Chapter 5

Analysis of Gene Expression in Lymphoid Development

The study of gene regulatory mechanisms controlling cell proliferation and differentiation is central in developmental biology. In particular, the development of lymphoid cells is well studied, as individual cell populations are easy to obtain and due to clinical interest [140, 177]. In Lymphoid development [25], all starts with the Hematopoietic stem cell (HSC), which differentiates into the Lymphoid progenitor, and later into B-cell, T-cell or Natural Killer cell lineages (see Figure 5.1 for a developmental tree). Recently, several studies have analyzed expression profiling of lymphoid cells in their distinguishable developmental stages [3, 34, 98, 100, 105, 156, 165, 220, 229]. Our main focus is on the analysis of patterns of gene expression in the distinct stages of the developmental tree, the developmental profiles of genes. In particular, we are interested in finding groups of genes displaying a particular pattern of expression, e.g., over-expression in T cells but under-expression in B cells.

As one of the major contribution of this thesis, we propose here a method for analyzing patterns of gene expressions in the course of development. Ideally, such method should exploit inherent dependencies arising from the data, as in methods for analyzing gene expression time-courses (see Chapter 4). We assume that, in development, the sequence of changes from a stem cell to a particular mature cell, as described by a developmental tree, are the most important in modeling gene expression from developmental processes. Motivated by this, we propose dependence trees (`DTree`) to model expression during the course of development [50]. We investigate here two approaches for obtaining the structure of dependence trees. In the first approach, we assume that the structure of the dependence tree is equal to the developmental tree as known by the biologists [50]. In a second approach, we additionally estimate the dependence tree structure from the data [49].

To find groups of co-expressed developmental profiles we use dependence trees in a mixture model [143]. Also, to minimize problems related to over-fitting, we propose Maximum-a-posteriori (MAP) estimates of parameters [80]. By doing so, we obtain a robust and flexible statistical model for clustering genome-wide mRNA expression data sets, which takes the intrinsic dependencies between developmental stages explicitly into account.

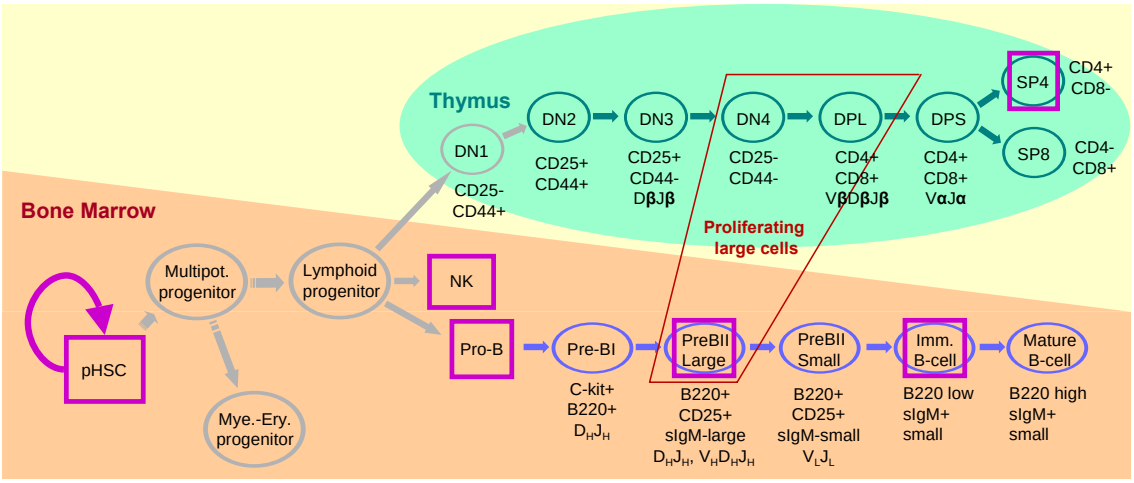


Figure 5.1: *Schematic view of lymphocyte cell development. Developmental stages are depicted as nodes and arrows indicate transition from one stage to another, i.e., specialization. Self-renewing hematopoietic stem cells give rise to T cells in the thymus (green), B cells in the bone marrow (blue) and natural killer cells (NK) via intermediate stages. DN stands for CD4-/CD8- double negative cells, DPL for CD4+/CD8+ double positive large cells, and DPS for CD4+/CD8+ double positive small cells. Cell surface antigens and rearrangement events are partially annotated. Some expression data sets investigated in this Chapter are denoted as follows: green ovals for T Cell and blue ovals for B Cell.*

This chapter is organized as follows. In Section 5.1, we give an overview of related work. Then, we present the dependence tree and the estimation of its parameters in Section 5.2. In Section 5.3, we describe mixtures of dependence trees, and derive the parameters of the MAP estimates (Section 5.3.2). Next, in Section 5.4, we show the results of the analysis of gene expression from lymphoid development. For the mixture of dependence trees with fixed tree structures (Section 5.4.1), we analyze two detailed data sets from B cells [100] and T cells [99]. Furthermore, we explore plausible regulatory roles of microRNAs known to be involved in hematopoiesis. For mixture of dependence trees with estimated structures (Section 5.4.2), we analyze a gene expression compendia with data from hematopoietic stem cells, T cells, B cells and Natural Killer cells. We perform a comparison of several clustering methods on a score based on enrichment analysis of biological pathways. For both methods, results on simulated data show the conditions under which our method has advantages. In Chapter 7, we present final remarks and future work.

5.1 Related Work

Dependence trees were first introduced for discrete variables by Chow and Liu [43], which showed that efficient computation using a maximum weight spanning tree algorithm is

possible. They applied the method for pattern recognition of handwritten digits. Mixtures of dependence trees were first proposed in [148]. The authors also proposed extensions to the basic structure estimation algorithm from [43] for sparse data and the use of priors in the tree structure. This also allowed forests (or disconnected trees) to be estimated. It was also shown that the estimated structures of the dependence trees were a good indicator of relevant dependencies between variables. Both studies [43, 148], however, were only concerned with discrete variables, in contrast to our approach, which regards continuous variables.

Another closely related method is the mixture of directed acyclic graphs (DAG) [213]. Indeed, the mixture of DAGs is a more general graphical model than the mixture of dependence trees. The use of DAGs as component models allows to model high order dependencies. However, there is no exact solution for the structure estimation of DAGs. Thus, its estimation is based on heuristics and requires larger computational effort than mixture of dependence trees. Another related research field is the estimation of covariance matrices with zero entries. In [40], an iterative conditional fitting method was applied for computing sparse covariance matrices from arbitrary undirected graphs. While the method obtained better estimates than classical statistical approaches, such as [7], it does not offer a solution for inferring the graph structure. In [182], a similar problem in the context of gene dependence (or association) networks was investigated. The authors applied a shrinkage factor in an efficient way for defining zero entries in the covariance matrix, while keeping it well-conditioned. Both methods have a high computational cost. They are also not able to find association networks, which are specific for particular gene modules, as performed by mixtures of dependence trees.

In the context of mixtures, our method represents an alternative to the parameterization of the covariance matrix of a mixture of multivariate Gaussians (MoG) not previously characterized [37, 79] (see Section 5.2.4 for a discussion). When computing MLE estimates, the dependence tree model essentially imposes zeros in the inverse of the covariance matrix reducing the number of free parameters to $O(L)$. If we considered all the covariances between observations for L developmental stages, it would be straightforward to represent the data distribution by an L-variate Gaussian model with full covariance matrix. However, this parameterization has $O(L^2)$ parameters, which are often unreliable even for small values of L. Moreover, the parameter estimation is prone to over-fit to outliers often found in noisy and scarce data [143]. This was also indicated in our results with simulated data (Section 5.4.1 and Section 5.4.2), where mixtures of Gaussians with full covariance matrix were outperformed by most of the methods. Additionally, in a study in the context of gene expression time courses [232], MoG with full covariance matrix was outperformed by simpler parameterizations of the covariance matrices.

The estimation of the structure of mutagenic trees is a related problem in bioinformatics [62, 63, 223]. In this application, one is interested in inferring the mutation events occurring in cells, such as cancer, which follow a tree-like event structure. For this particular problem, the root is known a priori (a wild type cell without mutations) and only

variables with observed mutation events are included as nodes in the tree [63]. The tree structure is estimated with a maximum weight branching algorithm (Edmonds' branching algorithm [46]). Recently, mixture of mutagenic trees, which combined the individual tree estimation from [63] with the EM algorithm, was applied to infer mutation events in HIV strains [23].

