



---

# Journal of Statistical Software

March 2008, Volume 25, Issue 4.

<http://www.jstatsoft.org/>

---

## clValid: An R Package for Cluster Validation

**Guy Brock**

Universtiy of Louisville

**Vasyl Pihur**

Universtiy of Louisville

**Susmita Datta**

Universtiy of Louisville

**Somnath Datta**

Universtiy of Louisville

---

### Abstract

The R package **clValid** contains functions for validating the results of a clustering analysis. There are three main types of cluster validation measures available, “internal”, “stability”, and “biological”. The user can choose from nine clustering algorithms in existing R packages, including hierarchical,  $K$ -means, self-organizing maps (SOM), and model-based clustering. In addition, we provide a function to perform the self-organizing tree algorithm (SOTA) method of clustering. Any combination of validation measures and clustering methods can be requested in a single function call. This allows the user to simultaneously evaluate several clustering algorithms while varying the number of clusters, to help determine the most appropriate method and number of clusters for the dataset of interest. Additionally, the package can automatically make use of the biological information contained in the Gene Ontology (GO) database to calculate the biological validation measures, via the annotation packages available in Bioconductor. The function returns an object of S4 class “clValid”, which has summary, plot, print, and additional methods which allow the user to display the optimal validation scores and extract clustering results.

*Keywords:* clustering, validation, R package, stability measures, biological annotation, functional categories.

---

## 1. Introduction

Clustering is an unsupervised technique used to group together objects which are “close” to one another in a multidimensional feature space, usually for the purpose of uncovering some inherent structure which the data possesses. Clustering is commonly used in the analysis of high-throughput genomic data, with the aim of grouping together genes or proteins which have similar expression patterns and possibly share common biological pathways (DeRisi *et al.*

1997; Chu *et al.* 1998; Eisen *et al.* 1998; Bhattacharjee *et al.* 2007). A plethora of clustering algorithms currently exist, many of which have shown some promise in the analysis of genomic data (Herrero *et al.* 2001; McLachlan *et al.* 2002; Dembele and Kastner 2003; Fu and Medico 2007). Deciding which clustering method to use can therefore be a daunting task for the researcher conducting the experiment. An additional, related problem is determining the number of clusters that are most appropriate for the data. Ideally, the resulting clusters should not only have good statistical properties (compact, well-separated, connected, and stable), but also give results that are biologically relevant.

A variety of measures aimed at validating the results of a clustering analysis and determining which clustering algorithm performs the best for a particular experiment have been proposed (Kerr and Churchill 2001; Yeung *et al.* 2001; Datta and Datta 2003). This validation can be based solely on the internal properties of the data or on some external reference, and on the expression data alone or in conjunction with relevant biological information (Gibbons and Roth 2002; Gat-Viks *et al.* 2003; Bolshakova *et al.* 2005; Datta and Datta 2006). The article by Handl *et al.* (2005), in particular, gives an excellent overview of cluster validation with post-genomic data and provides a synopsis of many of the available validation measures.

In this paper, we present an R package **clValid** which contains a variety of methods for validating the results from a cluster analysis. The main function is `clValid()`, and the available validation measures fall into the three general categories of “internal”, “stability”, and “biological”. The user can simultaneously select multiple clustering algorithms, validation measures, and numbers of clusters in a single function call, to determine the most appropriate method and an optimal number of clusters for the dataset. Additionally, the **clValid** package makes use of the biological information contained in the Gene Ontology (GO, <http://www.geneontology.org/>) database via the annotation packages in Bioconductor (<http://www.bioconductor.org/>) in order to automate the calculation of the biological validation measures. The package also contains a function for implementing the self-organizing tree algorithm (SOTA Dopazo and Carazo 1997), which to our knowledge has not been previously available in R packages on the Comprehensive R Archive Network (CRAN, <http://CRAN.R-project.org>). The function returns an object of S4 class “clValid”, which has a variety of methods available to plot and summarize the validation measures, display the optimal scores along with the corresponding cluster method and number of clusters, and extract the clustering results for a particular algorithm.

The rest of this paper is organized as follows. Section 2 contains a detailed description of the validation measures that are available. Section 3 describes the clustering algorithms which are available to use with the **clValid** package. Section 4 contain an example using mouse gene expression data from Bhattacharjee *et al.* (2007) that illustrates the use of the **clValid** package functions and objects. Finally, Section 5 discusses some additional validation software which is available, and some of the benefits our software provides in comparison.

## 2. Validation measures

The **clValid** package offers three types of cluster validation, “internal”, “stability”, and “biological”. Internal validation measures take only the dataset and the clustering partition as input and use intrinsic information in the data to assess the quality of the clustering. The stability measures are a special version of internal measures. They evaluate the consistency of

a clustering result by comparing it with the clusters obtained after each column is removed, one at a time. Biological validation evaluates the ability of a clustering algorithm to produce biologically meaningful clusters. We have measures to investigate both the biological homogeneity and stability of the clustering results.

## 2.1. Internal measures

For internal validation, we selected measures that reflect the compactness, connectedness, and separation of the cluster partitions. Connectedness relates to what extent observations are placed in the same cluster as their nearest neighbors in the data space, and is here measured by the connectivity (Handl *et al.* 2005). Compactness assesses cluster homogeneity, usually by looking at the intra-cluster variance, while separation quantifies the degree of separation between clusters (usually by measuring the distance between cluster centroids). Since compactness and separation demonstrate opposing trends (compactness increases with the number of clusters but separation decreases), popular methods combine the two measures into a single score. The Dunn index (Dunn 1974) and silhouette width (Rousseeuw 1987) are both examples of non-linear combinations of the compactness and separation, and with the connectivity comprise the three internal measures available in **clValid**. The details of each measure are given below, and for a good overview of internal measures in general see Handl *et al.* (2005).

### *Connectivity*

Define  $nn_{i(j)}$  as the  $j$ th nearest neighbor of observation  $i$ , and let  $x_{i,nn_{i(j)}}$  be zero if  $i$  and  $nn_{i(j)}$  are in the same cluster and  $1/j$  otherwise. Then, for a particular clustering partition  $\mathcal{C} = \{C_1, \dots, C_K\}$  of the  $N$  observations into  $K$  disjoint clusters, the connectivity is defined as

$$Conn(\mathcal{C}) = \sum_{i=1}^N \sum_{j=1}^L x_{i,nn_{i(j)}} ,$$

where  $L$  is a parameter that determines the number of neighbors that contribute to the connectivity measure. The connectivity has a value between zero and  $\infty$  and should be minimized.

### *Silhouette width*

The silhouette width is the average of each observation's silhouette value. The silhouette value measures the degree of confidence in the clustering assignment of a particular observation, with well-clustered observations having values near 1 and poorly clustered observations having values near  $-1$ . For observation  $i$ , it is defined as

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)} ,$$

where  $a_i$  is the average distance between  $i$  and all other observations in the same cluster, and  $b_i$  is the average distance between  $i$  and the observations in the "nearest neighboring cluster", i.e.

$$a_i = \frac{1}{n(C(i))} \sum_{j \in C(i)} dist(i, j) , \quad b_i = \min_{C_k \in \mathcal{C} \setminus C(i)} \sum_{j \in C_k} \frac{dist(i, j)}{n(C_k)} ,$$

where  $C(i)$  is the cluster containing observation  $i$ ,  $dist(i, j)$  is the distance (e.g. Euclidean, Manhattan) between observations  $i$  and  $j$ , and  $n(C)$  is the cardinality of cluster  $C$ . The silhouette width thus lies in the interval  $[-1, 1]$ , and should be maximized. For more information, see the help page for the `silhouette()` function in package `cluster` (Rousseeuw *et al.* 2007).

