Searching for robust and clinically relevant subtypes in high density molecular data

Eva Budinská[1,2,3]

[1] *Institute of Biostatistics and Analyses, Masaryk University; e-mail: budinska@iba.muni.cz*
[2] *Swiss Institute of Bioinformatics, Lausanne, Switzerland*
[3] *Masaryk Memorial Cancer Institute, Brno*

Abstract

Identification of clinically relevant molecular subtypes became an important tool in elucidating tumor biology. Robustness of the analytical approach employed for this task and consequently the robustness of derived subtypes is of major concern, if further experiments are to be conducted to confirm derived hypotheses. Here, we discuss multiple novel techniques for the control of robustness in cluster analysis designed for analysis of high-density molecular data.

Key words

Gene expression, molecular subtyping, consensus clustering, dynamic tree cut

1. Introduction

Subtyping based on high density molecular data, such as microarrays, aims at identifying groups of samples with similar molecular patterns. These can be for instance similar patterns of gene expression, microRNA expression or methylation. Finding molecular subtypes is very relevant mainly in medicine, especially in diseases that appear homogenous histo-pathologically, yet give a very heterogeneous response in terms of treatment outcome or survival. Recently, a lot of effort is dedicated to molecular subtyping of different cancers. Breast cancer for instance, was the first one where gene expression subtyping was applied, revealing a set of groups, that serve until now a basis for treatment consideration (Perou et al., 2002). Molecular subtypes help to elucidate the underlying biological mechanisms responsible for heterogeneity in tumour behaviour and help to focus the research on the subtype specific drugs targets, with hope to optimize treatment and ensure better prognosis of the given cancer as a whole. In order to make molecular subtypes clinically relevant, many additional analyses elucidating the biological, clinical and prognostic inference of subtypes are needed. The analysis becomes a fairly complex process involving different data-mining and statistical tools, together with thorough bio-medical interpretation of results.

Hereby, we will focus on the most important part of the subtyping procedure – robust clustering.

2. Example dataset

Throughout this article, we will use two datasets:

golub dataset - a microarray derived gene expression dataset available in R package `multtest` under the name `golub`, comprising 38 samples of three groups of acute leukaemia (AML – acute myeloid leukaemia, ALL-B – acute lymphoid leukaemia B cell

type and ALL-T – acute lymphoid leukaemia T-cell type) and gene expression values of 3051 genes. This was the first dataset used to demonstrate the use of gene expression data in cancer studies (Golub et al., 1999).

random dataset - a matrix of 1000 features and 100 samples, randomly sampled from normal distribution with 0 mean value and standard deviation equal to 1. This dataset will serve an example of a dataset without particular inner structure.

3. Robust clustering

Clustering – or, so called usupervised learning - is the analytical approach used for subtype derivation. The main objective of clustering is to find distinct, preferably non-overlapping subpopulations within the large population of interest, members of which share similar pattern. Different basic clustering techniques exist and can be divided into model-based and distance-based methods. The model based methods use parametric assumptions on data distribution and often provide probabilities of cluster assignment. The distance-based methods are based on a similarity measure and can be further split into hierarchical and non-hierarchical, according to the algorithm they apply in order to group the samples. The detailed description of these methods and discussion on the choice of metrics is beyond the scope of this article and can be found elsewhere (Budinska et al., 2009).

In large genomic studies, hierarchical clustering is a particularly preferred method, because of its pattern visualization advantage. Often, not only clusters of samples, but also clusters of features – molecular entities - that underpin biological differences are of importance. Heatmap – a colored two dimensional plot with rows and columns representing samples and genes, ordered according to the hierarchical clustering dendrogram is one of the most often published type of figure in the field of large-scale molecular data (with the exception of DNA sequencing).

It is well known that the choice of clustering algorithm and metrics affects the final results, because clustering algorithms are biased towards partitions in accordance with their own clustering criterion. Moreover, clustering algorithms are designed to provide a data partition, even in non-existence of such a pattern, and the significance of these results must be assessed ad-hoc. While the clustering algorithms and corresponding metrics can be selected a-priori, based on the data type, our experience or published recommendations, two main issues are still to be addressed: i) the determination of the number of clusters and ii) the assessment of the confidence of the selection of number of clusters and cluster assignment for individual samples. Missing the external measure of class assignment (ground truth), the evaluation of clusters is based solely on internal validation measures, estimating the quality based on the intrinsic data values.

