The Stochastic QT–Clust Algorithm: Evaluation of Stability and Variance on Time–Course Microarray Data*

Theresa Scharl[1] and Friedrich Leisch[2]

[1] Department of Statistics and Probability Theory, Vienna University of Technology, Wiedner Hauptstraße 8-10/1071, A-1040 Wien, Austria; Theresa.Scharl@ci.tuwien.ac.at

[2] Department of Statistics, University of Munich, Ludwigstraße 33, D-80539 München, Germany; Friedrich.Leisch@stat.uni-muenchen.de

Summary. Clustering time–course microarray data is an important tool to find co–regulated genes and groups of genes with similar temporal or spacial expression patterns. Depending on the distance measure and cluster algorithm used different kinds of clusters will be found. In a simulation study on a publicly available dataset from yeast the quality–based cluster algorithm QT–Clust is compared to a stochastic variant of the original algorithm using different distance measures. For that purpose the stability and sum of within cluster distances are evaluated. Finally stochastic QT–Clust is compared to the well–known k–means algorithm.

Key words: cluster analysis, time–course microarray data, distance measures, R

1 Introduction

The interpretation of enormous amounts of data from microarrays has been a challenging task in statistics and bioinformatics for the past few years. Time–course microarray experiments make it possible to look at the gene expression of thousands of genes at several time points simultaneously. Genes with similar expression pattern are co–expressed genes which are likely to be co–regulated. Hence clustering gene expression patterns may help to find groups of co–regulated genes or to identify common temporal or spatial expression patterns. Finally cluster results can suggest functional pathways and interaction between genes.

In this simulation study the quality-based cluster algorithm QT–Clust [HKY99] for the maximum radius problem is compared to its stochastic variant using dif-

* This paper was published as: Theresa Scharl and Friedrich Leisch. The Stochastic QT–Clust Algorithm: Evaluation of Stability and Variance on Time–Course Microarray Data. In Alfredo Rizzi and Maurizio Vichi, editors, Compstat 2006—Proceedings in Computational Statistics, pages 1015–1022. Physica Verlag, Heidelberg, Germany, 2006.

ferent distance measures. For that purpose two evaluation criteria are chosen. To investigate the stability of a partition pairwise comparisons of cluster results are conducted. As a measure of the quality of a partition the sum of within cluster distances is observed.

In a final step stochastic QT–Clust is compared to the well–known k–means algorithm. Five distance measures which are commonly used in the context of clustering time–course microarray data were chosen for the comparison: Euclidean, Manhattan and Maximum distance and two correlation–based distances "1-correlation" and "1-Jackknife correlation". All algorithms used are implemented in R (**http://www.r-project.org**) package `flexclust` [Lei06] available from CRAN (**http://cran.r-project.org**). `flexclust` is a flexible toolbox for clustering which allows to try out various distance measures with only minimal programming effort.

A publicly available dataset from yeast was used in this simulation study, the seventeen time point mitotic cell cycle data [CCW+98] available at **http://genome-www.stanford.edu**. This dataset was preprocessed adapting the instructions given by [HKY99]. First the outlier time points 10 and 11 were removed from the original 17 variables. Then genes that were either expressed at very low levels or did not vary significantly over the time points were removed. This procedure yields gene expression data on $G = 3722$ genes (observations) for $T = 15$ time points (variables).

2 Methods

2.1 Cluster algorithms

In this study three algorithms are compared. One is the well–known k-means algorithm, the second one is QT–Clust [HKY99] which is a quality-based cluster algorithm specially designed to identify co–expressed genes and the third is stochastic QT–Clust. For stochastic QT–Clust [Lei06] one has to define the quality of clusters which is given by the maximum diameter of the clusters and the minimal number of points that form a single cluster. This means that the number of clusters can only be controlled indirectly through these two parameters. Additionally the number `ntry` of times steps 1 to 4 should be run in each step has to be defined.