The problem approached in this chapter is closely related to the gene expression time-course analysis discussed in Chapter 4. Dependence trees can also be used for analyzing short time courses. We can define the dependence tree structure to be a linear chain connecting consecutive time points. In this scenario, `DTrees` will model only first-order temporal dependencies, but ignore higher order dependencies often present in gene expression time-courses (see Section 4.2.1). On the other hand, models employed in time course analysis [14, 185] cannot be extended to modeling tree like dependency structures arising in developmental processes.

Mixture of dependence trees with estimated structure has some relation to bi-clustering. Bi-clustering methods find not only co-expressed genes but also similarity of expression in the biological conditions. However, bi-clustering methods do not make explicit use of any dependencies (developmental or temporal) in these data sets (see [139] for a survey on bi-clustering algorithms). One of such method, Samba [210], is graph-based and finds strongly connected subgraphs in a bi-partite graph. The bi-partite graph has genes and biological conditions as nodes. The edges between nodes representing genes and biological conditions are weighted proportional to the gene expression value of the given gene in that particular biological condition. Another relevant approach is the use of a non-negative matrix factorization (NMF) [31]. This method decomposes the gene expression matrix in two matrices: one representing the K most significant "meta-conditions" and the other the K most significant "meta-genes". The authors proposed a consensus clustering method for choosing K (or the number of clusters) automatically and minimizing problems related to the random initialization of the method.

Regarding lymphoid development, lymphocyte cell populations can be purified by fluorescence activated cell sorting (FACS) exploiting the large variety of cell surface antigens, which appear in specific order during differentiation as the result of a linear sequence of genomic rearrangements at the T and B cell receptor loci [98, 100]. Based on this, lineage-specific expression and roles of transcription factors have been studied extensively [140, 177, 224]. Recently, a new class of regulatory RNAs, microRNAs, have been identified as being involved in lymphocyte cell development [41, 151, 171].

Several studies [3, 34, 98, 100, 105, 156, 165, 220, 229] have combined FACS mediated cell sorting and mRNA expression profiling to derive a more comprehensive picture of the lymphocytes in distinguishable developmental stages. Nevertheless, prior work on the analysis of gene expression from lymphoid development relies mostly on classical clustering methods, such as self-organizing maps [98, 100], hierarchical clustering [156, 220], k-means [3], principal component analysis (PCA) [229] or on performing tests of differential expression between cell types of interest [165]. One particular interesting study was pro-

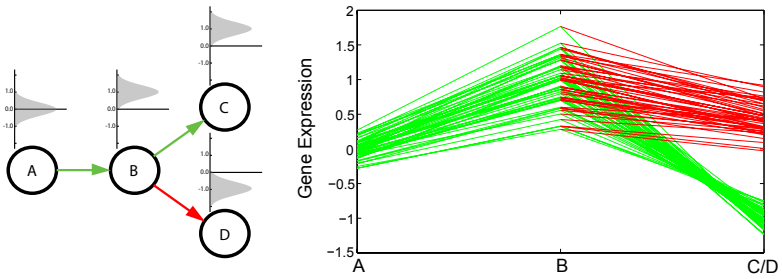


Figure 5.2: *Example of a simple developmental tree and a group of developmental profiles. On the left, we depict a simple developmental tree, where arrows represent dependencies between variables. Above each tree variable, we depict a pdf related to it. On the right, we display the gene expression values (y-axis) in the distinct development stages (x-axis). Each line corresponds to the developmental profile of a given gene of a particular path of the tree on the left, as in a time-course plot. Distinct paths have different colors, according to the tree on the left. In this particular example, we have the path A, B and C in green and B and D in red. By superimposing the lines corresponding to paths B to C and B to D, we can contrast the differences in expression values of genes in these two alternative differentiation lineages.*

posed in [105], where several publicly available data from lymphoid cells were combined and made available for further analyses through an interactive web tool. The authors applied PCA analysis to explore similarities of lymphoid cells based on their gene expression signatures. Furthermore, a simple method based on the correlation measure was used for inferring "networks" of genes. However, that work did not address any developmental aspect of lymphoid cells, as it was restricted to gene expression profiles from lymphoid cells at mature or immature cell stages (later developmental stages). Other studies concentrated on small-scale data, where selected genes are used to infer regulatory networks. One of these studies applied a state-space model to infer networks of T cell activation [173]. Troncale and colleagues adopted Petri Nets to model and infer regulatory networks of early pHSC development [216], while Basso and colleagues proposed a novel algorithm for a similar task [18].

5.2 Dependence Trees

The main assumption underlying dependence trees (`DTree`) is that expression levels of a particular developmental stage depend primarily on expression levels of the immediately preceding stage. For example, given the tree structure depicted in Figure 5.2, we assume the following approximation of the joint probability density function (pdf) of the observation vector from four random variables (x_A, x_B, x_C, x_D)

$$p(x_A, x_B, x_C, x_D) \approx p^T(x_A, x_B, x_C, x_D) = p(x_A)p(x_B|x_A)p(x_C|x_B)p(x_D|x_B). \quad (5.1)$$

In other words, we condition the probability of a given variable on its immediate predecessor in accordance with the tree structure shown in Figure 5.2. In Figure 5.2 right, a group of hypothetical genes with similar developmental profiles is illustrated. The genes display average expression in stage A, up-regulation in stage B, down-regulation in stage C and up-regulation in stage D. Furthermore, the genes have distinct expression intensities, but similar relative expression changes. Genes strongly up-regulated in B are also strongly down-regulated in C and strongly up-regulated in D. These dependencies are reflected in the correlation between these stages. For example, A and B (or B and D) are positively correlated, and stages B and C are negatively correlated. A statistical model for such developmental profiles should include these dependencies between subsequent stages, as it is provided by `DTrees`.

Formally, let $X = (X_1, ..., X_u, ..., X_L)$ be an L-dimensional continuous random vector where the variable X_u denotes the expression values of the developmental stage u and $x = (x_1, ..., x_L)$ denotes an observation of X representing a developmental profile of a gene. Consider a directed graph (V, E), where each vertex in V represents a variable in X, $|V| = L$, and a directed edge $(v, u) \in E$ indicates that variable X_u is dependent on variable X_v. The structure of a `DTree` is represented by a directed tree. A directed tree is a connected directed graph, whose vertices except the root have in-degree equal to 1, and there are no cycles in the graph. For simplicity, we represent the `DTree` structure by the parent map, $pa : \{1, ..., L\} \mapsto \{1, ..., L\}$, where $pa(u) = v$ indicates that $(v, u) \in E$. The root of the `DTree`, which has no incoming edges is represented by $pa(u) = u$. We define the pdf of a `DTree` as

$$p^T(x|\theta) = \prod_{u=1}^{L} p(x_u|x_{pa(u)}, \tau_u). \tag{5.2}$$

We denote the model parameters by $\theta = (pa, \tau_1, ..., \tau_u, ...\tau_L)$. Note, that a `DTree` can be also regarded as an approximation of the joint pdf of a L-dimensional continuous random vector by a product of $L - 1$ second order pdfs [43].

