LMU
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Theresa Scharl, Bettina Grün & Friedrich Leisch

Mixtures of Regression Models for Time-Course Gene Expression Data: Evaluation of Initialization and Random Effects

Technical Report Number 071, 2009
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Mixtures of Regression Models for Time-Course Gene Expression Data: Evaluation of Initialization and Random Effects

Theresa Scharl
Vienna University of Technology, Austria
University of Natural Resources and Applied Life Sciences, Vienna, Austria

Bettina Grün
Wirtschaftsuniversität Wien, Austria

Friedrich Leisch
Ludwig-Maximilians-University, Munich, Germany

This is a preprint of an article that has been accepted for publication in *Bioinformatics* on 07-Dec-2009. Please use the journal version for citation.

Abstract

Summary: Finite mixture models are routinely applied to time course microarray data. Due to the complexity and size of this type of data the choice of good starting values plays an important role. So far initialization strategies have only been investigated for data from a mixture of multivariate normal distributions. In this work several initialization procedures are evaluated for mixtures of regression models with and without random effects in an extensive simulation study on different artificial datasets. Finally these procedures are also applied to a real dataset from *E. coli.*
Availability: The latest release versions of R packages flexmix, gcExplorer and kernlab are always available from CRAN (http://cran.r-project.org/).

1 Introduction

Finite mixtures of regression models are the state-of-the-art technique for modeling time course microarray data. The Expectation-Maximization (EM) algorithm (Dempster *et al.*, 1977) is the most common method for maximum likelihood (ML) estimation despite its drawbacks such as convergence only to a local optimum in dependence of the initialization. Good starting values are therefore crucial for the EM algorithm to perform well. A common strategy is to use random initialization and to run the algorithm several times in order to overcome this convergence to local optima already determined by the initialization.

Up to our knowledge different initialization strategies have only been investigated for mixtures of multivariate normal distributions in a model–based clustering setting. In this study the performance of the initialization strategies proposed for this setting is investigated for mixtures of regression models with respect to time course microarray data.

Biernacki *et al.* (2003) give an overview of simple initialization strategies including random initialization, classification EM (CEM) algorithms, stochastic EM (SEM) algorithms and preliminary short runs of EM itself. Their aim is to identify a simple method that gives the highest likelihood in a fixed number of iterations.

Wehrens *et al.* (2004) present an approach for large datasets where a random sub-sample is clustered prior to applying the model to the whole dataset (sampling method). A modification of this method is given in Fraley *et al.* (2005). They propose incremental model-based clustering for large datasets with small clusters and apply it to image data (incremental method). The situation of large datasets with small clusters is also characteristic of microarray data.

A completely different approach is given by spectral clustering (e.g., Ng *et al.*, 2001) which does not make any assumptions on the form of the clusters. The cluster solution can be used as a starting value for EM.

In this paper several initialization strategies are investigated in an extensive simulation study on different artificial time course datasets, i.e., random initialization, classification EM, stochastic EM, short runs of EM itself, the sampling and the incremental method as well as spectral clustering. The aim of this study is to find good initialization strategies for clusterwise regression and to evaluate the differences between mixtures of linear models and mixtures of linear mixed models. Finally these procedures are also applied to a real dataset from *E. coli.*

2 Methods

2.1 Model specification

The mixture density h of a finite mixture model with K components is given by

$$h(y|x, \psi) = \sum_{k=1}^{K} \pi_k f(y|x, \theta_k).$$

y is the response, x are the predictors and ψ denotes the vector of all parameters for the mixture density h. For the component weights π_k it holds that $\pi_k > 0$ for all k and $\sum_{k=1}^{K} \pi_k = 1$. θ_k is the component-specific parameter vector for the component-specific density function f.

Mixtures of mixed-effects models (e.g., Celeux *et al.*, 2005 or Ng *et al.*, 2006) are used to account for different kinds of heterogeneity between individuals. The components of the mixture correspond to different groups with distinct parameterizations while the random effects allow for individual differences which cluster around a common mean value.

The data of each individual i is given by (Y_i, X_i, Z_i) which consists of n_i observations of the dependent variables $Y_i = (y_{ij})_{j=1,\dots,n_i}$, the covariates for the fixed effects $X_i = (x'_{ij})_{j=1,\dots,n_i}$ and the covariates for the random effects $Z_i = (z'_{ij})_{j=1,\dots,n_i}$. The finite mixture density of mixed effects models with K components is given for the observations of individual i by

$$h(Y_i|X_i, Z_i, \psi)$$

$$= \textstyle\sum_{k=1}^{K} \pi_k \int \prod_{j=1}^{n_i} \phi_1(y_{ij}; x'_{ij}\beta_k + z'_{ij}b_i^k, \sigma_k^2)\phi_q(b_i^k; 0, \Psi_k)db_i^k$$

$$= \textstyle\sum_{k=1}^{K} \pi_k \phi_{n_i}(Y_i; X_i\beta_k, Z_i\Psi_k Z'_i + \sigma_k^2 I_{n_i}).$$

$\phi_d(.; \mu, \Sigma)$ denotes the d-dimensional multivariate normal distribution with mean μ and variance-covariance matrix Σ.

- Fixed effects: $x'_{ij}\beta_k$ with β_k deterministic.

- Random effects: $z'_{ij}b^k_i$ with $b^k_i \sim N(0, \Psi_k)$.

Splines are frequently included in mixtures of mixed-effects models to treat the gene expression level as a continuous function of time without requiring the specification of the functional relationship. These are used with B-splines (Luan and Li, 2003; Bar-Joseph *et al.*, 2003) and smoothing splines (Ma *et al.*, 2006). For smoothing splines the degree of smoothness is chosen automatically by cross-validation.

In this study smoothing splines and B-splines are used to fit finite mixtures of linear regression models to time course gene expression data. Mixtures of linear models in combination with smoothing splines are compared to mixtures of linear mixed models with B-splines using various initialization strategies.

2.2 Parameter Estimation

The EM algorithm (Dempster *et al.*, 1977) is the standard tool for ML estimation of finite mixture models. In the E-step the expectation of the complete likelihood is taken, i.e., the a-posteriori probabilities are computed. In the M-step the expected complete likelihood is maximized where the missing component memberships are replaced by the a-posteriori probabilities. The likelihood is increased in each step and convergence of the algorithm is guaranteed for bounded likelihoods. Detection of the global optimum however cannot be ensured.

2.3 Initialization Strategies

- **True cluster membership:** For simulated data the true cluster memberships can be used for initialization in order to investigate the behavior of the EM algorithm when started in the optimal solution.

- **Random initialization:** A commonly used approach is to run EM t times with random starting values and to select the solution maximizing the likelihood among those t runs.

- **Classification EM algorithm:** CEM (Celeux and Govaert, 1992) is a three step procedure where the E-step is equivalent to the standard algorithm. In the C-step a partition is derived by assigning each individual to the component with the maximum a-posteriori probability. In the M-step the ML estimates are computed for the mixture components using the sub-sample induced by the partition of the C-step. CEM converges in a finite number of iterations and tends to produce a mixture with well separated components (Biernacki *et al.*, 2003). It is not maximizing the observed likelihood but the complete likelihood.

  As an initialization strategy CEM is run from t random starting positions and the one providing the highest data log-likelihood is chosen as an initial solution for EM. CEM is started with K much larger than the desired number of clusters in the data as hard classification tends to omit too small clusters whereas the large ones dominate.

- **Stochastic EM algorithm:** SEM (Diebolt and Ip, 1996) includes a restoration of the unknown component labels by drawing them at random from their current a-posteriori probabilities. The E-step is equivalent to the standard algorithm. In the S-step a partition is desired by assigning each point at random to one of the mixture components according to the multinomial distribution with parameter equal to the a-posteriori probabilities. In the M-step the ML estimates are computed for the mixture components using the sub-sample induced by the partition of the S-step. Random drawing at each iteration prevents the SEM from being trapped in local optima.

For initialization SEM is run t times keeping the position leading to the highest maximum likelihood value. The stopping criterion for SEM is the maximum number of iterations which is set to 100.