### *Dunn index*

The Dunn index is the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance. It is computed as

$$D(\mathcal{C}) = \frac{\min_{C_k, C_l \in \mathcal{C}, C_k \neq C_l} \left( \min_{i \in C_k, j \in C_l} dist(i, j) \right)}{\max_{C_m \in \mathcal{C}} diam(C_m)},$$

where  $diam(C_m)$  is the maximum distance between observations in cluster  $C_m$ . The Dunn index has a value between zero and  $\infty$ , and should be maximized.

## 2.2. Stability measures

Let  $N$  denote the total number of observations (rows) in a dataset and  $M$  denote the total number of columns, which are assumed to be numeric (e.g., a collection of samples, time points, etc.). The stability measures compare the results from clustering based on the full data to clustering based on removing each column, one at a time (Datta and Datta 2003; Yeung *et al.* 2001). These measures work especially well if the data are highly correlated, which is often the case in high-throughput genomic data. The included measures are the average proportion of non-overlap (APN), the average distance (AD), the average distance between means (ADM), and the figure of merit (FOM). In all cases the average is taken over all the deleted columns, and all measures should be minimized.

### *Average proportion of non-overlap (APN)*

The APN measures the average proportion of observations not placed in the same cluster by clustering based on the full data and clustering based on the data with a single column removed. Let  $C^{i,0}$  represent the cluster containing observation  $i$  using the original clustering (based on all available data), and  $C^{i,\ell}$  represent the cluster containing observation  $i$  where the clustering is based on the dataset with column  $\ell$  removed. Then, the APN measure is defined as

$$APN(\mathcal{C}) = \frac{1}{MN} \sum_{i=1}^N \sum_{\ell=1}^M \left( 1 - \frac{n(C^{i,\ell} \cap C^{i,0})}{n(C^{i,0})} \right).$$

The APN is in the interval  $[0, 1]$ , with values close to zero corresponding with highly consistent clustering results.

### *Average distance (AD)*

The AD measure computes the average distance between observations placed in the same cluster by clustering based on the full data and clustering based on the data with a single

column removed. It is defined as

$$AD(\mathcal{C}) = \frac{1}{MN} \sum_{i=1}^N \sum_{\ell=1}^M \frac{1}{n(C^{i,0})n(C^{i,\ell})} \left[ \sum_{i \in C^{i,0}, j \in C^{i,\ell}} dist(i, j) \right].$$

The AD has a value between zero and  $\infty$ , and smaller values are preferred.

#### *Average distance between means (ADM)*

The ADM measure computes the average distance between cluster centers for observations placed in the same cluster by clustering based on the full data and clustering based on the data with a single column removed. It is defined as

$$ADM(\mathcal{C}) = \frac{1}{MN} \sum_{i=1}^N \sum_{\ell=1}^M dist(\bar{x}_{C^{i,\ell}}, \bar{x}_{C^{i,0}}),$$

where  $\bar{x}_{C^{i,0}}$  is the mean of the observations in the cluster which contain observation  $i$ , when clustering is based on the full data, and  $\bar{x}_{C^{i,\ell}}$  is similarly defined. Currently, ADM only uses the Euclidean distance. It also has a value between zero and  $\infty$ , and again smaller values are preferred.

#### *Figure of merit (FOM)*

The FOM measures the average intra-cluster variance of the observations in the deleted column, where the clustering is based on the remaining (undeleted) samples. This estimates the mean error using predictions based on the cluster averages. For a particular left-out column  $\ell$ , the FOM is

$$FOM(\ell, \mathcal{C}) = \sqrt{\frac{1}{N} \sum_{k=1}^K \sum_{i \in C_k(\ell)} dist(x_{i,\ell}, \bar{x}_{C_k(\ell)})},$$

where  $x_{i,\ell}$  is the value of the  $i$ th observation in the  $\ell$ th column, and  $\bar{x}_{C_k(\ell)}$  is the average of cluster  $C_k(\ell)$ . Currently, the only distance available for FOM is Euclidean. The FOM is multiplied by an adjustment factor  $\sqrt{\frac{N}{N-K}}$ , to alleviate the tendency to decrease as the number of clusters increases. The final score is averaged over all the removed columns, and has a value between zero and  $\infty$ , with smaller values equaling better performance.

### **2.3. Biological measures**

Biological validation evaluates the ability of a clustering algorithm to produce biologically meaningful clusters. A typical application of biological validation is in microarray data, where observations correspond to genes (where “genes” could be open reading frames (ORFs), express sequence tags (ESTs), serial analysis of gene expression (SAGE) tags, etc.). There are two measures available, the biological homogeneity index (BHI) and biological stability index (BSI), both originally presented in [Datta and Datta \(2006\)](#).

#### *Biological homogeneity index (BHI)*

As its name implies, the BHI measures how homogeneous the clusters are biologically. Let  $\mathcal{B} = \{B_1, \dots, B_F\}$  be a set of  $F$  functional classes, not necessarily disjoint, and let  $B(i)$  be

the functional class containing gene  $i$  (with possibly more than one functional class containing  $i$ ). Similarly, we define  $B(j)$  as the function class containing gene  $j$ , and assign the indicator function  $I(B(i) = B(j))$  the value 1 if  $B(i)$  and  $B(j)$  match (any one match is sufficient in the case of membership to multiple functional classes), and 0 otherwise. Intuitively, we hope that genes placed in the same statistical cluster also belong to the same functional classes. Then, for a given statistical clustering partition  $\mathcal{C} = \{C_1, \dots, C_K\}$  and set of biological classes  $\mathcal{B}$ , the BHI is defined as

$$BHI(\mathcal{C}, \mathcal{B}) = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k(n_k - 1)} \sum_{i \neq j \in C_k} I(B(i) = B(j)).$$

Here  $n_k = n(C_k \cap B)$  is the number of annotated genes in statistical cluster  $C_k$ . The BHI is in the range  $[0, 1]$ , with larger values corresponding to more biologically homogeneous clusters.

### *Biological stability index (BSI)*

The BSI is similar to the other stability measures, and inspects the consistency of clustering for genes with similar biological functionality. Each sample is removed one at a time, and the cluster membership for genes with similar functional annotation is compared with the cluster membership using all available samples. The BSI is defined as

$$BSI(\mathcal{C}, \mathcal{B}) = \frac{1}{F} \sum_{k=1}^F \frac{1}{n(B_k)(n(B_k) - 1)M} \sum_{\ell=1}^M \sum_{i \neq j \in B_k} \frac{n(C^{i,0} \cap C^{j,\ell})}{n(C^{i,0})},$$

where  $F$  is the total number of functional classes,  $C^{i,0}$  is the statistical cluster containing observation  $i$  based on all the data, and  $C^{j,\ell}$  is the statistical cluster containing observation  $j$  when column  $\ell$  is removed. The BSI is in the range  $[0, 1]$ , with larger values corresponding to more stable clusters of the functionally annotated genes.

## 3. Clustering algorithms

The R statistical computing project ([R Development Core Team 2007](#)) has a wide variety of clustering algorithms available in the base distribution and various add-on packages. We make use of nine algorithms from the base distribution and add-on packages **cluster** ([Rousseeuw et al. 2007](#); [Kaufman and Rousseeuw 1990](#)), **kohonen** ([Wehrens 2007](#)), and **mclust** ([Fraley and Raftery 2007](#); [Fraley and A. E. Raftery 2003](#)), and in addition provide a function for implementing SOTA ([Dopazo and Carazo 1997](#)) in the **clValid** package. A brief description of each clustering method and its availability is given below.