5.2.1 Equivalence of Dependence Trees

We can use the formalism of graphical models and Bayesian networks, which `DTrees` are a particular case, for analyzing characteristics of the model [125]. One interesting aspect is the existence of several `DTrees` with equivalent pdfs. Intuitively, the main information contained in the `DTree` structure are the connected pair of variables, but not the directions of the edges. For example, we can obtain an equivalent `DTree` pdf using an undirected tree representation. Formally, we can apply a graph factorization [125] to the undirected representation of the `DTree` structure, which yields the following pdf [148]

$$p^T(x|\theta) = \frac{\prod_{(u,v)\in E} p(x_u, x_v)}{\prod_{v\in V} p(x_v)^{\deg(v)-1}}, \tag{5.3}$$

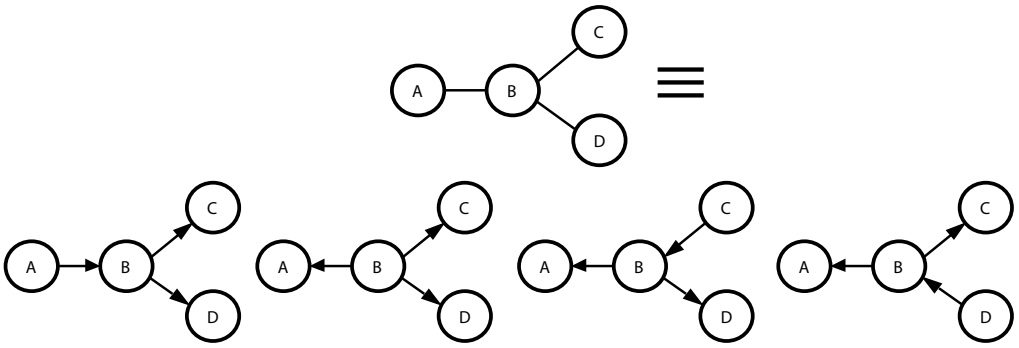


Figure 5.3: *We depict the undirected tree structure of the graph from Figure 5.2 (top), and the four possible directed versions obtained by choosing respectively edges A, B, C and D as a root (bottom).*

where deg(v) is the number of edges of v.

It can be shown with the application of the Bayes rule that the pdfs from Eq. 5.2 and Eq. 5.3 are equivalent,

$$\begin{aligned} p^T(x_A, x_B, x_C, x_D) &= \frac{p(x_A, x_B)p(x_B, x_C)p(x_B, x_D)}{p(x_B)p(x_B)} \\ &= \frac{p(x_A)p(x_B|x_A)p(x_B, x_C)p(x_B, x_D)}{p(x_B)p(x_B)} \\ &= p(x_A)p(x_B|x_A)p(x_C|x_B)p(x_D|x_B). \end{aligned}$$

For any undirected tree structure, we can also obtain a directed tree by choosing a vertex as a root, and directing the edges away from the root. Any arbitrary choice of root will lead to equivalent decompositions of the tree pdfs. For instance, in Figure 5.3 left-middle, we have X_B as a root, which leads to the following pdf

$$p^T(x_A, x_B, x_C, x_D) = p(x_B)p(x_A|x_B)p(x_C|x_B)p(x_D|x_B). \tag{5.4}$$

By Bayes rule we can show that Eq. 5.4 can be easily transformed into Eq. 5.1

$$\begin{aligned} p^T(x_A, x_B, x_C, x_D) &= p(x_B)p(x_A|x_B)p(x_C|x_B)p(x_D|x_B) \\ &= p(x_A, x_B)p(x_C|x_B)p(x_D|x_B) \\ &= p(x_A)p(x_B|x_A)p(x_C|x_B)p(x_D|x_B). \end{aligned}$$

In summary, any directed representation of an underlying undirected tree will lead to equivalent tree pdfs [148]. See [125] for a formal treatment based on the equivalence of pdfs (or distributions) in chain graphs. Given the simplicity and the intuitive representation, this chapter will mostly use directed versions of the tree structures. The choices of the direction of edges are based on the prior knowledge of the data, i.e., the underlying developmental tree.

5.2.2 Parameterization of Dependence Trees

We use conditional Gaussian density functions [126] as conditional densities, denoted by $p(x_u|x_{pa(u)}, \tau_u)$ in Eq. 5.2. Hence, for a given developmental profile x and a non-root developmental stage u with $pa(u) = v$, the pdf takes the following form

$$p(x_u|x_v, \tau_u) = (\sqrt{2\pi}\sigma_{u|v})^{-1} \exp\left(\frac{-(x_u - \mu_u - w_{u|v}(x_v - \mu_v))^2}{2\sigma^2_{u|v}}\right), \tag{5.5}$$

where $\tau_u = (\mu_u, w_{u|v}, \sigma^2_{u|v})$ are the parameters for one conditional density in the model.

For a given expression data set $\mathbf{X}$ consisting of N gene observations at L developmental stages, let $x_i = (x_{i1}, \ldots, x_{iu}, \ldots, x_{iL})$ be the developmental profile of gene i, and x_{iu} be the expression value of the gene i in development stage u for $1 \leq i \leq N$ and $1 \leq u \leq L$. The maximum likelihood estimates (MLE) for the parameters of the conditional Gaussian are [125],

$$\hat{\mu}_u = \frac{\sum_{i=1}^N x_{iu}}{N}, \tag{5.6}$$

$$\hat{w}_{u|v} = \frac{\hat{\sigma}_{uv}}{\hat{\sigma}^2_v}, \text{ and} \tag{5.7}$$

$$\hat{\sigma}^2_{u|v} = \hat{\sigma}^2_u - \hat{w}^2_{u|v}\hat{\sigma}^2_v. \tag{5.8}$$

These terms can be computed from the sufficient statistics as follows

$$\hat{\sigma}^2_u = \frac{\sum_{i=1}^N (x_{iu} - \hat{\mu}_u)^2}{N}, \text{ and} \hat{\sigma}_{uv} = \frac{\sum_{i=1}^N (x_{iu} - \hat{\mu}_u)(x_{iv} - \hat{\mu}_v)}{N}. \tag{5.9}$$

The conditional normal pdf can be seen as estimating a linear fit between X_u and X_v, where $w_{u|v} > 0$ indicates a positive linear correlation and $w_{u|v} < 0$ a negative linear correlation between variables; $w_{u|v} = 0$ if the variables are independent. Furthermore, $w_{u|v}$ and $\sigma^2_{u|v}$ are related because the better the linear fit the smaller the variance. For the special case of the root (recall that $pa(u) = u$), $w_{u|u}$ is set to zero, and the conditional density is effectively an univariate normal. The model has $3L - 1$ free parameters. See Section 5.3.2 for the complete derivation of MAP estimates of the conditional Gaussians.

Returning to our example, the model estimates given the developmental tree and expression profiles from Figure 5.2 are the following

$$\begin{aligned}
\tau_A = &\quad (\hat{\mu}_A, \hat{w}_{A|A}, \hat{\sigma}^2_{A|A}) &&= (-0.01, 0, 0.02),\\
\tau_B = &\quad (\hat{\mu}_B, \hat{w}_{B|A}, \hat{\sigma}^2_{B|A}) &&= (0.97, 2.2, 0.02),\\
\tau_C = &\quad (\hat{\mu}_C, \hat{w}_{C|B}, \hat{\sigma}^2_{C|B}) &&= (-0.99, -0.3, 0.01), \text{ and}\\
\tau_D = &\quad (\hat{\mu}_D, \hat{w}_{D|B}, \hat{\sigma}^2_{D|B}) &&= (0.45, 0.53, 0.01).
\end{aligned}$$

As expected, $\hat{w}_{B|A}$ and $\hat{w}_{D|B}$ are positive, indicating a linear dependence between these

variables. On the other hand, $\hat{w}_{C|B}$ is negative, which indicates a negative correlation between X_B and X_C.