- **Short runs of EM:** This procedure is suggested by Biernacki *et al.* (2003). EM is run t times from random starting positions before passing to EM without waiting for convergence using the threshold value $|L_q - L_{q-1}|/(|L_q| + 0.1) < tol$, where the tolerance ($tol$) is set to 10^{-2}. L_q is the log-likelihood at the qth iteration.

- **Sampling:** Wehrens *et al.* (2004) modify the simple strategy to cluster larger datasets by clustering a small random sample of the data and to apply the resulting estimated model to the full dataset. The sampling method starts with drawing t samples of size 100 from the full dataset. Next, the EM algorithm is run 3 times on the t samples and the ML solution is used to initialize the EM algorithm for the full dataset. Finally, the ML solution of the t models is selected.

- **Incremental Method:** Fraley *et al.* (2005) developed incremental model-based clustering which is an extension of the sampling method. The method starts by drawing a random sample of the data, selecting and fitting a clustering model to the sample that underestimates the number of components, and extending the model to the full dataset by additional EM iterations. New clusters are added incrementally, initialized with the observations that are poorly fit by the current model. The algorithm stops if adding further components does not increase the log-likelihood or if an a-priori fixed maximum number of components is reached.

  In this simulation the incremental method is started on t samples of size 100 with K equal to 6. As in the sampling method EM is started 3 times and the ML solution is applied to the full dataset. New clusters are added incrementally and initialized with those observations with the lowest 5% log-likelihoods.

- **Spectral clustering:** In spectral clustering (e.g., Ng *et al.*, 2001) data points are clustered using eigenvectors of matrices derived from the data. The success of spectral clustering is mainly based on the fact that it does not make any assumptions on the form of the clusters.

  In this simulation study the algorithm of Ng *et al.* (2001) is applied where the K eigenvectors are used simultaneously. The cluster solution from spectral clustering is used as starting value for the EM algorithm.

An overview of the investigated initialization strategies is given in Table 1. Throughout all computations the minimum component weight of clusters is 0.005 (Leisch, 2004) and the maximum number of iterations is 5000 (except for SEM where it is 100). The convergence criterion for the EM algorithm is $|L_q - L_{q-1}|/(|L_q| + 0.1) < tol$, where the tolerance tol is

Table 1: Overview of the initialization strategies and parameters used where e.g. cem.em indicates the procedure of initializing the EM algorithm in the cluster solution of CEM providing the highest log-likelihood.

	Method	K	t
true	True cluster membership	16	1
rep.em	Random initialization	16	10
cem	Classification EM (CEM)	30	10
cem.em	CEM.EM	result from cem	1
sem	Stochastic EM (SEM)	16	10
sem.em	SEM.EM	result from sem	1
tol	Short runs of EM (Short)	16	10
tol.em	Short.EM	result from tol	1
sam	Sampling	16	$10 \cdot 3$
inc	Incremental Method	6	$10 \cdot 3$
sc	Spectral clustering (SC)	16	1
sc.em	SC.EM	result from sc	1

set to 10^{-6}. In Table 1 K is the number of clusters the algorithm starts with and t is the number of times the algorithm is started keeping only the solution with maximum likelihood. For the incremental method the number of starts is $10 \cdot 3$, i.e., 10 samples are drawn from the full datasets and the algorithm is started 3 times with random initialization on each of the samples. Due to the minimum component weight of clusters smaller components are omitted during the run of the EM algorithm and the number of components in the cluster solutions is often smaller than the number of clusters K the algorithm starts with.

2.4 Evaluation

For the comparison of the different methods the adjusted Rand index is used (Hubert and Arabie, 1985) as well as the log-likelihood, AIC and BIC. Additionally the runtimes of the different procedures are compared.

3 Simulation Study

The performance of the different cluster methods is first evaluated on artificial datasets which are designed to resemble time course gene expression patterns. The number of clusters is set to 15 (as used in Thalamuthu *et al.* (2006)) plus an additional noise cluster of genes. The number of time points is set to 16 (equal to the number of time points in the *E. coli* dataset also used in this paper in the next section). This is a common length for time series microarray data (see for example Cho *et al.*, 1998). The cluster sizes vary between 10 and 100 yielding a total of 630 genes with defined cluster patterns.

Typical time course microarray data have the following form

$$y_{ij} = b_i + \epsilon_{ij},$$

where $b_i \sim N(\mu_k, \sigma_b^2)$ and $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$.

Table 2: Overview of the varying noise parameters.

	Noise level	low	medium	high
N	number of noise genes	100	500	1000
σ_ϵ	SD of mean of genes	0.1	0.3	0.5
σ_m	SD of mean of noise genes	0	1	2
σ_b	SD of RI	0.1	0.7	1.5

The expression pattern y of each gene i at time point j in a given cluster k is assumed to follow the shape of the cluster center μ_k but with a gene specific shift b_i (specified by the noise parameter "SD of RI" (σ_b) where SD denotes standard deviation and RI is the Random Intercept). Additionally a normally distributed measurement error ϵ_{ij} (specified by the noise parameter "SD of mean of genes" (σ_ϵ)) is added to each observation (time point) j. For simplicity σ_b^2 and σ_ϵ^2 are constant across all K components in the simulations.

As typical gene clusters do have arbitrary cluster sizes all simulated datasets consist of clusters of sizes between 10 and 100. Finally an additional noise set of genes of specified size N (given by the noise parameter "number of noise genes") is added to the data. For each noise gene $\mu_k \sim N(0, \sigma_m^2)$ and $\sigma_b \sim U(0.1, 0.3)$. σ_m is specified by the noise parameter "SD of mean of noise genes".

An overview of the different noise parameters used is given in Table 2. One set of cluster centers is used to generate 81 datasets using all possible combinations of noise parameters.

The framework of this simulation study is the following:

1. Add the 81 different combinations of noise to the cluster centers (as given in Table 2).

2. Perform clusterwise regression using the different initialization strategies.

3. Evaluate the performance of initialization strategies on the datasets where the noise set of genes is omitted using the adjusted Rand index (Hubert and Arabie, 1985), the log-likelihood, AIC and BIC.

In this simulation setup cluster centers are created using integrated autoregressive (AR) models. These have been used before to describe gene expression time series (e.g., Ramoni *et al.*, 2002) as AR processes resemble the shape of gene expression over time observed in real time course data very well. An AR process A_j of order 1 is defined by

$$A_j = \alpha A_{j-1} + \epsilon_j$$

where ϵ_j is a series of uncorrelated random variables with mean 0 and variance σ^2. It describes how each observation is a function of the previous observation.

An integrated AR(1) process is a process whose dth difference is an AR(1) process. If $d = 0$ the observations are modeled directly, if $d = 1$ the differences between consecutive observations are modeled, i.e.,

$$A_j = A_{j-1} + \alpha(A_{j-1} - A_{j-2}) + \epsilon_j.$$

If $d = 2$ the differences of differences are modeled, etc.