1. Start with a randomly chosen center.
2. Iteratively add the gene that minimizes the increase in cluster diameter.
3. Continue until no gene can be added without surpassing the diameter threshold.
4. Repeat from 1. for `ntry` $-$ 1 further centers.
5. Select the largest candidate cluster and remove the genes it contains from further consideration.
6. Goto 1. on the smaller dataset.
7. Stop when the largest remaining cluster has fewer than some prespecified number of elements.

Using QT–Clust outliers will not be added to any cluster as opposed to k-means where every gene will be part of a cluster. If `ntry` is equal to the number of genes G the original QT–Clust algorithm [HKY99] is obtained. Stochastic QT–Clust speeds up the procedure and we get different local maxima of the objective function. The original algorithm will always converge in the same local optimum. In the following the term *QT–Clust* always refers to stochastic QT–Clust.

2.2 Distance measures

Five distance measures were chosen which are commonly used in the context of clustering time–course microarray data (see for example [SMS+05] or [CHT03]). Three geometric distances (Euclidean distance, Manhattan distance, and Maximum distance) and two *correlation*–based distance measures ("1–correlation" and "1–Jackknife correlation") are used. The Jackknife correlation was introduced by [HKY99]

$$d_{xy} = 1 - \min(\rho_{xy}^{(1)}, \rho_{xy}^{(2)}, \ldots, \rho_{xy}^{(T)})$$

where $\rho_{xy}^{(t)}$ is the correlation of pair x,y computed with the tth time point deleted. x and y are T–dimensional vectors where T is the number of time points in the experiment. This distance measure can handle one outlier time point which is very helpful in the context of microarray experiments because technical problems can easily distort the data.

2.3 Implementation

R package `flexclust` contains extensible implementations of the generalized k–means and QT–Clust algorithm in order to make it easy for data analysts to try out a variety of distance measures. Function `kcca` uses a family concept similar to the implementation of generalized linear models in S [CH92]. A KCCA family consists of the following two parts:

dist : A function taking N observations and K centroids as inputs and returning the NxK matrix of distances between all observations and centroids.

cent : An (optional) function computing the centroid for a given subset of the observations.

An example for a new distance measure is given by

```
distJackknife <- function(x, centers)
{
   m = array(dim=c(ncol(centers),nrow(x),nrow(centers)))
   for(i in 1:ncol(centers)){
     m[i,,] = distCor(x[,-i,drop=FALSE],centers[,-i,drop=FALSE])
   }
   apply(m,2:3,min)
}
```

For Jackknife Correlation a closed form for centroid computation does not exist up to our knowledge. If `cent` is not specified a general purpose optimizer is used (at some speed and precision penalty). The canonical centroids cluster–wise means and cluster–wise medians are used for Euclidean distance and for Manhattan distance. The KCCA family objects are also used for function `qtclust`.

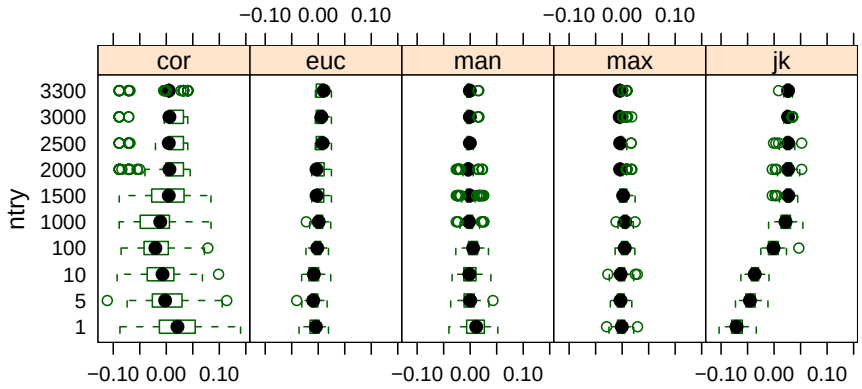

Fig. 1. Sum of within cluster distances of QT–Clust for increasing values of the hyper parameter `ntry` after rescaling.

