Supplementary materials to ”Mixture models with multiple levels, with application to the analysis of multi-factor gene expression data”

Rebecka Jörnsten*
Department of Statistics
Rutgers University
501 Hill Center
Piscataway, NJ 08854, USA

Sündüz Keleş
Department of Statistics,
Department of Biostatistics
and Medical Bioinformatics
University of Wisconsin-Madison
1300 University Avenue
Madison, WI 53706, USA

November 26, 2007

1 Design matrices for various parameterizations

We parameterize the cluster mean $\mu_{kl} = W\beta_{kl}$, where $\beta_{kl} = (\alpha_{k1}, \alpha_{k2}, \alpha_{k3}, \gamma_{kl1}, \gamma_{kl2}, \gamma_{kl3})$. The three parameterizations discussed in section 2 of the paper are shown here:

$$W_I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$W_{II} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

*Corresponding Author: rebecka@stat.rutgers.edu, telephone +1 732 445-3145, fax +1 732 445-3428

$$W_{III} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

2 Details of the $\mathcal{MIX}_{\mathcal{L}}$ algorithm

2.1 The PEM algorithm.

1. **M-step-Aggregate.** Compute gene specific membership for level 1 by aggregating $\hat{\eta}_{gkl}$.
$$\hat{\tau}_{gk} = \sum_{l=1}^{L_k} \hat{\eta}_{gkl}.$$

2. **M-step-Profile-1.** This step concerns the profiling of the multivariate normal density for $\mathbf{x}_g$.

    (a) Consider the part of the expected complete data likelihood that involves the marginal distribution of $\mathbf{x}_g$. We have
    $$\sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{l=1}^{L_k}\left(-\frac{1}{2}\hat{\eta}_{gkl}(\mathbf{x}_g - W_K\alpha_k)'\mathbf{\Sigma}_k^{-1}(\mathbf{x}_g - W_K\alpha_k) - \frac{1}{2}\hat{\eta}_{gkl}\log|\mathbf{\Sigma}_k^X|\right) = \tag{1}$$
    $$\sum_{g=1}^{G}\sum_{k=1}^{K}\left(-\frac{1}{2}\hat{\tau}_{gk}(\mathbf{x}_g - W_K\alpha_k)'\mathbf{\Sigma}_k^{-1}(\mathbf{x}_g - W_K\alpha_k) - \frac{1}{2}\hat{\tau}_{gk}\log|\mathbf{\Sigma}_k^X|\right),$$
    where W_K is the upper left diagonal block of design matrix W (corresponding to the 1st level data).

    (b) Given μ_k, we profile with respect to $\mathbf{\Sigma}_k^X$ and get
    $$\mathbf{\Sigma}_k^{X^{(r)}} = \frac{\sum_{g=1}^G \hat{\tau}_{gk}(\mathbf{x}_g - \mu_k^{(r)})(\mathbf{x}_g - \mu_k^{(r)})'}{\sum_{g=1}^G \hat{\tau}_{gk}},$$
    where r refers to the r-th iteration of the EM steps. A regularized version of $\mathbf{\Sigma}_k^X$ is obtained as
    $$\tilde{\mathbf{\Sigma}}_k^{X^{(r)}} = \frac{\Delta_p^X(\nu) + \mathbf{\Sigma}_k^{X^{(r)}} n_k}{\nu + n_k},$$
    where $n_k = \sum_{g=1}^G \hat{\tau}_{gk}$, and $\Delta_p^X = \frac{\sum_{g=1}^G (\mathbf{x}_g - \bar{x_g})(\mathbf{x}_g - \bar{x_g}))'}{GK^{2/T}}$. The choice of scale parameter, ν, is discussed in section 2.3 (see also Fraley and Raftery (2004)).

    (c) Holding $\mathbf{\Sigma}_k^X$ fixed, the maximizer over α_k can be obtained by first performing a weighted least squares fit of the form
    $$x_g = W_K\alpha_{gk} + \epsilon, \quad \text{where} \quad cov(\epsilon) = \mathbf{\Sigma}_k^X,$$

and then taking a weighted average of the estimates of α_{gk} as

$$\alpha_k^{(r)} \equiv \hat{\alpha}_k = \frac{\sum_{g=1}^G \hat{\tau}_{gk}\hat{\alpha}_{gk}}{\sum_{g=1}^G \hat{\tau}_{gk}}.$$

Finally, we update μ_k as $\mu_k^{(r)} = W_K\hat{\alpha}_k$.

3. **M-step-Profile-2.** Next, we consider profiling the conditional distribution of $\mathbf{y}_g$ given $\mathbf{x}_g$. The expected complete data likelihood that involves the conditional distribution of $\mathbf{y}_g$ given $\mathbf{x}_g$ is given by

$$\sum_{g=1}^G\sum_{k=1}^K\sum_{l=1}^{L_k}\left(-\frac{1}{2}\eta_{gkl}(\mathbf{y}_g - \mu_{kl}^{Y|X})'\mathbf{\Sigma}_{kl}^{Y|X^{-1}}(\mathbf{y}_g - \mu_{kl}^{Y|X}) - \frac{1}{2}\eta_{gkl}\log|\mathbf{\Sigma}_k^{Y|X}|\right),$$

where

$$\begin{aligned}\mu_{kl}^{Y|X} &= \mathbf{\Sigma}_{kl}^{XY}(\mathbf{\Sigma}_k^X)^{-1}(\mathbf{x}_g - \mu_k),\\ \mathbf{\Sigma}_{kl}^{Y|X} &= \mathbf{\Sigma}_{kl}^{Y} - \mathbf{\Sigma}_{kl}^{YX}(\mathbf{\Sigma}_k^X)^{-1}\mathbf{\Sigma}_{kl}^{XY}.\end{aligned}$$

(a) The second profiling step starts with updating $\mathbf{\Sigma}_{kl}^{YX}$and $\mathbf{\Sigma}_{kl}^{Y}$ as follows.

$$\begin{aligned}\mathbf{\Sigma}_{kl}^{YX^{(r)}} &= \frac{\sum_{g=}^G \hat{\eta}_{gkl}(\mathbf{y}_g - \mu_{kl}^{Y^{(r)}})(\mathbf{x}_g - \mu_k^{X^{(r)}})'}{\sum_{g=1}^G \hat{\eta}_{gkl}},\\ \mathbf{\Sigma}_{kl}^{Y^{(r)}} &= \frac{\sum_{g=}^G \hat{\eta}_{gkl}(\mathbf{y}_g - \mu_{kl}^{Y^{(r)}})(\mathbf{y}_g - \mu_{kl}^{Y^{(r)}})'}{\sum_{g=1}^G \hat{\eta}_{gkl}}.\end{aligned}$$

The regularized versions of these covariance estimates are

$$\begin{aligned}\tilde{\mathbf{\Sigma}}_{kl}^{YX^{(r)}} &= \frac{\Delta_p^{YX}(\nu) + \mathbf{\Sigma}_{kl}^{YX^{(r)}}n_{kl}}{\nu + n_{kl}},\\ \tilde{\mathbf{\Sigma}}_{kl}^{Y^{(r)}} &= \frac{\Delta_p^{Y}(\nu) + \mathbf{\Sigma}_{kl}^{Y^{(r)}}n_{kl}}{\nu + n_{kl}},\end{aligned}$$

where

$$\begin{aligned}\Delta_p^Y &= \frac{\sum_{i=1}^G(\mathbf{y}_g - \bar{\mathbf{y}})(\mathbf{y}_g - \bar{\mathbf{y}})'}{\sum_{k=1}^K {L_k}^{2/d}},\\ \Delta_p^{YX} &= \frac{\sum_{i=1}^G(\mathbf{y}_g - \bar{\mathbf{y}})(\mathbf{x}_g - \bar{\mathbf{x}})'}{\sum_{k=1}^K {L_k}^{2/d}}.\end{aligned}$$