### *Hierarchical*

Hierarchical clustering is an agglomerative clustering algorithm that yields a dendrogram which can be cut at a chosen height to produce the desired number of clusters ([Kaufman and Rousseeuw 1990](#)). Each observation is initially placed in its own cluster, and the clusters are successively joined together in order of their ‘‘closeness’’. The closeness of any two clusters is determined by a dissimilarity matrix, and can be based on a variety of agglomeration methods. Hierarchical clustering is included with the base distribution of R in function `hclust()`, and is also implemented in the `agnes()` function in package **cluster**.

### *K-means*

*K*-means is an iterative method which minimizes the within-class sum of squares for a given number of clusters (Hartigan and Wong 1979). The algorithm starts with an initial guess for the cluster centers, and each observation is placed in the cluster to which it is closest. The cluster centers are then updated, and the entire process is repeated until the cluster centers no longer move. Often another clustering algorithm (e.g., hierarchical) is run initially to determine starting points for the cluster centers. *K*-means is implemented in the function `kmeans()`, included with the base distribution of R.

### *DIANA*

DIANA is a divisive hierarchical algorithm that initially starts with all observations in a single cluster, and successively divides the clusters until each cluster contains a single observation. Along with SOTA, DIANA is one of a few representatives of the divisive hierarchical approach to clustering. DIANA is available in function `diana()` in package **cluster**.

### *PAM*

Partitioning around medoids (PAM) is similar to *K*-means, but is considered more robust because it admits the use of other dissimilarities besides Euclidean distance. Like *K*-means, the number of clusters is fixed in advance, and an initial set of cluster centers is required to start the algorithm. PAM is available in the **cluster** package as function `pam()`.

### *CLARA*

CLARA is a sampling-based algorithm which implements PAM on a number of sub-datasets (Kaufman and Rousseeuw 1990). This allows for faster running times when a number of observations is relatively large. CLARA is also available in package **cluster** as function `clara()`.

### *FANNY*

This algorithm performs fuzzy clustering, where each observation can have partial membership in each cluster (Kaufman and Rousseeuw 1990). Thus, each observation has a vector which gives the partial membership to each of the clusters. A hard cluster can be produced by assigning each observation to the cluster where it has the highest membership. FANNY is available in the **cluster** package (function `fanny()`).

### *SOM*

Self-organizing maps (Kohonen 1997) is an unsupervised learning technique that is popular among computational biologists and machine learning researchers. SOM is based on neural networks, and is highly regarded for its ability to map and visualize high-dimensional data in two dimensions. SOM is available as the `som()` function in package **kohonen**.

### *Model-based clustering*

Under this approach, a statistical model consisting of a finite mixture of Gaussian distributions is fit to the data (Fraley and Raftery 2001). Each mixture component represents a cluster, and the mixture components and group memberships are estimated using maximum

likelihood (EM algorithm). The function `Mclust()` in package **mclust** implements model based clustering.

### *SOTA*

Self-organizing tree algorithm (SOTA) is an unsupervised network with a divisive hierarchical binary tree structure. It was originally proposed by [Dopazo and Carazo \(1997\)](#) for phylogenetic reconstruction, and later applied to cluster microarray gene expression data in ([Herrero et al. 2001](#)). It uses a fast algorithm and hence is suitable for clustering a large number of objects. SOTA is included with the **clValid** package as function `sota()`.

## 4. Example: Mouse mesenchymal cells

To illustrate the cluster validation measures in package **clValid**, we use data from an Affymetrix microarray experiment comparing gene expression of mesenchymal cells from two distinct lineages, neural crest and mesoderm-derived. The dataset consists of 147 genes and ESTs which were determined to be significantly differentially expressed between the two cell lineages, with at least a 1.5 fold increase or decrease in expression. There are three samples for each of the neural crest and mesoderm-derived cells, so the expression matrix has dimension  $147 \times 6$ . In addition, the genes were grouped into the functional classes according to their biological description, with categories ECM/receptors (16), growth/differentiation (16), kinases/phosphatases (7), metabolism (8), stress-induced (6), transcription factors (28), and miscellaneous (25). The biological functions of 10 genes were unknown, and 31 of the “genes” were ESTs. For further description of the dataset and the experiments the reader is referred to [Bhattacharjee et al. \(2007\)](#).

We begin by loading the package, then loading the dataset.

```
R> library("clValid")
R> data("mouse")
```

This dataset has the typical format found in microarray data, with the rows as genes (variables) and the columns as the samples. Although this is a transposition of the data structure used for more conventional statistics (rows are samples, columns are variables), in both cases the typical goal is to cluster the rows based on the columns (although, in microarray data analysis the samples are also sometimes clustered). Hence, the **clValid** function assumes that the rows of the input matrix are the intended items to be clustered.

We want to evaluate the results from a clustering analysis, using all the available clustering algorithms. Since the genes fall into one of two groups, up or down-regulated in the neural crest vs. mesoderm-derived tissue, the numbers of clusters is varied from 2 to 6. The distance metric (both for the applicable clustering methods and validation measures) is set to “euclidean”; other available options are “correlation” and “manhattan”. The agglomeration method for hierarchical clustering is set to “average”. We illustrate each category of validation measure separately, but it should be noted that the user can request all three types of validation measures at once (which would also be more computationally efficient).

### *Internal validation*

The internal validation measures are the connectivity, silhouette width, and Dunn index. The



neighborhood size for the connectivity is set to 10 by default, the `neighbSize` argument can be used to change this. Note that the clustering method “agnes” was omitted, since this also performs hierarchical clustering and would be redundant with the “hierarchical” method.

```
R> express <- mouse[, c("M1", "M2", "M3", "NC1", "NC2", "NC3")]
R> rownames(express) <- mouse$ID
R> intern <- clValid(express, 2:6, clMethods = c("hierarchical",
+       "kmeans", "diana", "fanny", "som", "pam", "sota", "clara",
+       "model"), validation = "internal")
```

To view the results of the analysis, `print`, `plot`, and `summary` methods are available for the `clValid` object `intern`. The summary statement will display all the validation measures in a table, and also give the clustering method and number of clusters corresponding to the optimal score for each measure.

```
R> summary(intern)
```

Clustering Methods:

```
hierarchical kmeans diana fanny som pam sota clara model
```

Cluster sizes:

```
2 3 4 5 6
```

Validation Measures:

		2	3	4	5	6
hierarchical	Connectivity	5.3270	14.2528	20.7520	27.0726	30.6194
	Dunn	0.1291	0.0788	0.0857	0.0899	0.0899
	Silhouette	0.5133	0.4195	0.3700	0.3343	0.3233
kmeans	Connectivity	13.2548	17.6651	37.3980	43.2655	50.6095
	Dunn	0.0464	0.0873	0.0777	0.0815	0.0703
	Silhouette	0.4571	0.4182	0.3615	0.3367	0.3207
diana	Connectivity	18.7552	25.1187	38.1242	38.8143	45.1349
	Dunn	0.0315	0.0358	0.0492	0.0577	0.0646
	Silhouette	0.4601	0.3705	0.3538	0.3378	0.3316
fanny	Connectivity	19.8925	32.7579	42.7421	42.7992	55.6552
	Dunn	0.0401	0.0430	0.0623	0.0700	0.0632
	Silhouette	0.4332	0.3401	0.2877	0.2765	0.3624
som	Connectivity	13.2548	27.5270	40.7056	44.8294	35.8317
	Dunn	0.0464	0.0664	0.0554	0.0612	0.0816
	Silhouette	0.4571	0.4097	0.3529	0.3210	0.3968
pam	Connectivity	18.7917	27.9651	30.9302	44.9671	32.9667
	Dunn	0.0391	0.0597	0.0510	0.0761	0.0816
	Silhouette	0.4271	0.3489	0.3563	0.3530	0.4152
sota	Connectivity	22.7690	30.1794	32.6333	41.8321	47.7548
	Dunn	0.0351	0.0446	0.0459	0.0459	0.0509
	Silhouette	0.4395	0.3682	0.3169	0.2887	0.3236

clara	Connectivity	18.7028	27.9651	44.8234	35.5159	26.1238
	Dunn	0.0287	0.0597	0.0660	0.0761	0.0857
	Silhouette	0.4257	0.3489	0.3304	0.3636	0.3836
model	Connectivity	23.7373	121.6671	89.2726	111.0246	96.4258
	Dunn	0.0240	0.0304	0.0232	0.0332	0.0231
	Silhouette	0.3291	0.2131	-0.0106	0.0902	0.0694