3 Simulation

3.1 Evaluation of stability and variance of QT–Clust

First of all the sum of within cluster distances and stability of the stochastic variant of QT–Clust are investigated in order to compare it to the original algorithm. The sum of within cluster distances W is given by

$$W = \sum_{j=1}^{k} \sum_{x \in X_j} d(x, c_j)$$

where x is an element of cluster X_j, d is the distance measure and c_j is the center of cluster j. Now 100 replicates of QT–Clust each are computed for increasing values of the hyper parameter `ntry` between 1 and 3300. For `ntry` equal to the number of genes the algorithm is deterministic and equivalent to the original algorithm. Figure 1 shows boxplots of the sum of within cluster distances for all five distance measures after rescaling. On this dataset the parameter `ntry` has major impact on the quality of the partition. Even though on this dataset the variability is higher for small values of `ntry` this may lead to better results. Smaller values of the sum of within cluster distances can be obtained using small values of `ntry`. Additionally the different distance measures show different patterns and all suggest different values of `ntry`. Values between 1 and 100 always lead to to better results than the original algorithm on this dataset and therefore further simulations on different datasets need to be conducted. Figure 2 shows boxplots of all consecutive pairwise comparisons of cluster results for 100 replicates using the adjusted Rand index [HA85]. A Rand index of 1 corresponds to identical partitions and a Rand index of 0 corresponds to agreement by chance given cluster size. The stability of QT–Clust increases for

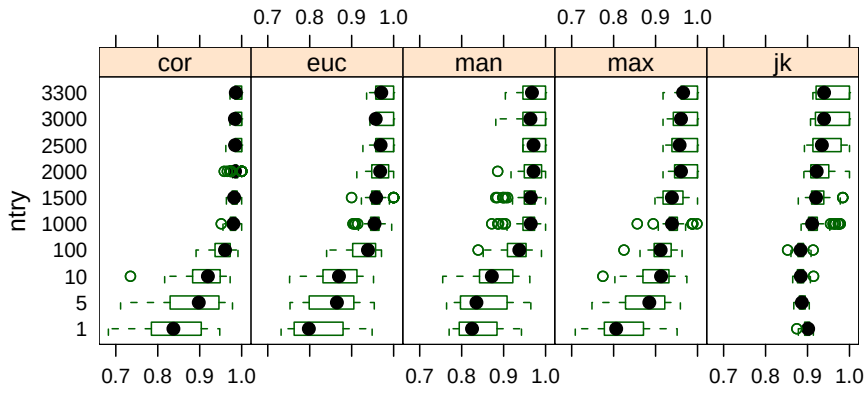

Fig. 2. Stability of the stochastic approximation of QT–Clust for increasing values of the hyper parameter `ntry`. Pairwise comparison of the results using boxplots of the Rand indices.

increasing values of `ntry` except for "1-Jackknife correlation". This distance measure is different to all others as it allows single outliers. It would be interesting to see if Jackknife versions of Euclidean, Manhattan and Maximum distance show similar characteristics. So if one is interested in clusters with small sum of within cluster distances stochastic QT–Clust performs better than the original algorithm on this dataset. If reproducibility is important then the deterministic algorithm is preferable.

As there is no general "best" value for `ntry` different numbers have to be tested for each dataset and each distance measure in order to find a suitable value. Now the simulations for `ntry` = 5, 100, 1000 and 3300 were compared for all distances. Figure 3 shows boxplots of the stability within a single value of `ntry` and between results for different values of `ntry`. "1–Jackknife correlation" reveals fewer differences than the other distance measures. This may indicate that the instability originates from outliers in single coordinates. For all further simulations on this dataset `ntry` = 5 is used to obtain cluster results with small sum of within cluster distances. Additionally the use of smaller values of `ntry` speeds up the procedure.