Then, the conditional mean of $\mathbf{y}_g$ and the covariance matrix are updated as follows:

$$\begin{aligned}\mu_{kl}^{Y|X^{(r)}} &= \mathbf{\Sigma}_{kl}^{XY^{(r)}}(\mathbf{\Sigma}_k^{X^{(r)}})^{-1}(\mathbf{x}_g - \mu_k^{(r)}),\\ \mathbf{\Sigma}_{kl}^{Y|X^{(r)}} &= \mathbf{\Sigma}_{kl}^{Y^{(r)}} - \mathbf{\Sigma}_{kl}^{YX^{(r)}}(\mathbf{\Sigma}_k^{X^{(r)}})^{-1}\mathbf{\Sigma}_{kl}^{XY^{(r)}}.\end{aligned}$$

(b) Similar to the **M-step-Profile-1** step above, for fixed $\mathbf{\Sigma}_{kl}^{Y|X}$, we have a weighted least squares formulation given by

$$\mathbf{y}_g^* = W_L\gamma_{gkl} + \epsilon, \quad cov(\epsilon) = \mathbf{\Sigma}_{kl}^{Y|X},$$

where $\mathbf{y}_g^* = \mathbf{y}_g - \mu_{kl}^{Y|X^{(r)}} - W_{LK}\hat{\alpha}_k^{(r)}$ and W_L represents the lower diagonal block of the W matrix corresponding to L-level parameters whereas W_{LK} represents the lower off-diagonal block of the W matrix.

We obtain

$$\hat{\gamma}_{kl} = \frac{\sum_{g=1}^{G} \hat{\eta}_{gkl} \hat{\gamma}_{gkl}}{\sum_{g=1}^{G} \hat{\eta}_{gkl}},$$

and set $\hat{\beta}_{kl} = (\hat{\alpha}_k, \hat{\gamma}_{kl})$, and $\mu_{kl}^{(r)} = W \hat{\beta}_{kl}$.

2.2 Model selection

2.2.1 Cluster parameterizations and subset selection

Let us first consider the case with K and $\mathbf{L}_K = \{L_k, k = 1, \cdots, K\}$ fixed. We want to select the sparsest representation of each cluster mean. This will enable us to better interpret the meaning of each cluster. For example, is a particular cluster model representing (i) a static cell-line difference, or (ii) a dynamic one, and if so for which time-points do the cell-lines really differ?

Recently, several papers have appeared on the topic of variable selection for model based clustering. These papers focus on the selection of a subset of variables, or dimensions of the feature vector, that can discriminate between cluster components (e.g., Friedman and Meulman (2002), Law et al. (2004), Raftery and Dean (2006), Hoff (2006), Tadesse et al. (2005)).

Raftery et al. (Raftery and Dean; 2006) proposed an iterative algorithm, considering deletions or additions to the set of discriminative variables. Consider the addition of a set of variables. The two models that are compared are; (1) a cluster mixture model for the new set of variables (original set and the set under consideration), and a cluster independent model of the excluded variables, and (2) a cluster model for the original set, with a cluster independent model for the set under consideration and the excluded variables. The decision to accept a new set of variables is made using Bayes factors.

Hoff (Hoff; 2006) models the cluster means with cluster specific contrasts. Let us consider a d-dimensional data set with global mean $\mathbf{u}$ (d-dimensional) and covariance Σ. At the cluster level, we define parameters $\mathbf{u}_k = \mu + \delta^k$, where δ_k represents a set of contrasts between the global mean and the cluster mean. Hoff considers the case where only a subset of the d-dimensional vector δ_k are non-zero, and that this subset may vary across clusters. The model is fit via a hierarchical Bayesian scheme with priors on the cluster specific subsets of non-zero contrasts.

In our parametrization of the cluster means, as outlined in the section below, we deviate from the above approaches. Our parametrization, and the corresponding sparsest representation we select, allows for cluster specific descriptions of contrasts between variables *within* a cluster, as well as *between* clusters. We model all dimensions within the clustering model. However, for each cluster we allow for only a subset of *parameters* to be non-zero. The subset of coefficients that are set to zero do not necessarily correspond to a dimension that is irrelevant for clustering.

How do we then perform subset selection within each cluster model? Clearly, a full combinatorial search of all possible subsets is not feasible. For each combination of subset models, the EM algorithm has to be re-run to adapt to the reduced complexity of some of the clusters. Object posterior probabilities are affected by the cluster specific models.

We take a backward selection approach to selecting the optimal subset models. We begin with the full model for each cluster $\{k, l\}$. We then visit each cluster, one at a time, and threshold the posterior probabilities η_{gkl} to obtain a cluster specific data set of size n_{kl} (or n_k for a 1st level cluster k). We perform backward selection at a 1st level cluster k using only the 1st level data. We formulate the model selection as a generalized linear regression problem, where $x_g = W_K \alpha_k + \epsilon$, $\epsilon \sim N(0, \Sigma_K^X)$. We hold Σ_K fixed during the model selection, and the estimated covariance matrix is used in the weighted least squares fit. We use the local BIC to select the optimal cluster specific model. After backward selection we thus obtain a sparse solution α_k^* for each internal node. We then re-run the EM steps with the sparse restrictions on β (i.e. using a subset of the columns of matrices W_K for each cluster k). Thus we obtain an updated allocation between all $\{k, l\}$ clusters given the selected subset model class.

To perform model selection at the $\{k, l\}$ 2nd level clusters we use the profile likelihood, as was done in the corresponding M-step of the fitting algorithm. For each cluster $\{k, l\}$ we compute the conditional mean $\mu_{l(k)}$ and covariance $\Sigma_{k,l}^{Y|X}$. We can write the profile likelihood in terms of the 2nd level specific parameters only (γ_{kl}). We perform backward selection in a generalized linear regression problem; $y_g^* = W_L \gamma_{kl} + \epsilon_L$, $\epsilon_L \sim N(0, \Sigma_{k,l}^{Y|X})$. y_g^* is defined in supplementary section 2.1, PEM step 3(b). We obtain the optimal sparse solution γ_{kl}^*. We then re-run the EM steps with the sparse restrictions on γ_{kl} (a subset of columns of W_L for each sub-cluster $l(k)$). We thus obtain an updated allocation among all clusters.

Finally, to reduce the impact of such a greedy and directed search, we re-run the whole selection strategy from the most recent allocation, starting yet again from the full model and searching backwards. In practice, we found that iterations of the subset selection algorithm rarely produced a different final result.

We outline the subset selection algorithm here:

I Initialize with the full model at each node z, where z is one from the set of internal ($k = 1, \cdots, K$) or leaf-nodes ($\{k, l\}, k = 1, \cdots, K, l = 1, \cdots, L_k$).
Set the current design matrix of each node z to the full W; $W_K(k)$ for the internal nodes, $W_L(k, l)$ for the leaf-nodes.
(The number of columns of a design matrix, $col(W(z))$, corresponds to the number of non-zero parameters at node z.)
Run the EM-algorithm.