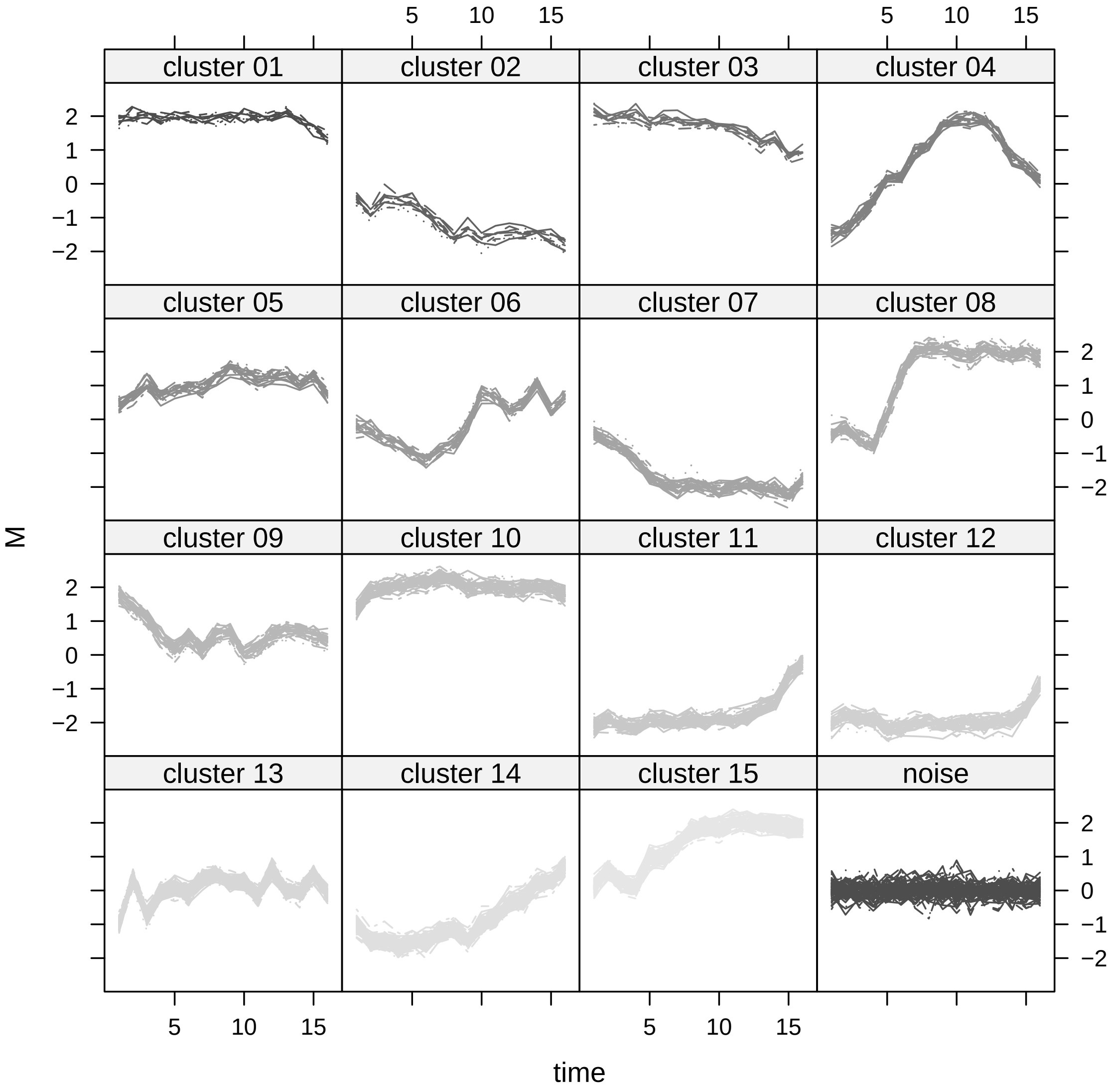

Figure 1: Artificial dataset with low noise level where integrated AR processes are used to create cluster centers.

In this study parameter d is either 1 or 2 in order to get different degrees of smoothness. Half of the generated time series are then reversed and finally transformed to the range of typical gene expression profiles.

One set of cluster centers consists of 15 expression patterns yielding datasets of 15 clusters with dimension (number of time points) 16. The framework in this setting is to generate 50 sets of cluster centers and to perform the simulations on all $50 \cdot 81$ datasets. An artificial dataset where low noise is added to the cluster centers is given in Figure 1 where cluster 16 is a noise cluster of genes showing no differential expression.

Each cluster of the artificial datasets is generated by adding noise to the cluster center. Therefore the underlying cluster structure is known throughout all simulations on artificial data. For simplicity this a priori known starting number of clusters $K = 16$ (15 clusters and an additional noise set of genes) is used for all initialization strategies. Exceptions are CEM and the incremental method where the starting number of clusters are 30 and 6 respectively (see Section 2.3 on CEM).

4 Software

All computations are performed in the statistical computing environment R (R Development Core Team, 2009). The EM algorithm for ML estimation of finite mixture models is implemented in the R package **flexmix** (Grün and Leisch, 2008). The E-step is treated as fixed whereas arbitrary models can be fitted by modifying the M-step. For mixtures of linear mixed models `FLXMRlmer()` and for mixture of linear models with smoothing splines `FLXMRsmooth.spline()` are used as model drivers for the M-step. Spectral clustering is implemented in the R package **kernlab** (Karatzoglou *et al.*, 2004).

R package **gcExplorer** (Scharl and Leisch, 2009) contains functionality to generate a wide range of time course gene expression data. The idea is to start with a set of cluster centers which are created by some data generating process. The following data generating processes are currently possible:

- simulate from a normal distribution,
- simulate from an integrated AR process, and
- manually define patterns.

The gene cluster simulator `gcSim()` is used as follows to generate the set of centers

```
cent <- gcSim(sim = "arima", time = 16, sd = 0.1,
              sd.ri = 0, size = 1, n = 15)
```

where

- `sim:` data generating process
- `time:` number of time points
- `sd:` SD of the mean of genes
- `sd.ri:` SD of RI

- `size`: number of genes in a cluster
- `n`: number of clusters

A set of centers can be used to form 81 different datasets using all possible combinations of noise parameters (as given in Table 2), e.g. with

```
data1 <- gcData(
   gcSim(sim = "pattern", cent = cent,
         sd = 0.1, sd.ri = 0.1,
         size = rep(c(10, 20, 30, 50, 100),
         each = 3)),
   gcSim(sim = "noise", time = 16, size = 100))
```

where `cent` is the set of centers

5 Results

In the following the performance of the different initialization strategies is analyzed in detail.

5.1 Mixtures of linear models

First the cluster results of the models without random intercept (RI) are summarized (in the following called “mixtures of LMs”). Figure 2 shows the adjusted Rand index of cluster solutions of the different initialization strategies and the true cluster membership when low, medium and high noise level is present in the data. For a low noise level starting in the true cluster solution yields the best results as expected. In this case sampling and spectral clustering are also good initialization strategies whereas the CEM variants performs worst. For medium or high noise level the performance of all models is not good and even starting in the true solution yields adjusted Rand indices smaller than 0.5. For these noisy datasets spectral clustering outperforms clusterwise regression.

In Figure 3 the adjusted Rand index is used to compare the performance of the different methods when only one type of noise is present in the data. The corresponding log-likelihoods and runtimes are displayed in Figures 4 and 5. Figure 4 shows that hardly any increase in log-likelihood is observed when starting EM in the solution of CEM, SEM or short runs of EM. The same conclusions can be drawn from the boxplots of the corresponding AIC and BIC values (cf. supplementary material). Runtimes are only shown in Figure 5 for the three noise scenarios where the number of genes is the same, i.e., large SD of mean of genes, large SD of RI and large mean of noise genes. As the number of noise genes added to a dataset is much larger in the forth noise scenario (yielding a total of 1630 genes) the longer runtimes cannot directly be compared to the other scenarios where the number of genes is always 730.

In the case of a large SD of the mean of genes the true cluster solution is the best starting partition, followed by SEM and spectral clustering (see Figure 3). The overall performance is good. However, the runtimes of sampling and the incremental method are the longest, followed by SEM and random initialization.

