BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–3

Supplementary material to computational cluster validation in post-genomic data analysis

Julia Handl*, Joshua Knowles and Douglas B. Kell

School of Chemistry, University of Manchester, Faraday Building, Sackville Street PO Box 88, Manchester M60 1QD, UK

1 IMPLEMENTATION OF CLUSTERING ALGORITHMS

1.1 Preprocessing

All algorithms are run using Euclidean distance and no normalization.

1.2 k-means

Starting from a random partitioning, the k-means algorithm [8] repeatedly (i) computes the current cluster centres (that is, the average vector of each cluster in data space) and (ii) reassigns each data item to the cluster whose centre is closest to it. It terminates when no more reassignments take place. By this means, the intra-cluster variance, that is, the sum of squares of the differences between data items and their associated cluster centres is locally minimized. Our implementation of the k-means algorithm is based on the batch version of k-means, that is, cluster centres are recomputed only after the reassignment of all data items. As k-means can sometimes generate empty clusters, these are identified every iteration and are randomly re-initialized. This enforcement of the correct number of clusters can prevent convergence, and we therefore set the maximum number of iterations to 100. To reduce suboptimal solutions k-means is run repeatedly (100 times) using random initialization (which is known to be an effective initialization method [9]) and only the best result in terms of intra-cluster variance is returned.

1.3 Self-organizing maps

SOMs [7] are two-layered unsupervised neural networks that adapt a set of weight vectors to approximately model the input data. Each of these weight vectors is associated with one of the neurons in the neural layer, which are arranged in the form of a grid, and are randomly initialized.

During training, individual input vectors $\vec{x}$ are successively presented to the network, and, for each of these stimuli, the best matching unit (BMU) $\vec{y}$ (that is, the most similar weight vector) and its local neighbours are determined and updated according to the learning rule

$$\vec{w} \leftarrow \vec{w} + \epsilon \cdot h(\vec{y}, \sigma)(\vec{x} - \vec{w}).$$

Here, ϵ is the learning rate and $h(\vec{y}, \sigma)$ is a *neighbourhood function* (centred around the BMU $\vec{y}$) whose spread is determined by the parameter σ.

Our implementation of SOM is based on the guidelines given in the description of the SOM Toolbox [14]. The correct number of clusters k is used to set the grid resolution to $1 \times k$ rectangular grid cells; the weight vectors are uniformly randomly initialized. The SOM is trained in two training phases, a first 'coarse' approximation phase and a second fine-tuning phase. The first phase starts with a neighbourhood size of $\sigma_1^{start} = \max(1.0, \frac{1}{4}k)$, which is exponentially decreased to $\sigma_1^{end} = \max(1.0, \frac{1}{4}\sigma_1^{start})$. The learning rate during this phase is $\epsilon_1 = 0.5$. The second phase starts with the final neighbourhood size of the first phase, that is, $\sigma_2^{start} = \sigma_1^{end}$, and continues to decrease it to $\sigma_2^{end} = 1.0$. The learning rate in this second phase is $\epsilon_2 = 0.05$. The number of iterations for each phase are $it_1 = 10$ and $it_2 = 40$ respectively, and, in each iteration, all data items are presented to the SOM in random order. Finally, in the classification step all data items are assigned to the best matching output neuron. Each output neuron is interpreted as one cluster.

1.4 Self-organizing tree algorithm

The self-organizing tree algorithm (SOTA, [4]) is a divisive method that combines aspects of self-organizing maps and hierarchical algorithms. Starting from a single cell (cluster), SOTA iteratively divides 'leaf' cells (cells without descendants), thus giving rise to a binary tree structure. Each cell within this binary structure is associated with a weight vector. In each 'cycle', the next cell to be split is determined as the leaf cell with the largest intra-cluster variance.

The actual splitting operation is then realized as follows: The weight vectors associated with the two new 'sister' nodes are initialized to be the same as the weight vector of their parent node. All the input data associated with the parent node are then repeatedly presented to the two sister nodes. For a given input vector $\vec{x}$, the weight vector $\vec{w}$ of the best matching unit[1] (BMU) is updated as

$$\vec{w} \leftarrow \vec{w} + \epsilon_w(\vec{x} - \vec{w}),$$

the weight vector of the parent of the BMU is updated as

$$\vec{w} \leftarrow \vec{w} + \epsilon_a(\vec{x} - \vec{w}),$$

and the weight vector of the sister of the BMU is updated as

$$\vec{w} \leftarrow \vec{w} + \epsilon_s(\vec{x} - \vec{w}),$$

where $\epsilon_w = 0.01$, $\epsilon_a = 0.005$ and $\epsilon_s = 0.001$.