Optimal Scores:

	Score	Method	Clusters
Connectivity	5.3270	hierarchical	2
Dunn	0.1291	hierarchical	2
Silhouette	0.5133	hierarchical	2

Hierarchical clustering with two clusters performs the best in each case. The validation measures can also be displayed graphically using the `plot()` method. Plots for individual measures can be requested using the `measures` argument. A legend is also included with each plot. The default location of the legend is the top right corner of each plot, this can be changed using the `legendLoc` argument. Here, we combine all three plots into a single figure and so suppress the legends in each individual plot.

```
R> op <- par(no.readonly = TRUE)
R> par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))
R> plot(intern, legend = FALSE)
R> plot(nClusters(intern), measures(intern, "Dunn")[, , 1], type = "n",
+       axes = F, xlab = "", ylab = "")
R> legend("center", clusterMethods(intern), col = 1:9, lty = 1:9,
+       pch = paste(1:9))
R> par(op)
```

The plots of the connectivity, Dunn index, and silhouette width are given in Figure 1. Recall that the connectivity should be minimized, while both the Dunn index and the silhouette width should be maximized. Thus, it appears that hierarchical clustering outperforms the other clustering algorithms under each validation measure, for nearly every number of clusters evaluated. Somewhat surprisingly, model-based clustering does not perform well on any of the measures. Regardless of the clustering algorithm, the optimal number of clusters seems to be two using the connectivity and silhouette width. For the Dunn index the best choice for the number of clusters is less clear.

### *Stability validation*

The stability measures include the APN, AD, ADM, and FOM. The measures should be minimized in each case. Stability validation requires more time than internal validation, since clustering needs to be redone for each of the datasets with a single column removed.

```
R> stab <- clValid(express, 2:6, clMethods = c("hierarchical", "kmeans",
+     "diana", "fanny", "som", "pam", "sota", "clara", "model"),
+     validation = "stability")
```

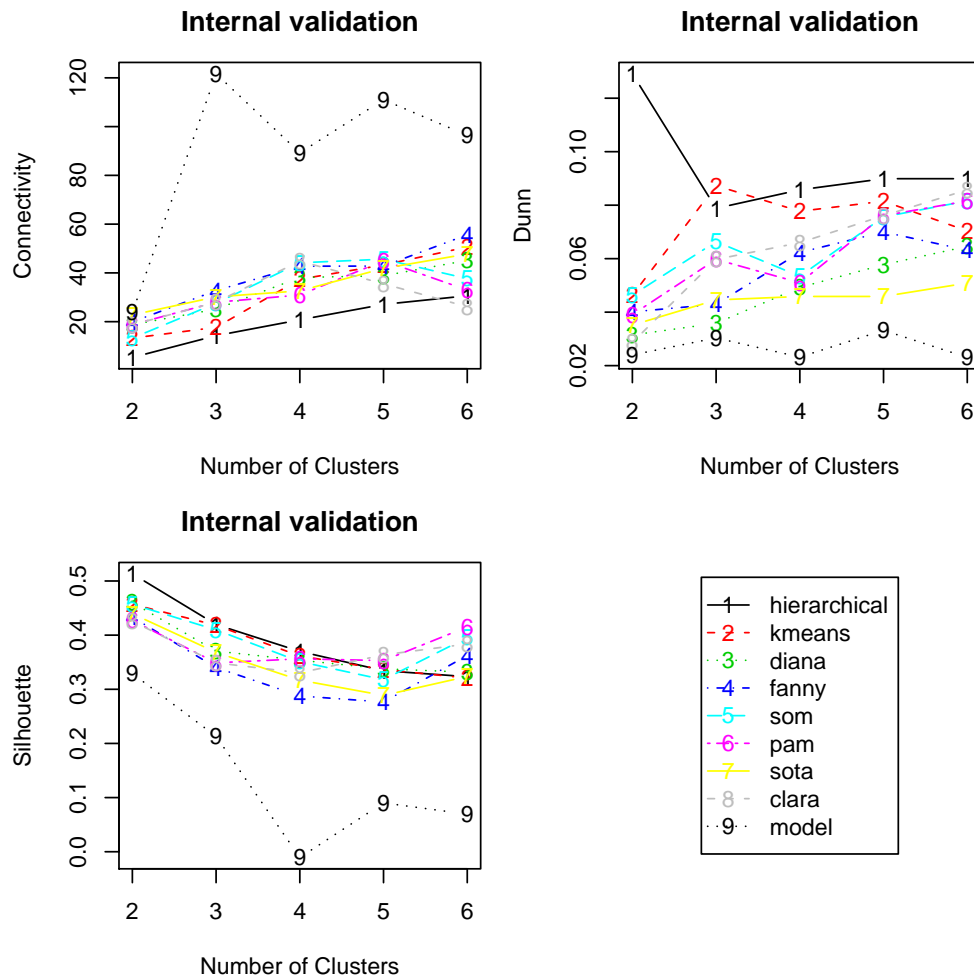


Figure 1: Plots of the connectivity measure, the Dunn index, and the silhouette width.

Instead of viewing all the validation measures via the `summary()` method, we can instead just view the optimal values using the `optimalScores()` method.

```
R> optimalScores(stab)
```

	Score	Method	Clusters
APN	0.04781010	hierarchical	2
AD	1.52717887	pam	6
ADM	0.14007952	pam	6
FOM	0.51580323	pam	6

For the APN measures, hierarchical clustering with two clusters again gives the best score. However, for the other three measures PAM with six clusters has the best score. It is illustrative to graphically visualize each of the validation measures. The plot of the FOM measure is very similar to the AD measure, so we have omitted it from the figure.