II **(a)** Visit each internal node k, and perform a hard threshold operation on τ_{gk} to obtain the node specific data.

(b) If $W(k)$ is empty, go to the next node k.
Otherwise, perform backward selection for the weighted least squares fit at node k. Obtain the sparse solution α_k^* via the local BIC, and update the current design matrix at node k to $W_K(k) = W_K^*(k)$ (i.e. drop the columns that correspond to $\alpha_k^* = 0$).

(c) Re-run the EM algorithm with the updated $W_K(k)$ constraints.

III **(a)** Visit each leaf node $\{k, l\}$, and perform a hard threshold operation on η_{gkl} to obtain the node specific data.

(b) If $W_L(l(k))$ is empty, go to the next node $\{k, l\}$.
Otherwise, perform backward selection for the weighted least squares fit at node $\{k, l\}$ using the profile likelihood. Obtain the sparse solution γ_{kl}^* via the local BIC, and update the current design matrix at node $\{k, l\}$ to $W_L(l(k)) = W_L^*(l(k))$ (i.e. drop the columns that correspond to $\gamma_{kl}^* = 0$).

(c) Re-run the EM algorithm with the $W_K(k)$ and updated $W_L(l(k))$ constraints.

IV Go to I and iterate until convergence.

2.2.2 Selecting the number of clusters.

The selection of the number of clusters is usually approached as a complexity allocation problem using criteria such as BIC, CIC or MDL (e.g. Fraley and Raftery (2002), Raftery and Dean (2006)). Recently, Zhu and Zhang (2004) developed a general statistical hypothesis testing formulation to select the number of clusters. Here we take the complexity allocation route, using BIC to select the number of clusters. Let us consider a multi-level parametrization where the dimensionality of the data vectors at the 1st level is $Dim(1)$, and at the 2nd level $Dim(2)$. We denote the model coefficients at the 1st level by $\alpha_k, k = \{1, \cdots, K\}$, and the model coefficients at the 2nd level by $\gamma_{kl}, l = \{1, \cdots, L_k\}$ for all $k = \{1, \cdots, K\}$. In the previous section we considered subset model selection for each node $\{k, l\}$ of the multi-level clustering. Thus, the number of non-zero coefficients $\alpha_k \neq 0$ may be less than $Dim(1)$, and similarly for γ_{kl}. We denote the number of non-zero coefficients at each node $\{k, l\}$ by $(dim(\alpha_k), \mathrm{dim}(\gamma_{kl}))$ respectively. We gather all parameters of a multi-level fit into a set $\Theta(K, \mathbf{L}_K) = \{\pi_{kl}, \alpha_k, \gamma kl, \Sigma_{kl}, \forall k = \{1, \cdots, K\}, l = \{1, \cdots, L_k\}\}$. Then the total model complexity is given by

$$p(\Theta(K, \mathbf{L}_K)) = \left[\sum_{k=1}^{K} \left(dim(\alpha_k) + \sum_{l=1}^{L_k} dim(\gamma_{kl}) \right)\right]_{(1)} + \\ + \left[\frac{K Dim(1)(Dim(1) - 1)}{2}\right]_{(2)} + \left[(\sum_{k=1}^{K} L_k) - 1\right]_{(3)}$$

$$\left[\left(\sum_{k=1}^{K} L_k\right)\left(Dim(1)Dim(2)+\frac{Diml(2)(Dim(2)-1)}{2}\right)\right]_{(4)},$$

where term (1) is the number of mean parameters estimated at the 1st and 2nd levels, term (2) is the 1st-level covariance estimates, term (4) is the 2nd-level covariance estimates and cross-covariance estimates between the 1st and 2nd levels, and term (3) is the number of estimated cluster proportions. For each given K and $\mathbf{L}_K$ we can compute the log-likelihood:

$$l(\Theta(K,\mathbf{L}_K))=\sum_{g=1}^{G}\log\left(\sum_{k=1}^{K}\sum_{l=1}^{L_k}\pi_{kl}\phi((\mathbf{x}_g,\mathbf{y}_g);W\beta_{kl},\Sigma_{kl})\right).$$

We then compute the BIC value as

$$BIC(K,\mathbf{L}_K)=-2l(\Theta(K,\mathbf{L}_K))+p(\Theta(K,\mathbf{L}_K))\log(G).$$

We explored several different search strategies for identifying the optimal multi-level model. The best performance was obtained using a backward search. In the flow-chart below, M refers to the total number of clusters ($M=\sum_k L_k$).

```
I   Initialize with the null model $M = 1, L_1 = 0$ and set the $BIC$ to an arbitrarily large value.
II  Set $M = M + 1$.
    (a) Outer loop
        • Set $K = M$ and $\mathbf{L}_K = \{L_k = 1, \forall k = \{1, \cdots, M\}\}$.
          Run the EM algorithm.
          Record the corresponding BIC value: $BIC(new)$.
          Go to Inner Loop II-b.
    (b) Inner Loop
        • Set K = K - 1
          For $b = \{1, \cdots, B\}$
            – group the $M$ 1st level parameters from the single-level clustering (II-a): $(\mu_{\mathbf{k}}, \Sigma_k)$ into $K$ groups. The corresponding grouping defines the set $\mathbf{L}^b_K(new) = \{L^b_k, k = 1, \cdots, K\}$.
            – run the EM algorithm for $K$ and $\mathbf{L}^b_K(new)$ and record $BIC^b(K)$.
        • Set $b^* = argmin_b BIC^b(K)$, and set $BIC(K) = BIC^{b^*}(K)$. Retain the best multi-level clustering with $K$ 1st level clusters and the corresponding grouping $\mathbf{L}_K(new) = \mathbf{L}^{b^*}_K(new)$.
    (c) • If $BIC(K) \geq BIC(new)$ go to step III (the optimal number of sub-clusters has been exceeded).
        • If $BIC(K) < BIC(new)$, accept the best multi-level model model $K$ and the corresponding set $\mathbf{L}_K = \mathbf{L}_K(new)$, $BIC(new) = BIC(K)$.
```

Go to Inner Loop step **II-b**.

III

- If $BIC(new) \geq BIC$, STOP (the optimal number of clusters have been exceeded)
- If $BIC(new) < BIC$, set $BIC = BIC(new)$ and go to **II-a** (consider increasing the total number of clusters).

For both subset selection, and the selection of the number of clusters, we adopt greedy searches. While it is true that such schemes can converge to local optima, a fully exhaustive search is computationally prohibitive. A stochastic search may remedy the problem of local optima. We did not consider stochastic searches here, but do run the full algorithm several times while initiating from different starting values.