The training continues until convergence is reached (that is, the relative change in variance falls below a given threshold $E = 0.0001$), or a maximum number of 'epochs' $I_{max} = 1000$ is exceeded. In a last step, the input data associated with the parent node

*to whom correspondence should be addressed

[1] In the case of a tie between the two sister nodes, the left one is defined as the BMU.

is then presented to an entire subtree of the network (starting from $M = 5$ levels below the parent node) and associated with the best matching leaf cell.

In our experiments these cycles of iterative splitting are repeated until the desired number of clusters is reached.

1.5 Hierarchical clustering

As a fourth and fifth method, two agglomerative hierarchical clustering algorithms [15] are implemented. Both follow the same scheme, but employ different linkage metrics, namely average link and single link. In general, an agglomerative clustering algorithm starts with the finest partitioning possible (that is, singletons) and, in each iteration, merges the two least distant clusters. For the linkage metric of average link, the distance between two clusters C_i and C_j is computed as the average dissimilarity between all possible pairs of data elements i and j with $i \in C_i$ and $j \in C_j$. For the linkage metric of single link, the distance between two clusters C_i and C_j is computed as the smallest dissimilarity between all possible pairs of data elements i and j with $i \in C_i$ and $j \in C_j$. The algorithm terminates when the target number of clusters has been obtained.

2 DEFINITION OF VALIDITY MEASURES

2.1 F-measure

The F-measure [13] adopts the ideas of precision and recall from information retrieval. Each class t (inherent to the data) is regarded as the set of N_t items desired for a query; each cluster C_k (generated by the algorithm) is regarded as the set of N_k items retrieved for a query; N_{tk} gives the number of elements of class t within cluster C_k. For each class t and cluster C_k, precision and recall are then defined as $P(t, C_k) = \frac{N_{tk}}{N_k}$ and $R(t, C_k) = \frac{N_{tk}}{N_t}$, and the corresponding value under the F-measure is

$$F(t, C_k) = \frac{(b^2 + 1) \cdot P(t, C_k) \cdot R(t, C_k)}{b^2 \cdot P(t, C_k) + R(t, C_k)},$$

where equal weighting for $P(t, C_k)$ and $R(t, C_k)$ is obtained by setting $b = 1$. The overall F-measure value for the partitioning is computed as

$$F(C) = \sum_{t \in T} \frac{N_t}{N} \cdot \max_{C_k \in C} F(t, C_k).$$

It is limited to the interval $[0, 1]$ and should be maximized.

2.2 Adjusted Rand Index

The Rand Index is based on counting the number of pair-wise co-assignments of data items. Given the partitions U and V, the quantities a, b, c and d are computed for all possible pairs of data points i and j, and their respective cluster assignments $c_{U(i)}$, $c_{U(j)}$, $c_{V(i)}$ and $c_{V(j)}$, where

$$a = |\{i, j \mid c_{U(i)} = c_{U(j)} \wedge c_{V(i)} = c_{V(j)}\}|$$
$$b = |\{i, j \mid c_{U(i)} = c_{U(j)} \wedge c_{V(i)} \neq c_{V(j)}\}|$$
$$c = |\{i, j \mid c_{U(i)} \neq c_{U(j)} \wedge c_{V(i)} = c_{V(j)}\}|$$
$$d = |\{i, j \mid c_{U(i)} \neq c_{U(j)} \wedge c_{V(i)} \neq c_{V(j)}\}|$$

Hence, a and d keep track of correspondences between the two partitionings, whereas b and c count clear deviations. The Rand Index [10] is then defined as

$$R(U, V) = \frac{a + d}{a + b + c + d}.$$

Using a different representation based on the contingency table defined by U and V, the adjusted Rand Index [5, 16] is given as

$$R(U, V) = \frac{\sum_{lk} \binom{n_{lk}}{2} - \left[\sum_l \binom{n_{l\cdot}}{2} \cdot \sum_k \binom{n_{\cdot k}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_l \binom{n_{l\cdot}}{2} + \sum_k \binom{n_{\cdot k}}{2}\right] - \left[\sum_l \binom{n_{l\cdot}}{2} \cdot \sum_k \binom{n_{\cdot k}}{2}\right] / \binom{n}{2}},$$

where n_{lk} denotes the number of data items that have been assigned to both cluster l and cluster k.

Both the Rand Index and the adjusted Rand Index are limited to the interval $[0, 1]$ and should be maximized.