3.2 Comparison of distances

Now the different distance measures are compared using k–means and QT–Clust. The goal is to investigate the influence of distance measures and cluster algorithms used to be able to answer biological questions more precisely. In order to evaluate the effects of different distance measures for k–means the following procedure is used

1. draw 100 bootstrap samples from the original data,
2. cluster them into 50 clusters using each of the five distance measures, and
3. compare the obtained results using the adjusted Rand index.

Restarting k-means several times with different initializations for each data set and keeping only the best solution did not make much of a difference, probably due to the

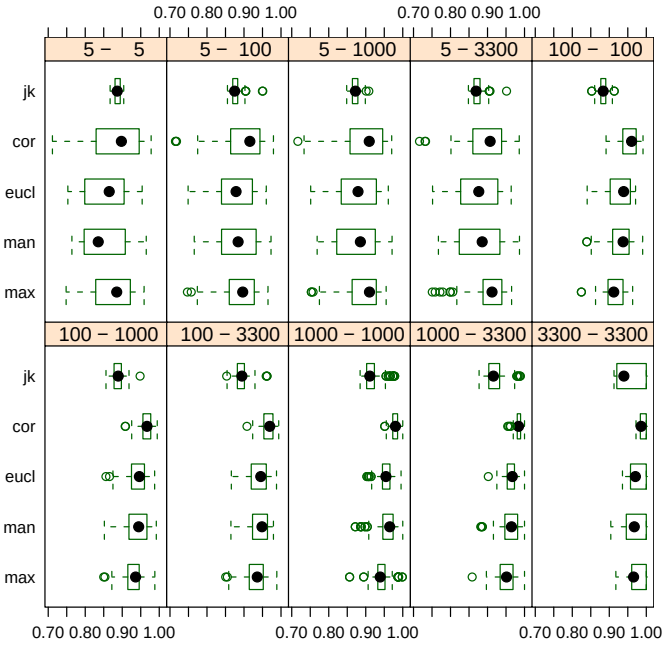

Fig. 3. Rand index for pairwise comparison of `ntry` = 5, 100, 1000 and 3300 on 100 replicates within and between values of `ntry`.

relatively large number of cluster centers used. The number of clusters for k–means is chosen arbitrary but it was tried to find an appropriate radius for QT–Clust to get similar numbers of clusters. For QT–Clust 100 replicates of the algorithm were computed for each distance measure on the original data because the algorithm has no prediction step.

First the differences between distance measures are explored within cluster algorithm (see Figure 4). On this dataset it was found that for QT–Clust "1-correlation" and "1-Jackknife correlation" are most similar and disagree most with Maximum distance. For k–means the agreement between distances is much higher than for QT–Clust and the two correlation–based distances also provide the highest similarity.

Subsequently it was examined for which distance measure the two algorithms differ most and it was found that the results on this dataset vary most for Maximum distance and agree best for "1-correlation" and "1-Jackknife correlation" distance (see Figure 5).

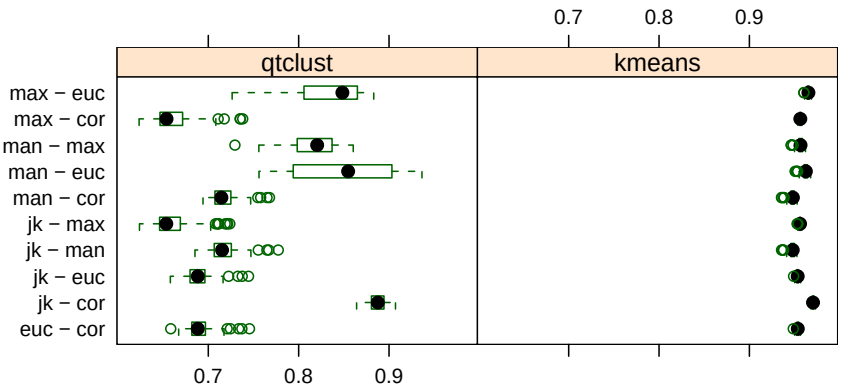

Fig. 4. Rand index for pairwise comparison of 100 replicates of QT–Clust (left panel) and 100 bootstrap samples of k–means (right panel) using different distance measures.

