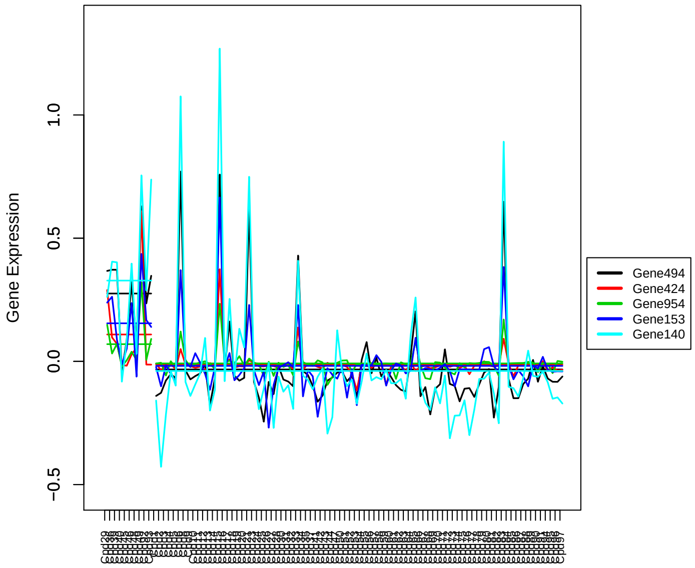


Figure 102: *Gene Profile of Top 5 of Interesting Genes for Cluster 6 of CECa*$_{0.5}$*, CECa*$_{0.6}$*, SNFa and Weighted*$_{0.9}$ *- Inhouse1*

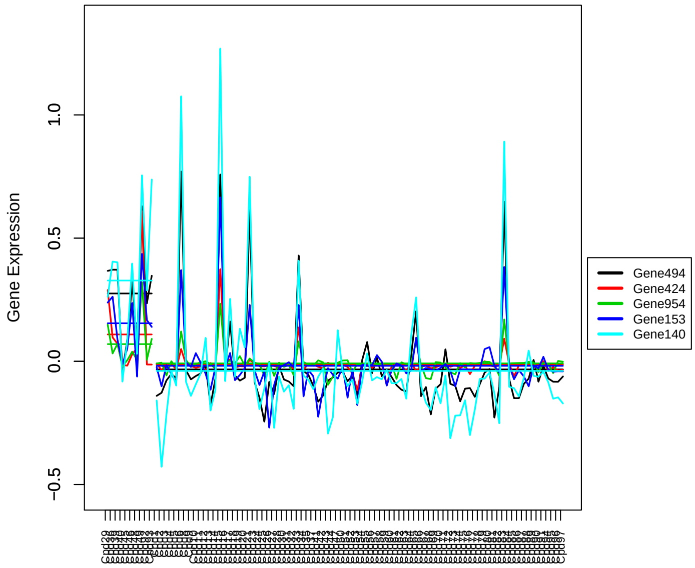


Figure 103: *Gene Profile of Top 5 of Interesting Genes for Cluster 6 of SNFb and SNFc - Inhouse1*

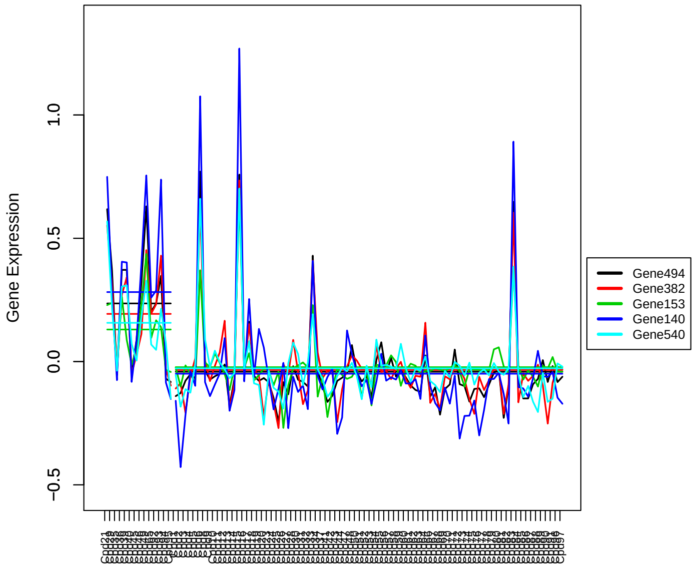


Figure 104: *Gene Profile of Top 5 of Interesting Genes for Cluster 6 of Weighted*$_{0.5}$*, Weighted*$_{0.7}$*, CECc*$_{0.1}$ *and CECc*$_{0.0}$ *- Inhouse1*

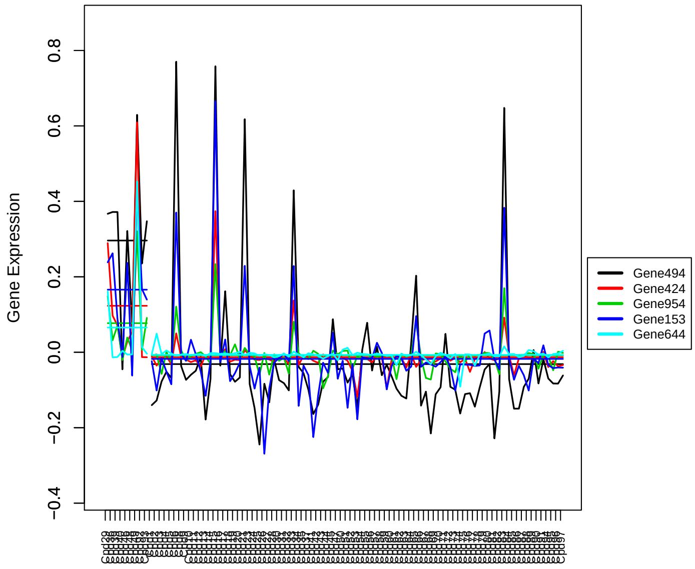


Figure 105: *Gene Profile of Top* 5 *of Interesting Genes for Cluster* 6 *of CECa*$_{1.0-0.7}$ *- Inhouse1*

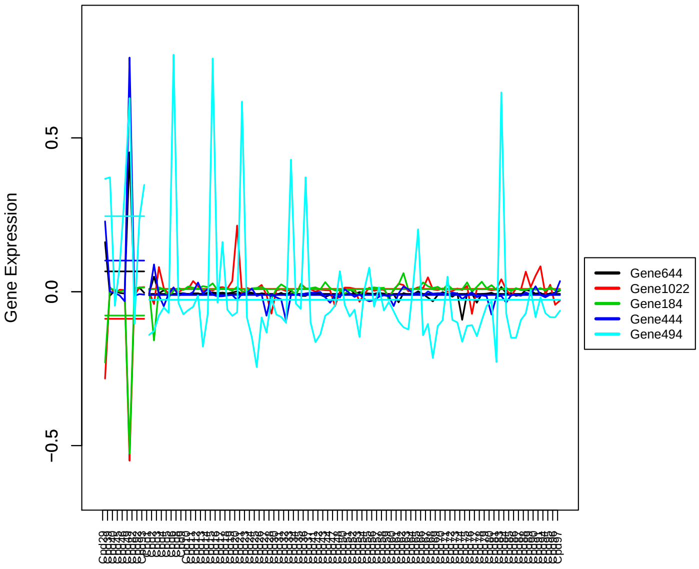


Figure 106: *Gene Profile of Top* 5 *of Interesting Genes for Cluster* 6 *of CECb*$_0$.4 *- Inhouse1*

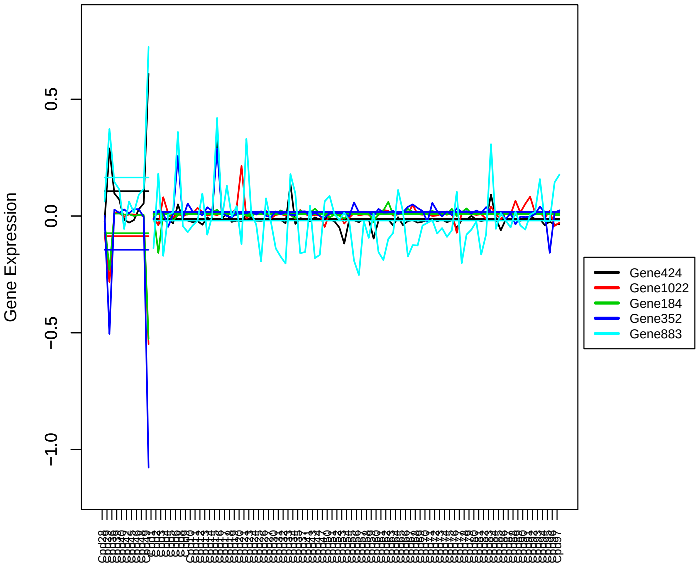


Figure 107: *Gene Profile of Top* 5 *of Interesting Genes for Cluster* 6 *of CECc*$_0$.9 *- Inhouse1*

9.5 Functions

For this thesis project, a function was written for each of the methods mentioned in the methodology in order to facilitate and speed up the analysis. In what follows, every function is explained separately, however the functions are also combined into an *Ultimate function* in which the user can select one or more methods to be applied to its data set. The functions were designed such that these can be used for a continuation of the research and potential expansion of their use. All functions were written in R version 3.1.0.

Required packages

For the functions to work properly, the following packages available in *CRAN* or *Bioconductor* should be installed: *cluster, ade4, samr, a4, MLP, org.Hs.eg.db, plyr, limma, ggplot2, gplots, lattice, SNFtool, plotrix.*

Since the help files are those of the package, from this point onwards, the number of the pages is reset.