2.3 Minkowski Score

The Minkowski Score [6] computes the agreement between two partitionings U and V, based on their cophenetic matrices C_U and C_V. A cophenetic matrix describing a crisp partitioning is a binary matrix with $C_U(i, j) = 1$ if data items i and j are in the same cluster, and $C_U(i, j) = 1$ otherwise. The Minkowski Score is defined as

$$M(U, V) = \frac{||C_U - C_V||}{||C_U||}.$$

It is limited to the interval $[0, +\infty]$ and should be minimized.

2.4 Variance

Intra-cluster variance establishes the compactness of a given partitioning. Here, it is computed as the root mean square distance between data items and their corresponding cluster centre

$$V(C) = \sqrt{\frac{1}{N} \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \mu_k))},$$

where N is the size of the data set, C is the set of all clusters, μ_k is the centroid of cluster C_k and $\delta(.,.)$ is the distance function employed. Intra-cluster variance is limited to the interval $[0, +\infty]$ and should be minimized.

2.5 Connectivity

Connectivity [3] reflects the degree of connectedness of a cluster by evaluating the degree to which neighbouring data-points have been placed in the same cluster. It is computed as

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

where

$$x_{i, nn_{i(j)}} = \left\{ \begin{array}{ll} \frac{1}{j} & \text{if } \nexists C_k : \; i \in C_k \wedge nn_{i(j)} \in C_k \\ 0 & \text{otherwise.} \end{array} \right\}$$

Here, $nn_{i(j)}$ is the jth nearest neighbour of datum i, and L is a parameter determining the number of neighbours that contribute to the connectivity measure. Connectivity is limited to the interval $[0, +\infty]$ and should be minimized.

2.6 Silhouette Width

The Silhouette Width [11] for a partitioning is computed as the average Silhouette value over all data items. The Silhouette value for an individual data item i, which reflects the confidence in this particular cluster assignment, is computed as

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

where a_i denotes the average distance between i and all data items in the same cluster, and b_i denotes the average distance between i and all data items in the closest other cluster (which is defined as the one yielding the minimal b_i). The Silhouette Width is limited to the interval $[-1, 1]$ and should be maximized.

2.7 Dunn Index

The Dunn Index [1] measures the ratio between the smallest cluster distance and the largest intra-cluster distance in a partitioning. It is defined as

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

where $diam(C_m)$ is the maximum intra-cluster distance within cluster C_m and $dist(C_k, C_l)$ is the minimal distance between pairs of data items i and j with $i \in C_k$ and $j \in C_l$. The Dunn Index is limited to the interval $[0, +\infty]$ and should be maximized.

2.8 Stability

A standard implementation of stability is used. The data set is randomly split into two halves, and both halves are clustered. The partitioning of the first half is used to predict the partitioning of the second half by means of a nearest-neighbour classifier (a centroid-based classifier was also considered for the use with k-means, but yielded similar results): for each data item in the second half, its nearest neighbour in the first half is determined. It is then predicted to belong to the same cluster as this nearest neighbour. The predictive power (the agreement between the partitioning and the prediction on the second half) is computed as suggested by Tibshirani et al [12]:

$$P(C) = \min_{C_k \in C} \frac{a}{a + b},$$

where a and b are computed as defined in Section 2.2 above. The entire process is repeated 20 times, and the average predictive power obtained is returned. Stability is limited to the interval $[0, 1]$ and should be maximized.

3 ENLARGED GRAPHS IN COLOUR FROM THE PAPER

The following pages provide colour versions of Figure 5, Figure 7, Figure 8, Figure 9, Figure 10 and Figure 11 from the paper.

REFERENCES

[1]J. C. Dunn. Well separated clusters and fuzzy partitions. *Journal on Cybernetics*, 4:95–104, 1974.

[2]C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature*, pages 584–593. Springer-Verlag, 1996.

[3]J. Handl and J. Knowles. Exploiting the trade-off — the benefits of multiple objectives in data clustering. In *Proceedings of the Third International Conference on Evolutionary Multicriterion Optimization*, pages 547–560. Springer-Verlag, 2005.

[4]J. Herrero, A. Valencia, and J. Dopazo. A hierarchical unsupervised growing neural network for clustering gene expression data. *Bioinformatics*, 17:126–136, 2001.

[5]A. Hubert. Comparing partitions. *Journal of Classification*, 2:193–198, 1985.

[6]N. Jardine and R. Sibson. *Mathematical Taxonomy*. John Wiley and Sons, 1971.

[7]T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 2001.

[8]L. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

[9]J. M. Pena, J. A. Lozana, and P. Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20:1027–1040, 1999.