4 Summary and outlook

The deterministic QT–Clust algorithm has been compared to the stochastic QT–Clust algorithm on a microarray dataset from yeast using different distance measures. For that purpose the stability and sum of within cluster distances were evaluated for increasing values of the hyper parameter `ntry`. On this dataset it was found that stochastic QT–Clust leads to better results if one is interested in a small sum of within cluster distances. If stability and reproducibility are more important then the original algorithm is preferable even though it results only in a local optimum. Finally the results of stochastic QT–Clust were compared to k–means. As expected the different methods used have major impact on the resulting clusters of this dataset. The method of choice depends on the data and on the biological questions asked. In any case we found that the influence of distance measure is much higher for QT–Clust than for k–means. There are still open questions for the future like the characteristics of Jackknife versions of Euclidean, Manhattan and Maximum distance. Furthermore it would be interesting to see how the results change when the dataset changes slightly. Therefore further simulations on different datasets will be conducted. So far we already did the same simulations on a dataset from E.Coli and found similar results.

Acknowledgement. This work was supported by the Austrian K_{ind}/K_{net} Center of Biopharmaceutical Technology (ACBT) and the Austrian Science Foundation (FWF) under grant P17382. Some of the simulation runs were done by Gregor Siedler.

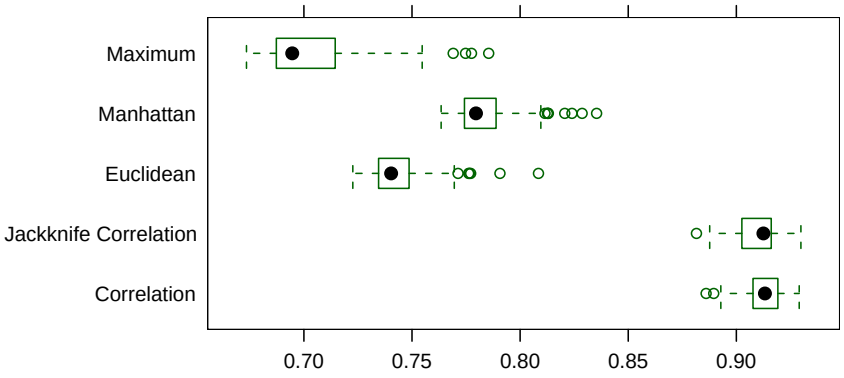

Fig. 5. Rand index for pairwise comparison between k–means and QT–Clust of 100 cluster results using different distance measures.

References

[CCW+98] Raymond J. Cho, Michael J. Campbell, Elizabeth A. Winzeler, Lars Steinmetz, Andrew Conway, Lisa Wodicka, Tyra G. Wolfsberg, Andrei E. Gabrielian, David Landsman, David J. Lockhart, and Ronald W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, 1998.

[CH92] J.M. Chambers and T.J. Hastie. *Programming with data: A guide to the S language*. Springer Verlag, Berlin, Germany, 1992.

[CHT03] Hugh Chipman, Trevor J. Hastie, and Robert Tibshirani. Clustering microarray data. In Terence P. Speed, editor, *Statistical Analysis of Gene Expression Microarray Data*, chapter 4, pages 159–200. Chapman & Hall/CRC Press, 2003.

[HA85] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[HKY99] Laurie J. Heyer, Semyon Kruglyak, and Shibu Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[Lei06] Friedrich Leisch. A toolbox for k-centroids cluster analysis. *Computational Statistics and Data Analysis*, 2006. Accepted for publication.

[SMS+05] Qizheng Sheng, Yves Moreau, Frank De Smet, Kathleen Marchal, and Bart De Moor. Advances in cluster analysis of microarray data. In Francisco Azuaje and Joaquin Dopazo, editors, *Data Analysis and Visualization in Genomics and Proteomics*. John Wiley & Sons, Ltd, 2005. ISBN 0-470-09439-7.