Package 'IntClust'

September 6, 2014

Type Package

Title Integrative Data Analysis via Clustering

Version 1.0

Date 2014-08-31

Author Marijke Van Moerbeke

Maintainer Marijke Van Moerbeke <vanmoerbeke.marijke@gmail.com>

Description The package contains several integrative data methods to be applied with clustering. For now, only agglomerative hierarchical clustering is implemented.Visualization functions are available to visualize and compare results of the different methods.

License GPL4

R topics documented:

IntClust-package 2
ADClust 3
ADECa 3
ADECb 4
ADECc 5
bioassayMat 5
CECa 6
CECb 7
CECc 7
Cluster 8
ClusterCols 9
ClusterDistribution 10
Clusterplot 11
Colorpalette 11
ColorsNames 12
ComparePlot 13
DiffGenes 14
DiffGenes.2 15
DiffGenesSelection 15
distanceheatmaps 16
FindCluster 17
FindGenes 17

fingerprintMat1 . . . 18
fingerprintMat2 . . . 18
GeneInfo . . . 19
geneMat1 . . . 19
geneMat2 . . . 19
Geneset.intersect . . . 20
Geneset.intersectSelection . . . 20
GS . . . 21
HeatmapCols . . . 21
LabelCols . . . 22
Labelplot . . . 23
MatrixFunction . . . 23
my_palette1 . . . 24
my_palette1a . . . 25
my_palette2 . . . 25
my_palette3 . . . 25
my_palette3a . . . 26
my_palette4 . . . 26
Pathways . . . 26
Pathways.2 . . . 28
PathwaysIter . . . 28
PathwaysIterSelection . . . 29
PathwaysSelection . . . 30
ProfilePlot . . . 32
ProfilePlotAll . . . 33
Shared . . . 33
SharedComps . . . 34
SharedLimma . . . 35
SharedMLP . . . 35
SharedSelection . . . 36
SharedSelectionLimma . . . 36
SimilarityMeasure . . . 37
SNFa . . . 37
SNFb . . . 38
SNFc . . . 39
targetMat . . . 39
Ultimate . . . 40
WeightedClust . . . 41
WonM . . . 42

Index **43**

`IntClust-package` *Integrative Data Analysis via Clustering*

Description

The package contains several integrative data methods to be applied with clustering. For now, only agglomerative hierarchical clustering is implemented. Visualization functions are available to visualize and compare results of the different methods.

Author(s)

Marijke Van Moerbeke
Maintainer: `<vanmoerbeke.marijke@gmail.com>`

ADClust *Aggregated Data Clustering*

Description

In order to perform aggregated data clustering, the `ADClust` function was written. The function requires as input a list of data matrices of the same type which are combined into a single (larger) matrix. Hierarchical clustering is performed with the `agnes` function and ward link on the resulting data matrix and an applicable distance measure is indicated by the user.

Usage

```
ADClust(List, distmeasure = "tanimoto", clust = "agnes", linkage = "ward")
```

Arguments

`List`	A list of data matrices containing the data. It is assumed the rows are corresponding with the objects.
`distmeasure`	Choice of metric for the dissimilarity matrix.
`clust`	Choice of clustering function. Defaults to `agnes`.
`linkage`	Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output of `ADClust` is a list with 3 elements. The first element AllData is the combined data matrix of all the data sources. The second element DistM is the distance matrix computed from the combined data matrix with the provided distance measure. The final element is the Clust element and contains the resulting clustering. This is the element that will be of interest to further applications.

ADECa *Aggregated Data Clustering - version a*

Description

Function `ADECa` performs aggregated data ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. Unless the number of features is prespecified by the user.

Usage

```
ADECa(List, distmeasure = "tanimoto", t = 10, r = NULL, nclusters = NULL, clust = "agnes",
      linkage = "ward")
```

Arguments

`List` A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.

`distmeasure` The distance measure to be used on the fused data matrix.

`t` The number of iterations.

`r` Optional. The number of features to take for the random sample.

`nclusters` The number of clusters to cut the dendrogram in.

`clust` Choice of clustering function. Defaults to `agnes`.

`linkage` Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output is a list with 3 elements. The first element AllData is the fused data matrix of the provided data sources. The second element S is the resulting co-association matrix and the final element Clust is the result of performing hierarchical clustering on S.

ADECb *Aggregated Data Clustering - version b*

Description

Function `ADECb` performs aggregated data ensemble clustering in which in every iteration the total number of features are used in the clustering procedure. However, the function is capable of cutting the resulting dendrogram several times, each time into a different number of cluster.

Usage

```
ADECb(List,distmeasure="tanimoto",nclusters=seq(5,25,1),clust="agnes",
       linkage="ward")
```

Arguments

`List` A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.

`distmeasure` The distance measure to be used on the fused data matrix.

`nclusters` A sequence of the number of clusters to cut the dendrogram in.

`clust` Choice of clustering function. Defaults to `agnes`.

`linkage` Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output is a list with 3 elements. The first element AllData is the fused data matrix of the provided data sources. The second element S is the resulting co-association matrix and the final element Clust is the result of performing hierarchical clustering on S.

ADECc *Aggregated Data Clustering - version c*

Description

Function ADECc performs aggregated data ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. Unless the number of features is fixed beforehand by the user. Further, each resulting dendrogram can be cut numerous times into a different specific number of clusters.

Usage

```
ADECc=function(List,distmeasure="tanimoto",t=10,r=NULL,nclusters=NULL,clust="agnes",
          linkage="ward")
```

Arguments

List	A list of data matrices of the same type. It is assumed the rows are corresponding with the objects.
distmeasure	The distance measure to be used on the fused data matrix.
t	The number of iterations.
r	Optional. The number of features to take for the random sample.
nclusters	A sequence of the number of clusters to cut the dendrogram in.
clust	Choice of clustering function. Defaults to agnes.
linkage	Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output is a list with 3 elements. The first element AllData is the fused data matrix of the provided data sources. The second element S is the resulting co-association matrix and the final element Clust is the result of performing hierarchical clustering on S.

bioassayMat *Bio-assays score of the Inhouse1 data.*

Description

Bio-assays score of the Inhouse1 data for 13 variables.

Usage

```
data(bioassayMat)
```

Format

The format is: num [1:13, 1:94] 5 7.34 5 7.35 5 5.6 5.73 6 8.15 8.07 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:13] "bio1" "bio2" "bio3" "bio4" ... ..$: chr [1:94] "Cpd1" "Cpd2" "Cpd3" "Cpd4" ...

CECa *Complementary Ensemble Clustering - version a*

Description

Function CECa performs complementary ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. Unless the number of features is prespecified by the user.

Usage

```
CECa(List, distmeasure = c("tanimoto", "tanimoto"), t = 10, r = NULL, nclusters = NULL,
     weight = NULL, clust = "agnes", linkage = "ward", Clustweight = 0.5)
```

Arguments

List	A list of the data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A vector of the distance measures to be used on each data matrix.
t	The number of iterations.
r	Optional. A vector of The number of features to take for each data set.
nclusters	Vector of the number of clusters to cut the dendrogram in of each data source.
weight	Optional. A specific weight to give to the first co-association matrix. If NULL, the weight is sequence from 0 to 1 and a result is produced for each.
clust	Choice of clustering function. Defaults to agnes.
linkage	Choice of inter group dissimilarity. Defaults to Ward link.
Clustweight	A weight for which the result will be put aside of the other results. This was done for comparative reason and easy access.

Value

The output is a list with 4 elements. The first element Incidence contains the summed incidence matrices for each data source. The second element is IncidenceComb and contains the resulting co-association matrix after a weighted addition of the matrices in Incidence for each specified weight. Results is the third element and consists of the resulting hierarchical clustering of each co-association matrix given in IncidenceComb. The final element Clust is the result of CEC for the weight specified in Clustweight.

CECb *Complementary Ensemble Clustering - version b*

Description

Function CECb performs complementary ensemble clustering in which in every iteration the total number of features are used in the clustering procedure. However, the function is capable of cutting the resulting dendrogram several times, each time into a different number of cluster.

Usage

```
CECb(List, distmeasure = c("tanimoto", "tanimoto"), nclusters=seq(5,25,1), weight = NULL,
     clust = "agnes", linkage = "ward", Clustweight = 0.5)
```

Arguments

List	A list of the data matrices. It is assumed the rows are corresponding with the objects.
distmeasure	A vector of the distance measures to be used on each data matrix.
nclusters	Sequence of the number of clusters to cut the dendrogram.
weight	Optional. A specific weight to give to the first co-association matrix. If NULL, the weight is sequence from 0 to 1 and a result is produced for each.
clust	Choice of clustering function. Defaults to agnes.
linkage	Choice of inter group dissimilarity. Defaults to Ward link.
Clustweight	A weight for which the result will be put aside of the other results. This was done for comparative reason and easy access.

Value

The output is a list with 4 elements. The first element Incidence contains the summed incidence matrices for each data source. The second element is IncidenceComb and contains the resulting co-association matrix after a weighted addition of the matrices in Incidence for each specified weight. Results is the third element and consists of the resulting hierarchical clustering of each co-association matrix given in IncidenceComb. The final element Clust is the result of CEC for the weight specified in Clustweight.

CECc *Complementary Ensemble Clustering - version c*

Description

Function CECc performs complementary ensemble clustering in which in every iteration the number of random samples taken is randomly set between m/2 and m-1 with m the total number of features. Unless the number of features is fixed beforehand by the user. Further, each resulting dendrogram can be cut numerous times into a different specific number of clusters.

