Gene expression

Mixtures of regression models for time course gene expression data: evaluation of initialization and random effects

Theresa Scharl¹,²,*, Bettina Grün³ and Friedrich Leisch⁴

¹Department of Statistics and Probability Theory, Vienna University of Technology, Wiedner Hauptstr. 8-10, A-1040 Vienna, ²Department of Biotechnology, University of Natural Resources and Applied Life Sciences, Muthgasse 18, A-1190 Vienna, ³Department of Statistics and Mathematics, Wirtschaftsuniversität Wien, Augasse 2-6, A-1090 Vienna, Austria and ⁴Department of Statistics, University of Munich, Ludwigstr. 33, D-80539 Munich, Germany

Received on October 4, 2009; revised on December 1, 2009; accepted on December 7, 2009
Advance Access publication December 29, 2009
Associate Editor: Martin Bishop

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 Escherichia coli.

Availability: The latest release versions of R packages flexmix, geExplorer and kernlab are always available from CRAN (http://cran.r-project.org/).
Contact: theresa.scharl@ci.tuwien.ac.at
Supplementary information: Supplementary data are available at Bioinformatics online.

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 subsample 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 (SC) (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 article, several initialization strategies are investigated in an extensive simulation study on different artificial time course datasets, i.e. random initialization, CEM, SEM, short runs of EM itself, the sampling and the incremental method as well as SC.

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 Escherichia coli.

2 METHODS
2.1 Model specification
The mixture density \( h(y; \pi, \psi) = \sum_{k=1}^{K} \pi_k f(y; \theta_k) \) is given by

\[ h(y; \pi, \psi) = \sum_{k=1}^{K} \pi_k f(y; \theta_k) \]

where \( y \) is the response, \( x \) denotes the predictors and \( \psi \) is the predictor of all parameters for the mixture density \( h \). For the component weights \( \pi_k \), it holds that \( \pi_k > 0 \) for all \( k \) and \( \sum_{k=1}^{K} \pi_k = 1 \). \( \theta_k \) is the component-specific parameter vector for the component-specific density function \( f \).

Mixture 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, whereas 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 \( Y_i = (y_{ijk})_{k=1, \ldots , K} \), the covariates for the fixed effects \( X_i = (x_{ijk})_{k=1, \ldots , K} \) and the covariates for the random effects \( Z_i = (z_{ijk})_{k=1, \ldots , K} \). The finite mixture density of mixed effects models with \( K \) components is given for the observations of individual \( i \) by

\[
h(Y_i | X_i, Z_i, \psi) = \sum_{k=1}^{K} \pi_{nk} \left[ \prod_{j=1}^{n_i} \phi_k(y_{ijk} | \beta, \phi) \right] = \sum_{k=1}^{K} \pi_{nk} \left[ \prod_{j=1}^{n_i} \phi_k(y_{ijk} | \beta, \phi) \right]
\]

\( \phi_k : (\mu, \Sigma) \) denotes the \( d \)-dimensional multivariate normal distribution with mean \( \mu \) and variance-covariance matrix \( \Sigma \).

- Fixed effects: \( \beta_k \) with \( \beta_k \) deterministic.
- Random effects: \( z_{ijk} \) with \( z_{ijk} \) normally distributed.

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 (Bar-Joseph et al., 2003; Luan and Li, 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 with 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 \( r \) times with random starting values and to select the solution maximizing the log-likelihood among those \( r \) runs.
- CEM 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 subsample 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 \( r \) 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.

- SEM algorithm: SEM (Diebolt and Ip, 1996) includes a restoration of the unknown component labels by sampling 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 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 subsample 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 \( r \) times keeping the position leading to the highest ML 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) and used in the sampling method.
- 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 \( r \) samples of size 100 with \( K \) equal to 6. As in the sampling method EM is run three 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.

- SC: in SC (e.g. Ng et al., 2001) data points are clustered using eigenvectors of matrices derived from the data. The success of SC 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 SC 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 \( |L_{t+1} - L_T|/(|L_0| + 0.1) < \) tol, where the tolerance tol is set to \( 10^{-5} \). In Table 1, \( K \) is the number of clusters the algorithm starts with and \( r \) is the number of times the algorithm is started keeping only the solution with ML. For the incremental method, the number of starts is 10, i.e. 10 samples are drawn from the full datasets and the algorithm is started three 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.
Additionally, the runtimes of the different procedures are compared.

Table 1. Overview of the initialization strategies and parameters used

<table>
<thead>
<tr>
<th>Method</th>
<th>K</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>rep.em</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>cem</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>cem.em</td>
<td>Result from cem</td>
<td>1</td>
</tr>
<tr>
<td>sem</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>sem.em</td>
<td>Result from sem</td>
<td>1</td>
</tr>
<tr>
<td>tol</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>tol.em</td>
<td>Result from tol</td>
<td>1</td>
</tr>
<tr>
<td>sam</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>sc</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>sc.em</td>
<td>Result from sc</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2. Overview of the varying noise parameters

<table>
<thead>
<tr>
<th>Noise level</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>100</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>(\sigma^2)</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>(\sigma_{\text{ME}})</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>(\sigma^2_{\text{RI}})</td>
<td>0.1</td>
<td>0.7</td>
<td>1.5</td>
</tr>
</tbody>
</table>

For example, cem.em indicates the procedure of initializing the EM algorithm in the cluster solution of CEM providing the highest log-likelihood.