5.2.3 Estimation of the Structure of Dependence Trees

As described in the previous section, in developmental processes the developmental tree structure is already known a priori. Although the developmental tree is an interesting candidate for modeling dependencies, we are also interested in the case of estimating the tree structure from the data. We summarize here our extension to continuous variables of the solution proposed in [43], which considers trees on discrete distributions. The solution is based on finding the `DTree` structure that minimizes the relative entropy between $p(x)$ and the approximation $p^T(x)$

$$p^{T*} = \text{argmin}_{p^T} \text{D}(p||p^T). \tag{5.10}$$

The relative entropy between p and p^T is defined as [54],

$$\text{D}(p||p^T) = \int_X p(x) \log \frac{p(x)}{p^T(x)}.$$

Replacing $p^T(x)$ by Eq. 5.2, we obtain,

$$\begin{aligned} \text{D}(p||p^T) &= \int_X p(x) \log p(x) - \int_X p(x) \sum_{u=1}^{L} \log p(x_u|x_{pa(u)}), \\ &= \text{H}(X) - \int_X p(x) \sum_{u=1}^{L} \log p(x_u) - \int_X p(x) \sum_{u=1}^{L} \log \frac{p(x_u|x_{pa(u)},)}{p(x_u)} \end{aligned}$$

We can simplify the previous equation by applying the Bayes rule and the definition of entropy (H) and mutual information (I) [54],

$$\begin{aligned} \text{D}(p||p^T) &= \text{H}(X) - \sum_{u=1}^{L} \text{H}(X_u) - \int_X \sum_{u=1}^{L} p(x_u, x_{pa(u)}) \log \frac{p(x_u, x_{pa(u)})}{p(x_u)p(x_{pa(u)})}, \\ &= \text{H}(X) - \sum_{u=1}^{L} \text{H}(X_u) - \sum_{u=1}^{L} \text{I}(X_u, X_{pa(u)}). \end{aligned} \tag{5.11}$$

Since $\text{H}(X)$ and $\text{H}(X_u)$ are independent of p^T, then Eq. 5.10 can be reduced as follows,

$$pa^* = \text{argmax}_{pa} \sum_{u=1}^{L} \text{I}(X_u, X_{pa(u)}). \tag{5.12}$$

The solution to this problem can be efficiently computed by applying a maximum weight

spanning tree algorithm on a fully connected undirected graph, where vertices represent the variables and the weights of edges are equal to the mutual information between variables [43]. The computational complexity of this algorithm is $O(L^2 \log L)$.

Finally, we need to compute $\mathrm{I}(X_u, X_{pa(u)})$ for multivariate Gaussian. Given that $pa(u) = v$, the mutual information is defined as [54]

$$\mathrm{I}(X_u, X_v) = \int_{X_u} \int_{X_v} p(x_u, x_v) \log \frac{p(x_u, x_v)}{p(x_u)p(x_v)} dx_u dx_v. \tag{5.13}$$

Expanding the terms, we obtain

$$\begin{aligned}\mathrm{I}(X_u, X_v) = \; & \int_{X_u} \int_{X_v} p(x_u, x_v) \log p(x_u, x_v) dx_u dx_v - \int_{X_u} \int_{X_v} p(x_u, x_v) \log p(x_u) dx_u dx_v \\ & - \int_{X_u} \int_{X_v} p(x_u, x_v) \log p(x_v) dx_u dx_v,\end{aligned}$$

and by definition of $\mathrm{H}(X)$, it follows that

$$\mathrm{I}(X_u, X_v) = \mathrm{H}(X_u) + \mathrm{H}(X_v) - \mathrm{H}(X_u, X_v). \tag{5.14}$$

The entropy of an L dimensional multivariate Gaussian pdf is defined as [54],

$$\mathrm{H}(X) = \frac{1}{2} \log(2\pi e)^L |\Sigma_X|, \tag{5.15}$$

where Σ_X is the covariance matrix of X. By substituting Eq.5.15 into Eq.5.14, we obtain

$$\mathrm{I}(X_u, X_v) = \frac{1}{2} \log(2\pi e \sigma^2_{X_u}) + \frac{1}{2} \log(2\pi e \sigma^2_{X_v}) - \frac{1}{2} \log((2\pi e)^2 |\Sigma_{X_u, X_v}|),$$

and, as $|\Sigma_{X_u, X_v}| = \sigma^2_u \sigma^2_v - (\sigma_{u,v})^2$, it follows that

$$\mathrm{I}(X_u, X_v) = \frac{1}{2} \log \left(\frac{(2\pi e)^2}{(2\pi e)^2} \right) - \frac{1}{2} \log \left(\frac{\sigma^2_u \sigma^2_v - \sigma^2_{u,v}}{\sigma^2_u \sigma^2_v} \right),$$

and hence,

$$\mathrm{I}(X_u, X_v) = -\frac{1}{2} \log \left(1 - \frac{\sigma^2_{u,v}}{\sigma^2_u \sigma^2_v} \right). \tag{5.16}$$

Note that the mutual information is proportional to the correlation coefficient $\rho_{u,v} = \frac{\sigma^2_{u,v}}{\sigma^2_u \sigma^2_v}$. That is, it measures the dependence between the two variables; $\mathrm{I}(X_u, X_v) = 0$ if both variables are independent. Moreover, the mutual information is symmetric ($\mathrm{I}(X_u, X_v) = \mathrm{I}(X_v, X_u)$). Therefore, the estimation method does not determine direction of edges. To obtain a directed tree, we select one particular edge as root and direct all edges away from it (as discussed in Section 5.2.1, any direction choice would lead to equivalent `DTree` pdfs).

We propose a "treeness" index for evaluating how well a `DTree` performs in capturing

dependence in the data. Intuitively, we measure the proportion of the mutual information represented in the tree edges, in comparison to the total mutual information on all pairs of variables. That is, for a tree structure pa the treeness index is defined as follows

$$T(pa) = \frac{\sum_{u=1}^{L} \mathrm{I}(X_u, X_{pa(u)})}{\sum_{u=1}^{L} \sum_{v=u+1}^{L} \mathrm{I}(X_u, X_v)}. \tag{5.17}$$

A value of zero indicates that no dependence is captured by the `DTree` and 1 indicates that all dependence is captured by the `DTree`.

5.2.4 Dependence Trees and Multivariate Gaussians

There is a close correspondence between the pdfs of multivariate Gaussians and `DTrees`. Given that $pa(u) = v$, a `DTree` pdf is equivalent to a multivariate Gaussian with mean vector $\mu = (\mu_1, ..., \mu_L)$, and entries of the covariance matrix (Σ^T) of the form [179]

$$\sigma^T_{u,v} = \sum_{t=pa(v)} w_{v|t} \sigma^T_{u,t} + \mathbf{1}(u = v)\sigma_v \tag{5.18}$$

For the example, for the `DTree` shown in Figure 5.2, the corresponding covariance matrix Σ^T is as follows

$$\begin{Bmatrix} \sigma^2_A & w_{B|A} * \sigma^2_A & w_{C|B} * w_{B|A} * \sigma^2_A & w_{D|B} * w_{B|A} * \sigma^2_A \\ w_{B|A} * \sigma^2_A & \sigma^2_{B|A} - w^2_{B|A}\sigma^2_A & w_{C|B} * (\sigma^2_{B|A} - w^2_{B|A}\sigma^2_A) & w_{D|B} * (\sigma^2_{B|A} - w^2_{B|A}\sigma^2_A) \\ w_{C|B} * w_{B|A} * \sigma^2_A & w_{C|B} * (\sigma^2_{B|A} - w^2_{B|A}\sigma^2_A) & \sigma^2_{C|B} - w^2_{C|B}\sigma^2_B & w_{C|B} * w_{D|B} * (\sigma^2_{B|A} - w^2_{B|A}\sigma^2_A) \\ w_{D|B} * w_{B|A} * \sigma^2_A & w_{D|B} * (\sigma^2_{B|A} - w^2_{B|A}\sigma^2_A) & w_{D|B} * w_{C|B} * (\sigma^2_{B|A} - w^2_{B|A}\sigma^2_A) & \sigma^2_{D|B} - w^2_{D|B}\sigma^2_B \end{Bmatrix}.$$

This represents a type of covariance matrix parameterization not yet characterized before (see Section 2.3.3 for a discussion and [10, 37] for others covariance matrix parameterizations).

5.3 Mixture of Dependence Trees

In order to find clusters of co-expressed genes, we combine several `DTrees` in a mixture model. Each `DTree` is a representation of a cluster or group of genes with co-expressed developmental profiles, i.e., each `DTree` models distinct patterns of gene expression in the course of development (see Figure 5.4 for an example). Throughout this chapter we refer to the proposed method as `MixDTrees`.