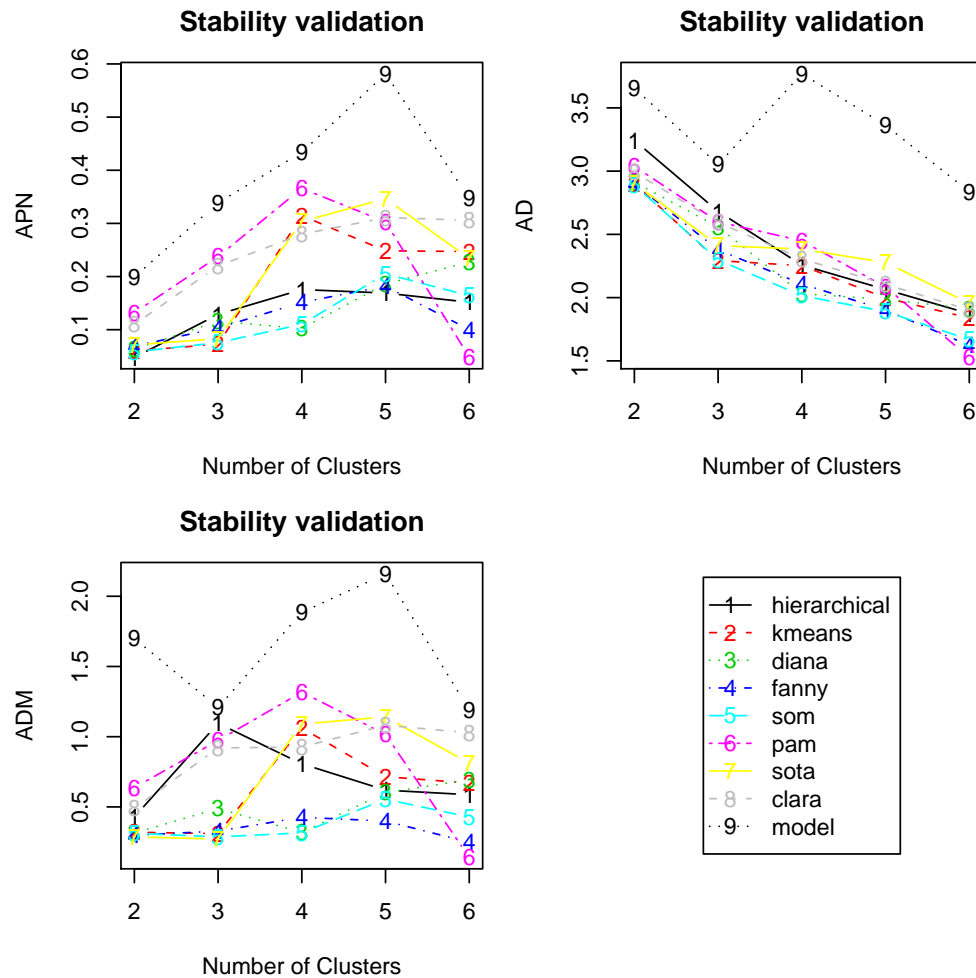


Figure 2: Plot of the APN, AD, and APN measures.

```
R> par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))
R> plot(stab, measure = c("APN", "AD", "ADM"), legend = FALSE)
R> plot(nClusters(stab), measures(stab, "APN")[, 1], type = "n",
+       axes = F, xlab = "", ylab = "")
R> legend("center", clusterMethods(stab), col = 1:9, lty = 1:9,
+       pch = paste(1:9))
R> par(op)
```

The plots of the APN, AD, and ADM are given in Figure 2. The APN measure shows an interesting trend for many of the clustering methods, in that it initially increases from two to four clusters but subsequently decreases. Though hierarchical clustering with two clusters has the best score, PAM with six clusters is a close second. The AD and FOM measures tend to decrease as the number of clusters increases. PAM with six clusters has the best overall score, but over the entire range of clusters evaluated SOM, K-means, and DIANA have better overall performance. Similarly, for the ADM measure SOTA has a more stable and better

overall performance when compared to PAM over the entire range for the number of clusters.

### *Biological validation*

There are two options for biological validation using the BHI and BSI measures. The first option is to explicitly specify the functional clustering of the genes. This requires the user to predetermine the functional classes of the genes, e.g. using an annotation software package like **FatiGO** (Al-Shahrour *et al.* 2004) or **FunCat** (Ruepp *et al.* 2004).

The functional categorization of the genes in the dataset **mouse** were previously determined in Bhattacharjee *et al.* (2007), so these will be used initially to define the functional classes.

```
R> fc <- tapply(rownames(express), mouse$FC, c)
R> fc <- fc[!names(fc) %in% c("EST", "Unknown")]
R> bio <- clValid(express, 2:6, clMethods = c("hierarchical", "kmeans",
+     "diana", "fanny", "som", "pam", "sota", "clara", "model"),
+     validation = "biological", annotation = fc)
```

Recall that both the BHI and BSI should be maximized. The optimal values for each measure are given below.

```
R> optimalScores(bio)
```

	Score	Method	Clusters
BHI	0.2533592	model	6
BSI	0.6755826	hierarchical	2

Model-based clustering with six clusters has the best value of the BHI, while for the BSI hierarchical clustering with two clusters again does well. Plots of the measures are given in Figures 3 and 4.

```
R> plot(bio, measure = "BHI", legendLoc = "topleft")
R> plot(bio, measure = "BSI")
```

Model-based clustering appears to have the best BHI score over the range for the number of clusters, while hierarchical clustering is slightly better than model-based overall for the BSI scores.

The other option for biological validation is to use the annotation packages available in Bioconductor (<http://www.bioconductor.org/>, Gentleman *et al.* 2004). This option uses the annotation packages to map the genes to their corresponding GO terms. There are three main ontologies, cellular component (“CC”), biological process (“BP”), and molecular function (“MF”), which can be selected via the `GOcategory` argument. The user must download, at a minimum, the **Biobase** (Gentleman *et al.* 2007), **annotate** (Gentleman 2007), and **GO** (Liu *et al.* 2007a) packages from Bioconductor, then load them during the R session. In addition, any specific annotation packages that are required will need to be downloaded (e.g., experiments using the Affymetrix GeneChip hgu95av2 would require the **hgu95av2** package Liu *et al.* 2007b). Once the appropriate annotation packages are downloaded, they can be

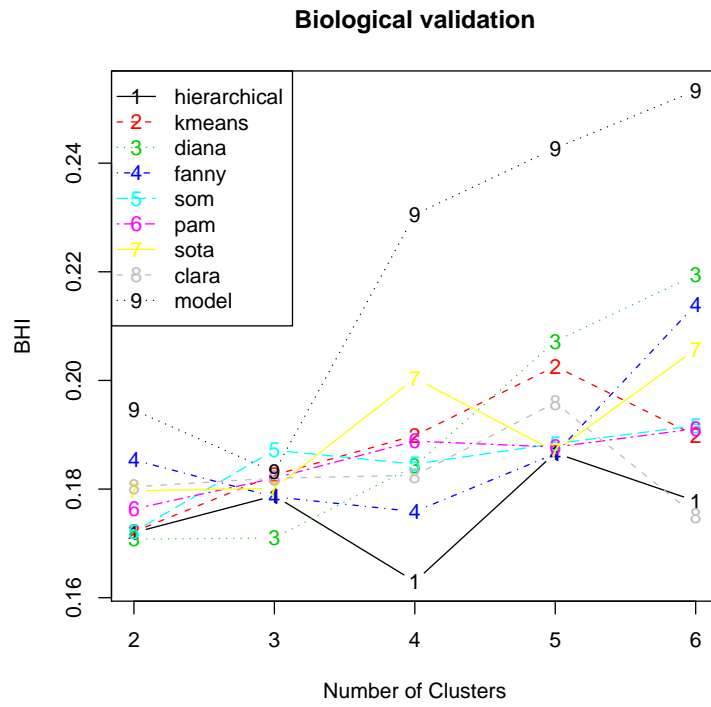


Figure 3: Plot of the BHI measure, using predetermined functional classes.