[10]W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[11]P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[12]R. Tibshirani, G. Walther, D. Botstein, and P. Brown. Cluster validation by prediction strength. Technical report, Department of Statistics, Stanford University, 2001. Available at http://www-stat.stanford.edu/ tibs/ftp/predstr.ps.

[13]C. van Rijsbergen. *Information retrieval, second edition*. Butterworths, 1979.

[14]J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parkankangas. SOM Toolbox for Matlab 5. Technical Report A57, Neural Networks Research Centre, Helsinki University of Technology, Espoo, Finland, April 2000. Available at http://www.cis.hut.fi/projects/somtoolbox/package/papers/techrep.pdf.

[15]E. Vorhees. *The effectiveness and efficiency of agglomerative hierarchical clustering in document retrieval*. PhD thesis, Department of Computer Science, Cornell University, 1985.

[16]K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.

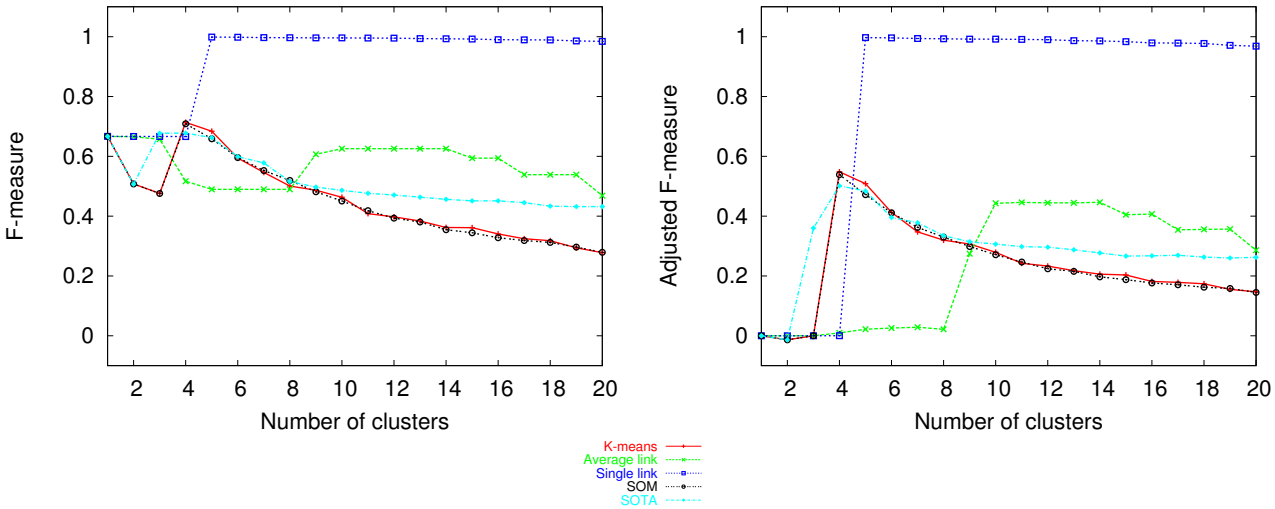

Fig. 6. Illustration of the biases of external validation measures. Shown are the results for k-means, SOM, SOTA, average link and single link on the Long data set under (left) the F-measure, and (right) the adjusted F-measure (averages over 21 runs). The original F-measure values indicate that both single and average link clearly outperform k-means, SOM and SOTA for $k = 2$, by a margin of up to 0.2. However, this conceals the fact that, for this cluster number, all five algorithms have equally failed to identify the correct cluster structure on Long. While k-means, SOM and SOTA have split both clusters in the middle (minimizing variance), both agglomerative clustering algorithms have isolated outliers in one cluster and merged the bulk of the data in the second cluster. Only for $k \geq 5$ does single link succeed in separating the two core clusters. However, the poor performance of all five algorithms for $k = 2$ becomes evident in the plot of the adjusted F-measure. In general, the normalization may not only correct the estimated absolute degree of quality, but may also correct the ordering between the solutions obtained for different numbers of clusters (for example, average link for $k = 2$ and $k = 10$) or for different algorithms (for example, average link and k-means for $k = 9$).