Usage

```
CECc(List, distmeasure = c("tanimoto", "tanimoto"), t = 10, r = NULL, nclusters = NULL,
     weight = NULL, clust = "agnes", linkage = "ward", Clustweight = 0.5)
```

Arguments

`List`	A list of the data matrices. It is assumed the rows are corresponding with the objects.
`distmeasure`	A vector of the distance measures to be used on each data matrix.
`t`	The number of iterations.
`r`	Optional. A vector of The number of features to take for each data set.
`nclusters`	Sequence of the number of clusters to cut the dendrogram in.
`weight`	Optional. A specific weight to give to the first co-association matrix. If NULL, the weight is sequence from 0 to 1 and a result is produced for each.
`clust`	Choice of clustering function. Defaults to `agnes`.
`linkage`	Choice of inter group dissimilarity. Defaults to Ward link.
`Clustweight`	A weight for which the result will be put aside of the other results. This was done for comparative reason and easy access.

Value

The output is a list with 4 elements. The first element Incidence contains the summed incidence matrices for each data source. The second element is IncidenceComb and contains the resulting co-association matrix after a weighted addition of the matrices in Incidence for each specified weight. Results is the third element and consists of the resulting hierarchical clustering of each co-association matrix given in IncidenceComb. The final element Clust is the result of CEC for the weight specified in Clustweight.

`Cluster` *Clustering on a single source*

Description

The function `Cluster` was written to perform clustering on a single source of information, i.e one data matrix. For now, the only option is to carry out agglomerative hierarchical clustering with the ward link as it was implemented in the `agnes` function in the cluster library of R. The option is available to compute the gap statistic to determine the optimal number of clusters. The gap statistic is determined by the criteria described by the cluster package: firstSEmax, globalSEmax, firstmax,globalmax, Tibs2001SEmax. The number of iterations is set to a default of 500. The implemented distances to be used for the dissimilarity matrix are jaccard, tanimoto and euclidean. The jaccard distances were computed with the `dist.binary(...,method=1)` function in the ade4 package and the euclidean ones with the `daisy` function in again the cluster package. The Tanimoto distances were implemented manually following the formula in the methodology.

Usage

```
Cluster(Data, distmeasure = "tanimoto", clust = "agnes", linkage = "ward", gap = TRUE,
        maxK = 50)
```

Arguments

`Data`	A matrix containing the data. It is assumed the rows are corresponding with the objects.
`distmeasure`	Choice of metric for the dissimilarity matrix.
`clust`	Choice of clustering function. Defaults to `agnes`.
`linkage`	Choice of inter group dissimilarity. Defaults to Ward link.
`gap`	Logical. Indicator if gap statistics should be computed. Setting to $FALSE$ will greatly reduce the computation time.
`maxK`	The maximum number of clusters to be considered during the gap.

Value

The output will be a list with $2$ components. The first element DistM is the distance matrix to be used as input for the clustering function. This is provided if another clustering technique is preferred to the `agnes` clustering. The second element Clust contains the output from the `agnes` function and thus the resulting clustering. If the gap option was indicated to be true, another 3 elements are joined to the list. Clust_gap contains the output from the function to compute the gap statistics and gapdata is a subset of this output. Both can be used to make plots to visualize the gap statistic. The final component is k which is a matrix containing the optimal number of cluster determined by each criterion mentioned earlier.

`ClusterCols` *Internal Function of* `Clusterplot`

Description

The `ClusterCols` function is capable of coloring the leaves of a dendrogram of one method with the colors (and thus cluster) the compounds had in the clustering of another method. This helps to see how compounds have changed clusters between these two methods. Given a dendrogram to color of method 1 and the resulting clustering of method 2 (the output of an `agnes` function), each leaf (compound) will be given the color of the cluster it belongs to in method 2. If groups of compounds with the same color appear in the dendrogram, this implies that the clustering of these is similar to the original clustering. If method 1 and method 2 are the same, the dendrogram is just colored by cluster.

Usage

```
ClusterCols(x, Data, nclusters = 7, cols = my_palette2)
```

Arguments

`x`	The dendrogram of method 1 to be colored
`Data`	The output of an `agnes` function, i.e. the resulting clustering of method 2 on which the colors should be based.
`nclusters`	The number of clusters to cut the dendrogram in.
`cols`	The colors for the clusters.

Value

A dendrogram of method 1 with colored leaves of which the colors are determined by method 2.

ClusterDistribution *Distribution of a Cluster over Methods*

Description

It is often desired to track a specific selection of object over the different methods and/or weights. This can be done with the `ClusterDistribution`. For every method, it is tracked where the objects of the selections are situated. This provided with extra information as which compounds of the original selection can be found in this cluster and which are extra. Further, plots of the distribution of the compounds can be produced. One plot follows the complete distribution of the cluster while another one focuses on either the maximum number of compounds or a specific cluster whatever is specified. It are the number of compounds that are plotted. A table can be produced as well, that separates the objects that are shared over all methods from those extra in the original selection and extra for the other methods. The `MatrixFunction` is applied to make sure that the clusters are comparable over the methods.

Usage

```
ClusterDistribution(List, Selection, nrclusters, followMaxComps = FALSE,
                    followClust = TRUE, fusionsLog = TRUE, WeightClust = TRUE,
                    names = NULL, reverse = FALSE, Plot = TRUE, Table = TRUE,
                    CompletePlot = FALSE, cols)
```

Arguments

`List`	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
`Selection`	The selection of objects to follow.
`nrclusters`	The number of clusters to cut the dendrogram in.
`followMaxComps`	Logical for plot. Whether to follow the maximum of objects.
`followClust`	Logical for plot. Whether to follow the specific cluster.
`fusionsLog`	To be handed to `MatrixFunction`.
`WeightClust`	To be handed to `MatrixFunction`.
`names`	Optional. Names of the methods.
`reverse`	Logical. Should the last element of the List be used as reference? Mostly to be used for CEC or weighted clustering.
`Plot`	Logical. Should a plot be produced. Depending on followMaxComps and followClust it focuses on the maximum of compounds or a cluster.
`Table`	Logical. Should a table with the compounds per method and the shared compounds be produced?
`CompletePlot`	Logical. Should the complete distribution of the selection be plotted?
`cols`	The colors used for the different clusters.

Value

The output is a list with an element for every method. This element contains the selection, the number of clusters the selection is divided over, the minimum and maximum number of compounds found together and per found cluster also information n which of the selection it contain and which are extra to this cluster. Depending on whether followMaxComps or followClust is specified, the cluster of interest is mentioned separately as well for easy access. If the option was specified to create a table, this can be found under the “Table" element. Each plot that was specified to be created is plotted in a new window in the graphics console.

Clusterplot *Color the Leaves of a Dendrogram*

Description

The above described function `ClusterCols` is used in the function `Clusterplot` which actually plots the dendrogram made by `ClusterCols`. Further, given the outputs of any other functions, it is capable of selection the elements needed for `ClusterCols`.

Usage

```
Clusterplot(Data1, Data2, nclusters = 7, cols = my_palette2, ...)
```

Arguments

`Data1`	The resulting list method 1 which contains the dendrogram to be colored.
`Data2`	The resulting list method 2 , i.e. the resulting clustering on which the colors should be based.
`nclusters`	The number of clusters to cut the dendrogram in.
`cols`	The colors for the clusters.
`...`	Other options which can be given to the plot function.

Value

A plot of the dendrogram of method 1 with colored leaves of which the colors are determined by method 2.

Colorpalette *Colorpalette*

Description

In order to facilitate the influence of the different methods on the clustering of the compounds colors can be used. The function `Colorpalette` is able to pick out as many colors as there are clusters. This is done with the help of the `colorRampPalette` function of the grDevices package

Usage

```
Colorpalette(colors = c("red", "green"), ncols = 5)
```

Arguments

`colors`	A vector containing the colors of choice.
`ncols`	The number of colors to be specified. If higher than the number of colors, it specifies ncols in the region between the colors.

Value

A vector containing the hex codes of the chosen colors.

Note

The function `cutree` is often used to cut the dendrogram into a specific number of clusters. This function numbers the clusters in the order of the names of the compounds in the data and not in the ordering in which clusters are formed. Therefore, the ordering of the colors does not agree to the ordering of the cluster. To make sure that the color number corresponds to the cluster number, the numbering of the colors was adapted in the visualization functions `ClusterCols` and `distanceheatmaps`.

`ColorsNames` *Internal function of ComparePlot*

Description

The `ColorsNames` function is used on the output of the `MatrixFunction` and matches the cluster numbers indicated by the cell with the names of the colors. This is necessary to produce the plot of the `ComparePlot` function.

Usage

```
ColorsNames(MatrixColors, cols = my_palette2)
```

Arguments

`MatrixColors`	A matrix which is the output of the MatrixFunction function.
`cols`	The hex codes of the colors to be used.

Value

A vector containing the hex code of the color that corresponds to each cell of the matrix. This function is called upon by the `ComparePlot function`.

ComparePlot *Comparison of Clustering Results over Multiple Methods*

Description

A visual comparison of all methods is handy to see which compounds will always cluster together independent of the applied methods. To this aid the function `ComparePlot` has been written.