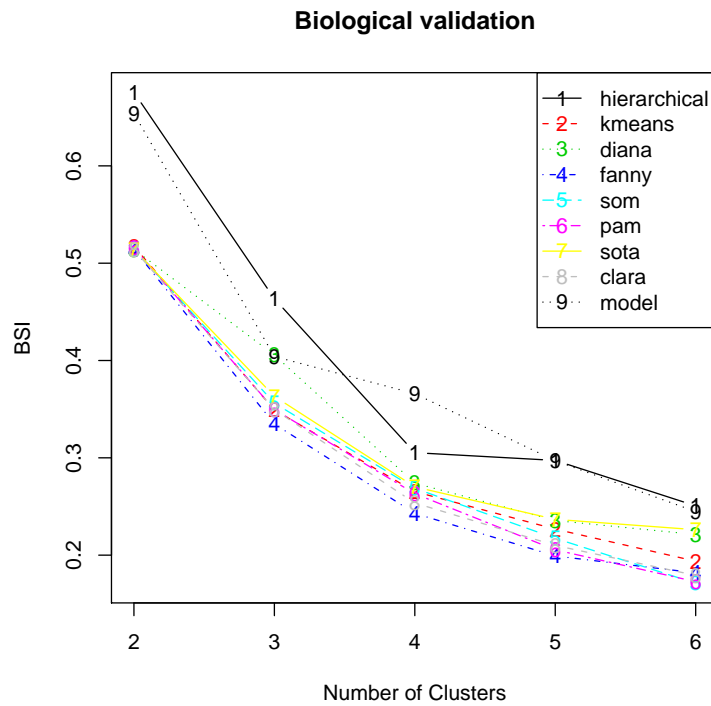


Figure 4: Plot of the BSI measure, using predetermined functional classes.

specified in the function call via the `annotation` argument. The `goTermFreq` argument is used to select a threshold, so that only GO terms with a frequency in the dataset above the threshold are used to determine the functional classes.

To illustrate, the identifiers in the dataset `mouse` are from the Affymetrix Murine Genome 430a GeneChip Array, with corresponding annotation package `moe430a` (Liu *et al.* 2007c) available from Bioconductor. We leave the `goTermFreq` argument at its default level of 0.05, and use all available GO categories (`GOcategory="all"`) for annotation.

```
R> library("Biobase")
R> library("annotate")
R> library("GO")
R> library("moe430a")
R> bio2 <- clValid(express, 2:6, clMethods = c("hierarchical", "kmeans",
+      "diana", "fanny", "som", "pam", "sota", "clara", "model"),
+      validation = "biological", annotation = "moe430a", GOcategory = "all")

R> optimalScores(bio2)
```

	Score	Method	Clusters
BHI	0.1655374	model	6
BSI	0.7938518	hierarchical	2

The optimal method and number of clusters for the two measures agree with those found using the predetermined functional classes, and the plots of the measures given in Figures 5 and 6 are also very similar to the previous plots. Notice again that hierarchical clustering has the best performance on the BSI measurement over the range for the number of clusters, but generally does poorly under the BHI validation measure.

```
R> plot(bio2, measure = "BHI", legendLoc = "topleft")
R> plot(bio2, measure = "BSI")
```

### *Further analysis*

Hierarchical clustering consistently performs well for many of the validation measures. The clustering results from any method can be extracted from a `clValid` object for further analysis, using the `clusters()` method. Here, we extract the results from hierarchical clustering, to plot the dendrogram and view the observations that are grouped together at the various levels of the topology. The dendrogram is plotted in Figure 7, with the genes belonging to the “Growth/Differentiation” (GD) and “Transcription factor” (TF) functional classes labeled. The genes belonging to the top two clusters are cross-classified with their functional annotation given in the dataset. Of potential interest, the second cluster contains no genes in the “EST” or “Miscellaneous” categories. Further inspection of the results is left to a subject matter expert.

```
R> hc <- clusters(bio, "hierarchical")
```

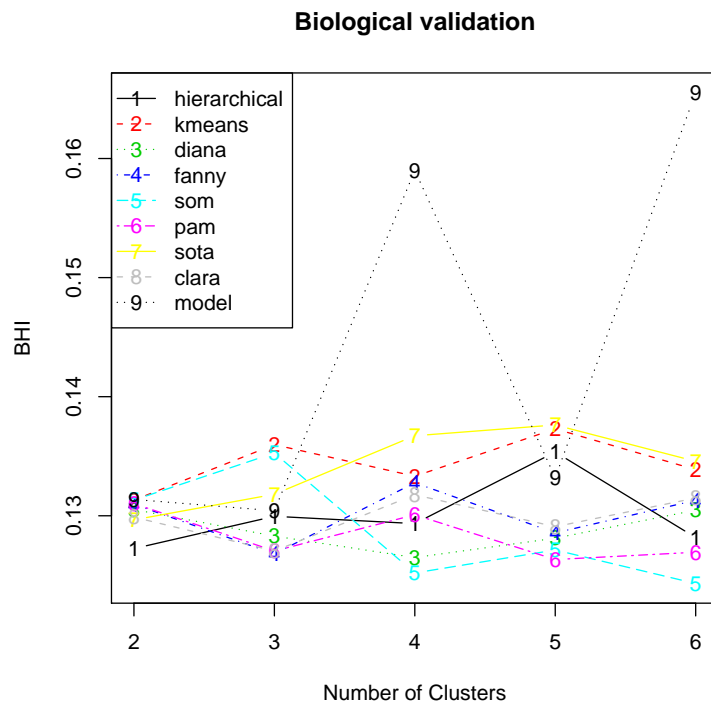


Figure 5: Plot of the BHI measure, using annotation package **moe430a** in Bioconductor.

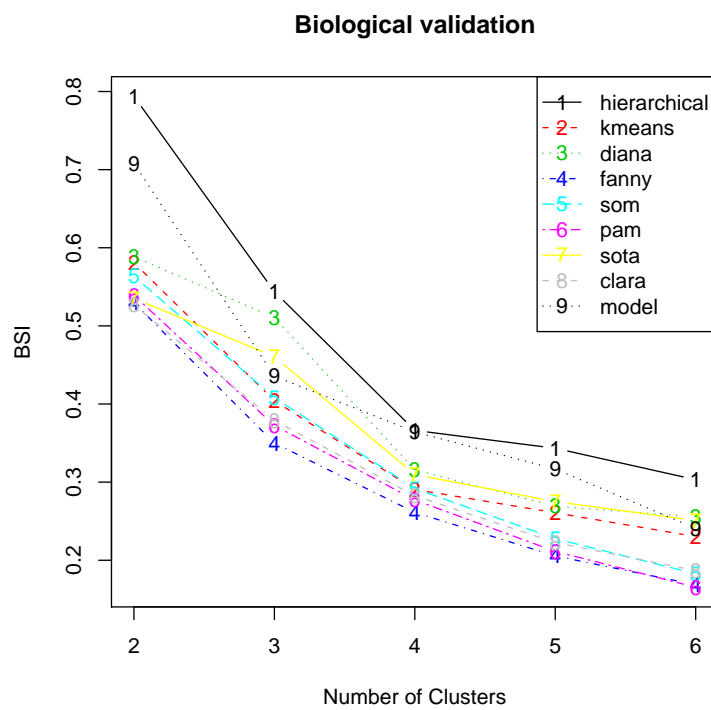


Figure 6: Plot of the BSI measure, using annotation package **moe430a** in Bioconductor.