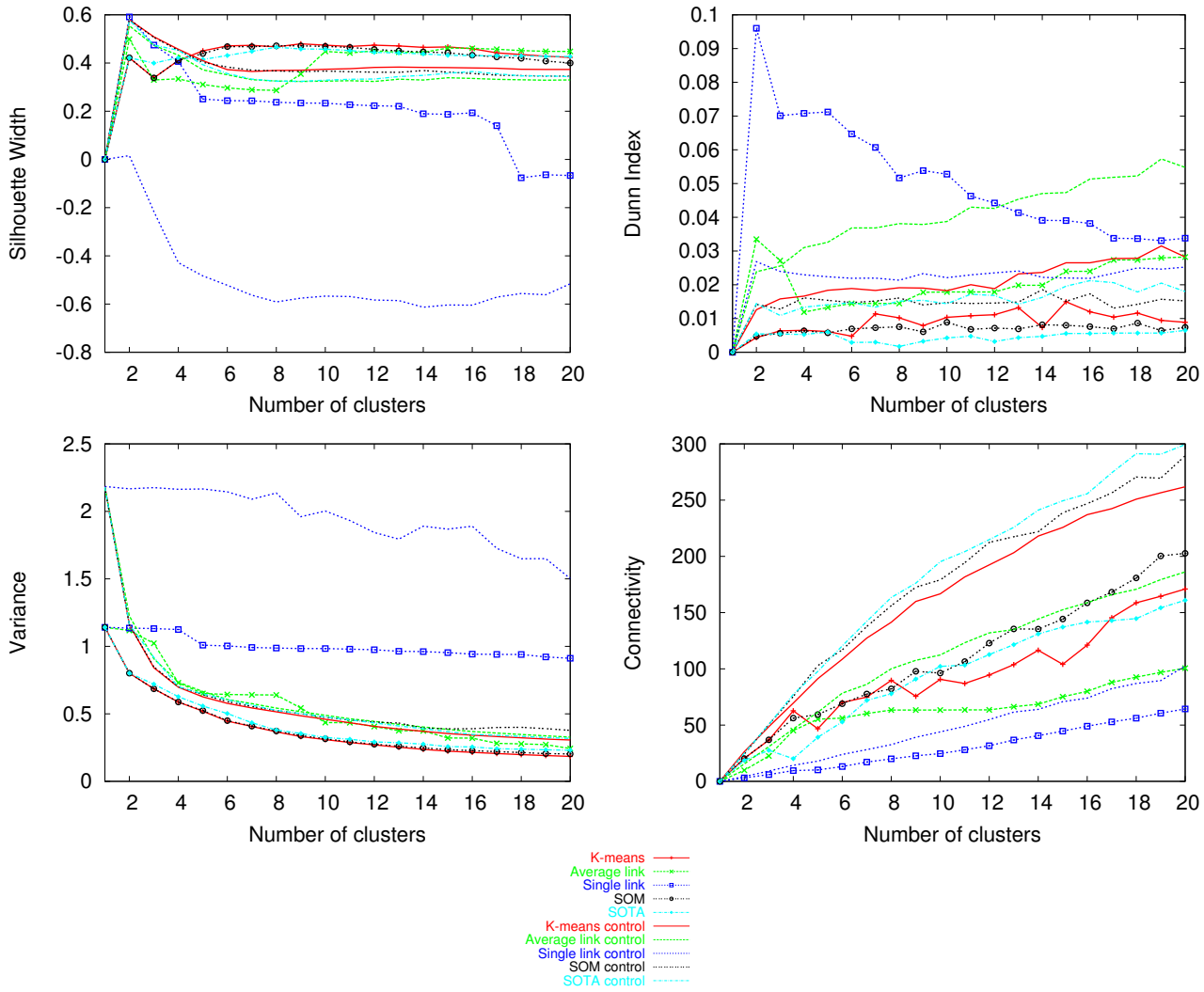

Fig. 8. Illustration of the biases of internal validation measures. A simple null model is used: the random control data have been generated from a uniform random distribution within the bounds of the original data. Shown are the results for k-means, SOM, SOTA, average link and single link on the Long data set under (top left) the Silhouette Width, (top right) the Dunn Index, (bottom left) the variance, and (bottom right) the connectivity measure (averages over 21 runs) on the original data and uniformly random control data. The performance curves obtained for both original and uniformly random control data are plotted to permit visual comparison. Considering the Silhouette Widths only, the plot for the original data seems to indicate that $k = 2$ yields a good partitioning for all five algorithms, with the agglomerative algorithms being the best performers. Moreover, single link is assessed to perform worse than both other methods for all $k \geq 5$, a result that stands in obvious contrast to its true performance as verified by the adjusted F-measure. Only a comparison to the values obtained for random control data reveals that, for $k = 2$ and $k = 3$, k-means, SOM, SOTA and average link do in fact perform worse than for random data. In comparison to its performance on random control data, single link seems to perform well, but it is not possible to derive the correct number of clusters from the plot (there is no peak at $k = 5$). Interpretation of the other three measures is given in the text.