Usage

```
ComparePlot(List, nclusters = 7, cols = my_palette2, fusionsLog = FALSE, WeightClust = FALSE,
            names = NULL, reverse = FALSE, margins = c(8.1, 3.1, 3.1, 4.1), ...)
```

Arguments

`List`	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
`nclusters`	The number of clusters to cut the dendrogram in.
`cols`	The hex codes of the colors to be used.
`fusionsLog`	To be handed to `MatrixFunction`.
`WeightClust`	To be handed to `MatrixFunction`.
`names`	Optional. Names of the methods to be used as labels for the columns.
`reverse`	Logical. Should the last element of the List be used as reference? Mostly to be used for CEC or weighted clustering.
`margins`	Optional. Margins to be used for the plot.
`...`	

Details

This functions makes use of the functions `MatrixFunction` and `Colorsnames`. Given a list with the outputs of several methods, the first step is to call upon `MatrixFunction` and to produce a matrix of which the columns are ordered according to the ordering of the objects of the first method in the list. Each cell represent the number of the cluster the object belongs to for a specific method indicated by the rows. The clusters are arranged in such a way that these correspond to that one cluster of the referenced method that they have the most in common with. The function `color2D.matplot` produces a plot of this matrix but needs a vector indicating the names of the colors to be used. This is where `ColorsNames` comes in. A vector of the color names of the output of the `MatrixFunction` is created and handed to `color2D.matplot`. It is optional to adjust the margins of the plot and to give a vector with the names of the methods which will be used as labels for the rows in the plot. The labels for the columns are the names of the object in the order of clustering of the referenced method. Further, the similarity measures of the methods compared to the reference will be computed and shown on the right side of the plot.

Value

A plot which translates the matrix output of the function `MatrixFunction` in which the columns represent the objects in the ordering the referenced method and the rows the outputs of the given methods. Each cluster is given a distinct color. This way it can be easily observed which objects will cluster together. The labels on the right side of the plot are the similarity measures computed by `SimilarityMeasure`.

DiffGenes *Differential Gene Expression*

Description

It was decided to use the limma method to find possible genes of interest. The function DiffGenes will, given the output of a certain method, look for genes that are differentially expressed for each cluster by applying the limma function to that cluster and compare it to all other clusters simultaneously. If a list of outputs of several methods is provided, DiffGenes will perform the limma function for each method. The function rearranges the clusters of the methods to a reference method such that a comparison is made easier. Given a list of methods, it calls upon MatrixFunction to rearrange the number of clusters according to the first element of the list which will be used as the reference.

Usage

```
DiffGenes(List, GeneExpr = geneMat, nclusters = 7, method = "limma", sign = 0.05,
          top = NULL, fusionsLog = TRUE, WeightClust = TRUE, names = NULL)
```

Arguments

List	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
GeneExpr	The gene expression matrix of the objects.
nclusters	he number of clusters to cut the dendrogram in.
method	The method to applied to look for DE genes. For now, only the limma method is available
sign	The significance level to be handled.
top	Overrules sign. The number of top genes to be shown.
fusionsLog	To be handed to MatrixFunction.
WeightClust	To be handed to MatrixFunction.
names	Optional. Names of the methods.

Value

The output is a list in which there is element for each method. Per method, there is a list per cluster and this contains another list of length 2. The first element Compounds contains the objects in said cluster and the second element Genes the genes that are differentially expressed according to the specified significance level or a specific number of genes specified with top.

DiffGenes.2 *DiffGenes.2*

Description

Internal function of DiffGenes if only 1 method is specified

Usage

```
DiffGenes.2(Data, GeneExpr = geneMat, nclusters = 7, method = "limma", sign = 0.05,
            top = NULL)
```

Arguments

Data	Data
GeneExpr	The gene expression matrix of the objects.
nclusters	he number of clusters to cut the dendrogram in.
method	The method to applied to look for DE genes. For now, only the limma method is available
sign	The significance level to be handled.
top	Overrules sign. The number of top genes to be shown.

Value

The significant genes per cluster for the method.

DiffGenesSelection *Differential Gene Expression for a Selection*

Description

The function `DiffGenesSelection` performs the same procedure as `DiffGenes` but only for a specific selection of compounds and only for the cluster that contains the maximum number of compounds of the selection.

Usage

```
DiffGenesSelection(List, Selection, GeneExpr = geneMat, nclusters = 7, method = "limma",
                  sign = 0.05, top = NULL, fusionsLog = TRUE, WeightClust = TRUE,
                    names = NULL)
```

Arguments

`List` A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.

`Selection` The selection of objects to follow.

`GeneExpr` The gene expression matrix of the objects.

`nclusters` he number of clusters to cut the dendrogram in.

`method` The method to applied to look for DE genes. For now, only the limma method is available

`sign` The significance level to be handled.

`top` Overrules sign. The number of top genes to be shown.

`fusionsLog` To be handed to `MatrixFunction`.

`WeightClust` To be handed to `MatrixFunction`.

`names` Optional. Names of the methods.

Value

The output is a list in which there is element for each method. Per method, there is only the cluster that contains the maximum number of compounds and this contains another list of length 2. The first element Compounds contains the objects in said cluster and the second element Genes the genes that are differentially expressed according to the specified significance level or a specific number of genes specified with top.

`distanceheatmaps` *Internal function of HeatmapCols*

Description

Another way to compare to methods is via an adaptation of heatmaps. The input of this function is the resulting clustering (the Clust element of the list) of two methods and can be seen as: method 1 versus method 2. The dendrograms are cut into a specific number of clusters. Each cluster of method 2 and its members are given a distinct color represented by a number. These are the clusters to which a comparison is made. A matrix is set up of which the columns are determined by the ordering of clustering of method 2 and the rows by the ordering of method 1. Every column represent one object just as every row and every column represent the color of its cluster. A function visits every cell of the matrix. If the objects represented by the cell are still together in a cluster, the color of the column is passed to the cell. This creates the distance matrix which can be given to the `HeatmapCols` function to create the heatmap.

Usage

```
distanceheatmaps(Data1, Data2, names, nclusters = 7)
```

Arguments

`Data1` The resulting clustering of method 1.

`Data2` The resulting clustering of method 2.

`names` The names of the objects in the data sets.

`nclusters` The number of clusters to cut the dendrogram in.

Value

A matrix indicating whether or not objects are still in the same cluster in method 1 compared to method 2.

FindCluster *Finding Clusters*

Description

The `FindCluster` function is helpful in selecting compounds of a specific cluster. After performing the `MatrixFunction`, it will return the compounds in the cluster of the selected row.

Usage

```
FindCluster(List, nclusters, select = c(1, 4), fusionsLog = TRUE, WeightClust = TRUE,
            names = NULL)
```

Arguments

`List`	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
`nclusters`	The number of clusters to cut the dendrogram in.
`select`	The row and the number of the cluster to select.
`fusionsLog`	Logical indicator for the fusion of clusters.
`WeightClust`	Optional. To be used for the outputs of CEC or WeightedClust. Then only the result of the Clust element is considered.
`names`	Optional. Names of the methods.

Value

The compounds of the selected cluster.

FindGenes *Find significant Genes*

Description

Due to the shifting of compounds over the clusters for the different methods. it is possible that the same gene is found significant for a different cluster in another method. These can be tracked with the `FindGenes` function. Per method and per cluster, it will take note of the genes found significant and investigate if these were also find for another cluster in another method.

Usage

```
FindGenes(DataLimma, names = NULL)
```

Arguments

`DataLimma`	Output of the `DiffGenes` function
`names`	Optional. Names of the methods.

Value

A list with an element per cluster and per cluster one for every gene. Per gene, a vector is given which contain the methods for which the gene was found. If the cluster is changed compared to the reference method of DataLimma, this is indicated with an underscore.

`fingerprintMat1` *The fingerprints for the MCF7 data.*

Description

The 250 fingerprints for the MCF7 data.

Usage

```
data(fingerprintMat1)
```

Format

The format is: logi [1:56, 1:250] FALSE FALSE FALSE FALSE FALSE FALSE ... - attr(*, "dimnames")=List of 2 ..$: chr [1:56] "metformin" "phenformin" "phenyl biguanide" "estradiol" ... ..$: chr [1:250] "-2147375257" "-2147119955" "-2146474760" "-2145840573" ...

`fingerprintMat2` *The fingerprints for the Inhouse1 data.*

Description

The 324 fingerprints for the Inhouse1 data.

Usage

```
data(fingerprintMat2)
```

Format

The format is: logi [1:324, 1:94] FALSE FALSE FALSE FALSE TRUE FALSE ... - attr(*, "dimnames")=List of 2 ..$: chr [1:324] "FP1" "FP2" "FP3" "FP4" ... ..$: chr [1:94] "Cpd1" "Cpd2" "Cpd3" "Cpd4" ...

`GeneInfo` *Gene info*

Description

The entrezIdentifiers of the used genes in MCF7.

Usage

```
data(GeneInfo)
```

Format

The format is: chr [1:2434, 1:3] "10001" "100129361" "10015" "100188893" ... - attr(*, "dimnames")=List of 2 ..$: chr [1:2434] "1" "2" "3" "4" ... ..$: chr [1:3] "ENTREZID" "SYMBOL" "GENENAME"

`geneMat1` *The gene expression for MCF7*

Description

The gene expression of 2434 genes for MCF7

Usage

```
data(geneMat1)
```

Format

The format is: num [1:2434, 1:56] -0.0772 -0.0698 -0.055 -0.0498 -0.0597 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:2434] "MED6" "LOC100129361" "PDCD6IP" "TOMM6" ... ..$: chr [1:56] "metformin" "phenformin" "phenyl biguanide" "estradiol" ...

`geneMat2` *The gene expression for Inhouse1*

Description

The gene expression of 1056 genes for Inhouse1.

Usage

```
data(geneMat2)
```

Format

The format is: num [1:1056, 1:94] 10.03 4.81 7.76 5.45 7.31 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:1056] "Gene1" "Gene2" "Gene3" "Gene4" ... ..$: chr [1:94] "Cpd1" "Cpd2" "Cpd3" "Cpd4" ...

Geneset.intersect *Intersection over resulting gene sets of PathwaysIter*

Description

The function Geneset.intersect puts per method the results of the PathwaysIter function together for each cluster and takes the intersection over the iterations per cluster per method. This is to see if over the different resamplings of the data, similar pathways were discovered.