When a large SD of the RI is present in the data the models without RI cannot identify the components no matter what initialization strategy is used. In this case the runtimes

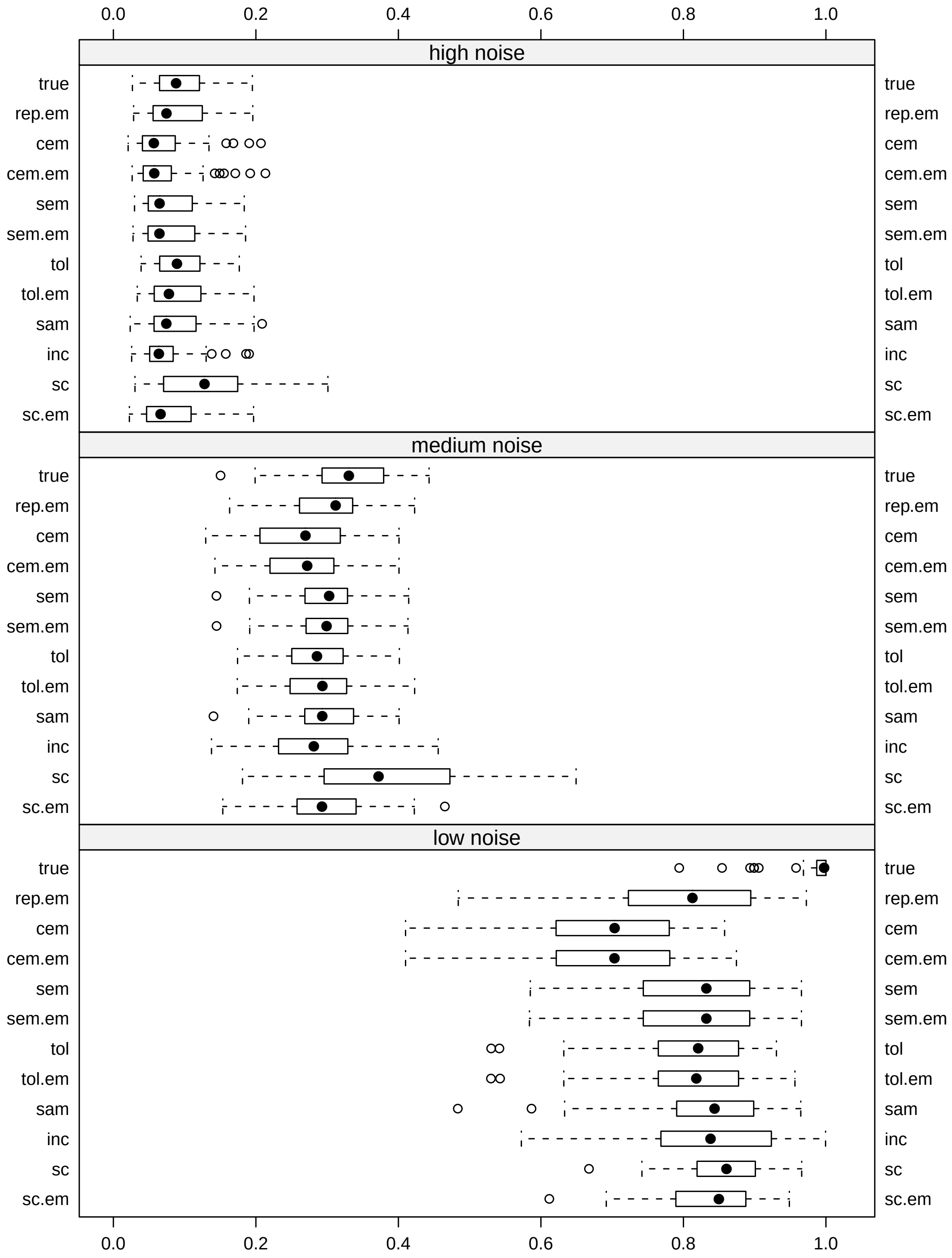

Figure 2: Adjusted Rand index of the different initialization strategies for mixtures of LMs for low, medium and high noise level.

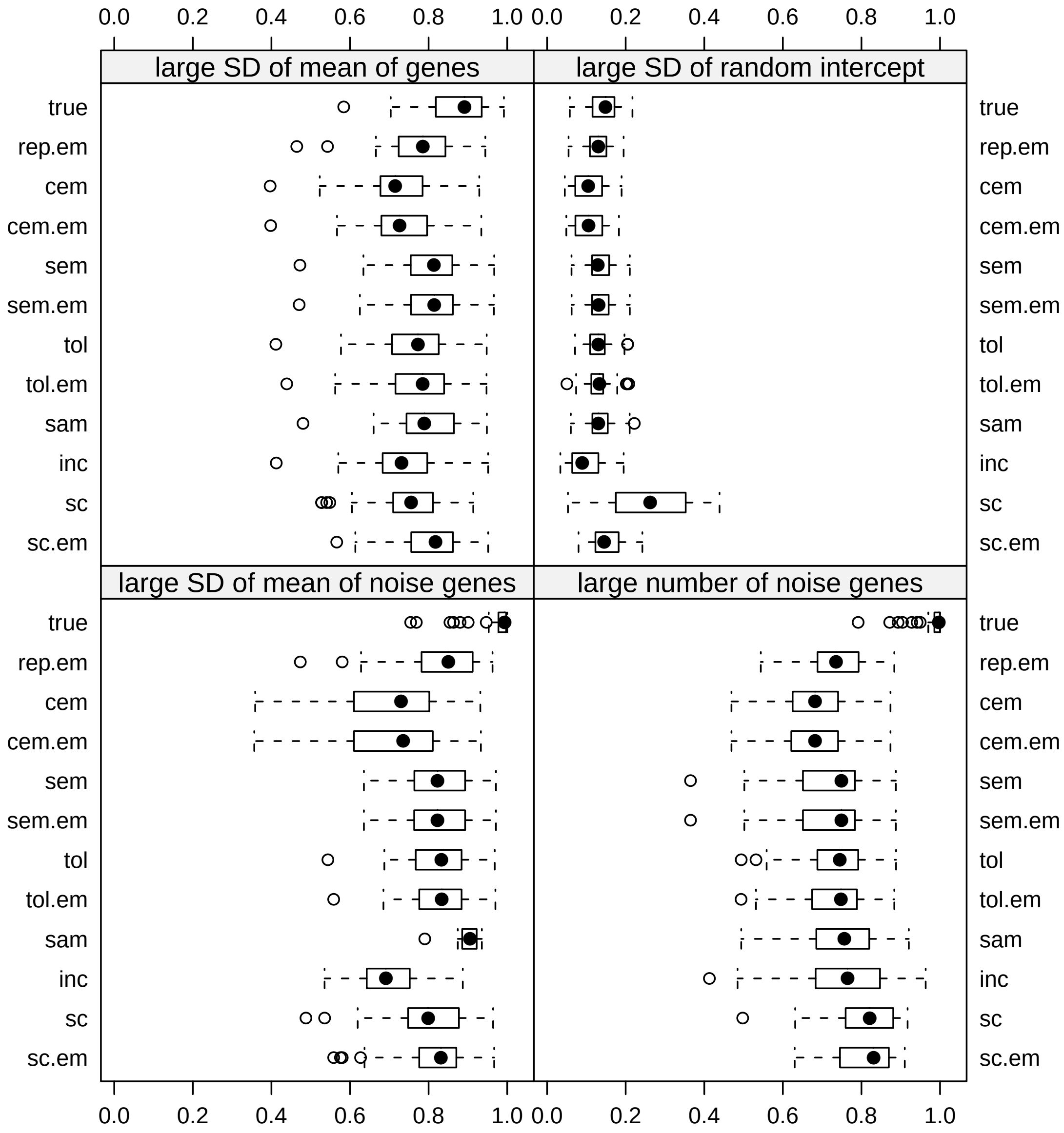

Figure 3: Adjusted Rand index of the different initialization strategies for mixtures of LMs when only one type of noise is present in the data.