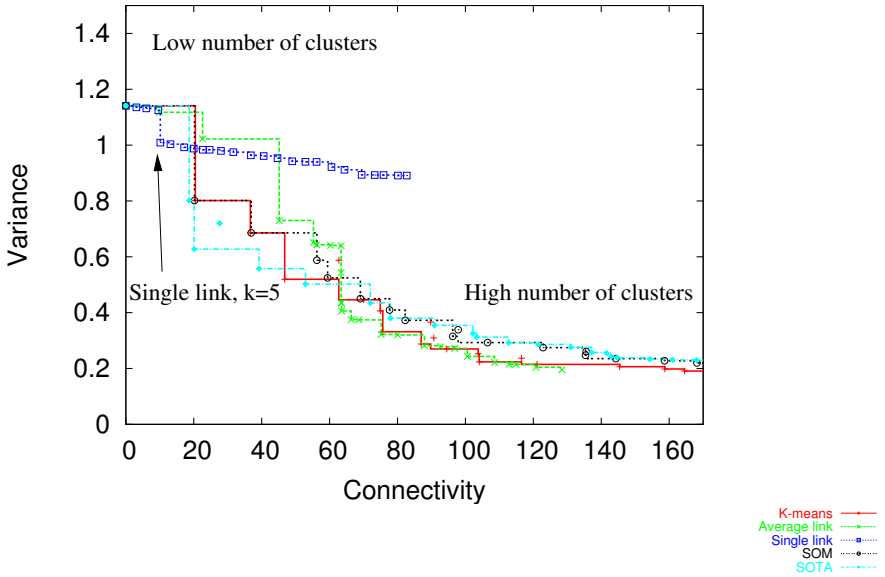

Fig. 9. Illustration of the solution visualization in two-objective space. Shown are the solutions (averages over 21 runs) for k-means, SOM, SOTA, average link and single link on the Long data set in a plot of connectivity versus variance (both to be minimized). The set of solutions returned by each clustering algorithm is summarized by an attainment surface [2], which is the boundary in objective space that separates the region dominated by the attained solutions from the region that is not dominated. Due to the trends of the two objectives (variance decreases for a higher number of clusters, while connectivity increases), the number of clusters in the solutions generally increases from the top left to the bottom right. The correct number of clusters for a given algorithm is expected to show as a strong 'knee' in its attainment front. This is because for the correct number of clusters we expect a relatively large drop in variance at little additional cost in connectivity. In the above plot, a clear 'knee' can be observed for single link at $k = 5$.