Formally, we combine a set of K `DTrees` in a mixture model

$$p(x|\Theta) = \sum_{k=1}^{K} \alpha_k p^T_k(x|\theta_k), \tag{5.19}$$

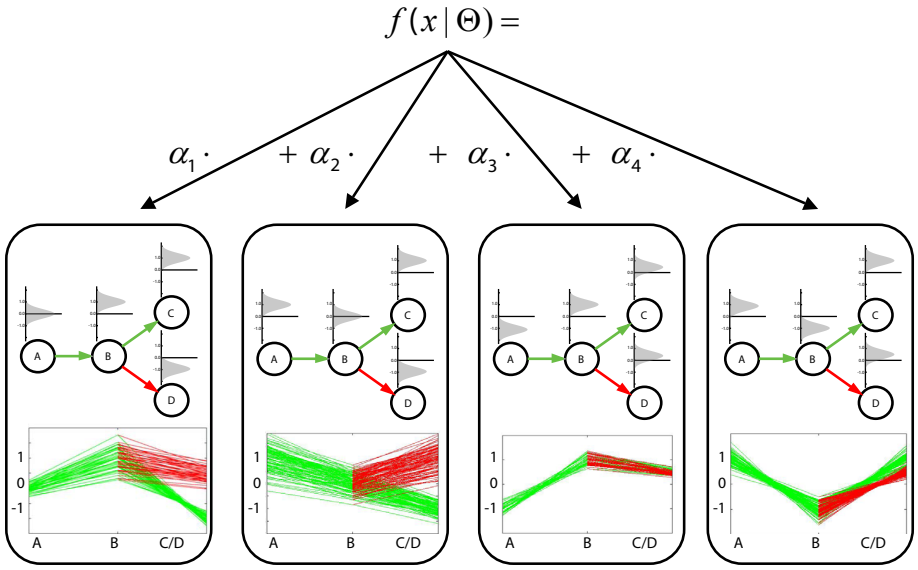


Figure 5.4: *Example of a mixture of four* `DTrees` *with the structure defined in Figure 5.2. Each of these* `DTrees` *models distinct developmental profiles found in the data set employed as example. Furthermore, clusters can have distinct sizes proportional to their α_i's. Note also that it is not necessary that clusters have distinct expression values in branching stages. For example, stages C and D have similar expression values for cluster 3 and 4. This can be interpreted as the genes being equally expressed in the two alternative lineages.*

where α_k is the mixture coefficient (see Section 2.2), $p_k^T(x|\theta_k)$ is the density corresponding to the kth `DTree` as defined in Eq. 5.2, and $\Theta = (\alpha_1, ..., \alpha_K, \theta_1, ..., \theta_K)$.

5.3.1 `MixDTrees` with Developmental Tree as Structure

The differentiation of cells in the course of development is conveniently represented as a developmental tree. The structures of these trees are well-known for most data sets under investigation. Thus, one approach explored in this work is the use of the developmental tree as prior knowledge, that is to define all `DTrees` structures in the mixture to be the same as in the developmental tree. We will call this method `MixDTrees-Dev`. For estimating `MixDTrees-Dev`, we apply the EM algorithm described in Section 2.3.1. In order to do so, we need to define the `DTree` estimates of the M-Step of the EM algorithm. We choose to use maximum-a-posteriori (MAP) estimates, as these minimize problems related to over-fitting [80].

5.3.2 Maximum-a-posteriori Estimates

To prevent over-fitting of the `DTree`, we propose the use of a maximum-a-posteriori point estimate (MAP) approach, which regularizes the estimates from Eq. 5.7 and Eq. 5.8. In

practice, we define prior distributions for these parameters, penalizing parameters with undesirable values. For example, a low $\sigma^2_{u|v,k}$ is an indication of over-fitting and should be avoided, unless there is enough data (or evidence) for that particular component. Maximum-a-posteriori estimates can be used in the EM algorithm. This can achieved by changing Eq. 2.8 to maximize the expected a posteriori distribution, instead of the complete likelihood function.

More precisely, our aim is to find estimates maximizing the posterior distribution

$$p(\Theta|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{X},\mathbf{Y}|\Theta)p(\Theta)}{p(\mathbf{X},\mathbf{Y})} \tag{5.20}$$

where $\mathbf{X}$ is the observed data, $\mathbf{Y}$ indicates which mixture component generated a given observation and Θ are the model parameters. The pdf $p(\mathbf{X},\mathbf{Y}|\Theta)$ is the complete data likelihood (Eq. 2.7), $p(\Theta)$ is the prior distribution on the parameters Θ and $p(\mathbf{X},\mathbf{Y})$ is the prior of the data. We can ignore the last term ($p(\mathbf{X},\mathbf{Y})$) in our problem, as it is independent of Θ, and will be constant for a fixed data set.

Since `MixDTrees` are based on first-order dependencies, it is sufficient to find the parameters in a simple bivariate scenario $(X_u, X_{pa(u)})$, where $pa(u) = v$ and $\mathbf{X}_u$ corresponds to the observed data from variable X_u. This simplifies Eq. 5.20 to

$$p(\Theta|\mathbf{X}_u,\mathbf{X}_v,\mathbf{Y}) \approx p(\mathbf{X}_u,\mathbf{X}_v,\mathbf{Y}|\Theta)p(\Theta). \tag{5.21}$$

where

$$p(\mathbf{X}_u,\mathbf{X}_v,\mathbf{Y}|\Theta) = \prod_{k=1}^{K}\prod_{i=1}^{N}(\alpha_k \cdot p_k^T(\mathbf{X}_u,\mathbf{X}_v|\Theta_k))^{r_{ik}},$$

as shown in Section 2.3.1 and

$$p(\Theta) = \prod_{k=1}^{K} p(\Theta_k) = \prod_{k=1}^{K} p(w_{u|v}|\sigma^2_{u|v,k},\alpha_k)p(\sigma^2_{u|v,k}|\alpha_k)p(\alpha_k),$$

where $\alpha_k = \sum_{i=1}^{N} r_{ik}/N$, and $r_{ik} = p(y_i = k|x_i)$ is the posterior probability (Eq. 2.16) that observation i belongs to `DTree` k.

Priors on Parameters. We use conjugate priors to regularize the parameters $w_{u|v,k}$ and $\sigma^2_{u|v,k}$ and to avoid over-fitting, when there is low evidence for a given component model (or low α_k).

For simplicity of computation, we work with a precision parameter $\lambda_{u|v,k} = (\sigma^2_{u|v,k})^{-1}$. We define the prior of $\lambda_{u|v,k}$ to be proportional to

$$p(\lambda_{u|v,k}|\nu_{u|v,k},\alpha_k) \sim \text{Exponential}\left(\frac{\lambda_{u|v,k}}{\alpha_k}\right) = \frac{\sum_{i=1}^{N} r_{ik}}{\lambda_{u|v,k}} \exp\left(-\frac{\sum_{i=1}^{N} r_{ik}}{\lambda_{u|v,k}}\right) \quad (5.22)$$

where $\nu_{u|v,k}$ is a hyper-parameter. Intuitively, this prior penalizes variables with low variances and low evidence, enforcing higher $\sigma^2_{u|v,k}$.

The prior of $w_{u|v,k}$ is defined as follows

$$p(w_{u|v,k}|\lambda_{u|v,k},\sigma^2_{v|k},\alpha_k,\beta_{u|v,k}) = N(0,\beta_{u|v,k}(\lambda_{u|v,k}\alpha_k\sigma^2_{v|k})^{-1}), \quad (5.23)$$

which is invariant to the scale of the variables X_u and X_v, and has $\beta_{u|v,k}$ as a hyper-parameter. Intuitively, this prior penalizes variables with high covariance and low evidence, enforcing smaller $w_{u|v,k}$ values.