These issues are of particular importance in the data analysis of high density molecular data, which suffer from the curse of dimensionality problem. The small number of samples (tens to hundreds) and relatively huge number of molecular features (thousands, tens of thousands) makes clustering techniques susceptible to over-fitting, due to the sensitivity to noise, which is in these data much more abundant. This highly affects the robustness of the clustering to the sampling variability.

Resampling of the original dataset is away to simulate sampling variability. Although the idea of resampling in clustering is not new (Jain and Moreau, 1988), in the case of more noisy high-density molecular data, the preference is to avoid sampling with replacement, because replicated values can be artificially considered a separate cluster (Monti et al.,

2003). Multiple methods have been recently suggested to address these problems in the concept of microarray data analysis, mainly based on repeated resampling and consequent re-clustering of the original dataset, in order to study the behavior of the results when data is disturbed. This approach is simulating possible differences between different datasets, presumably resulting in a more robust result (for a review, see e.g. Handl et al., 2005).

For example a prediction-based resampling method *Clest* was designed (Dudoit and Fridlyand, 2002) in order to robustly estimate the number of clusters, showing the superiority of its performance in microarray data over six other methods, including more conventional such as *Silhouette* (Kaufman, Rousseeuw, 1990), or more recent such as *gap* (Tibshirani et al., 2001). However, this method does not solve the problem of the assessment of the confidence of cluster assignment for individual samples. A new method assessing both problems – *consensus clustering* (Monti et al., 2003) - was suggested and was successfully applied in different cancer subtyping analyses. In a comparative study (Giancarlo et al., 2008) this method was also evaluated the best method in terms of performance and algorithm independency. We will dedicate the following subsection to the description of this method.

3.1. Consensus clustering

Is a resampling and re-clustering based method designed to represent the *consensus* across multiple runs of a clustering algorithm (Monti et al., 2003), in order to:

- determine the number of clusters in the data and to assess the stability of the discovered clusters
- represent the consensus over multiple runs of a clustering algorithm with random restart, so as to account for its sensitivity to the initial conditions.

In addition, it serves a visualization tool for the evaluation cluster number, membership, and boundaries.

The basic principle is to disturb the structure of the original $N \times P$ data matrix by random selection of a subset of samples and/or features. The new dataset is then consequently clustered, given the selected clustering algorithm, similarity measure and number of clusters or tree cut height. This resampling and clustering is repeated L times. In the l-th run, the cluster membership of samples is recorded and two $N \times N$ matrices are created:

- *connectivity matrix* $C^{(l)}$ that stores for each pair of samples *i,j* the information whether they were clustered together, e.g. $C_{ij}^{(l)} = 1$ if sample *i* and *j* belong to the same cluster, 0 otherwise
- *indicator matrix* $I^{(l)}$ that stores for each pair *i,j* the information whether they were both selected in the resampling, e.g. $I_{ij}^{(l)} = 1$ if sample *i* and *j* were in the same selection, 0 otherwise

After all l runs, the *consensus matrix* M is calculated by dividing the number of times two features were found together in the same cluster by the number of times that they have been selected together in the sampling subsets. A consensus matrix is therefore a N × N matrix

that stores for each pair of items the weighted proportion of clustering runs in which the two items were clustered together:

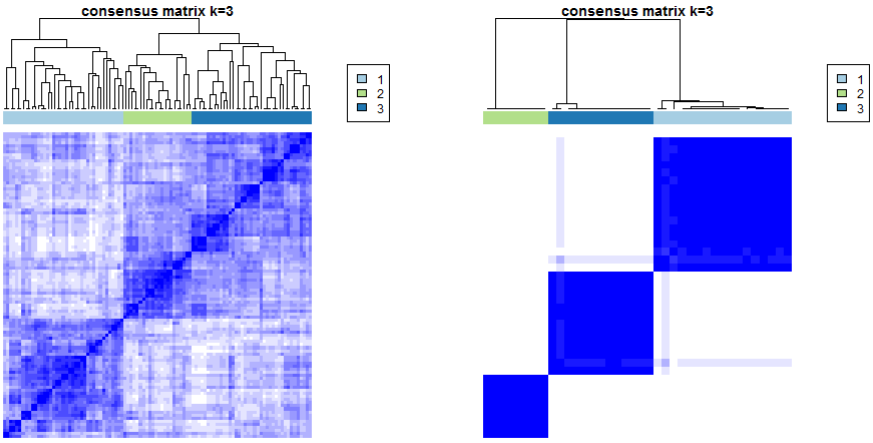
$$M_{ij} = \frac{\sum_{l=1}^{L} C_{ij}^{(l)}}{\sum_{l=1}^{L} I_{ij}^{(l)}}$$