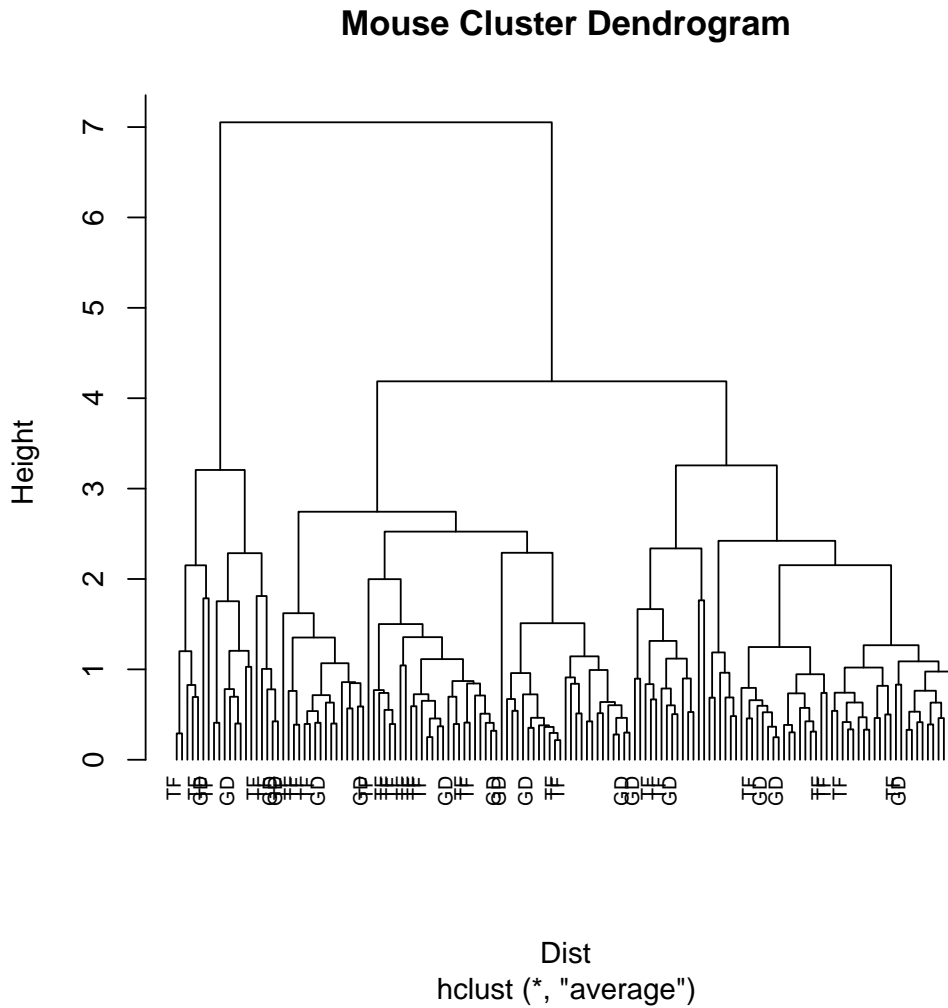


Figure 7: Plot of the dendrogram for hierarchical clustering. Growth/Differentiation (GD) and transcription factor (TF) genes are labeled.

```
R> mfc <- factor(mouse$FC, labels = c("Re", "EST", "GD", "KP", "Met",
+   "Mis", "St", "TF", "U"))
R> tf.gd <- ifelse(mfc %in% c("GD", "TF"), levels(mfc)[mfc], "")
R> plot(hc, labels = tf.gd, cex = 0.7, hang = -1, main = "Mouse Cluster Dendrogram")
```

```
R> two <- cutree(hc, 2)
R> xtabs(~mouse$FC + two)
```

	two	
mouse\$FC	1	2
ECM/Receptors	12	4

EST	31	0
Growth/Differentiation	12	4
Kinases/Phosphatases	4	3
Metabolism	7	1
Miscellaneous	25	0
Stress-induced	4	2
Transcription factor	23	5
Unknown	9	1

## 5. Discussion

We have developed an R package, **clValid**, which contains measures for validating the results from a clustering procedure. We categorize the measures into three distinct types, “internal”, “stability”, and “biological”, and provide plot, summary, and additional methods for viewing and summarizing the validation scores and extracting the clustering results for further analysis. In addition to the object-oriented nature of the language, implementing the validation measures within the R statistical programming framework provides the additional advantage in that it can interface with numerous clustering algorithms in existing R packages, and accommodate further algorithms as they are developed and coded into R libraries. Currently, `clValid()` accepts up to ten different clustering methods. This permits the user to simultaneously vary the number of clusters and the clustering algorithms to decide how best to group the observations in her/his dataset. Lastly, the package makes use of the annotation packages available in Bioconductor (<http://www.bioconductor.org/>) to calculate the biological validation measures, so that the information contained in the GO database can be used to assist in the cluster validation process.

The illustration for the **clValid** package we have given here focuses on clustering genes, but it is common in microarray analysis to cluster both genes and samples to create a “heatmap”. Though the “biological” validation measures are specifically designed for validation of clustering genes, the other measures could also be used with clustering of samples in a microarray experiment. Also, for microarray data, it is a good idea to limit the number of genes being clustered to a small subset (100 ~ 600) of the thousands of expression measures routinely available on a microarray, both for computational and visualization purposes. Typically, some initial pre-selection of the genes based on *t*-statistics, *p*-values, or expression ratios is performed.

There are several R packages that also perform cluster validation and are available from CRAN (<http://CRAN.R-project.org/>) or Bioconductor (<http://www.bioconductor.org/>). Examples include the `clustIndex()` function in package **cclust** (Dimitriadou 2007), which performs 14 different validation measures in three classes, `cluster.stats()` and `clusterboot()` in package **fpc** (Hennig 2007), the **clusterRepro** (Kapp and Tibshirani 2006) and **clusterSim** (Walesiak and Dudek 2007) packages, and the **clusterStab** (MacDonald *et al.* 2007) package from Bioconductor. The `cl_validity()` function in package **clue** (Hornik 2005) does validation for both partitioning methods (“dissimilarity accounted for”) and hierarchical methods (“variance accounted for”), and function `fclustIndex()` in package **e1071** (Dimitriadou *et al.* 2007) has several fuzzy cluster validation measures. However, to our knowledge none of these packages offers biological validation or the unique stability measures which we present here.

Handl *et al.* (2005) provides C++ code for the validation measures which they discuss, and the **Caat** tool available in the **GEPAS** (<http://gepas.bioinfo.cipf.es/>) software suite offers a web-based interface for visualizing and validating (using the silhouette width) cluster results. However, neither of these two tools are as flexible for interfacing with the variety of clustering algorithms that are available in the R language, or can automatically access the annotation information which is available in Bioconductor. Hence, the **clValid** package is a valuable addition to the growing collection of cluster validation software available for researchers.

## Acknowledgments

The authors would like to thank Dr. Bhattacharjee and his lab for sharing their data, and the editor, an associate editor, and two anonymous reviewers for their comments which helped improve the quality of this manuscript. This research was supported in part by NIH grant 1P30ES014443 (Guy Brock and Susmita Datta), NSF grants MCB-0517135 (Susmita Datta) and DMS-0706965 (Somnath Datta), and NSA grant H98230-06-1-0062 (Somnath Datta).