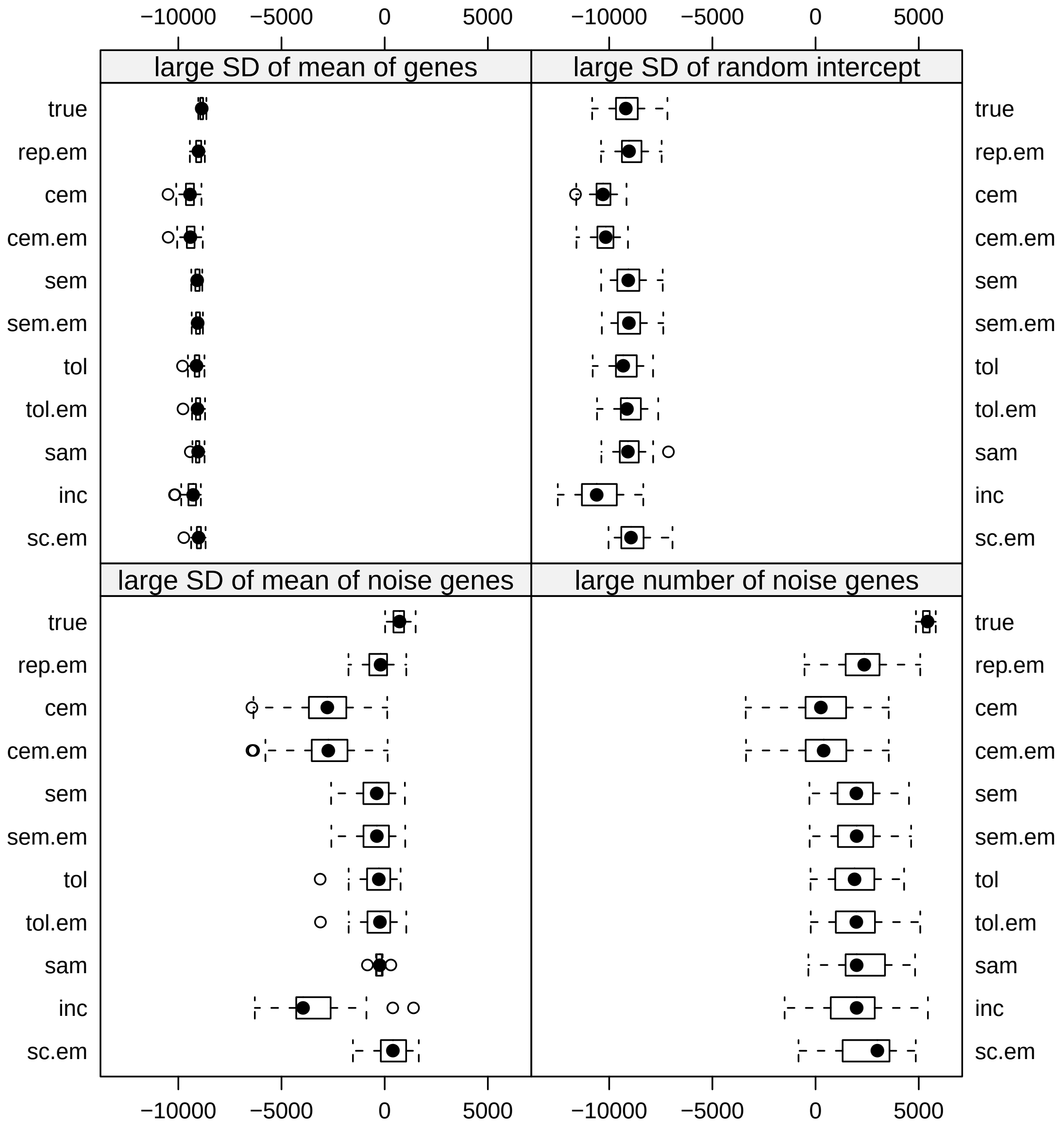

Figure 4: Log-likelihood of the different initialization strategies for mixtures of LMs when only one type of noise is present in the data.

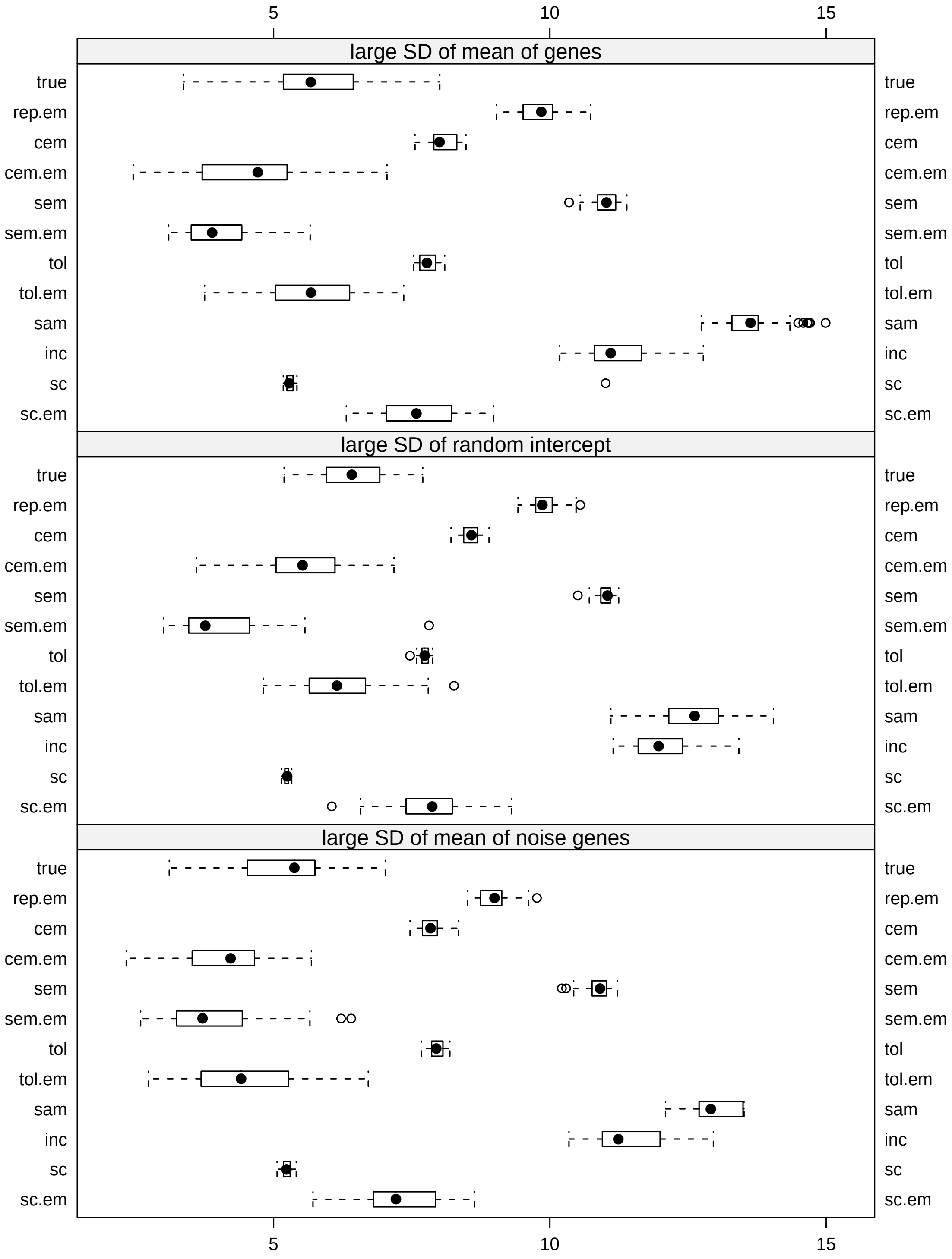

Figure 5: log_2-transformed system time of the different initialization strategies for mixtures of LMs when only one type of noise is present in the data, i.e., large SD of mean of genes, large SD of RI or large SD of mean of noise genes.

of sampling, the incremental method, SEM, and random initialization are again very large. Furthermore, spectral clustering outperforms clusterwise regression.

For clusters with large SD of the mean of the noise genes the agreement between the cluster solutions and the true cluster memberships is in general very high. This indicates that noise genes do not affect the clustering of differentially expressed genes. In this case CEM and the incremental method yield the worst results. Again, sampling and the incremental method have the longest runtimes.

Finally, in the case of a large number of noise genes starting in the true cluster solution clearly outperforms the other initialization strategies but the performance of all methods is very good.

5.2 Mixtures of linear mixed models

Next the cluster results of mixture models with RI (in the following called "mixtures of LMMs") are summarized in boxplots. Figure 6 shows the adjusted Rand index of cluster solutions of the different initialization strategies and the true cluster membership when low, medium and high noise level is present in the data. In contrast to the model without RI (Figure 2) where the quality of the cluster solutions decreases tremendously when medium or high noise is added to the datasets now the overall impression of the cluster solutions is much better. Even for high noise level the corrected Rand index is about 0.6. CEM and the incremental method perform among the worst for low noise level whereas starting in the true cluster solution and SEM yield the best results. In addition mixture models with a RI clearly outperform spectral clustering.

The performance of the mixture of LMMs when only one type of noise is present in the data (see Figures 7 and 8) is also much better compared to the mixtures of LMs (Figure 3). Again, the results for the log-likelihoods are very similar to those using AIC and BIC and hence, the boxplots of AIC and BIC are omitted here. As expected the performance of all initialization strategies is very good for data generated with a large SD of the RI.