The idea behind this approach is that samples that are frequently found in the same cluster represent more reliable cluster members than those who cluster together less frequently, being more sensible to the random noise and changes in feature selection. Each entry of the consensus matrix is a *consensus index* of a given pair of samples, with values from 0 (no consensus, samples were never members of the same cluster) to 1 (perfect consensus, clusters were members of the same cluster 100% times). Consensus matrix M represents a robust similarity measure and 1-M is a distance matrix, that can be used as an entry to hierarchical clustering in order to obtain a final robust clustering. Figure 1 demonstrates a result of hierarchical clustering applied on consensus matrix on our example data. While the consensus matrix of the *random* dataset is very unstructured, a very clear three-class structure is visible for the *golub* dataset.

The consensus matrix between samples can be directly used to define statistics of stability of clusters and cluster sample assignments. If I_k be a set of indices of samples belonging to cluster k, the consensus measure of a cluster k - *cluster consensus* - can be defined as an average consensus index between all pairs of samples belonging to the same cluster:

$$m^k = \frac{1}{N_l(N_l - 1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} M_{ij}$$

Figure 1. Heatmap representation of the consensus matrix for the *random* dataset – left and the *golub* dataset – right, for three clusters. The colour ranges from white representing 0 consensus to bright blue, representing the consensus of 1.



The corresponding *sample consensus* for each sample s_i and cluster l can be defined as:

$$m_i^k = \frac{1}{N_l - 1\{s_i \in I_k\}} \sum_{\substack{j \in I_l \\ j \neq i}} M_{ij} ,$$

where $1\{s_i \in I_k\}$ is the indicator function that equals 1 if $\{s_i \in I_k\}$ is true, 0 otherwise. The sample consensus is the average consensus index of the sample to all members of the cluster. Both measures can be used to identify outliers – either clusters with relatively low consensus, suggesting remaining heterogeneity in the cluster, or samples, that could be considered outliers because of very small consensus to any other sample in the dataset.

Consensus matrix can be also used to estimate the optimal number of clusters. For details, see section 2.2.

3.1.1. Other consensus clustering techniques

Multiple variations of the consensus clustering method exist and are a natural extension of the original algorithm.

Method called *merged consensus clustering* (Swift et al., 2004), in contrast to the method of Monti et al., creates the consensus matrix as a function of runs of consensus clustering with multiple different algorithms. This should eliminate the possible negative effect a single algorithm, which might not be suitable for the particular type of data.

Weighted clustering (Deohdar and Ghosh, 2006) builds on the idea that the clusterings produced within a consensus clustering procedure are not necessarily of the same quality. If an external metrics of quality exists, one should be able to integrate this in order to weigh the contributions of each clustering to the final consensus matrix, which is then calculated as

$$M_{ij} = \sum_{k=1}^{K} w_k C_{ij}^{(k)} ,$$

where w_k is weight of the particular clustering. This method also uses different clustering algorithms and different distance measures.

For the comparison of different consensus clustering algorithms, see for example (Goder and Filkov, 2008).

3.1.2. R-packages for consensus clustering

Two major packages are available in R for consensus clustering. The package `ConsensusClusterPlus` (Wilkerson, 2011) provides all the algorithms and metrics as described in (Monti et al., 2003). It is a part of Bioconductor repository and can be installed directly from R console using command:

```
>source("http://bioconductor.org/biocLite.R")
>biocLite("ConsensusClusterPlus")
```

Package `clusterCons` implements the merged clustering of (Swift et al., 2004). It can be installed directly from R console by using the `install.packages()`command.