Usage

```
Geneset.intersect(list.output, sign, names = NULL, seperatetables = FALSE,
                  separatepvals = FALSE)
```

Arguments

list.output	The output of the PathwaysIter function.
sign	The significance level to be handled for cutting of the pathways.
names	Optional. Names of the methods.
seperatetables	If TRUE, a separate element is created per cluster. containing the pathways for each iteration.
separatepvals	If TRUE, the p-values of the each iteration of each pathway in the intersection is given. If FALSE, only the mean p-value is provided.

Value

The output is a list with an element per method. For each method, it is portrayed per cluster which pathways belong to the intersection over all iterations and their corresponding mean p-values.

Geneset.intersectSelection *Intersection over resulting gene sets of PathwaysIterSelection*

Description

The function Geneset.intersectSelection performs the same procedure as Geneset.intersect but only for a specific selection of compounds and only for the cluster that contains the maximum number of compounds of the selection. It works on the output of PathwaysIterSelection.

Usage

```
Geneset.intersectSelection(list.output, sign, names = NULL, seperatetables = FALSE,
                           separatepvals = FALSE)
```

Arguments

`list.output`	The output of the `PathwaysIterSelection` function.
`sign`	The significance level to be handled for cutting of the pathways.
`names`	Optional. Names of the methods.
`seperatetables`	If TRUE, a separate element is created per cluster. containing the pathways for each iteration.
`separatepvals`	If TRUE, the p-values of the each iteration of each pathway in the intersection is given. If FALSE, only the mean p-value is provided.

Value

The output is a list with an element per method. For each method, it is portrayed for the cluster with the maximum number of compounds which pathways belong to the intersection over all iterations and their corresponding mean p-values.

`GS` *GeneSets for the Pathway Analysis of the MCF7 Data*

Description

GeneSets for the Pathway Analysis of the MCF7 Data

Usage

```
data(GS)
```

Format

The format is: List of 7804 $ GO:0000002: chr [1:17] "291" "1763" "1890" "3980" ...

$ GO:0000723: chr [1:65] "142" "472" "641" "1736" ... [list output truncated] - attr(*, "species")= chr "Human" - attr(*, "geneSetSource")= chr "GOBP" - attr(*, "descriptions")= Named chr [1:11972] "mitochondrial genome maintenance" "reproduction" "single strand break repair" "regulation of DNA recombination" ... ..- attr(*, "names")= chr [1:11972] "GO:0000002" "GO:0000003" "GO:0000012" "GO:0000018" ... - attr(*, "class")= chr [1:2] "geneSetMLP" "list"

`HeatmapCols` *Comparing 2 Clustering Results*

Description

The `HeatmapCols` function performs the `distanceheatmaps` function given the outputs of two clustering methods and plots the resulting heatmap. The function `heatmap.2` is called upon to make the actual plot of the heatmap. It is noted that for this function the number of colors should be one more than the number of clusters to color the so calles zero cells in the distance matrix.

Usage

```
HeatmapCols(Data1, Data2, names = rownames(fingerprintMat), nclusters = 7,
            cols = my_palette)
```

Arguments

`Data1`	The resulting list of method 1.
`Data2`	The resulting list of method 2.
`names`	The names of the objects in the data sets.
`nclusters`	The number of clusters to cut the dendrogram in.
`cols`	The colors to be used for the clusters.

Value

A heatmap based on the distance matrix created by `distanceheatmaps` with the dendrogram of method 2 on top of the plot and the one from method 1 on the left. The names of the compounds are depicted on the bottom in the order of clustering of method 2 and on the right by the ordering of method 1. Vertically the cluster of method 2 can be seen while horizontally those of method 1 are portrayed.

`LabelCols` *Internal function of Labelplot*

Description

Sometimes only one or two particular clusters are of interest if for example these contain an object of great importance. Then it would be nice not to color all clusters but just the compounds of these clusters and see where these are now. This can be done with the function `LabelCols`

Usage

```
LabelCols(x, Sel1, Sel2 = NULL, col1, col2 = NULL)
```

Arguments

`x`	The dendrogram to be colored.
`Sel1`	The selection of objects to be colored.
`Sel2`	An optional second selection to be colored.
`col1`	The color for the first selection.
`col2`	The color for the optional second selection.

Value

A dendrogram of which the leaves of the selection(s) are colored.

Labelplot *Coloring Specific Leaves of a Dendrogram*

Description

Just as the function ClusterCols, LabelCols as its own plotting function Labelplot which plots the dendrogram.

Usage

```
Labelplot(Data, Sel1, Sel2 = NULL, col1, col2 = NULL, ...)
```

Arguments

Data	The resulting list of a method which contains the dendrogram to be colored.
Sel1	The selection of objects to be colored.
Sel2	An optional second selection to be colored.
col1	The color for the first selection.
col2	The color for the optional second selection.
...	Other options which can be given to the plot function.

Value

A plot of the dendrogram of which the leaves of the selection(s) are colored.

MatrixFunction *Rearranging Clusters for Comparison.*

Description

When multiple methods are performed on a data set, it is interesting to compare their results. However, a comparison is not easily done since a different methods leads to a different ordering of the objects. The MatrixFunction rearranges the cluster to a reference method.

Usage

```
MatrixFunction(List, nclusters = 7, fusionsLog = FALSE, WeightClust = FALSE,
               names = NULL)
```

Arguments

List	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
nclusters	The number of clusters to cut the dendrogram in.
fusionsLog	Logical indicator for the fusion of clusters.
WeightClust	Optional. To be used for the outputs of CEC or WeightedClust. Then only the result of the Clust element is considered.
names	Optional. Names of the methods.

Details

It is interesting to compare the results of the methods described in the methodology. All methods result in a dendrogram which is cut into a specific number of clusters with the `cutree` function. This results in an numbering of cluster based on the ordering of the names in the data and not on the order in which they are grouped into clusters. However, different methods lead to different clusters and it is possible that cluster $1$ of one method will not be the cluster that has the most in common with cluster 1 of another method. This makes comparisons rather difficult. Therefore the `MatrixFunction` function was written which takes one method as a reference and rearranges the cluster numbers of the other methods to this reference such that clusters are appointed to that cluster they have the most in common with. The result of this function is a matrix of which the columns are in the order of the clustering of the compounds of the referenced method and the rows represent the methods. Each cell contains the number of the cluster the compound is in for that method compared to the method used as a reference. This function is applied in the functions `SimilarityMeasure`, `DiffGenes`, `Pathways` and `ComparePlot`. It is a possibility that 2 or more clusters are fused together compared to the reference method. If this is true, the function will alert the user and will ask to put the parameter fusionsLog to true. Since `MatrixFunction` is often used as an internal function, also for visualization, it will print out how many more colors should be specified for those clusters that did not find a suitable match. This can be due to fusion or complete segregation of its compounds into other clusters.

Value

A matrix of which the cells indicate to what cluster the compounds belong to according to the methods.\ The `MatrixFunction` function was optimized for the situations presented by the data sets at hand. It is noted that the function might fail in a particular situation which results in a infinite loop.

`my_palette1` *Colors for the heatmaps of MCF7*

Description

Colors for the heatmaps of MCF7

Usage

```
data(my_palette1)
```

Format

The format is: chr [1:8] "#8EE5EE" "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" ...

`my_palette1a` *Colors for the ComparePlot for MCF7*

Description

Colors for the ComparePlot for MCF7

Usage

```
data(my_palette1a)
```

Format

The format is: chr [1:8] "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" "#0000EE" ...

`my_palette2` *Colors for the ComparePlot for MCF7*

Description

Colors for the ComparePlot for MCF7

Usage

```
data(my_palette2)
```

Format

The format is: chr [1:7] "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" "#0000EE" ...

`my_palette3` *Colors for the heatmaps of Inhouse1*

Description

Colors for the heatmaps of Inhouse1

Usage

```
data(my_palette3)
```

Format

The format is: chr [1:11] "#8EE5EE" "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" ...

my_palette3a *Colors for the ComparePlot for Inhouse1*

Description

Colors for the ComparePlot for Inhouse1

Usage

```
data(my_palette3a)
```

Format

The format is: chr [1:13] "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" "#0000EE" ...

my_palette4 *Colors for the ComparePlot for Inhouse1*

Description

Colors for the ComparePlot for Inhouse1

Usage

```
data(my_palette4)
```

Format

The format is: chr [1:10] "#D2691E" "#EE2C2C" "#EEAD0E" "#006400" "#0000EE" ...

Pathways *Pathway Analysis*

Description

Pathway Analysis over the cluster per method.

Usage

```
Pathways(List, GeneExpr = geneMat, nclusters = 7, method = c("limma", "MLP"),
         ENTREZID = GeneInfo[, 1], geneSetSource = "GOBP", top = NULL,
         GENESET = GS, sign = 0.05, fusionsLog = TRUE, WeightClust = TRUE,
         names = NULL)
```

Arguments

`List`	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
`GeneExpr`	The gene expression matrix of the objects.
`nclusters`	he number of clusters to cut the dendrogram in.
`method`	The method to applied to look for DE genes. For now, only the limma method is available
`ENTREZID`	Vector containing the ENTREZID's of the genes. If not provided, the rownames of the matrix will be considered.
`geneSetSource`	The source for the getGeneSets function ('GOBP', 'GOMF','GOCC', 'KEGG' or 'REACTOME').
`top`	Overrules sign. The number of genes to display for each cluster. If not specified, only the significant genes are shown.
`GENESET`	Optional. Can provide own candidate gene sets.~
`sign`	The significance level to be handled.
`fusionsLog`	To be handed to `MatrixFunction`.
`WeightClust`	To be handed to `MatrixFunction`.
`names`	Optional. Names of the methods.