The big disadvantage of mixture of LMMs are the long runtimes (not shown here) which are by a factor of 10 longer than the runtimes of the mixture of LMs (see Figure 5). However, the trend is the same for models with and without RI. Random initialization, SEM, the sampling and the incremental method cannot be recommended due to the extremely long runtimes.

5.3 Comparison of LMs vs. LMMs

A fair comparison of mixtures of LMs and mixtures of LMMs is only possible for data sets without gene-specific shift because only in this case also the smaller model is appropriate whereas otherwise the true data structure can only be captured by LMMs. However, assuming no individual-specific effect seems implausable and hence we use the data sets of the scenario 'low SD of RI' but 'large SD of mean of genes' where the gene-specific shift is only 0.1, for comparison. Even in the case of large SD of mean of genes which corresponds to clusters with negligible gene-specific shift but large measurement noise mixtures of LMMs seem to perform no worse than mixtures of LMs (see Figure 9). The adjusted Rand index is larger for mixtures of LMs when starting in the true cluster solution. On the other hand mixtures of LMMs yield better results for CEM. This implies that the unnecessary flexibility of the more complex model class does not deteriorate the results by capturing for example noise effects.

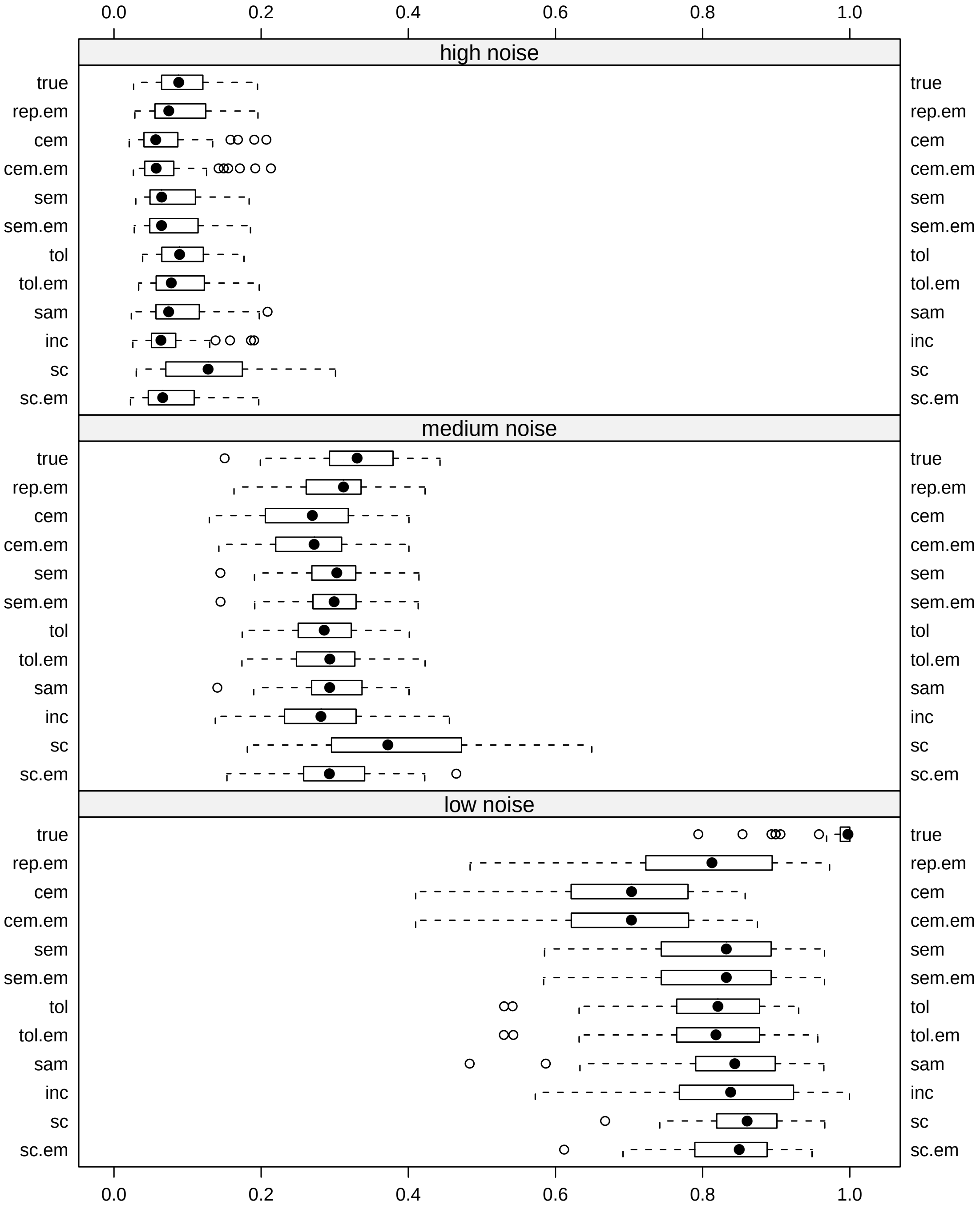

Figure 6: Adjusted Rand index of the different initialization strategies for mixtures of LMMs for low, medium and high noise level.

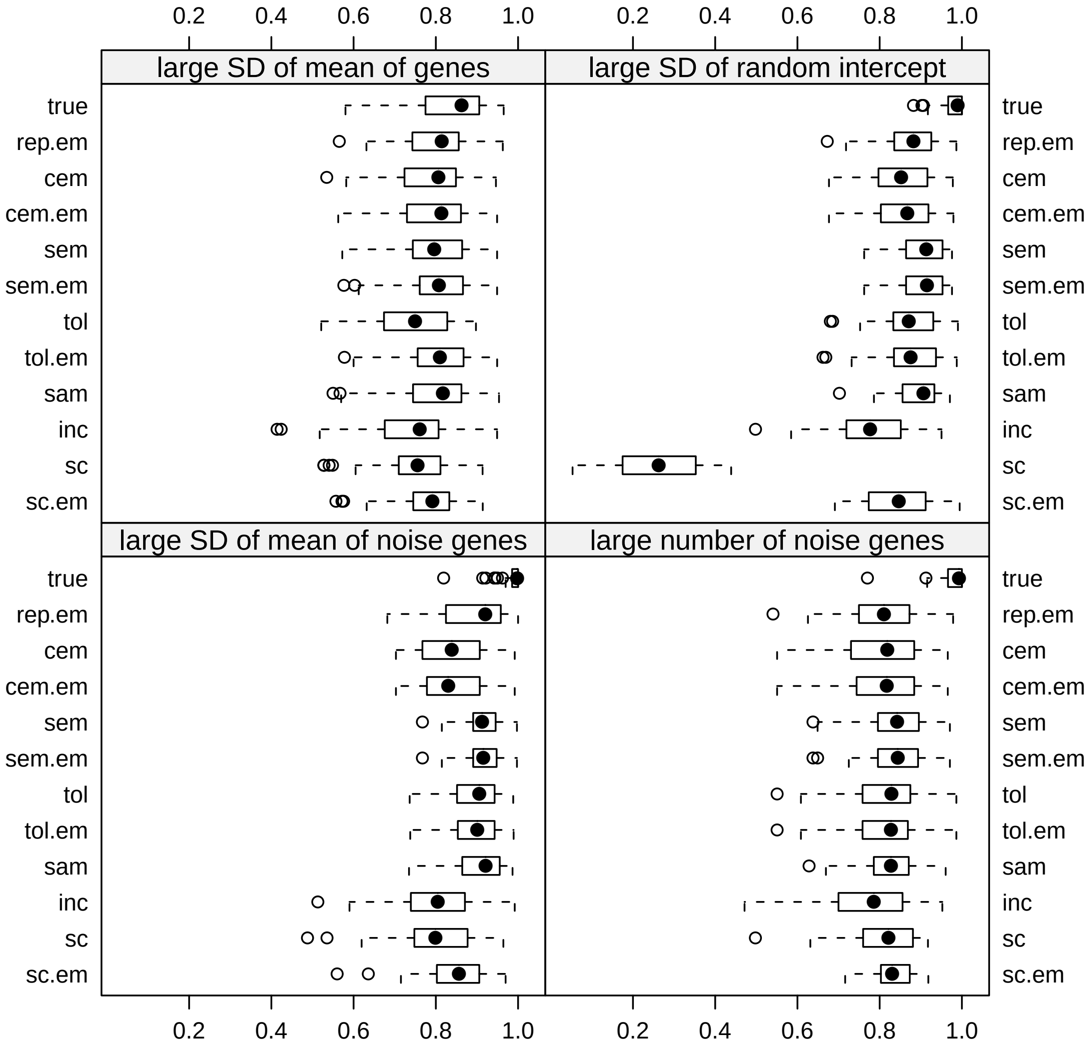

Figure 7: Adjusted Rand index of the different initialization strategies for mixtures of LMMs when only one type of noise is present in the data.