3.2. Determining the number of subtypes

In this section, two methods for determining the number of clusters are discussed. Both were developed specially for microarray data analysis and hierarchical clustering algorithm.

3.2.1. Consensus measure

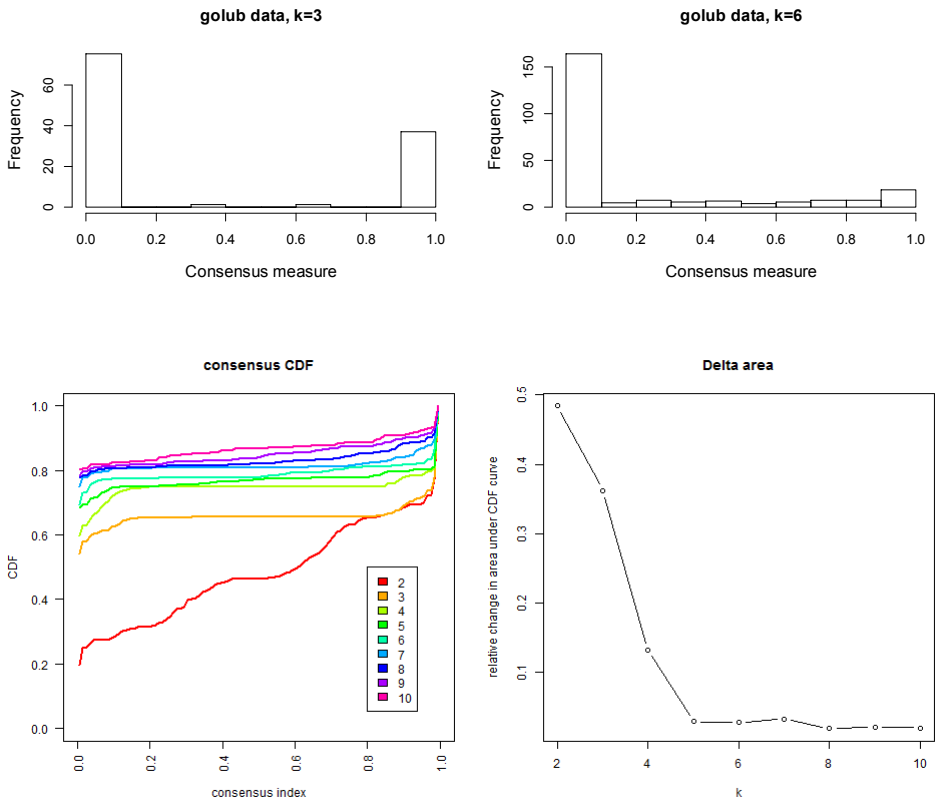
Consensus matrix – as described in section 2 - can be also used to estimate the optimal number of clusters. If consensus clustering is run for different cluster number values $k=1..K$, the decision criteria can be based on the calculation of for example the average intra-cluster consensus for each k. (Monti et al., 2003) propose another measure - the *empirical cumulative distribution* (CDF)

$$CDF^x = \frac{\sum_{i<j} 1\{M_{ij} \le x\}}{N(N-1)/2},$$

which compares the distribution of histograms of entries of consensus matrix M for each k. If clustering with k clusters represents a perfect partition, histogram of consensus matrix entries will consist of two bins over 0 (no consensus at all between samples from different clusters) and 1 (perfect consensus between clusters from different samples). The optimal number of clusters can then be decided by computing the area under CDF curve and by examining its relative change between different k (*delta area*). The CDF measure, however, is applicable mainly for hierarchical clustering, for which the method was designed. Figure 2 shows examples of histograms for k=3 and k=6 and CDF and delta area for the golub data. While histogram of consensus measures for the three cluster structure (heatmap on Figure 1 right) reveals indeed majority of values on 0 or 1, six cluster structure has a substantially decreased number of perfect consensus between samples and increased number of values between 0-1 suggesting instability of this number of clusters. The delta area plot shows that increasing number of clusters from 2 to 3, the area under CDF gains around 0.36, while

further increasing the number of clusters to 5 has no real impact on the area under CDF change and therefore the estimated value of k would be 3 or 4 subtypes.