2.3 Computational details

2.3.1 Regularizing the cluster covariance estimates

In Fraley and Raftery (2004), a regularized estimate of the cluster covariances are introduces as

$$\tilde{\boldsymbol{\Sigma}}_k^{X^{(r)}} = \frac{\Delta_p^X(\nu_p + d + 2) + \boldsymbol{\Sigma}_k^{X^{(r)}} n_k}{\nu_p + d + 2 + n_k}.$$

The motivation for this regularization comes from assuming a conjugate inverse Wishart prior distribution with scale matrix Δ_0 and degrees of freedom ν_p for $\boldsymbol{\Sigma}_{kl}^X$. Here, Δ_0 is estimated by the plug-in estimator

$$\Delta_p^X = \frac{\sum_{i=1}^{G} (\mathbf{x}_g - \bar{\mathbf{x}})(\mathbf{x}_g - \bar{\mathbf{x}})'}{K^{2/d}},$$

where $\bar{\mathbf{x}}$ represents the componentwise mean vector over all the G genes. ν_p is chosen as $\max\{0, n_{min}\} + d + 2$, where d is the dimension of the data, and n_{min} can be interpreted as the number of observations with variance Δ_p^X that are added to the clustered data.

The scaled global covariance matrix is not always a good choice to shrink toward. Consider a clustering in two dimensions, where K clusters means lie on the 45 degree line, and the cluster covariance are aligned at 135 degrees (i.e. orthogonal to the line connecting the cluster means). The global covariance will be aligned with the 45 degree line. The weighted average between the Δ_p^X and Σ_k^X can thus produce a very different cluster shape, even for moderately large clusters. To reduce the impact of "over-regularizing" the covariance estimates we take a frequentist approach. We numerically test the regularized estimates

$$\Sigma_k^{X^{(r)}} = \frac{\Delta_p^X(\nu) + \boldsymbol{\Sigma}_k^{X^{(r)}} n_k}{\nu + n_k}.$$

with $\nu = 0$ for singularity problems. We increase ν gradually until the regularized estimate is functional. Although this regularization no longer follows the Bayesian framework, we point out that the lack-of-fit of the over-regularized estimate can increase the deviance several orders of magnitude for every fixed number of clusters K, compared with the difference in deviance between different values of K! Thus, an aggressively regularized covariance estimate favors a small number of clusters K.

2.3.2 Starting values and running times

Mixture model fitting implemented via the EM algorithm is sensitive with respect to starting values, and $\mathcal{MIX}_{\mathcal{L}}$ is no exception. We initialize the single-level fit, with M clusters, using the k-means clustering algorithm. Each single-level fit is initialized from several k-means clustering outcomes, and the best fit is reported.

As mentioned above, we explored various multi-level initialization schemes (e.g. forward search, where a 1st level cluster is split in, and backward search, where a cluster is joined to form a 1st level cluster. The best results were obtained with a backward search strategy. We initialize the multi-level fit with a total of M clusters. We run the EM algorithm with $K = M$ and $L_k = 1, \forall k = \{1, \cdots, M\}$. We then cluster the M cluster means and covariances into K clusters, using only parameters defined at the 1st level data dimension, $Dim(1)$. This identifies clusters that can potentially form 1st level clusters, with sub-clusters defined over $Dim(2)$. The k-means clustering of the mean and covariance parameters from the M single-level fit identifies sub-cluster constellations $\mathbf{L}_K = \{L_k, k = 1, \cdots, K\}\}$, where $\sum_k L_k = M$. We run the multi-level EM algorithm from this initialization. To avoid convergence to local optima, we form at least B unique groupings of the M clusters into K 1st level clusters, and run the multi-level fit from all B initializations. The unique groupings are obtained by running k-means on the $Dim(1)$ parameter set repeatedly, and through random perturbations of the cluster allocations. It is absolutely necessary to run the multi-level clustering from several single-level initializations, and several groupings into K 1st level clusters, since the best single-level fit is not guaranteed to generate the best multi-level fit. In practice, we found that $B = 10$ alternative starting values for the single-level fit, and groupings into multi-level initializations, were sufficient. Since the above initialization procedure starts running the multi-level fit with starting values obtained from an unconstrained fit, the first iterations of the profile EM (for $L_k > 1$ for any k, or after subset selection) in general decreases the likelihood. After $1 - 5$ iterations, the EM steps reverse direction, and converge toward a constrained solution. In general, the multi-level fit converged after fewer than 50 iterations, whereas the EM run after subset selection converged after 25 iterations or less.

On an IBM thinkpad X60s, the run-time for a $\mathcal{MIX}_{\mathcal{L}}$ fit with $K = 7$ and $M = 9$ clusters total, including $B = 10$ alternative sub-cluster constellations, and including model selection of cluster parameters β_{kl}, is 2min 24seconds on average. This is using R version 2.2.1. A full model search requires a run such as the above to be applied for $M = M_{min}, \cdots, M_{max}$ and $K = M, \cdots, 1$. We considered $M_{min} = 3$ and $M_{max} = 15$ which allowed for a clear minimum BIC value to be identified. A complete run of the algorithm ($M \in [3, 15]$, $K \in [M, 1]$ with $B = 10$ alternative 1st level constellations, and including subset model selection for all clusters) took 1hour 22minutes on average. We ran each complete model search 10 times, initiating from different starting values. The R code is available from the corresponding author upon request.

3 Case study: time course gene expression data of proliferating stem cell lines

3.1 Pre-processing

Regulated mRNAs during differentiation of rat neural stem cells were analyzed using the ABI1700 microarray platform. This microarray, while technically advanced, suffers from the difficulty of integrating hybridization results into public databases for systems level analysis. This is particularly true for the rat array since many of the probes were designed for transcripts based on predicted human and mouse homologs. We analyzed a subset of 15,111 probes that were annotated with high level of confidence. Data extracted from the scanned arrays was processed using R/BioConductor scripts provided by Applied Biosystems. Raw data were quantile normalized (BM. et al. (2003)). The data set consisted of 6 separate experimental conditions (3 time points for each of two cell-lines), with 3 technical replicates each. A linear model was fit to the data, estimating both cell line and differentiation effects. We ranked genes based on the Welch F-statistic, and retained the 780 genes for which the Benjamini-Hochberg adjusted p-value was below 1%. We chose this conservative significance threshold to focus on the genes for which we were reasonably confident the differential effects over time and/or cell-line was real. However, a different testing procedure, e.g. the moderated F-test (Smyth (2004)), results in a slightly different gene list. With the conservative 1% threshold, the moderated F-test also selected 708 out of the 780 genes we focus on in the paper.

3.2 Examining the gene functional annotation of identified clusters

Supplementary Tables 1 to 4 report top 10 significant GO categories for each of the 9 clusters obtained with $\mathcal{MIX}_{\mathcal{L}}$.

Clusters 1 and 2: Cluster 1 corresponds to a set of genes that start out at baseline for both cell-lines, i.e. there is no pre-programming activity. In the glial like population, the expression of these genes increase rapidly over the course of the experiment. In supplementary Table 1 (middle) we see that some of these genes are in fact annotated as specific to gliogenesis. The set of genes in cluster 2 are always overexpressed in the glia population compared with neurons, and the expression in glia increases over time. Supplementary Table 1 (bottom) identifies this set of genes as appearing related to astrocyte formation (one type of glia), as well as transporter activity (of which chloride transport is a glial function).