Details

After finding differently expressed genes, it can be investigated whether pathways are related to those genes. This can be done with the help of the function `Pathways` which makes use of the `MLP` function of the MLP package. Given the output of a method, the `cutree` function is performed which results into a specific number of clusters. For each cluster, the limma method is performed comparing this cluster to the other clusters. This to obtain the necessary p-values of the genes. These are used as the input for the `MLP` function to find interesting pathways. By default the candidate gene sets are determined by the `getGeneSets` function in the MLP package. The default source will be GOBP, but this can be altered. Altering the default species of "human" was not implemented. Further, it is also possible to provide own candidate gene sets in the form of a list of pathway categories in which each component contains a vector of Entrez Gene identifiers related to that particular pathway. The default values for the minimum and maximum number of genes in a gene set for it to be considered were used. For MLP this is respectively 5 and 100. If a list of outputs of several methods is provided as data input, the cluster numbers are rearranged according to a reference method. The first method is taken as the reference and `MatrixFunction` is applied to get the correct ordering. When the clusters haven been re-appointed, the pathway analysis as described above is performed for each cluster of each method.

Value

The output of this function is a list with an element for each method and per method, one for each cluster. Each element of a cluster has three parts of which the first part Compounds contains the compounds of the cluster. The second part Genes contains either the significant genes determined with the parameter sign or a specific number of genes specified with top. The last part Pathways are the significant pathways found for that cluster. There will be a ranked.genesets.table component in which one can find a ranked table of all the significant genesets with their corresponding p-values provided by the MLP output. This table also contains a column with the description of the pathway. This function is limited to showing the significant pathways but can be altered to show the complete output of the `MLP` function.

Pathways.2 *Pathways.2*

Description

Internal function of Pathways if only 1 method is specified

Usage

```
Pathways.2(Data, GeneExpr = geneMat, nclusters = 7, method = c("limma", "MLP"),
           ENTREZID = GeneInfo[, 1], geneSetSource = "GOBP", top = NULL,
           GENESET = GS, sign = 0.05)
```

Arguments

`Data`	Data
`GeneExpr`	The gene expression matrix of the objects.
`nclusters`	he number of clusters to cut the dendrogram in.
`method`	The method to applied to look for DE genes. For now, only the limma method is available
`ENTREZID`	Vector containing the ENTREZID's of the genes. If not provided, the rownames of the matrix will be considered.
`geneSetSource`	The source for the getGeneSets function ('GOBP', 'GOMF','GOCC', 'KEGG' or 'REACTOME').
`top`	Overrules sign. The number of genes to display for each cluster. If not specified, only the significant genes are shown.
`GENESET`	Optional. Can provide own candidate gene sets.~
`sign`	The significance level to be handled.

Value

The significant pathways found for each cluster of the method.

PathwaysIter *Iteration of Pathway Analysis*

Description

The MLP method to perform pathway analysis is based on resampling of the data. Therefore it is recommended to perform the pathway analysis multiple times to observe how much the results are influenced by a different resample. The function `PathwaysIter` performs the pathway analysis as described in `Pathways` a specified number of times. The input can be one data set or a list as in `Pathways`.

Usage

```
PathwaysIter(List, GeneExpr = geneMat, nclusters = 7, method = c("limma", "MLP"),
             ENTREZID = GeneInfo[, 1], geneSetSource = "GOBP", top = NULL,
             GENESET = GS, sign = 0.05, niter = 10, fusionsLog = TRUE,
             WeightClust = TRUE, names = NULL)
```

Arguments

`List`	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
`GeneExpr`	The gene expression matrix of the objects.
`nclusters`	he number of clusters to cut the dendrogram in.
`method`	The method to applied to look for DE genes. For now, only the limma method is available
`ENTREZID`	Vector containing the ENTREZID's of the genes. If not provided, the rownames of the matrix will be considered.
`geneSetSource`	The source for the getGeneSets function ('GOBP', 'GOMF','GOCC', 'KEGG' or 'REACTOME').
`top`	Overrules sign. The number of genes to display for each cluster. If not specified, only the significant genes are shown.
`GENESET`	Optional. Can provide own candidate gene sets.~
`sign`	The significance level to be handled.
`niter`	The number of times to perform pathway analysis.
`fusionsLog`	To be handed to `MatrixFunction`.
`WeightClust`	To be handed to `MatrixFunction`.
`names`	Optional. Names of the methods.

Value

The output is a list with an element for each iteration. Per iteration, there is a list for each cluster of each method containing 3 components. The first component Compounds are the names of the object belonging to that cluster. The second element Genes are the differentially expressed genes and the third element Pathways are the pathways to be found significant.

`PathwaysIterSelection` *Iterations of Pathway Analysis for a Selection.*

Description

The function `PathwaysIterSelection` performs the same procedure as `PathwaysIter` but only for a specific selection of compounds and only for the cluster that contains the maximum number of compounds of the selection.

Usage

```
PathwaysIterSelection(List, Selection, GeneExpr = geneMat, nclusters = 7,
                      method = c("limma", "MLP"), ENTREZID = GeneInfo[, 1],
                      geneSetSource = "GOBP", top = NULL, GENESET = GS,
                      sign = 0.05, niter = 10, fusionsLog = TRUE,
                      WeightClust = TRUE, names = NULL)
```

Arguments

`List`	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
`Selection`	The selection of objects to follow.
`nclusters`	he number of clusters to cut the dendrogram in.
`method`	The method to applied to look for DE genes. For now, only the limma method is available
`ENTREZID`	Vector containing the ENTREZID's of the genes. If not provided, the rownames of the matrix will be considered.
`geneSetSource`	The source for the getGeneSets function ('GOBP', 'GOMF','GOCC', 'KEGG' or 'REACTOME').
`top`	Overrules sign. The number of genes to display for each cluster. If not specified, only the significant genes are shown.
`GENESET`	Optional. Can provide own candidate gene sets.~
`sign`	The significance level to be handled.
`niter`	The number of times to perform pathway analysis.
`fusionsLog`	To be handed to `MatrixFunction`.
`WeightClust`	To be handed to `MatrixFunction`.
`names`	Optional. Names of the methods.

Value

The output is a list with an element for each iteration. Per iteration, there is a list for the cluster with the maximum number of compounds of each method containing 3 components. The first component Compounds are the names of the object belonging to that cluster. The second element Genes are the differentially expressed genes and the third element Pathways are the pathways to be found significant.

`PathwaysSelection` *Pathway Analysis for a Selection.*

Description

The function `PathwaysSelection` performs the same procedure as `Pathways` but only for a specific selection of compounds and only for the cluster that contains the maximum number of compounds of the selection.

Usage

```
PathwaysSelection(List, Selection, GeneExpr = geneMat, nclusters = 7,
                  method = c("limma", "MLP"), ENTREZID = GeneInfo[, 1],
                  geneSetSource = "GOBP", top = NULL, GENESET = GS,
                  sign = 0.05, fusionsLog = TRUE, WeightClust = TRUE,
                  names = NULL)
```

Arguments

`List` A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.

`Selection` The selection of objects to follow.

`GeneExpr` The gene expression matrix of the objects.

`nclusters` he number of clusters to cut the dendrogram in.

`method` The method to applied to look for DE genes. For now, only the limma method is available

`ENTREZID` Vector containing the ENTREZID's of the genes. If not provided, the rownames of the matrix will be considered.

`geneSetSource` The source for the getGeneSets function ('GOBP', 'GOMF','GOCC', 'KEGG' or 'REACTOME').

`top` Overrules sign. The number of genes to display for each cluster. If not specified, only the significant genes are shown.

`GENESET` Optional. Can provide own candidate gene sets.~

`sign` The significance level to be handled.

`fusionsLog` To be handed to `MatrixFunction`.

`WeightClust` To be handed to `MatrixFunction`.

`names` Optional. Names of the methods.

Value

The output of this function is a list with an element for each method and per method, one for the cluster with the maximum number of compounds. Each element of the cluster has three parts of which the first part Compounds contains the compounds of the cluster. The second part Genes contains either the significant genes determined with the parameter sign or a specific number of genes specified with top. The last part Pathways are the significant pathways found for that cluster. There will be a ranked.genesets.table component in which one can find a ranked table of all the significant genesets with their corresponding p-values provided by the MLP output. This table also contains a column with the description of the pathway. This function is limited to showing the significant pathways but can be altered to show the complete output of the `MLP` function.

ProfilePlot *Plotting a Gene Profile*

Description

The function ProfilePlot shows the gene expression of a specific gene over the objects of the data set and has the ability to separate the values a set of objects from the others. It can be used for example to investigate the gene expression of a gene that was found to be shared between the clusters over the methods. The values of the shared objects of these clusters are then plotted in red at the left side of the plot while the other values gained a blue color and are put to the right. Further, it is optional to give the objects that belong to these clusters but are not common to the cluster over the methods, a specific color for each method. This way, it is known how much the gene expression of these objects resembles those of the shared components.

Usage

```
ProfilePlot(Gene, Comps = Comps, GeneExpr = geneMat, Clusters = NULL, cols = NULL,
          AddLegend = TRUE, names = NULL, margins = c(8.1, 4.1, 1.1, 6.5),
          extra, ...)
```

Arguments

Gene	The gene to be plotted.
Comps	The objects to be plotted or to be separated from the other objects.
GeneExpr	The gene expression matrix of the objects.
Clusters	Optional. A list of clusters to which Obs belongs for each method considered. The observations that are not part of Comps will be given a different color per method.
cols	Optional. The color to use for the objects in Clusters for each method.
AddLegend	Optional. Whether a legend of the colors should be added to the plot.
names	Optional but necessary when Addlegend=TRUE. Indicates the labels in the legend.
margins	Optional. Margins to be used for the plot.
extra	The space between the plot and the legend.
...	Optional parameter to be handed to the plot function.