Figure 2. Example of CDF derivation and selection of number of clusters on golub dataset. Consensus CDF and delta area plot are shown for k varying from 2 to 10.



3.2.2. Dynamic Tree Cut

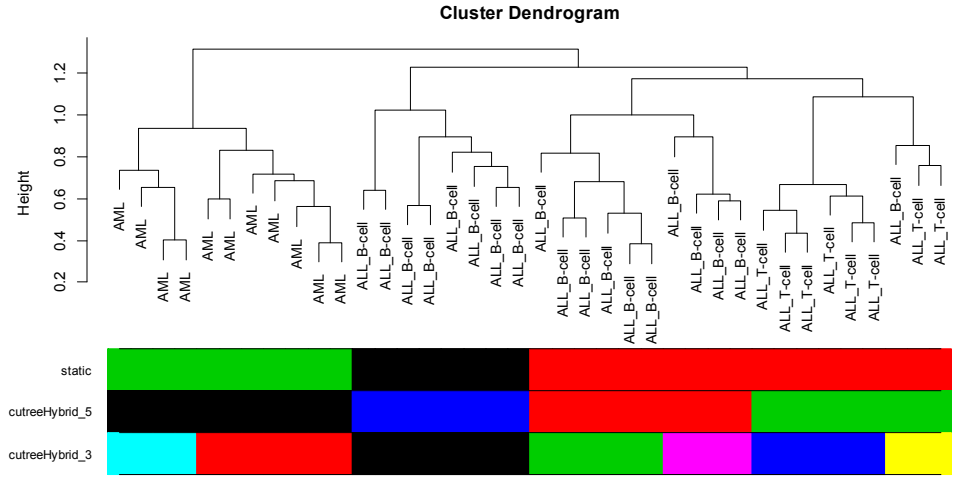
As already mentioned, hierarchical clustering has its particular importance in genomic data analysis. In comparison to other clustering techniques, clusters are defined ad-hoc, by cutting the branches of the hierarchically structured similarity tree – dendrogram – the output of this clustering – on a fixed height. All the branches below this cut are preserved and represent final clusters. The major disadvantage of this static cut approach is that often, different clusters are present on different cut heights – naturally presenting more or less similar groups of samples, and cutting low in order to obtain a cluster with high internal similarity results in the loss of structure of clusters with relatively lower similarity.

To address this problem, a set of novel dynamic branch cutting methods for detecting clusters in a dendrogram of hierarchical clustering was recently proposed (Langfelder et al, 2007). In this approach, clusters are being defined depending on their shape. The huge advantage is that the system of cluster determination is flexible – a set of parameters can be

used to control the resulting cut – such as for instance cut height, minimal cluster size or minimal intra cluster. First method called *Dynamic Tree* - this is a flexible extension of the static cut, works solely with the structure of the dendrogram. The second method *DynamicHybrid* dynamically crawls the dendrogram in the bottom-up direction and after defining clusters offers a possibility of additional assignment of the unassigned samples to the closest core clusters defined in the first step, if the requirements on cluster internal similarity are met. The description of both algorithms is fairly complex and I do strongly recommend the reader to consult the original paper for more details. Dynamic Tree Cut methods are implemented in R package `dynamicTreeCut`.

An example of comparison of a static and dynamic cut of dendrogram is demonstrated on the golub dataset in Figure 3. Both static cut and DynamicHybrid algorithm (represented by function `cutreeHybrid`) were run with cut height of 1.2. Minimal cluster size selected for cutreeHybrid was 3 and 5. While the static cut on the selected height identifies 3 clusters, cutHybrid with minimal cluster size of 5 identifies four major clusters. Decreasing the cluster size to 3 identifies further, yet still consistent splits.

Figure 3. Comparison of static and DynamicHybrid cut (as output by `cutreeHybrid` function of the R package `dynamicTreeCut`)on the dendrogram from hierarchical clustering with average linking algorithm and correlation-based distance between samples of golub data.



4. Other analytical challenges

Robustness of findings is one of the most important aspects of the applied research, and is indispensable for the clinical relevance. In order to call the subtypes robust, it is vital that the patterns defining the groups we find are not specific for a particular dataset, but can be found in other similar populations. We say that subtypes must be validated. However, the validation in a de-novo developed subtyping system has somewhat different meaning and is a much less evident analytical task than in the construction of classifiers. This is because it is not obvious to validate a pattern without existing objective class label (the ground truth). Without the ground truth, the validation can be done only indirectly, by the assessment of subtype specific differences in population characteristics that were not used for their

construction. Different survival experience or clinically relevant variables are examples of such characteristics.