Clusters 3 and 4: These clusters form a set of sub-clusters with neuron specific differential expression. To interpret these clusters, we rely on the following fact: it is known (from staining experiments) that the glial like cultures are heterogeneous. That is, in the cultures

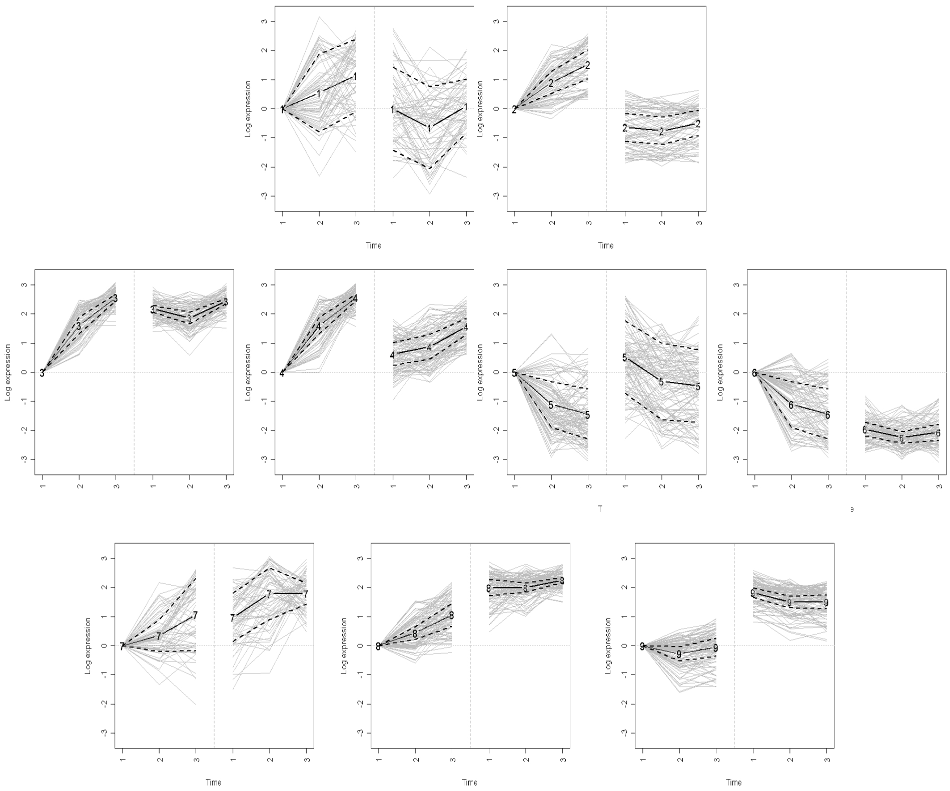


Figure 1: The 9 clusters generated by the best $\mathcal{MIX}_{\mathcal{L}}$ fit. The solid lines represent the cluster means, the dashed lines the mean plus/minus the standard deviation (point wise). The gray lines are the individual genes allocated to each cluster.

labeled ”glial like” we see a mixture of glia and neurons. In contrast, the neuron population is largely homogeneous, and almost all cells in these cultures become neurons.

Cluster 3 represents genes that start off high in neurons, whereas the set of genes in glia population approach (from below) neuron specific levels of activity. Cluster 3 thus highlights genes that are believed to be specific to neuron formation. These genes are activated in the glia culture among cells that converge to neurons (Goff et al. (2006)). Looking in supplementary Table 2, several GO categories that are overrepresented in cluster 3 correspond to neuron and neurite development, as well as activation of other neuron maturation processes (e.g. regulated by NFkappa-B). The neuron population has been ’pre-programmed’ to this cell fate, and these genes are thus highly expressed throughout the experiment for these cultures.

Cluster 4 represent genes that start off more highly expressed in the neuron population. In the glial like population we again pick up the gene activity associated with the sub-population converging to neurons. For these genes, activity is increasing in both populations. The GO categories associated with this cluster (supplementary Table 2) include growth cone, cy-

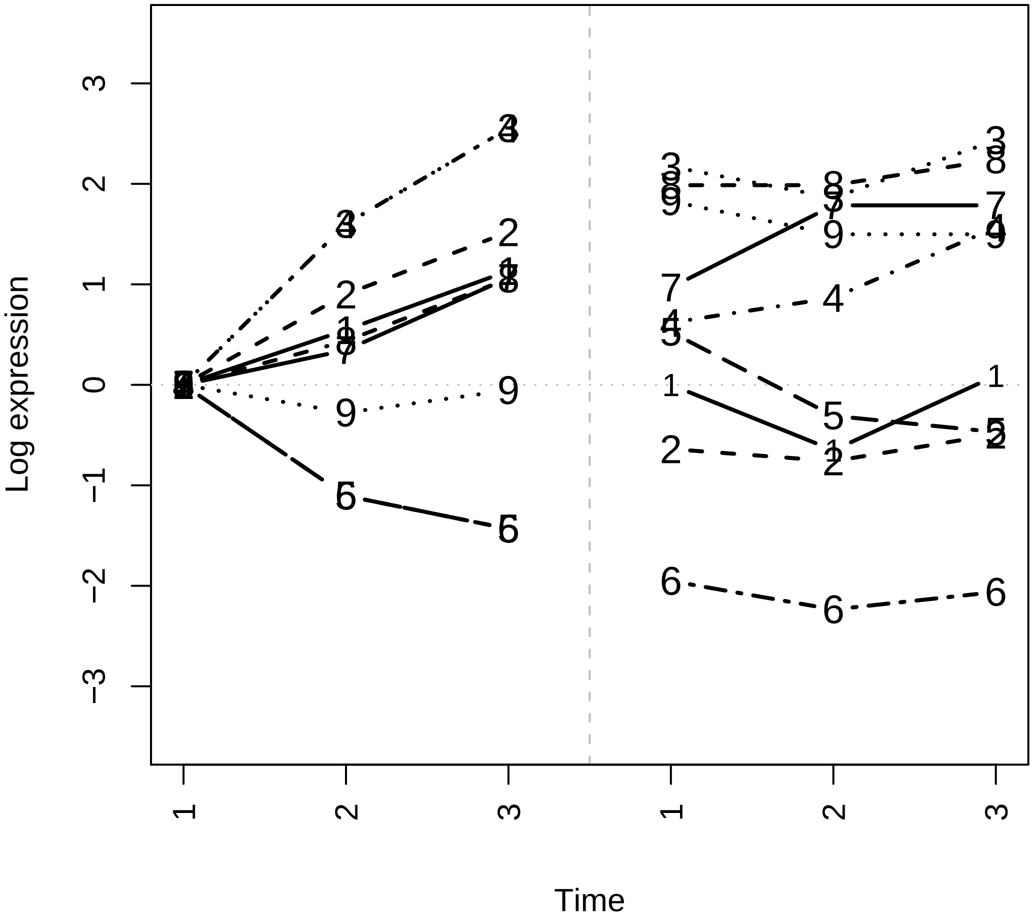


Figure 2: Cluster mean profiles of the best multi-level fit $K = 7$, $M = 9$, with two sets of sub-clusters (parametrization W_{III}).

toskeleton, and microtubule binding, which are associated with dendrite formation (Charych et al. (2006)). Dendrites are part of the more complex neuron structure which explains the later activity of these genes compared with the more basic neuronal developmental processes identified in cluster 3.