Value

A plot of the gene profile. A distinction is made by the use of different colors if not all objects are listed in the parameter Comps.

ProfilePlotAll *Plotting Multiple Gene Profiles*

Description

In `ProfilePlotAll`, the gene profiles of the significant genes for a specific cluster are shown on \$1\$ plot. Therefore, each gene is standardized by subtracting the mean for each gene.

Usage

```
ProfilePlotAll(Genes = Genes, Comps = Comps, GeneExpr = geneMat, Order = NULL,
               Clusters = NULL, cols = NULL, AddLegend = TRUE, margins =
               c(8.1, 4.1, 1.1, 6.5), extra = 5, ...)
```

Arguments

`Genes`	The genes to be plotted.
`Comps`	The objects to be plotted or to be separated from the other objects.
`GeneExpr`	The gene expression matrix of the objects.
`Order`	Optional. If the compounds are to set in a specific order of a specific method.
`Clusters`	Optional. A list of clusters to which Obs belongs for each method considered. The observations that are not part of Comps will be given a different color per method.
`cols`	Optional. The color to use for the objects in Clusters for each method.
`AddLegend`	Optional. Whether a legend of the colors should be added to the plot.
`margins`	Optional. Margins to be used for the plot.
`extra`	The space between the plot and the legend.
`...`	Optional parameter to be handed to the plot function.

Value

A plot which contains multiple gene profiles. A distinction is made between the values for the objects in Comps and the others.

Shared *Shared genes and pathways over the methods*

Description

It is interesting to investigate exactly which and how many differently expressed genes and pathways are shared by the clusters over the different methods. The function `Shared` will provide this information. Given the outputs of the `DiffGenes` function and/or the `PathwaysIter` function, it investigates how many genes and/or pathways are expressed by each cluster per method, how many of these are shared over the methods and which ones are shared including their respective p-values of each method and a mean p-value. This is very handy to look into the shared genes and pathways of clusters that share many objects but also of those that only share only a few. Further, the result also includes the number of compounds per cluster per method and how many of these are shared over the methods.

Usage

```
Shared(DataLimma = NULL, DataMLP = NULL, names = NULL)
```

Arguments

`DataLimma`	The output of `DiffGenes` function
`DataMLP`	The output of `Geneset.intersect` function.
`names`	Optional. Names of the methods.

Value

The result of the `Shared` function is a list with 2 elements. The first element Table is a table indicating how many genes and/or pathways were found to be differentially expressed and how many of these are shared. The table also contains the number of compounds shared between the clusters of the different methods. The second element Which is another list with a component per cluster. Each component consists of 3 vectors: sharedcomps indicating which objects were shared across the methods, sharedgenes represents the shared genes and the last one sharedpaths shows the shared pathways. The elements pvalsgenes and pvalspaths contain the mean p-values of the shared genes and pathways.

`SharedComps` *Finding Shared Commpounds over the Methods*

Description

The `SharedComps` function is an easy function to select the compounds over a number of methods. To be used on cluster elements of the `DiffGenes` elements.

Usage

```
SharedComps(List)
```

Arguments

`List`	A list of the outputs of a cluster for different methods.

Value

The shared compounds of all listed elements.

SharedLimma *Internal function of the* `Shared` *function.*

Description

Finds the shared genes of outputs of the `DiffGenes` function.

Usage

```
SharedLimma(DataLimma, names = NULL)
```

Arguments

`DataLimma`	The output of `DiffGenes` function
`names`	Optional. Names of the methods.

Value

The genes and shared genes over the methods per cluster.

SharedMLP *Internal function of the* `Shared` *function.*

Description

Finds the shared pathways of outputs of the `Geneset.intersect` function.

Usage

```
SharedMLP(DataMLP)
```

Arguments

`DataMLP`	The output of `Geneset.intersect` function.

Value

The pathyways and shared pathways over the methods per cluster.

SharedSelection *Shared genes and pathways over the methods for a Selection*

Description

The function SharedSelection performs the same procedure as Shared but only for a specific selection of compounds and only for the cluster that contains the maximum number of compounds of the selection. It works on the output of DiffGenesSelection and Geneset.intersectSelection.

Usage

```
SharedSelection(DataLimma = NULL, DataMLP = NULL, names = NULL)
```

Arguments

DataLimma	The output of DiffGenesSelection function.
DataMLP	The output of Geneset.intersectSelection function.
names	Optional. Names of the methods

Value

The result of the Shared function is a list with 2 elements. The first element Table is a table indicating how many genes and/or pathways were found to be differentially expressed and how many of these are shared. The table also contains the number of compounds shared between the cluster of the different methods. The second element Which is another list with one component. It consists of \$3\$ vectors: sharedcomps indicating which objects were shared across the methods, sharedgenes represents the shared genes and the last one sharedpaths shows the shared pathways. The elements pvalsgenes and pvalspaths contain the mean p-values of the shared genes and pathways.

SharedSelectionLimma *Internal function for the SharedSelection function*

Description

Finds the shared genes of outputs of the DiffGenesSelection function.

Usage

```
SharedSelectionLimma(DataLimma = NULL, names = NULL)
```

Arguments

DataLimma	The output of DiffGenesSelection function.
names	Optional. Names of the methods

Value

The genes and shared genes over the methods per cluster.

SimilarityMeasure *Similarity Measures*

Description

The function SimilarityMeasure computes the similarity of the methods. Given a list of outputs as input, the first element will be seen as the reference. Function MatrixFunction is called upon and the cluster numbers are rearranged according to the reference. Per method, SimilarityMeasure investigates which objects have the same cluster number in reference and said method. This number is divided by the total number of objects and used as a similarity measure.

Usage

```
SimilarityMeasure(List, nclusters = 7, fusionsLog = TRUE, WeightClust = TRUE,
                  names = NULL)
```

Arguments

List	A list of the outputs from the methods to be compared. The first element of the list will be used as the reference.
nclusters	The number of clusters to cut the dendrogram in.
fusionsLog	To be handed to MatrixFunction.
WeightClust	To be handed to MatrixFunction.
names	Optional. Names of the methods.

Value

A vector of similarity measures, one for each method given as input.

SNFa *Similarity Network Fusion - version a*

Description

Function SNFa calls upon the functions affinityMatrix and SNF of the SNFtool package.

Usage

```
SNFa(List, distmeasure = c("tanimoto", "tanimoto"), NN = 20, alpha = 0.5, T = 20,
     clust = "agnes", linkage = "ward")
```

Arguments

`List` A list of data matrices of the same type.It is assumed the rows are corresponding with the objects.

`distmeasure` A vector of the distance measures to be used on each data matrix.

`NN` The number of neighbours to be used in the procedure. The number is advised to be between 10 and 50.

`alpha` The parameter epsilon. The value is recommended to be between 0.3 and 0.8.

`T` The number of iterations.

`clust` Choice of clustering function. Defaults to `agnes`.

`linkage` Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output of `SNFa` is a list with 2 elements. The first element SNF_FusedM contains the fused similarity matrix and the second element Clust represents the results of performing hierarchical clustering on the fused matrix.

`SNFb` *Similarity Network Fusion - version b*

Description

Function `SNFb`, performs SNF but first determines the subsets of neighbours and then normalization is performed on the neighbours only.

Usage

```
SNFb(List, distmeasure = c("tanimoto", "tanimoto"), NN = 20, alpha = 0.5, T = 20,
     clust = "agnes", linkage = "ward")
```

Arguments

`List` A list of data matrices of the same type.It is assumed the rows are corresponding with the objects.

`distmeasure` A vector of the distance measures to be used on each data matrix.

`NN` The number of neighbours to be used in the procedure. The number is advised to be between 10 and 50.

`alpha` The parameter epsilon. The value is recommended to be between 0.3 and 0.8.

`T` The number of iterations.

`clust` Choice of clustering function. Defaults to `agnes`.

`linkage` Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output of `SNFb` is a list with 2 elements. The first element SNF_FusedM contains the fused similarity matrix and the second element Clust represents the results of performing hierarchical clustering on the fused matrix.

SNFc *Similarity Network Fusion - version c*

Description

Function `SNFc`, performs SNF but first a normalization over all objects is performed before taking the k neighbours of each object as a subset in obtaining the kernel matrix.

Usage

```
SNFc(List, distmeasure = c("tanimoto", "tanimoto"), NN = 20, alpha = 0.5, T = 20,
     clust = "agnes", linkage = "ward")
```

Arguments

`List`	A list of data matrices of the same type.It is assumed the rows are corresponding with the objects.
`distmeasure`	A vector of the distance measures to be used on each data matrix.
`NN`	The number of neighbours to be used in the procedure. The number is advised to be between 10 and 50.
`alpha`	The parameter epsilon. The value is recommended to be between 0.3 and 0.8.
`T`	The number of iterations.
`clust`	Choice of clustering function. Defaults to `agnes`.
`linkage`	Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output of `SNFc` is a list with 2 elements. The first element SNF_FusedM contains the fused similarity matrix and the second element Clust represents the results of performing hierarchical clustering on the fused matrix.

targetMat *The target predictions for the MCF7 data*

Description

The 477 target predictions for the MCF7 data

Usage

```
data(targetMat)
```

Format

The format is: num [1:56, 1:477] 0 0 0 0 0 0 0 0 0 0 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:56] "metformin" "phenformin" "phenyl biguanide" "estradiol" ... ..$: chr [1:477] "Arachidonate_15.lipoxygenase" "Estradiol_17.beta.dehydrogenase_2" "Estradiol_17.beta.dehydrogenase_1" "Lanosterol_synthase" ...