Often, a development of a subtype classifier is necessary in order to make the results applicable for the practice. Preferably, such a classifier will be accurate and robust to different technological platforms used to derive data and will be able to classify one sample only. This classifier can also serve to call subtypes in the validation set. However, in a complex analysis of subtyping, many decisions on types of methods and choice of parameters must be made. Although some can be subjected to sensitivity analysis exploring the effect of different choices on clustering results (such as similarity metrics or clustering algorithms), it is almost impossible to perform such an analysis for all considered parameters and algorithmic choices, due to the complexity of the problem. For this reason, simple application of the classifier on the validation set and consequent comparison of external characteristics of training vs. validation subtypes is not the optimal solution. Better solution would be to reproduce the subtyping on the validation set, using the same methods and parameters as selected for the training set.

5. Concluding remarks

We have seen a selection of state-of-the-art approaches for robust clustering in the molecular subtyping. However, the field is evolving very quickly and reader is strongly encouraged to search for the methodological improvements and critically review all the information provided with respect to the nature of the particular data analysed.

Some concepts remain, though, the same. The robustness and reproducibility in clinical research is indispensable. One should never search for the final and unchangeable answer – which is almost impossible to achieve because of the nature of biology and technological limitations - but rather focus on the extraction of the most essential information from the data that are available. In this respect, application of consensus clustering base methods seems inevitable, although in case of hierarchical clustering, one might consider to use rather dynamic Tree Cut for cluster assignment, as it allows for identification of core samples, without forcing the less representative samples to be assigned a cluster membership.

References

Budinska E, Bortlicek Z. 2009. E-learning E-learning in analysis of genomic and proteomic data. URL: http://telemedicina.med.muni.cz/genomic-proteomic-analysis/index-en.php

Deodhar M, Ghosh J. 2006. Weighted Consensus Clustering for Microarray Data Analysis. In: Dagli, CG et al. (ed): Intelligent Engineering Systems through Artificial Neural Networks, Volume 16, ACMA

Dudoit S, Fridlyand J. 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biol. 2002 Jun 25;3(7)

Goder A and Filkov V. Consensus Clustering Algorithms: Comparison and Refinement. 2008 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX) — San Francisco, 19 January 2008. Society for Industrial and Applied Mathematics.

Golub TR, Slonim DK, Tamayo P, Huard C, et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science vol. 286.

Giancarlo R, Scaturro D and Utro F. 2008 Computational cluster validation for microarray data analysis: experimental assessment of Clest, Consensus Clustering, Figure of Merit, Gap Statistics and Model Explorer. BMC Bioinformatics 2008, 9:462

Handl J, Knowles J and Kell DB. 2005. Computational cluster validation in post-genomic data analysis. Bioinformatics 21 (15): 3201-3212.

Jain AK. and Moreau J. 1988. Bootstrap techniques in cluster analysis. Pattern Recognition 20, 547–568

Kaufman L, Rousseeuw PJ. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. New York: Wiley

Langfelder P, Zhang B, Horvath S. 2007. Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics 24(5):719-720

Monti S, Tamayo P, Mesirov J, Golub T. 2003. Consensus clustering - A resampling-based method for class discovery and visualization of gene expression microarray dataAquatic Microbial Ecology 30: 83–89.

Perou CM, Sørlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA, Fluge O, Pergamenschikov A, Williams C, Zhu SX, Lønning PE, Børresen-Dale AL, Brown PO, Botstein D. 2000. Molecular portraits of human breast tumours. Nature 406:747–52

R Core Team. 2012. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Simpson TI, Armstrong JD and Jarman AP. 2010 Merged consensus clustering to assess and improve class discovery with microarray data. BMC Bioinformatics, 11:590.

Tibshirani, R., Walther, G. and Hastie, T. 2001. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63: 411–423

Wilkerson M. 2011. ConsensusClusterPlus: ConsensusClusterPlus. R package version 1.8.0.