Derivation of MAP Estimates. By replacing Eq. 5.5, 5.23 and 5.22 into Eq. 5.21 and taking the logarithm, we obtain

$$\begin{aligned}
\log p(\Theta|\mathbf{X}_u,\mathbf{X}_v,Y) \;=\; & -\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}\log(\lambda_{u|v,k}) \\
& -\sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}\left((x_{iu}-\mu_{u|k}-w_{u|v,k}(x_{iv}-\mu_{v|k}))^2\lambda_{u|v,k}/2\right) \\
& -\frac{1}{2}\sum_{k=1}^{K}\log\left(\frac{\beta_{u|v,k}}{\lambda_{u|v,k}\sigma^2_{v|k}\sum_{i=1}^{N} r_{ik}}\right) - \sum_{k=1}^{K}\frac{w^2_{u|v,k}\sigma^2_{v|k}\sum_{i=1}^{N} r_{ik}\lambda_{u|v,k}}{\beta_{u|v,k}} \\
& -\frac{1}{2}\sum_{k=1}^{K}\log\left(\frac{\nu_{u|v,k}}{\sum_{i=1}^{N} r_{ik}}\right) - \sum_{k=1}^{K}\frac{\lambda_{u|v,k}\sum_{i=1}^{N} r_{ik}}{\nu_{u|v,k}}.
\end{aligned}$$

We can take the derivate of the MAP with respect to $w_{u|v,k}$ as follows

$$\begin{aligned}
\frac{\partial \log p(\Theta|\mathbf{X}_u,\mathbf{X}_v,Y)}{\partial w_{u|v,k}} \;=\; & \sum_{i=1}^{N} r_{ik}\left((x_{iu}-\mu_{u|k}-w_{u|v,k}(x_{iv}-\mu_{v|k}))x_{iv}\lambda_{u|v,k}\right) \\
& -\frac{w_{u|v,k}\sum_{i=1}^{N} r_{ik}\sigma^2_{v|k}\lambda_{u|v,k}}{\beta_{u|v,k}},
\end{aligned}$$

and setting this equation to zero

$$0 = \sigma_{u,v|k} - w_{u|v,k}\sigma^2_{v|k} - \frac{w_{u|v,k}\hat{\sigma}^2_{v|k}}{\beta_{u|v,k}},$$

yields the MAP estimate,

$$\hat{w}_{u|v,k} = \frac{\hat{\sigma}_{u,v|k}}{\hat{\sigma}^2_{v|k}(1+\beta^{-1}_{u|v,k})}. \tag{5.24}$$

The MAP estimate of $\lambda_{u|v,k}$ can be derived in the following way,

$$\begin{aligned}\frac{\partial \log p(\Theta|\mathbf{X}_u, \mathbf{X}_v, Y)}{\partial \lambda_{u|v,k}} &= -\frac{1}{2}\sum_{i=1}^{N} r_{ik}(\lambda_{u|v,k})^{-1} \\ &\quad -\frac{1}{2}\sum_{i=1}^{N} r_{ik}(x_{iu} - \mu_{u|k} - w_{u|v,k}(x_{iv} - \mu_{v|k}))^2 \\ &\quad -\frac{w^2_{u|v,k}\sum_{i=1}^{N} r_{ik}\sigma^2_{v|k}}{\beta_{u|v,k}} + \frac{\sum_{i=1}^{N} r_{ik}}{\nu_{u|v,k}}.\end{aligned}$$

Setting it to zero yields

$$0 = -(\lambda_{u|v,k})^{-1} + \hat{\sigma}^2_{u|k} - w^2_{u|v,k}\hat{\sigma}^2_{v|k} - \frac{w^2_{u|v,k}\hat{\sigma}^2_{v|k}}{\beta_{u|v,k}} - \frac{1}{\nu_{u|v,k}},$$

$$(\lambda_{u|v,k})^{-1} = \hat{\sigma}^2_{u|v,k} = \hat{\sigma}^2_{u|k} - w^2_{u|v,k}\hat{\sigma}^2_{v|k}(1+\beta^{-1}_{u|v,k}) - \nu^{-1}_{u|v,k}. \tag{5.25}$$

When $\beta_{u|v,k} \to \infty$ and $\nu_{u|v,k} \to \infty$, the prior becomes non-informative, and MAP and ML estimates are equal. All the estimates make use of the following sufficient statistics

$$\hat{\mu}_{u|k} = \frac{\sum_{i=1}^{N} r_{ik}x_{iu}}{\sum_{i=1}^{N} r_{ik}}, \tag{5.26}$$

$$\hat{\sigma}^2_{u|k} = \frac{\sum_{i=1}^{N} r_{ik}(x_{iu} - \hat{\mu}_{u|k})^2}{\sum_{i=1}^{N} r_{ik}}, \tag{5.27}$$

$$\hat{\sigma}_{u,v|k} = \frac{\sum_{i=1}^{N} r_{ik}(x_{iv} - \hat{\mu}_{v|k})(x_{iu} - \hat{\mu}_{u|k})^2}{\sum_{i=1}^{N} r_{ik}}. \tag{5.28}$$

Hyper-parameters Estimates via Empirical Bayes. In an empirical Bayes approach [36], by derivating Eq. 5.21 in relation to the hyper-parameters, we can estimate the maximum a posteriori values of $\beta_{u|v,k}$ and $\nu_{u|v,k}$ from the data as follows

$$\frac{\partial \log p(\Theta|\mathbf{X}_u, \mathbf{X}_v, Y)}{\partial \beta_{u|v,k}} = \frac{1}{2\beta_{u|v,k}} - \frac{w_{u|v,k}^2 \sum_{i=1}^{N} r_{ik} \sigma_{v|k}^2}{2\beta_k^2 \sigma_{u|v,k}^2},$$

setting it to zero

$$0 = -\beta_{u|v,k} - \frac{\sigma_{v|k}^2 \sum_{i=1}^{N} r_{ik} w_{u|v,k}^2}{2\sigma_{u|v,k}^2}$$

and by definition of $\sigma_{u|v,k}^2$ and $w_{u|v,k}$, this yields

$$\hat{\beta}_{u|v,k} = \frac{\sum_{i=1}^{N} r_{ik}}{\frac{2\sigma_{u|k}^2 \sigma_{v|k}^2}{\sigma_{u,v|k}^2} - 2}. \tag{5.29}$$

For $\nu_{u|v,k}$, we have

$$\frac{\partial \log p(\Theta|\mathbf{X}_u, \mathbf{X}_v, Y)}{\partial \nu_{u|v,k}} = -\frac{1}{2\nu_{u|v,k}} - \frac{\sum_{i=1}^{N} r_{ik} \lambda_{u|v,k}}{2\nu_{u|v,k}^2},$$

setting this equation to zero, we obtain

$$\hat{\nu}_{u|v,k} = -\frac{\sum_{i=1}^{N} r_{ik}}{2\sigma_{u|v,k}^2} \tag{5.30}$$

Both empirical priors penalize variables with large variances or with low evidence enforcing respectively lower $w_{u|v,k}$ and higher $\sigma_{u|v,k}^2$.

5.3.3 `MixDTrees` with Estimated Structure

We do not expect that all genes in a particular developmental process will share the same dependence structure, nor that the most likely `DTree` will exactly match the developmental tree per se. Indeed, we expect that some genes will be particularly correlated in particular developmental lineages, but not in others. For example, group 1 from Figure 5.5 has genes tightly over-expressed in the blue lineage ($\{X_D, X_E, X_F\}$), as does group 2 in the orange lineage ($\{X_B, X_C\}$). We also expect that some genes, which are important for earlier developmental stages, to have similar expression profiles in stages near the root, but not in mature cell types (leaf vertices of a developmental tree). See for example group 3 in Figure 5.5, which exhibits over-expression in all earlier stages ($\{X_A, X_B, X_D\}$).

To infer these group-specific dependencies, we estimate a mixture of K `DTrees`, where each component have its tree structure estimated from the data. We will call this approach mixture of dependence trees with estimated structure (`MixDTrees-Str`). Note that the mixture of dependence trees with estimated structure corresponds to a relaxation of `MixDTrees-Dev` (Section 5.3.1), when a single dependence tree structure is assumed.