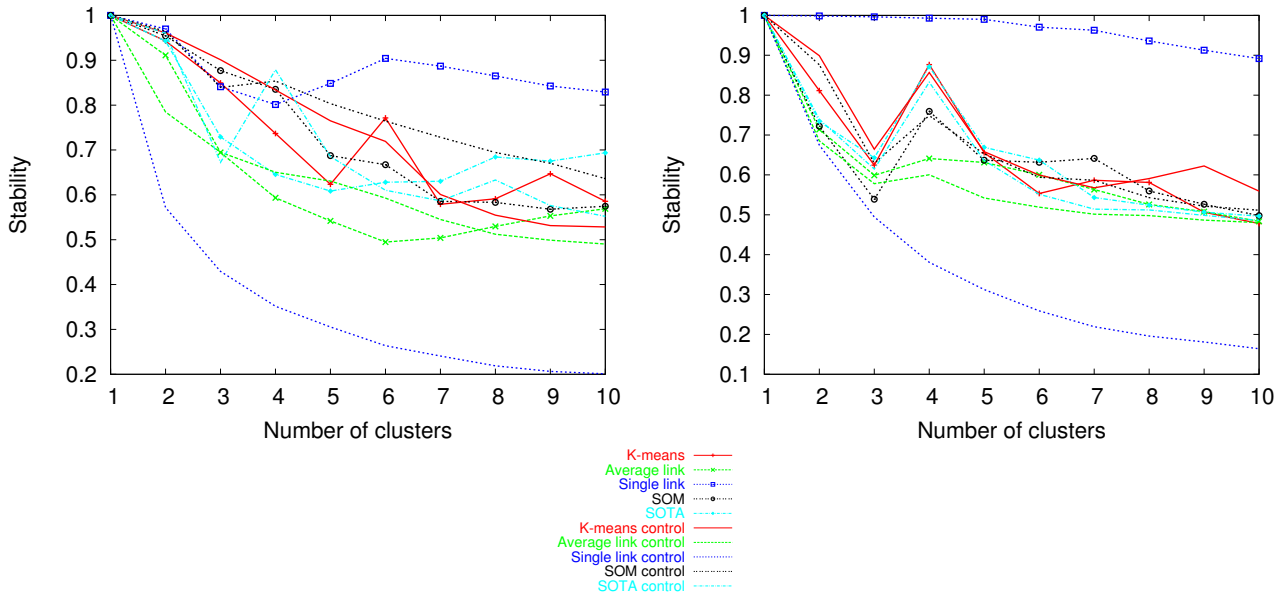

Fig. 10. Stability-based analysis (averages over 21 runs) for (left) the Long and (right) the Square data set. On Long, the stability-based method is evidently able to estimate the presence of a good single link clustering solution for $k \geq 5$ (the stability plot peaks at $k = 6$), and the absence of good average link solutions (no significant peak in the stability plot). Yet, the results obtained for k-means, SOM and SOTA are disconcerting. For k-means, the stability-based analysis pinpoints a highly stable six-cluster solution. Further analysis shows that this solution is stable only due to k-means' tendency to converge to spherically shaped clusters, and is in fact highly sub-optimal. For SOM and SOTA, the stability-based analysis pinpoints a stable four-cluster solution for the random control data. For Square, where k-means, SOM, SOTA and average link perform comparably well (results not shown), the stability-based analysis correctly identifies the four-cluster solution for average link, k-means, SOM and SOTA (the stability plots for all four algorithms peak at $k = 4$). However, a comparison to the control curves reveals that, for k-means, SOM and SOTA, a four cluster solution is recommended with comparable confidence for uniformly random data. This shows that the conspicuous stability peak obtained for k-means, SOM and SOTA on the Square data is mainly an artifact of the square shape of the underlying data manifold.