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 article in the next section). This is a common length for time series microarray data (e.g. 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

\[ y_{ij} = \beta_i + \epsilon_{ij}, \]

where \(\beta_i \sim N(\mu_i, \sigma_{\beta_i}^2)\) and \(\epsilon_{ij} \sim N(0, \sigma_{\epsilon}^2)\). The expression pattern \(\gamma\) of each gene \(i\) at time point \(j\) in a given cluster \(k\) is assumed to follow the shape of the cluster center \(\mu_k\) but with a gene-specific shift \(\beta_i\) [specified by the noise parameter ‘SD of RI’ (\(\sigma_\beta\)) where SD denotes standard deviation and RI is the Random Intercept]. Additionally, a normally distributed measurement error \(\epsilon_{ij}\) [specified by the noise parameter ‘SD of mean of genes’ (\(\sigma_\epsilon\)) is added to each observation (time point) \(j\). For simplicity, \(\sigma_{\beta_i}^2\) and \(\sigma_{\epsilon}^2\) 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 \(\mu_{\text{noise}} \sim N(0, \sigma_{\mu_{\text{noise}}}^2)\) and \(\epsilon_{\text{noise}} \sim U(0,1,0.5)\). \(\sigma_{\mu_{\text{noise}}}\) 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 \(\epsilon_j\) is a series of uncorrelated random variables with mean 0 and variance \(\sigma^2\). It describes how each observation is a function of the previous observation.

An integrated AR(1) process is a process whose \(d\)-th difference is an AR(1) process. If \(d=0\) the observations are modeled directly, if \(d=1\) the differences of 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.

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 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.

![Fig. 1. Artificial dataset with low noise level where integrated AR processes are used to create cluster centers.](image-url)
Mixtures of regression models for time-course gene expression data

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 \texttt{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
\texttt{FLXMRlinear()} and for mixture of linear models with smoothing splines
\texttt{FLXMRsmooth.spline()} are used as model drivers for the M-step. SC
is implemented in the R package \texttt{kernlab} (Karatzoglou et al., 2004).

R package \texttt{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 \texttt{gcSim()} is used as follows to generate the set
of centers

\begin{verbatim}
cent <- gcSim(sim = "arima", time = 16, sd = 0.1, sd.ri = 0, size = 1, n = 15)
\end{verbatim}

where

- \texttt{sim}: data generating process
- \texttt{time}: number of time points
- \texttt{sd}: SD of the mean of genes
- \texttt{sd.ri}: SD of RI
- \texttt{size}: number of genes in a cluster
- \texttt{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

\begin{verbatim}
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))
\end{verbatim}

where \texttt{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 RI are summarized
(in the following called ‘mixtures of LMs’). Figure 2 shows the adjusted Rand index of
class 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 SC 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, SC 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 with
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 SC
(Fig. 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 of sampling, the
Fig. 3. Adjusted Rand index of the different initialization strategies for mixtures of LMs when only one type of noise is present in the data.

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

incremental method, SEM and random initialization are again very large. Furthermore, SC 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.

Fig. 5. log2-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.

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 with the model without RI (Fig. 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 SC.

The performance of the mixture of LMMs when only one type of noise is present in the data (Figs 7 and 8) is also much better compared with the mixtures of LMs (Fig. 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
Mixtures of regression models for time-course gene expression data

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

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

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

5.3 Comparison of LMs versus LMMs

A fair comparison of mixtures of LMs and mixtures of LMMs is only possible for datasets 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 that no individual-specific effect seems implausible, we use the datasets 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 (Fig. 9). The adjusted Rand index is larger for mixtures of LMs when starting in the true cluster solution. On the contrary, 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. However, the big disadvantage of mixtures of LMMs in cases where they are not actually needed are the much longer runtimes.

5.4 Escherichia 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 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 ($P < 0.05, \log \text{ratio} M > 2$ 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
the numbers of b-splines are used for LMMs. This implies that for mixtures of LMs, the number of components found and the best likelihood or the likelihood penalized for model complexity is the number of degrees of freedom used. For LMs smoothing splines are used whereas the number of components found varies 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 is between 10 (incremental method) and 35 (random initialization and short runs of EM).

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 SC on noisy datasets. However, the user should be aware of the much longer runtimes.

(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 with 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 datasets with less time points will be performed. First, results on the E.coli data (cf.

Fig. 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'.

| 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    | 515   | 104   | 48    | 3814  | 9257  | 7029  |

The best (and second best) model for each criterion is given in bold.
Supplementary Material) are very promising and indicate that the cluster methods perform similar on smaller datasets.

ACKNOWLEDGEMENT
Special thanks to Gerald Striedner for providing us with the E.coli data.

Funding: Austrian KWF/Knet Center of Biopharmaceutical Technology (ACBT); Austrian Research Foundation (FWF) (Hertha-Firnberg grant T351).

Conflict of Interest: none declared.

REFERENCES