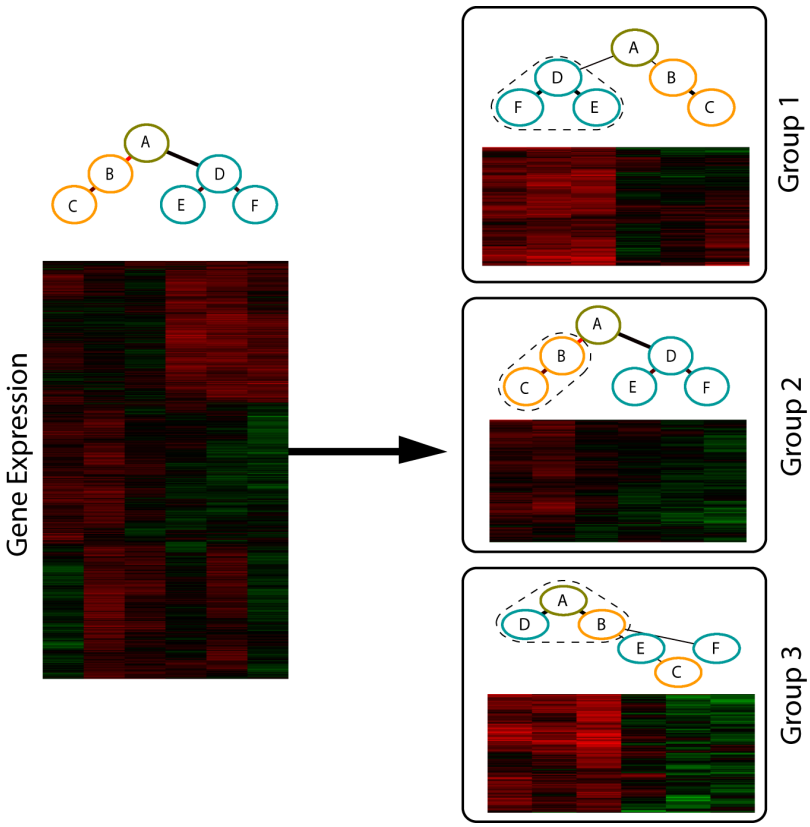


Figure 5.5: *Illustrative example of a developmental tree and its gene expression data (left). The developmental tree is constituted of a stem cell (stage A), an "orange" lineage (stages B and C) and a "blue" lineage (stages D, E and F). The red-green plot depicts the relative expression, where lines corresponds to gene profiles and columns to developmental stages ordered as in the above tree. In the right, we depict three groups of genes and their corresponding estimated tree structure as found by* `MixDTrees` *in the gene expression data in the left (see Section 5.3.3 for complete plot description).*

For estimation of `MixDTrees-Str`, we need to perform the method described in Section 5.2.3 for each `DTree` prior to the M-Step [148]. Once the structure is chosen, `DTree` parameters are set with the MAP estimates (see Section 5.3.2).

Visualization of `DTree` with Estimated Structure. The branches in the estimated tree structure reflect similarity in expression of developmental stages (stages in a same branch will share a similar expression profiles). To highlight these similarities, we propose the following plots. Gene clusters are depicted as a heat-map with red values indicating over-expression and green values indicating under-expression [71]. In this plot, the lines (gene profiles) are ordered as proposed in [16]. For the columns (developmental stage profiles), we compute all possible columns orderings and select the one that has a minimal difference in the mutual information of adjacent columns. To further help the interpretation of individual clusters, we compute strongly connected components [46] (SCC) in the graph

returned after thresholding the mutual information matrix. An optimal threshold parameter is obtained by evaluating the resulting SCC with the silhouette index [115]. SCC indicate within a `DTree`, which developmental stages in a particular branch have similar expression profiles.

5.4 Experiments

We describe in the Section 5.4.1 our analyses performed in [50], where `MixDTrees` with the developmental tree as structure(`MixDTrees-Dev`) is evaluated with two detailed studies covering several stages of the B and T cell development [99, 100]. Also, putative roles of microRNAs related to lymphoid development are investigated. In Section 5.4.2, we evaluate the use of `MixDTrees` with estimated structure (`MixDTrees-Str`) in a gene expression compendia containing early hematopoietic development cells and three lineages of lymphoid cells: B-cells, T-cells and Natural killer cells [3, 156, 165, 220, 229]. The method performance is compared with other methods via a score based on the enrichment of biological pathways. In both cases, in order to evaluate general characteristics of the methods, we use also simulated data sets.

5.4.1 `MixDTrees` with Developmental Tree Structure

Data

T Cell Development (`TCell`). This data set contains measurements of gene expression during the development of T cells in mouse [98]. Based on cell surface markers seven stages have been distinguished: CD4 and CD8 double negatives (`DN2, DN3, DN4`), large double positives (`DPL`), small double positives (`DPS`), single positive CD4 (`SP4`) and single positive CD8 (`SP8`) (see Figure 5.1 for the corresponding tree, and the original publication for details [98]). Affymetrix MU11k chips with four or five replicates are used to measure the expression levels of 13,104 mouse genes. We perform variance stabilization [104] on all chips, and compute the median values of replicates. To facilitate comparisons, we use the same list of 1,318 differentially expressed genes that was used by Hoffmann and colleagues [98]. Furthermore, we normalize the expression levels separately for each gene to mean zero and standard deviation one, as is routine in gene expression analysis. Finally, we map each probe set to a gene symbol if it exists in the respective chip platform annotation provided by the GEO database [69].

B Cell Development (`BCell`). This data set contains expression levels of five consecutive stages of the B cell lineage: Pre-BI, large Pre-BII, small Pre-BII, immature, and mature B cells [100]. This study was also conducted on Affymetrix MU11k chips. We pre-process the data exactly as it is described for `TCell`.

Lymphoid Development Related microRNAs (`LympMIR`). We collect 17 microRNAs that have been found to be involved in Lymphoid development or, at least, differentially expressed between distinguishable lymphocyte cell types [41, 44, 76, 151, 171]: mmu-miR-24, mmu-miR-26a, mmu-miR-142-3p, mmu-miR-146, mmu-miR-150, mmu-miR-155, mmu-miR-181a, mmu-miR-181b, mmu-miR-181c, mmu-miR-191, mmu-miR-221, mmu-miR-222, mmu-miR-223 and mmu-miR-342. Additionally, we include mmu-miR-15a, mmu-miR-15b, and mmu-miR-16, as they participate in the regulation of cell proliferation and apoptosis [35, 55]. Since in this work we refer exclusively to microRNAs of the mouse, the species prefix `mmu` is omitted throughout the text. The lists of candidate targets of these microRNAs are obtained in the miRBase Targets database [91] (Version 2.0), which uses the Miranda algorithm [72] to search for possible microRNA binding sites in the gene sequences.

Simulated Data (`SIM`). To generate this data set, we use `MixDTrees-Dev` with random parameterizations. All `DTrees` have theirs structure fixed to the tree represented in Figure 5.2. Then, we randomly chose $\mu_{u|v,k}$ from the range $[-1.5, 1.5]$ and $\sigma^2_{u|v,k}$ from $[0, 1]$. We create five experimental settings to inspect the performance of the method in the presence of distinct levels of dependence. For these five settings, we sample $w_{u|v,k}$ from $[-\epsilon, \epsilon]$ (independent data), $[-0.5, 0.5]$, $[-1, 1]$, $[-1.0, -0.5] \cup [0.5, 1]$ and $[-1, -1 + \epsilon] \cup [1 - \epsilon, 1]$ (tree dependent data), respectively, where $\epsilon = 0.001$. We set K to five and mixture coefficients α equal to $(0.1, 0.15, 0.2, 0.2, 0.35)$. For each experimental setting, we generate ten such mixtures, and sample 500 development profiles from each.

Results

We apply `MixDTrees-Dev` to two biological data sets: `TCell` and `BCell`. We compare our results with the ones obtained in [98, 100], which use Self-organizing maps (SOM) [120] as clustering method. For estimating `MixDTrees-Dev`, we perform the following. The mixture estimation method is initialized with K random `DTrees` (see Section 2.3.2). We, then, estimate then the mixture model using the EM-algorithm with MAP estimates. For both `TCell` and `BCell`, we use the same number of clusters (20) as [98, 100]. For evaluating the results, our analysis is complemented with information from OMIM [158], the Gene Ontology database [9] and from the literature. Furthermore, we perform a microRNA enrichment analysis in the clusters founds in both data sets to investigate putative roles of microRNAs related to lymphoid development.

We resort to simulated data to compare our method with established clustering methods, such as SOM, k-means and mixture of Gaussians, when inferring tree components in complex mixtures for varying levels of dependence between the individual variates. As we have class labels in the simulated data, we can evaluate the clusters with the use of external indices.