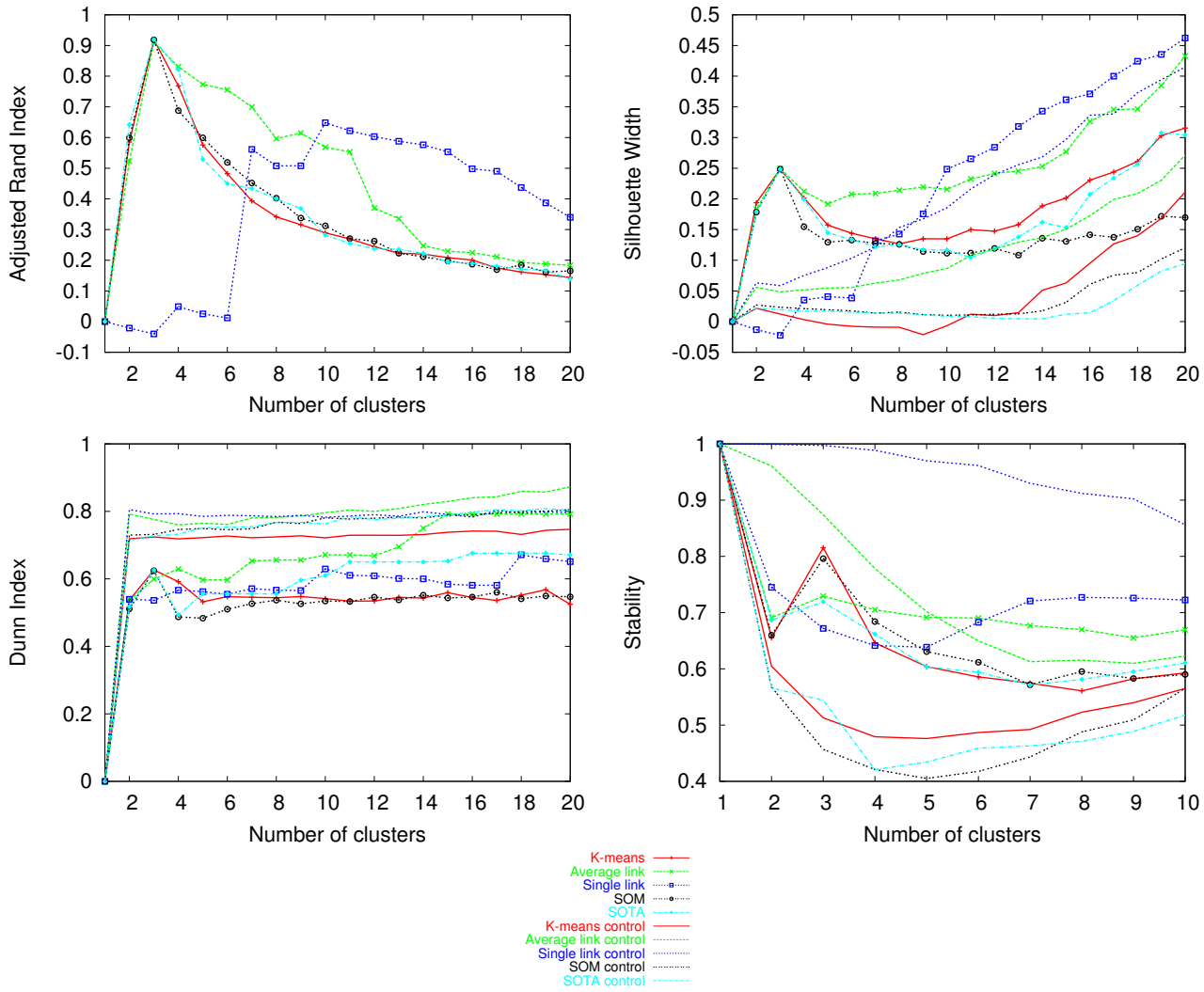

Fig. 11. Adjusted Rand Index, Silhouette Width, Dunn Index and stability (averages over 21 runs) for k-means, SOM, SOTA, average link and single link agglomerative clustering on the Leukemia test set. The evaluation under the adjusted Rand Index (comparing to the known class labels) shows that average link, k-means, SOM and SOTA perform robustly on this data. They identify the three main clusters (AML, B-lineage ALL and T-lineage ALL), and assign most of the samples correctly. Naturally, this is knowledge that would not be available in a real-life cluster analysis, and it is therefore interesting to see whether the results under the internal validation measures would have lead to the same conclusion. The performance curves under the Silhouette Width clearly indicates the high quality of the three-cluster solution. This result is best seen when comparing to the results obtained under the null model and 'removing' the bias towards large numbers of clusters, which arises due to the small size of the data set: for large numbers of clusters, the resulting partitionings contain many singleton clusters, which score highly under the Silhouette Width. The stability-based technique is less consistent: for k-means and SOM, the performance peak at $k = 3$ is well pronounced, but it is much weaker for SOTA and average link. Both the Silhouette Width and the stability-based method indicate the lack of structure in the single link solutions. The application of the Dunn Index is somewhat less successful: it fails to predict the insufficiency of single link, and it mis-estimates the number of clusters for average link.

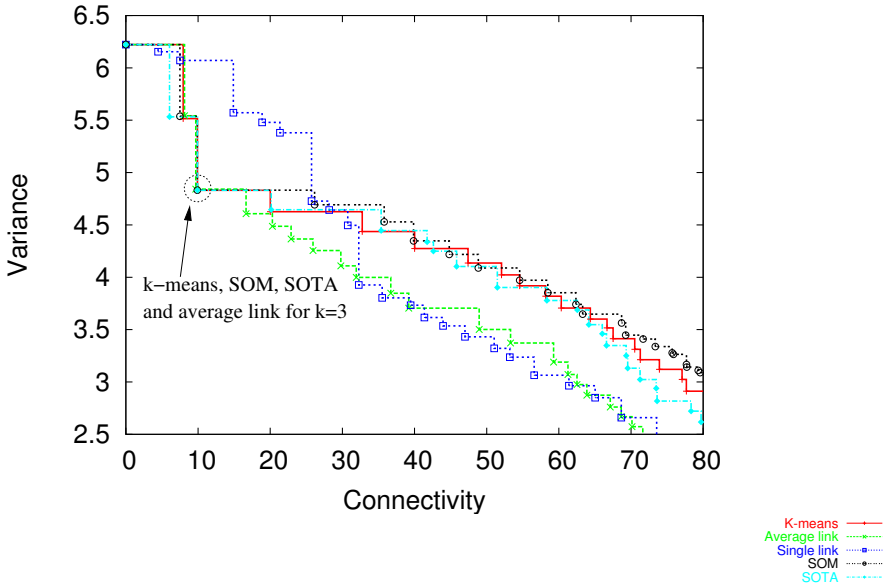

Fig. 12. Solution visualization in two-objective space. Shown are the solutions (averages over 21 runs) for k-means, SOM, SOTA, average link and single link on the Leukemia data set in a plot of connectivity versus variance. The knee corresponding to the three-cluster solution is clearly pronounced. The visualization also shows the consistency between the k-means, SOM, SOTA and average link solutions for $k = 2$ and $k = 3$, which further increases the confidence in the correctness of these partitionings.