However, the big disadvantage of mixtures of LMMs in cases where they are not actually needed are the much longer runtimes.

5.4 *E. coli* data

The goal of the *E. coli* experiment is the detailed investigation of the cellular response of *E. coli* BL21(DE3) to high level expression of recombinant human super–oxide–dismutase (SOD) on the transcriptional level. The experiment is available at ArrayExpress (http://www.ebi.ac.uk/microarray-as/ae/) with accession number E-MARS-19. The data consists of 530 genes at 16 time points after filtering genes not differentially expressed at least at one time point (p-value < 0.05, log ratio M > 2 and average intensity A > 8).

In the case of time course microarray data the definition of clusters is not clear and therefore the quality of a cluster solution is difficult to evaluate. Even the number of components is hard to specify as practitioners usually prefer small clusters which can easily be investigated. However, too many clusters are even harder to interpret. In an exploratory step different cluster solutions using random initialization were compared starting with 5 to 60 components and yielding up to 29 components. The likelihood criterion as well as AIC and BIC select the model where the EM algorithm was started with $K = 58$ and where 27 components are found. For the comparison of initialization strategies all algorithms are therefore started with $K = 58$ components.

The data was clustered using the methods investigated in the simulation study using 58 components for mixtures of regression models and 30 centers for spectral clustering. The primary goal of the comparison of the different cluster solutions is to find out which algorithm yields the best likelihood or the likelihood penalized for model complexity as the true cluster structure is unknown. Cluster results using the different initialization strategies are given in Table 3 where K is the number of components found and *df* is the number of degrees of freedom used. For LMs smoothing splines are used whereas b-splines are used for LMMs. This implies that for mixtures of LMs the numbers of df do not only depend on the number of components but also on the complexity of the smoothing splines in each of the components. For mixtures of LMs the number of components found is between 15 (sampling method) and 31 (short runs of EM). Initializing EM in the solution of short runs of EM is also the method with the largest log-likelihood and smallest AIC. BIC selects the solution of SEM where 23 clusters are found. In the case of mixture models with RI the number of components found varies between 10 (incremental method) and 35 (random initialization and short runs of EM). AIC, BIC and log-likelihood select the results of random initialization as the best solution.

6 Summary and Outlook

In this simulation study on artificial time course gene expression data commonly used initialization strategies were investigated to find the most appropriate ones for clusterwise regression.

Some general observations were made for this type of data:

1. For noisy datasets mixtures of LMs should not be used. Mixtures of LMMs clearly outperform mixtures of LMs and spectral clustering on noisy datasets. However, the user should be aware of the much longer runtimes.

Table 3: Results of the initialization strategies used on the *E. coli* dataset using mixtures of LMs and mixtures of LMMs when starting with $K = 58$ components.

	RI	K	df	time	iter	logLik	BIC	AIC
sc	N	30	-	16	-	-	-	-
rep.em	N	28	499	1305	39	-4045	12603	9088
cem	N	18	259	305	21	-5080	12500	10678
cem.em	N	18	320	99	68	-4954	12800	10547
sem	N	23	350	1638	96	-4437	**12038**	9573
sem.em	N	23	409	16	9	-4403	12502	9623
tol	N	**31**	550	260	9	-3893	12765	8886
tol.em	N	**31**	551	155	74	**-3683**	12347	**8468**
sam	N	15	266	24410	47	-5303	13010	11138
inc	N	18	314	4357	20	-5202	13243	11031
sc.em	N	30	535	104	48	-3814	12464	8698
rep.em	Y	**35**	279	31666	37	**-3113**	**8748**	**6783**
cem	Y	23	183	5115	21	-3799	9253	7964
cem.em	Y	23	183	2969	88	-3717	9090	7800
sem	Y	31	247	25464	96	-3267	8769	7029
sem.em	Y	31	247	568	14	-3261	8757	7017
tol	Y	**35**	279	5302	10	-3256	9036	7071
tol.em	Y	**35**	279	2251	48	**-3124**	8772	**6806**
sam	Y	13	103	11434	90	-4559	10049	9323
inc	Y	10	79	23870	12	-4990	10695	10139
sc.em	Y	30	239	2142	53	-3455	9072	7388

2. Running SEM, CEM or short runs of EM and using their best solution for the initialization of EM does hardly increase the performance already reached by these strategies. This was observed using the classification criterion as well as the likelihood criterion.

3. Computationally intensive methods like the sampling or incremental method are hardly worth the effort.

4. Random initialization yields very long runtimes compared to CEM or short runs of EM. However, the cluster results are similar.

5. The impact of the cluster method used is much larger than the impact of the initialization strategy.

6. For short runs of EM the tradeoff between the quality of the cluster solutions and runtime is very good.

In the future it would be interesting to extend this simulation study and compare the cluster results from mixture models to solutions from partitioning cluster algorithms. For this purpose further measures like a cluster stability score (e.g., Handl *et al.*, 2005) will be investigated beside the currently used methods.

Additionally, simulations on smaller data sets with less time points will be performed. First results on the *E. coli* data (cf. supplementary material) are very promising and indicate that the cluster methods perform similar on smaller data sets.

Acknowledgement

This work was supported by the Austrian $\mathrm{K}_{ind}/\mathrm{K}_{net}$ Center of Biopharmaceutical Technology (ACBT) and the Austrian Research Foundation (FWF) under Hertha-Firnberg grant T351. Special thanks to Gerald Striedner for providing us with the *E. coli* data.

Conflict of Interest: None declared

References

Bar-Joseph, Z., Gerber, G., Jaakkola, T. S., Gifford, D. K., and Simon, I. (2003). Continuous representations of time series gene expression data. *Journal of Computational Biology*, **3–4**, 341–356.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, **41**, 561–575.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm and two stochastic versions. *Computational Statistics & Data Analysis*, **14**, 315–332.

Celeux, G., Martin, O., and Lavergne, C. (2005). Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments. *Statistical Modelling*, **5**, 243–267.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, **2**(1), 65–73.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM-algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**, 1–38.

Diebolt, J. and Ip, E. (1996). Stochastic EM: method and application. In W. Gilks, S. Richardson, and D. Speigelhalter, editors, *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall.

Fraley, C., Raftery, A. E., and Wehrens, R. (2005). Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics*, **14**(3), 529–546.

Grün, B. and Leisch, F. (2008). Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, **28**(4), 1–35.

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

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, **2**, 193–218.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, **11**(9), 1–20.

Leisch, F. (2004). FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, **11**(8), 1–18.

Luan, Y. and Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, **19**(4), 474–482.

Ma, P., Castillo-Davis, C. I., Zhong, W., and Liu, J. S. (2006). A data-driven clustering method for time course gene expression data. *Nucleic Acids Research*, **34**(4), 1261–1269.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press.

Ng, S. K., McLachlan, G. J., Wang, K., Jones, L. B.-T., and Ng, S.-W. (2006). A mixture model with random-effects components for clustering correlated gene-expression profiles. *Bioinformatics*, **22**(14), 1745–1752.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ramoni, M. F., Sebastiani, P., and Kohane, I. S. (2002). Cluster analysis of gene expression dynamics. *Proc. Natl. Acad. Sci. USA*, **99**(14), 9121–9126.