Clusters 5 and 6: Clusters 5 and 6 again correspond to sub-clusters that are specific to activity in the neuron population. Cluster 5 corresponds to an overall higher activity in neurons compared with glia, and this activity is decreased in both populations. Cluster 6 corresponds to genes whose activity is always lower in neurons compared with glia, where again the glial activity is decreasing. In cluster 6, the glial gene expression is converging toward the neuron expression, suggesting that these genes are (de-)activated in the sub-population of cells in the glial population that form neurons. Cluster 5 is primarily associated with acid metabolism, whereas cluster 6 is associated with acid synthesis. Acid metabolism is a process by which neurons generate neurotransmitters. Glial cells are believed to synthesize some acids that assist in neuron development and migration. Therefore, one can largely associate genes in cluster 5 with neuron specific activity, which explains the under expression in glia.

Cluster 7, 8 and 9: Cluster 7 corresponds to a more rapid increase in expression in the neuron population compared with glia (as indicated by the selected cluster model with no time effect in neurons between $t = 1$ and $t = 3$). This cluster is the most sparsely populated, with a large cluster variance. The GO terms associated with these clusters are not easy to interpret, with the exception of "morphogenensis". Cluster 8 is associated with expression upregulated in the neuron population compared with the glial population at the onset. The glial expression is slowly converging toward the neuron population. Many of the top GO categories associated with cluster 8 are primarily centered on high level neuron functions (e.g. synaptic transmission). Cluster 9 consists of genes that are upregulated in neurons compared with glia at all times. The top GO categories in this cluster are linked to phosphorus binding. Phosphor is an activator of BDNF binding, a primary regulator of dendritic branching at the cell body (primary branching). If we compare clusters 9 and 4, we see that primary branching (cluster 9) is activated early in neurons ($t = 0$) and then decreasing, whereas genes associated with dendritic formation and higher levels of branching (cluster 4) is associated with increasing gene expression over the course of the experiment.

3.3 Mining the clustering results

Regulation of gene expression in a condition specific manner heavily relies on the activities of the transcription factors, i.e., DNA binding proteins, and mainly on their recognition of DNA in a sequence specific manner. The sites that the transcription factors bind to on DNA are usually 5-20 base pairs long and are referred to as DNA binding motifs or regulatory motifs. Identification of these sites is a challenging and not completely solved computational biology problem. Recently, several methods (Bussemaker et al.; 2001; Keleş et al.; 2002; Conlon et al.; 2003) illustrated that addressing this problem in a feature/variable selection framework is a powerful way of elucidating experiment/class specific binding sites. In these approaches, the key idea is to use regulatory motifs as covariates and generally gene expression (expressed versus not expressed) as an outcome of interest. Then, a linear regression model is typically built to link the motifs to the outcome. More recently, non-parametric regression approaches like logic regression (Ruczinski et al.; 2003) and MARS (Friedman; 1991) are also employed (Keleş et al.; 2004; Das et al.; 2004) instead of linear regression models.

In our analysis, we use the cluster assignment of each gene as a class label and consider all pairwise comparisons of the clusters in a logistic regression framework. Covariates in these regression models are based on the transcription factor database TRANSFAC (Wingender; 1994). For each gene, we construct a set of covariates utilizing the position specific probability matrix (PSPM) representations of the regulatory motifs. This representation corresponds to a 4 by length of the motif matrix where each (i,j)th entry corresponds to the probability of observing the ith nucleotide at the jth position of the motif (see Stormo (2000) for a comprehensive review of binding site representations). In order to construct the covariates, we extract first 1000 base pairs upstream of the transcription start site, i.e., regulatory region, for each gene. Then, these regions are scanned by each of the 795 regulatory motif PSPMs from TRANSFAC using the PATSER tool (Hertz and Stormo; 1999). As a result, we obtain, for each subsequence in the upstream sequence, a likelihood ratio score representing the likelihood of the subsequence under the regulatory motif model as opposed to a background model that assigns (0.3, 0.2, 0.2, 0.3) probabilities to the nucleotides A, C, G, and T, respectively.

The score of the best matching subsequence within the regulatory region is used as a covariate. Due to the high dimensional covariate space, elaborate variable selection schemes are required to identify the most relevant features.We utilize the recently developed `GLMpath` algorithm of (Park and Hastie; 2006). `GLMpath` fits L_1 regularized generalized linear models by solving the following minimization problem:

$$\hat{\beta}(\lambda) = \text{argmin}\{-\log L(\mathbf{y};\beta) + \lambda||\beta||_1\},$$

where λ is the regularization path and $L(\mathbf{y};\beta)$ represent the logistic regression likelihood parameterized by regression coefficients β in our framework. In our application, the regularization parameter is based on 5-fold cross-validation.

The number of discriminating position weight matrices identified for each pairwise comparison ranged from 0 to 9. The positions weight matrices identified from each pairwise comparison is displayed in supplementary Table 5. The empty cells between any pairwise comparison corresponds to an intercept only logistic regression model selected by `GLMpath`. Since TRANSFAC does not span the space of all position weight matrices relevant for rat, we indeed expect some of the pairwise comparisons not to have any discriminating position weight matrices. It has been previously noticed that although a linear regression analysis of gene expression as a function of regulatory sequences can elucidate major regulatory sequences affecting gene expression, such an analysis has typically low predictive power (Bussemaker et al.; 2001; Keleş et al.; 2002). Using a summary measure of gene expression, namely the clustering results, behaves similarly. Although we consider all pairwise comparisons, our main interest lies in the comparisons between the second level sub-clusters of the multi-level fit. As depicted in Figure 2(b) in the paper, sub-clusters 3 and 4 and sub-clusters 5 and 6 are obtained via a split in the second cell line. Examining the position weight matrices selected for these comparisons, we note that M00133 matrix which is identified in the comparison of clusters 3 and 4 corresponds to transcription factor Tst-1. Tst-1 is a member of the POU domain gene family and is expressed in specific neurons and in myelinating glia in the mammalian nervous system. This transcription factor, also called MeF2, has been identified by our collaborators in an independent biochemistry experiment (Goff et al. (2006)). MeF2 is believed to be a target of a neurogenesis regulating microRNA, and its association with a neuron specific expression pattern in our study lends support to this biological hypothesis. Further study of the identified neuron-specific transcription factors are now underway in collaboration with Professor R. Hart at Rutgers.

3.4 Simulation Results

In supplementary Figure 3, and Table 3 in the paper, we summarize the results from the simulation study. Supplementary Figure 3 (a) shows that indeed the BIC is always reduced after model selection, even after the EM steps are rerun with the selected parameter constraints. Thus, performing subset selection on a cluster by cluster basis, using the local BIC, always produces a better model in terms of the BIC validation index. In supplementary Figure 3 (b) we depict a histogram of the total number of selection errors (across all clusters) for the 50 simulated data sets. In the case of the single level model ($Mod(1)$) (top panel), the multi-level fit (MF) generates fewer selection errors than the single-level fit. This is an intriguing result, given that the multi-level fit for which these errors are compared is constrained to only have $L_k = 1$, i.e. no sub-clusters. The reason for the improved selection performance is that we visit internal (1st level) clusters, and leaf (2nd level) clusters separately, and are thus performing subset selection on $2 * (K = 8)$ clusters in the multi-level fit, compared with $K = 8$ clusters in the single-level fit.