Ultimate *Wrapper function for the Methods*

Description

The function Ultimate has the ability to perform multiple of the methods listed above simultaneously. The only necessary input are the data matrices and specification of the options. First, clustering is based on each data matrix separately after which the specified integrative analysis methods are conducted. A plot comparing the results is made automatically with ComparePlot. If weights are involved in the method, a comparison plot of the results for these weights is made as well.

Usage

```
Ultimate(List, distmeasure, NN = 20, alpha = 0.5, T = 20, t = 10, r = NULL, nclusters = 7,
       nclusterssep = c(7, 7), nclustersseq = NULL, weight = NULL, Clustweight = 0.5,
       clust = "agnes", linkage = "ward", gap = FALSE, maxK = 50, IntClust = c("ADC",
       "ADECa", "ADECb", "ADECc", "WonM", "CECa", "CECb", "CECc", "WeightedClust",
         "SNFa", "SNFb", "SNFc"), fusionsLog = TRUE, WeightClust = TRUE,
         PlotCompare = FALSE, cols = my_palette2, ...)
```

Arguments

`List`	A list of data matrices.
`distmeasure`	A vector of the distance measures to be used on each data matrix.
`NN`	The number of neighbours to be used in SNF.
`alpha`	The parameter epsilon in SNF.
`T`	The number of iterations in SNF.
`t`	The number of iterations in ADEC and CEC.
`r`	Optional. The number of features to take for the random sample in ADEC and CEC.
`nclusters`	The number of clusters to cut the dendrogram in for ADEC and the plot.
`nclusterssep`	Optional.Vector of the number of clusters to cut the dendrogram in of each data source. If NULL, the value of nclusters is used for each.
`nclustersseq`	The sequence of number of clusters to cut the dendrogram in for ADECb, CECb and WonM.
`weight`	The weights to be used in CEC and WeightedClust.
`Clustweight`	Optional. To be used for the outputs of CEC or WeightedClust. Then only the result of the Clust element is considered.
`clust`	Choice of clustering function. Defaults to `agnes`
`linkage`	Choice of inter group dissimilarity. Defaults to Ward link.
`gap`	Logical indicator if gap statistics should be computed. Setting to FALSE will greatly reduce the computation time.
`maxK`	The maximum number of clusters to be considered during the gap.
`IntClust`	Specification of the methods to be applied.
`fusionsLog`	To be handed to `MatrixFunction`.

`WeightClust`	To be handed to `MatrixFunction`.
`PlotCompare`	Logical. Should the plot over the methods and weight be produced?
`cols`	Color scheme to be used in the plots.
`...`	Options to be given to `ComparePlot`.

Value

The output of `Ultimate` is a list . The first element contains the results of the clustering of the first data source and the last element on the second data source . In between are the results of the integrative methods.

`WeightedClust` *Weighted Clustering*

Description

Weighted clustering is performed with the function `WeightedClust`. Given a list of the data matrices, a dissimilarity matrix is computed of each with the provided distance measures. These matrices are then combined resulting in a weighted dissimilarity matrix. Hierarchical clustering is performed on this weighted combination with the `agnes` function and the ward link.\

Usage

```
WeightedClust(List, distmeasure = c("tanimoto", "tanimoto"), weight = seq(1, 0, -0.1),
              Clustweight = 0.5, clust = "agnes", linkage = "ward")
```

Arguments

`List`	A list of data matrices of the same type.It is assumed the rows are corresponding with the objects.
`distmeasure`	A vector of the distance measures to be used on each data matrix.
`weight`	The weight is a sequence from 0 to 1 and a result is produced for each by default. A specific weight can be provided by the user.
`Clustweight`	A weight for which the result will be put aside of the other results. This was done for comparative reason and easy access.
`clust`	Choice of clustering function. Defaults to `agnes`.
`linkage`	Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output of `WeightedClust` is a list with 4 elements. The element DistM contains the distance matrices for each data matrix while WeightedDist contains the weighted distance matrices computed for each provided weight. Results consists of the resulting hierarchical clustering of each weighted dissimilarity matrix. The final element Clust is the result for the weight specified in Clustweight.

WonM *Weighting on Membership*

Description

Weighting on membership is performed with the WonM function. The first step is to compute the appropriate distance matrices for each data source and to use these for hierarchical clustering. This is executed with the agnes function and the ward link. The user may specify a range of values for the number of clusters to cut the resulting dendrograms in. For each value of number of clusters, an incidence matrix is computed and these are added for each data source separately. Eventually, the sums of the incidence matrices are joined together as well, resulting in 1 consensus matrix. Hierarchical clustering is performed on the consensus matrix to obtain the final clustering result.

Usage

```
WonM(List, distmeasure = c("tanimoto", "tanimoto"), nclusters = seq(5, 25, 1),
     clust = "agnes", linkage = "ward")
```

Arguments

List	A list of data matrices of the same type.It is assumed the rows are corresponding with the objects.
distmeasure	A vector of the distance measures to be used on each data matrix.
nclusters	A sequence of the number of clusters to cut the dendrogram in.
clust	Choice of clustering function. Defaults to agnes.
linkage	Choice of inter group dissimilarity. Defaults to Ward link.

Value

The output of WonM is a list with 4 elements. The element DistM contains the distance matrices of each data source and ClustSep the results of the hierarchical clustering performed on each. The computed consensus matrix over all data sources can be found in Consensus and the final clustering result is contained in the Clust element.

Index

∗Topic **Aggregated Data Ensemble Clustering**
ADECa, 3
ADECb, 4
ADECc, 5
∗Topic **ClusterDistribution**
ClusterDistribution, 10
∗Topic **Clustering**
Cluster, 8
ComparePlot, 13
FindCluster, 17
IntClust-package, 2
Ultimate, 40
∗Topic **Colors**
Colorpalette, 11
ColorsNames, 12
∗Topic **Comparison over Clustering results**
MatrixFunction, 23
∗Topic **Complementary Ensemble Clustering**
CECa, 6
CECb, 7
CECc, 7
∗Topic **Dendrogram**
ClusterCols, 9
Clusterplot, 11
LabelCols, 22
Labelplot, 23
∗Topic **Differential Gene Expression**
DiffGenes, 14
DiffGenes.2, 15
DiffGenesSelection, 15
Shared, 33
SharedComps, 34
SharedLimma, 35
SharedSelection, 36
SharedSelectionLimma, 36
∗Topic **Gene Profile**
ProfilePlot, 32
ProfilePlotAll, 33
∗Topic **Heatmap**
distanceheatmaps, 16
HeatmapCols, 21
∗Topic **Integrative Clustering**
ADClust, 3
∗Topic **Pathway Analysis**
Geneset.intersect, 20
Geneset.intersectSelection, 20
Pathways, 26
Pathways.2, 28
PathwaysIter, 28
PathwaysSelection, 30
Shared, 33
SharedMLP, 35
SharedSelection, 36
∗Topic **Pathyway Analysis**
PathwaysIterSelection, 29
∗Topic **Similarity Measure**
SimilarityMeasure, 37
∗Topic **Similarity Network Fusion**
SNFa, 37
SNFb, 38
SNFc, 39
∗Topic **Weighted Clustering**
WeightedClust, 41
∗Topic **Weighting on Membership**
WonM, 42
∗Topic **\textasciitildeFinding Genes**
FindGenes, 17
∗Topic **\textasciitildekwd2**
DiffGenesSelection, 15
MatrixFunction, 23
∗Topic **datasets**
bioassayMat, 5
fingerprintMat1, 18
fingerprintMat2, 18
GeneInfo, 19
geneMat1, 19
geneMat2, 19
GS, 21
my_palette1, 24
my_palette1a, 25
my_palette2, 25
my_palette3, 25
my_palette3a, 26

my_palette4, 26
targetMat, 39

ADClust, 3
ADECa, 3
ADECb, 4
ADECc, 5

bioassayMat, 5

CECa, 6
CECb, 7
CECc, 7
Cluster, 8
ClusterCols, 9
ClusterDistribution, 10
Clusterplot, 11
Colorpalette, 11
ColorsNames, 12
ComparePlot, 13

DiffGenes, 14
DiffGenes.2, 15
DiffGenesSelection, 15
distanceheatmaps, 16

FindCluster, 17
FindGenes, 17
fingerprintMat1, 18
fingerprintMat2, 18

GeneInfo, 19
geneMat1, 19
geneMat2, 19
Geneset.intersect, 20
Geneset.intersectSelection, 20
GS, 21

HeatmapCols, 21

IntClust (IntClust-package), 2
IntClust-package, 2

LabelCols, 22
Labelplot, 23

MatrixFunction, 23
my_palette1, 24
my_palette1a, 25
my_palette2, 25
my_palette3, 25
my_palette3a, 26
my_palette4, 26

Pathways, 26
Pathways.2, 28
PathwaysIter, 28
PathwaysIterSelection, 29
PathwaysSelection, 30
ProfilePlot, 32
ProfilePlotAll, 33

Shared, 33
SharedComps, 34
SharedLimma, 35
SharedMLP, 35
SharedSelection, 36
SharedSelectionLimma, 36
SimilarityMeasure, 37
SNFa, 37
SNFb, 38
SNFc, 39

targetMat, 39

Ultimate, 40

WeightedClust, 41
WonM, 42

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Integrated data anlaysis via clustering

Richting: **Master of Statistics-Bioinformatics**
Jaar: **2014**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Van Moerbeke, Marijke

Datum: **10/09/2014**