Scharl, T. and Leisch, F. (2009). gcExplorer: Interactive exploration of gene clusters. *Bioinformatics*, **25**(8), 1089–1090. doi: 10.1093/bioinformatics/btp099.

Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. C. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, **22**(19), 2405–2412.

Wehrens, R., Buydens, L. M., Fraley, C., and Raftery, A. E. (2004). Model-based clustering for image segmentation and large datasets via sampling. *Journal of Classification*, **21**, 231–253.

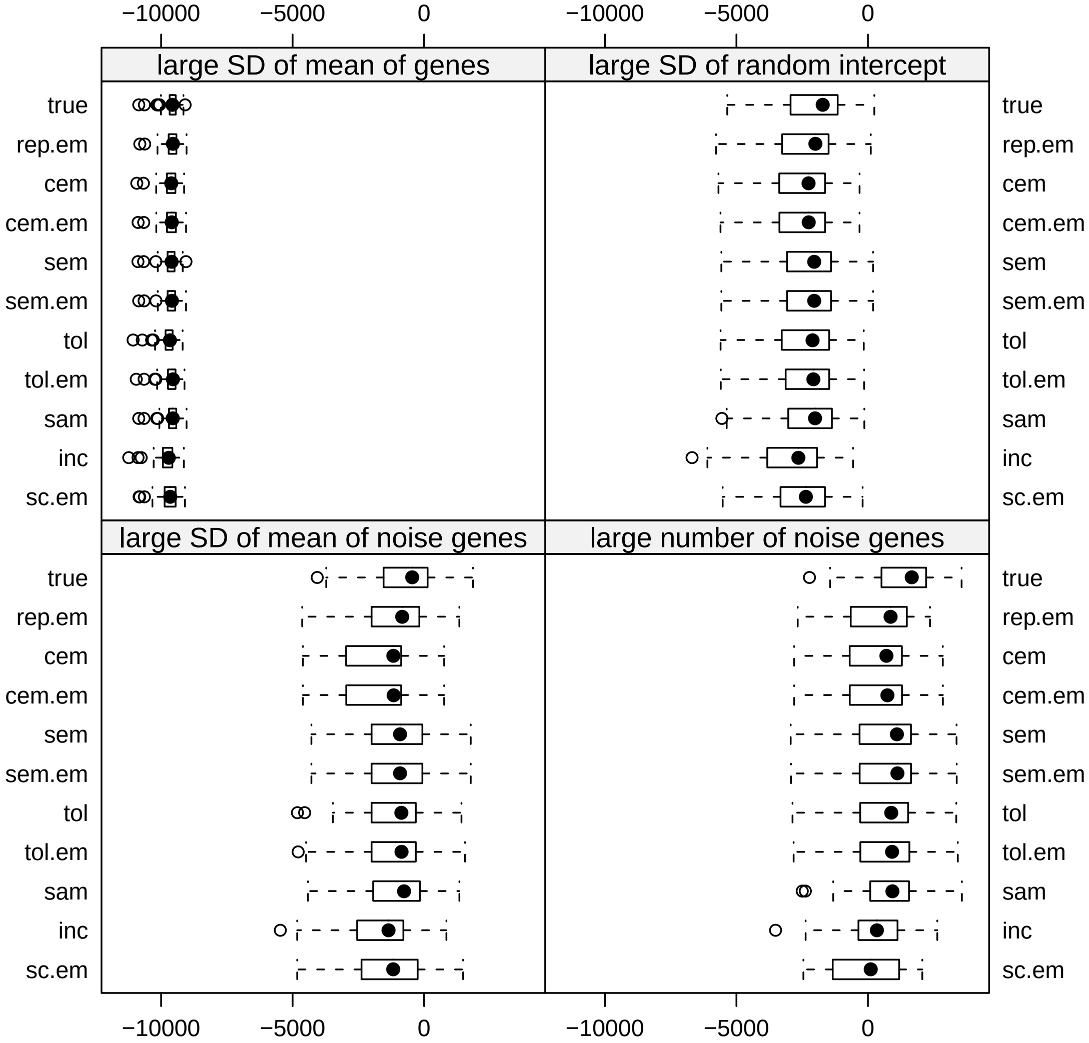

Figure 8: Log-likelihood of the different initialization strategies for mixtures of LMMs when only one type of noise is present in the data.

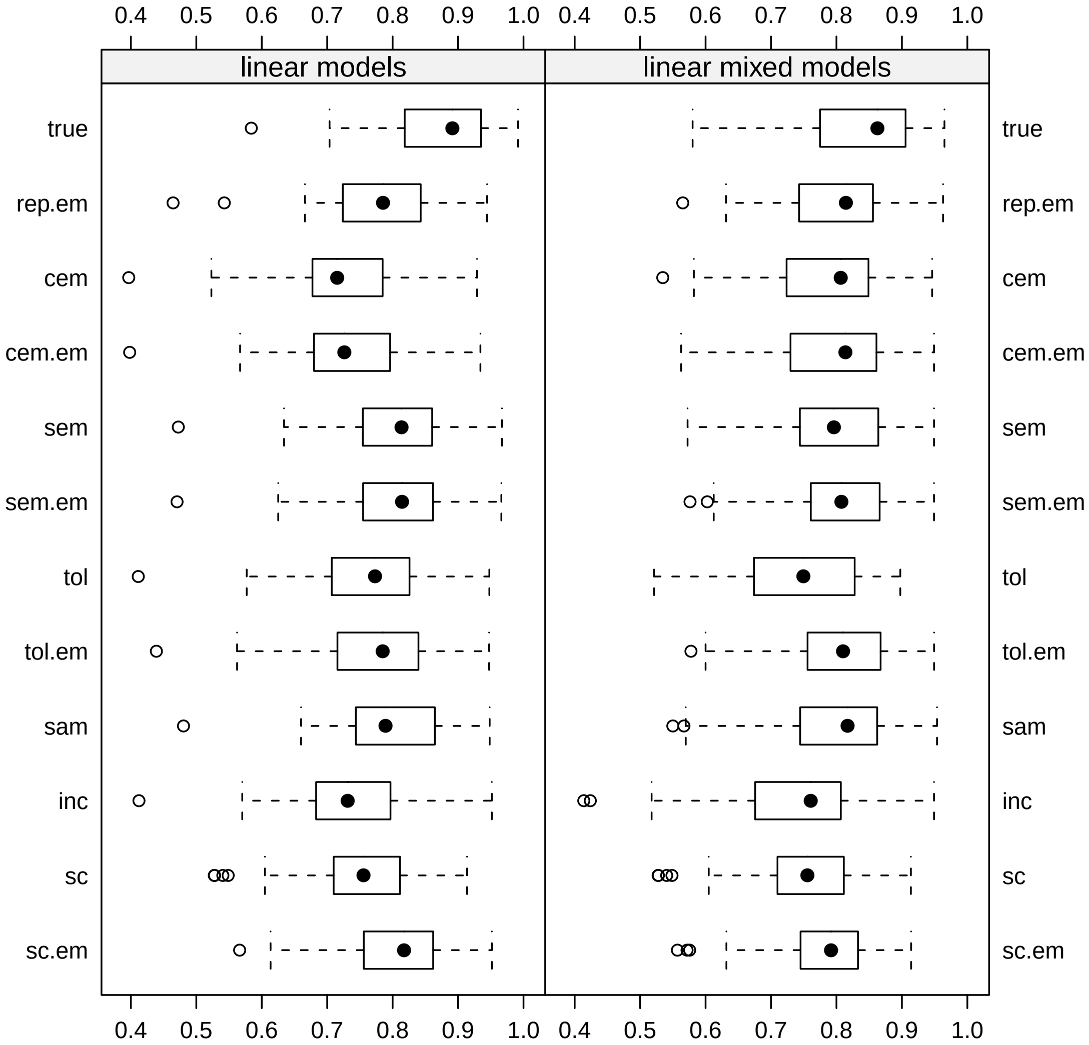

Figure 9: Adjusted Rand index of the different initialization strategies for mixtures of LMs versus mixtures of LMMs for the noise scenario ’low SD of RI’ but ’large SD of mean of genes’.