In supplementary Figure 3 (c) and (d) (lower panel), we depict the BIC reduction of the multi-level fit compared with the single-level fit, before and after the selection of the number of clusters, as well as after subset selection. In supplementary Figure 3 (c) we illustrate the results for the $Mod(1)$ (single-level fit is correct). We see that before subset selection, the single- and multi-level fits perform equally well (no difference in BIC value). After model selection, due to the increased number of clusters considered separately in the selection procedure (as stated above), the multi-level fit improves on the single-level fit. In supplementary Figure 3 (d), we illustrate the results from the $Mod(2)$ simulation (multi-level fit is correct). Here, the multi-level fit improves on the single-level fit both before and after selection. Occasionally, the multi-level fit will perform worse than the single-level fit. This is a direct result of the limitations of the simulation study. The multi-level fits require a more careful exploration across multiple starting values. However, for ease of computation, the single- and multi-level fits were only run from one starting value in the simulation study, which favors the single-level fit. Still, with the exception of a few rare cases, the multi-level fit provides a better solution for $Mod(2)$ data. The histograms in supplementary Figure 3 (b) (bottom panel) shows that the total number of selection errors is yet again smaller for the multi-level fit (MF).

References

BM., B., Irizarry, R., Astrand, M. and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics (Oxford, England)* **19**: 185–193.

Bussemaker, H., Li, H. and Siggia, E. (2001). Regulatory element detection using correlation with expression, *Nature Genetics* **27**: 167–171.

Charych, E. I., Akum, B. F., Goldberg, J., Jornsten, R. J., Rongo, C., Zheng, J. Q. and

Firestein, B. L. (2006). Activity-independent regulation of dendrite patterning by postsynaptic density protein psd-95, *Journal of Neuroscience* **26(40)**: 10164–76.

Conlon, E., Liu, X., Lieb, J. and Liu, J. (2003). Integrating regulatory motif discovery and genome-wide expression analy sis, *Proceedings of the National Academy of Sciences USA* **100**: 3339–3344.

Das, D., Banerjee, N. and Zhang, M. Q. (2004). Interacting models of cooperative gene regulation, *Proceedings of National Academy of Science, USA* **101**(46): 16234–16239.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation, *Journal of the American Statistical Association* **97**: 611–631.

Fraley, C. and Raftery, A. E. (2004). Bayesian regularization for normal mixture estimation and model-based clustering, *Technical Report 486*, University of Washington.

Friedman, J. H. (1991). Multivariate adaptive regression splines, *Annals of Statistics* **19**: 1–141.

Friedman, J. and Meulman, J. (2002). Clustering objects on subsets of attributes, *Technical report*, Department of Statistics, Stanford.

Goff, L. A., Davila1, J., Jörnsten, R., Keles, S., Li, H., Grumet, M. and Hart, R. P. (2006). Co-regulation of a single mir-9 locus and the adjacent mef2c gene during neuronal differentiation in neural stem cells., *submitted to Journal of Neuroscience* .

Hertz, G. Z. and Stormo, G. D. (1999). Identifying dna and protein patterns with statistically significant alignments of multiple sequences, *Bioinformatics* **15**(7): 563–577.

Hoff, P. (2006). Model-based subspace clustering, To appear in Bayesian Analysis.

Keleş, S., van der Laan, M. and Eisen, M. (2002). Identification of regulatory elements using a feature selection method, *Bioinformatics* **18**(9): 1167–1175.

Keleş, S., van der Laan, M. and Vulpe, C. (2004). Regulatory motif finding by logic regression, *Bioinformatics* **20**(16): 2799–2811.

Law, M. H., Figueiredo, M. A. and Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models, *IEEE Pattern Analysis and Machine Intelligence* **26**(9): 1154–1166.

Park, M. and Hastie, T. (2006). An l1 regularization-path algorithm for generalized linear models. a generalization of the lars algorithm for glms and the cox proportional hazard model. `http://www-stat.stanford.edu/~hastie/Papers/glmpath.pdf`.

Raftery, A. and Dean, N. (2006). Variable selection for model-based clustering, To appear in the *Journal of the American Statistical Association.*

Ruczinski, I., C., K. and M.L., L. (2003). Logic regression, *Journal of Computational and Graphical Statistics* **12**(3): 475–511.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments, *Statistical Applications in Genetics and Molecular Biology* **3**(1): Article 3.

Stormo, G. D. (2000). DNA binding sites: representation and discovery, *Bioinformatics* **16**(1): 16–23.

Tadesse, M. G., Sha, N. and Vannucci, M. (2005). Bayesian variable selection in clustering high-dimensional data., *Journal of the American Statistical Association* **100**: 602–617.

Wingender, E. (1994). Recognition of regulatory regions in genomic sequences, *Journal of Biotechnology* **35**: 273–280. `http://transfac.gbf.de/`.

Zhu, H. and Zhang, H. (2004). Hypothesis testing in mixture regression models, *Journal of the Royal Statistical Society - Series B* **66**(1): 3–16.

All clusters vs GO data base	
GO0048731	”System development”
GO0007399	”Nervous system development”
GO0030154	”Cell differentiation”
GO0006928	”Cell motility”
GO0051674	”Location of cell”
GO0040011	”Locomotion”
GO0022008	”Neurogenesis”
GO0051606	- ”Detection of stimulus”
GO0009582	- ”Detection of abiotic stimulus”
GO0030182	”Neuron differentiation”
Cluster 1 vs All clusters	
GO0006836	”Neurotransmitter transport”
GO0042063	”Gliogenesis”
GO0010001	”Glial cell differentiation”
GO0007399	”Nervous system development”
GO0031324	”Neg. regulation of cell metabolism”
GO0048737	”System development”
GO0006357	”Neg. reg. RNA polymerase transcription”
GO0001504	”Neurotransmitter uptake”
GO0048469	”Cell maturation”
GO0001764	”Neuron migration”
Cluster 2 vs All clusters	
GO0015290	”El.chem transport activity”
GO0015291	”Porter activity”
GO0015293	”Symporter actitivy”
GO0005416	”Amino acid symporter activity”
GO0048143	”Astrocyte formation”
GO0015103	”Anion transport activity”
GO0006820	”Anion transport”
GO0015380	”Anion exchange activity”
GO0015108	”Chloride transporter activity”
GO0015297	”Antiporter activity”

Table 1: Top 10 GO categories of all clusters, and clusters 1 and 2.

Cluster 3 vs All clusters	
GO0005694	”Chromosome”
GO0009966	”Reg. signal transduction”
GO0030900	”Forebrain development”
GO0007249	”NFkappa-B cascade”
GO0031175	”Neurite development”
GO0048666	”Neuron development”
GO0000785	”Chromatin”
GO0044427	”Chromosomal part”
GO0007242	”Intracell. signal cascade”
GO0007409	”Axonogenesis”
Cluster 4 vs All clusters	
GO0030427	”Site of polarized cone”
GO0030426	”Growth cone”
GO0015631	”Tubulin binding”
GO0005856	”Cytoskeleton”
GO0008017	”Microtubule binding”
GO0030018	”Z-disc”
GO0005886	”Plasma membrane”
GO0000267	”Cell fraction”
GO0044228	”Non-membrane-bound organelle”
GO0017111	”Nucleoside-triophasphate act.”

Table 2: Top 10 GO categories for clusters 3 and 4.