## References

- Al-Shahrour F, Diaz-Uriarte R, Dopazo J (2004). “**FatiGO**: A Web Tool for Finding Significant Associations of Gene Ontology Terms with Groups of Genes.” *Bioinformatics*, **20**(4), 578–80.
- Bhattacharjee V, Mukhopadhyay P, Singh S, Johnson C, Philipose JT, Warner CP, Greene RM, Pisano MM (2007). “Neural Crest and Mesoderm Lineage-Dependent Gene Expression in Orofacial Development.” *Differentiation*, **75**(5), 463–477.
- Bolshakova N, Azuaje F, Cunningham P (2005). “A Knowledge-Driven Approach to Cluster Validity Assessment.” *Bioinformatics*, **21**(10), 2546–7.
- Chu S, DeRisi J, Eisen M, Mulholland J, Botstein D, Brown PO, Herskowitz I (1998). “The Transcriptional Program of Sporulation in Budding Yeast.” *Science*, **282**(5389), 699–705.
- Datta S, Datta S (2003). “Comparisons and Validation of Statistical Clustering Techniques for Microarray Gene Expression Data.” *Bioinformatics*, **19**(4), 459–66.
- Datta S, Datta S (2006). “Methods for Evaluating Clustering Algorithms for Gene Expression Data using a Reference Set of Functional Classes.” *BMC Bioinformatics*, **7**, 397.
- Dembele D, Kastner P (2003). “Fuzzy *C*-Means Method for Clustering Microarray Data.” *Bioinformatics*, **19**(8), 973–80.
- DeRisi JL, Iyer VR, Brown PO (1997). “Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale.” *Science*, **278**(5338), 680–6.
- Dimitriadou E (2007). **cclust**: *Convex Clustering Methods and Clustering Indexes*. R package version 0.6-14, URL <http://CRAN.R-project.org/package=cclust>.

- Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A (2007). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.5-17, URL <http://CRAN.R-project.org/package=e1071>.
- Dopazo J, Carazo JM (1997). “Phylogenetic Reconstruction using a Growing Neural Network that Adopts the Topology of a Phylogenetic Tree.” *Journal of Molecular Evolution*, pp. 226–233.
- Dunn JC (1974). “Well Separated Clusters and Fuzzy Partitions.” *Journal on Cybernetics*, **4**, 95–104.
- Eisen MB, Spellman PT, Brown PO, Botstein D (1998). “Cluster Analysis and Display of Genome-Wide Expression Patterns.” *Proceedings of the National Academy of Sciences of the United States of America*, **95**(25), 14863–8.
- Fraley C, Raftery AE (2003). “Enhanced Model-based Clustering, Density Estimation, and Discriminant Analysis Software: MCLUST.” *Journal of Classification*, **20**(2), 263–286.
- Fraley C, Raftery AE (2001). “Model-based Clustering, Discriminant Analysis, and Density Estimation.” *Journal of the American Statistical Association*, **17**, 126–136.
- Fraley C, Raftery AE (2007). *mclust: Model-based Clustering / Normal Mixture Modeling*. R package version 3.1-2, URL <http://CRAN.R-project.org/package=mclust>.
- Fu L, Medico E (2007). “FLAME, a Novel Fuzzy Clustering Method for the Analysis of DNA Microarray Data.” *BMC Bioinformatics*, **8**, 3.
- Gat-Viks I, Sharan R, Shamir R (2003). “Scoring Clustering Solutions by their Biological Relevance.” *Bioinformatics*, **19**(18), 2381–9.
- Gentleman R (2007). *annotate: Using R Environments for Annotation*. R package version 1.16.1, URL <http://www.bioconductor.org/>.
- Gentleman R, Carey V, Morgan M, Falcon S (2007). *Biobase: Base functions for Bioconductor*. R package version 1.16.3, URL <http://www.bioconductor.org/>.
- Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Li FLC, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). “Bioconductor: Open Software Development for Computational Biology and Bioinformatics.” *Genome Biology*, **5**, R80. URL <http://genomebiology.com/2004/5/10/R80>.
- Gibbons FD, Roth FP (2002). “Judging the Quality of Gene Expression-based Clustering Methods using Gene Annotation.” *Genome Research*, **12**(10), 1574–81.
- Handl J, Knowles J, Kell DB (2005). “Computational Cluster Validation in Post-Genomic Data Analysis.” *Bioinformatics*, **21**(15), 3201–12.
- Hartigan JA, Wong MA (1979). “A K-means Clustering Algorithm.” *Applied Statistics*, **28**, 100–108.
- Hennig C (2007). *fpc: Fixed Point Clusters, Clusterwise Regression and Discriminant Plots*. R package version 1.2-3, URL <http://CRAN.R-project.org/package=fpc>.

- Herrero J, Valencia A, Dopazo J (2001). “A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns.” *Bioinformatics*, **17**(2), 126–36.
- Hornik K (2005). “A CLUE for CLUster Ensembles.” *Journal of Statistical Software*, **14**(12). URL <http://www.jstatsoft.org/v14/i12/>.
- Kapp A, Tibshirani R (2006). *clusterRepro: Reproducibility of Gene Expression Clusters*. R package version 0.5-1, URL <http://CRAN.R-project.org/package=clusterRepro>.
- Kaufman L, Rousseeuw PJ (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.
- Kerr MK, Churchill GA (2001). “Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions from Microarray Experiments.” *Proceedings of the National Academy of Sciences of the United States of America*, **98**(16), 8961–5.
- Kohonen T (1997). *Self-Organizing Maps*. Springer-Verlag, second edition.
- Liu TY, Lin C, Falcon S, Zhang J, MacDonald JW (2007a). *GO: A Data Package Containing Annotation Data for GO*. R package version 2.0.1, URL <http://www.bioconductor.org/>.
- Liu TY, Lin C, Falcon S, Zhang J, MacDonald JW (2007b). *hgu95av2: Affymetrix Human Genome U95 Set Annotation Data*. R package version 2.0.1, URL <http://www.bioconductor.org/>.
- Liu TY, Lin C, Falcon S, Zhang J, MacDonald JW (2007c). *moe430a: Affymetrix Mouse Expression Set 430 Annotation Data*. R package version 2.0.1, URL <http://www.bioconductor.org/>.
- MacDonald J, Ghosh D, Smolkin M (2007). *clusterStab: Compute Cluster Stability Scores for Microarray Data*. R package version 1.10.0, URL <http://CRAN.R-project.org/package=clusterStab>.
- McLachlan GJ, Bean RW, Peel D (2002). “A Mixture Model-based Approach to the Clustering of Microarray Expression Data.” *Bioinformatics*, **18**(3), 413–22.
- R Development Core Team (2007). *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>.
- Rousseeuw P, Struyf A, Hubert M, Maechler M (2007). *cluster: Cluster Analysis Extended Rousseeuw et al.* R package version 1.11.9, URL <http://CRAN.R-project.org/package=cluster>.
- Rousseeuw PJ (1987). “Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis.” *Journal of Computational and Applied Mathematics*, **20**, 53–65.
- Ruepp A, Zollner A, Maier D, Albermann K, Hani J, Mokejcs M, Tetko I, Guldener U, Mannhaupt G, Munsterkotter M, Mewes HW (2004). “The **FunCat**, a Functional Annotation Scheme for Systematic Classification of Proteins from Whole Genomes.” *Nucleic Acids Research*, **32**(18), 5539–45.

Walesiak M, Dudek A (2007). *clusterSim: Searching for Optimal Clustering Procedure for a Data Set*. R package version 0.33-1, URL <http://CRAN.R-project.org/package=clusterSim>.

Wehrens R (2007). *kohonen: Supervised and Unsupervised Self-Organising Maps*. R package version 2.0.2, URL <http://CRAN.R-project.org/package=kohonen>.

Yeung KY, Haynor DR, Ruzzo WL (2001). "Validating Clustering for Gene Expression Data." *Bioinformatics*, **17**(4), 309–18.

### **Affiliation:**

Guy Brock  
Department of Bioinformatics and Biostatistics  
School of Public Health and Information Sciences  
University of Louisville  
555 S Floyd St.  
Louisville, KY 40292, United States of America  
E-mail: [guy.brock@louisville.edu](mailto:guy.brock@louisville.edu)  
URL: <http://louisville.edu/~g0broc01/>