Cluster 5 vs All clusters	
GO0006767	”Vitamin metabolism”
GO0005739	”Mitochondria”
GO0019752	”Carb. acid metabolism”
GO0006082	”Organic acid metabolism”
GO0031975	”Envelope”
GO0031967	”Organelle envelope”
GO0044237	”Cell metabolism”
GO0043170	”Macromolecule metabolism”
GO0009058	”Biosynthesis”
GO0006865	”Amino acid transport”
Cluster 6 vs All clusters	
GO0044272	”Sulfur compound biosynthesis”
GO0008652	”Amino acid biosynthesis”
GO0000097	”Sulfur amino acid biosynthesis”
GO0006092	”Pathway of carbohydrate metabolism”
GO0050794	- ”Neg. reg. cell process”
GO0008217	”Blood pressure regulation”
GO0008202	”Steroid metabolism”
GO0005624	”Membrane fraction”
GO0005515	- ”Protein binding”
GO0000267	”Cell fraction”

Table 3: Top 10 GO categories for clusters 5 and 6.

Cluster 7 vs All clusters	
GO0048729	”Morphogenesis”
GO0050874	”Tissue development”
GO0009605	”Response to external stimulus”
GO0016042	”Lipid catabolism”
GO0050875	”Organ. phys. process”
GO0050896	”Response to stimulus”
GO0008081	”Phospholiric dieter hydrolase activity”
GO0042330	”Taxis”
GO0006935	”Chemotaxis”
GO0005543	”Phospholipid binding”

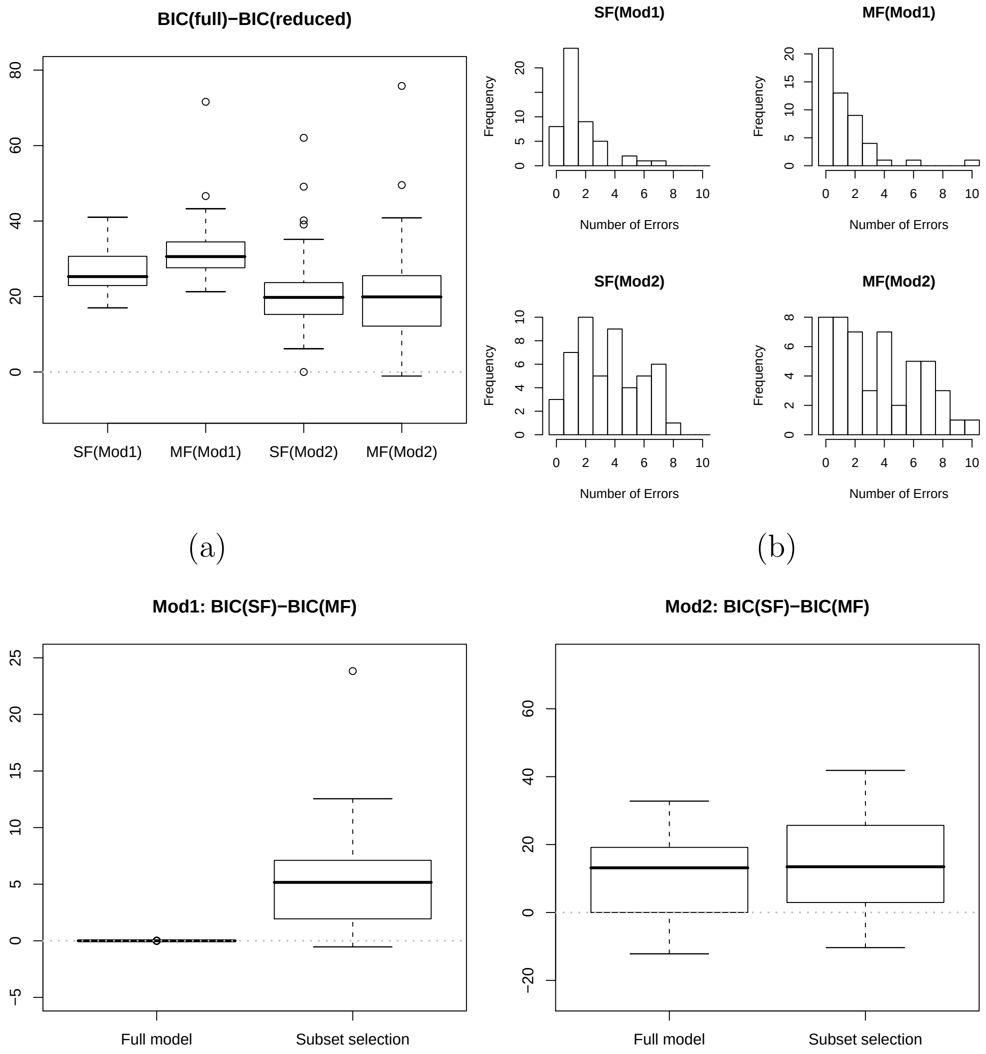
Cluster 8 vs All clusters	
GO0044421	”Extracell. region”
GO0043235	”Receptor complex”
GO0004720	”Protein-oxidase activity”
GO0007270	”Nerve-nerve synaptic transmission”
GO0044238	- ”Primary metabolism”
GO0005615	”Extracellular space”
GO0009653	”Morphogenesis”
GO0007271	Synaptic transmission
GO0005102	Receptor binding
GO0000902	Cellular morphogenesis

Cluster 9 vs All clusters	
GO0006797	”Phosphorus metabolism”
GO0006796	”Phosphate metabolism”
GO0006350	”Transcription”
GO0045449	”Reg. of transcription”
GO0006351	”DNA-dependent transcription”
GO0019219	”Reg. of nucleic acid metabolism”
GO0006468	”Protein amino acid phosphorylation”
GO0006464	”Protein modification”
GO0043412	”Biopolymer modification”
GO0044237	”Cellular metabolism”

Table 4: Top 10 GO categories for clusters 7, 8 and 9.

	2	3	4	5	6	7	8	9
1		M00115	M00115	M00303		M00339	M00197, M00701	M00257, M00446
2		M00639		M00334		M00227, M00257	M00218, M00257	M00257, M00334 M00377, M00503
3			M00100, M00133 M00320, M00361 M00362, M00721 M00733, M01013 M01043				M00197, M00357 M00461, M00717 M01087	M00728
4					M00362	M00721, M00733 M01013, M01043		
5					M00953	M00141, M00252 M00922, M01007		
6								M00241, M01086
7							M00274, M00291 M00422, M01007	M00189, M00252 M00291, M00339 M00340, M00354 M00454, M00922 M00942, M01007
8								M00016, M00179 M01078

Table 5: *Position weights matrices selected by the GLMpath algorithm.* 5-fold cross-validation is employed to select the optimal regularization parameter in each of the logistic regression models.



(a)

(b)

(c)

(d)

Figure 3: (a) The BIC value of the full model minus the BIC value after subset selection for both simulation settings: $(Mod(1), Mod(2))$, and both fitting strategies (single- (SF) and multi-level (MF) fits). The BIC is always smaller after subset selection. (b) Histograms of the total number of subset selection errors for the $Mod(1)$ data (40 parameters total) (top panel) and the $Mod(2)$ data (41 parameters total) (lower panel). The multi-level fit produce fewer selection errors in both cases. (c) The BIC of the single-level fit minus the BIC of the multi-level fit for $Mod(1)$ data, before and after subset selection. After subset selection, the multi-level fit improves on the single-level fit, even when the single-level model is correct. (d) The BIC of the single-level fit minus the BIC of the multi-level fit for $Mod(2)$ data, before and after subset selection. The multi-level fit improves on the single-level fit in almost all cases.