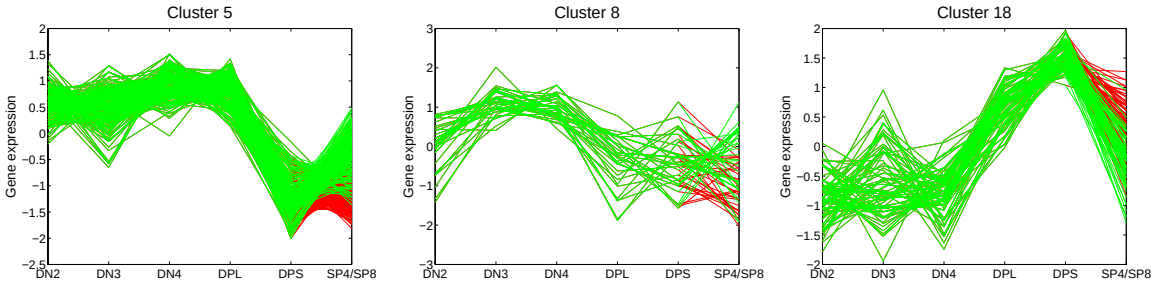


Figure 5.6: *Selected clusters from* `MixDTrees-Dev` *for* `TCell`*. We depict the clusters 5, 8 and 18 found in* `TCell`*, expression values on the y-axis, and cell types on the x-axis. Lines corresponding to developmental profile values between stages DN2, DN3, DN4, DPL, DPS and SP4 are in green and between DPS and SP8 in red.*

T Cell Development (`TCell`). `TCell` is a gene expression data set from seven differentiation stages of the T cell development (see Figure 5.1 for the developmental tree). The only branch in this tree is the final differentiation of DPS precursors into CD4 single positive SP4 cells and CD8 single positive SP8 cells. Most clusters found by `MixDTrees-Dev` from `TCell` show a distinctive pattern of differential expression along the developmental path, but they do not differ between SP4 and SP8 cells (clusters 4, 7, 11, 13, 14, 15, 16, 19, and 20). The most noticeable changes occur at the DPL stage in which the cells are proliferating and, subsequently, start to rearrange the TCRα-locus. This is also reflected in the overall correlation matrix[1]. Although the expression values of all neighboring stages are positively correlated, the correlation between the DPL stage and the DPS stage is much smaller in comparison to the double negative stages, all of which show high correlation. The correlation matrix suggests that SP4 and SP8 cells are more similar to each other than to their precursor DPS cells, which is expected since the two types of mature T cells share many cellular functions [98]. The largest differences with respect to SP4 and SP8 are found in clusters 5 and 18 (Figure 5.6). GO enrichment analysis shows that cell-cycle genes are clearly enriched in cluster 5. In contrast, cluster 18 mainly contains regulatory proteins involved in transcription and signaling (see Figure 5.6).

In order to demonstrate that our method is able to extract additional biological information, we concentrate our discussion on clusters showing distinct developmental profiles that could not be detected by the use of SOM [98]. For such a cluster, we assign functions to genes using the GO term annotation and complementary literature. In our analysis, we find that genes of cluster 8 are over-expressed in DN3 and DN4 cells 5.6), a developmental profile that had not been identified by SOM. With SOM, the genes of this cluster are dispersed over the two clusters (see Table B.1). Out of the 30 genes of cluster 8, seven are related to vesicle transport or to the Golgi/ER system. Additionally, we find five cell-cycle related genes, three involved in mitochondrial function, and seven genes of other functions,

[1] A simple way to check for similarities in the expression between developmental stages is to compute the correlation matrix of the data set at hand. As discussed in Section 5.2.3, the correlation matrix is proportional to the mutual information matrix.

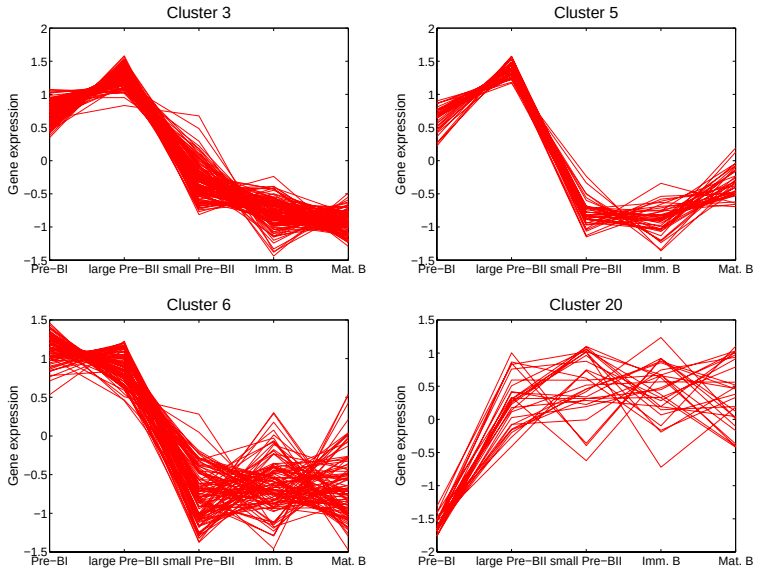


Figure 5.7: *Selected clusters from* `MixDTrees-Dev` *for* `BCell`*. We depict clusters 3, 5, 6 and 20 found in* `BCell`*, expression values on the y-axis, and cell types on the x-axis. Lines corresponding to developmental profile values between all stages are in red.*

which are mainly involved in signaling. These findings agree with the functions of DN3 and DN4 cells, which is the transport of precursor receptor molecules to the cell surface membrane and the initiation of proliferation. All these facts supports our claims that our method is able to identify functionally relevant groups of genes.

B Cell Development (`BCell`). Like in the `TCell` study, we investigated gene expression for five consecutive stages during B cell development (Figure 5.1). The correlation matrix of `BCell` suggests dependencies between gene expression values of successive stages, with the largest correlation between pre-BI and large pre-BII cells and between immature and mature B cells. When we compare, our clustering results to those obtained by SOM [98], we observe similar average developmental profiles, although the contingency table indicates differences in the cluster compositions (Table B.2). Clusters 3, 5 and 6, for example, contain genes that are up-regulated in pre-BI and large pre-BII cells and down-regulated in later developmental stages (Figure 5.7). Consistent with the phenotype of these cells, the function assigned to the genes of this cluster are mainly related to proliferation. GO categories that are associated with mitosis, cell-cycle and chromatin remodeling are clearly over-represented in these clusters.

Cluster 20 shows an average developmental profile that was not detected with SOM [98, 100]. The genes of this cluster are down-regulated in pre-BI cells, in which the first rearrangement of the D^H and J^H segments on the H chain loci has taken place, and up-regulated in all the following developmental stages (Figure 5.7). With SOM [98], these 23 genes are found distributed over the four clusters 11, 13, 14 and 17 (Table B.2). The most

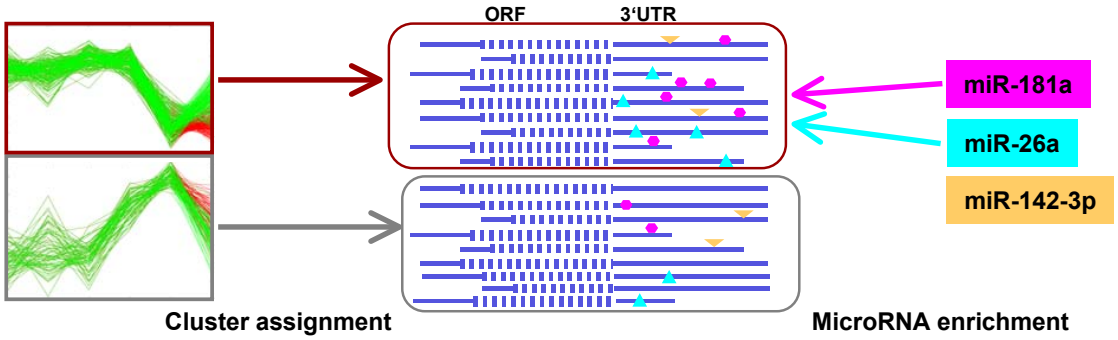


Figure 5.8: *Strategy to identify microRNAs and their target genes over-represented in clusters of co-expressed genes (indicated left) as part of a post-transcriptional regulatory mechanism. In the middle mRNAs clustered according to our mixture results are depicted and potential microRNA binding sites in their 3'UTRs are illustrated.*

plausible common function of some genes from cluster 20 is the regulation of survival and apoptosis during B cell development. The gene products *Nfkbia*, *Traf5* and the Src-family protein tyrosine kinases *Lyn* and *Syk* are known regulators of NF-kappa B activity, which in turn has been found to be involved in B cell fate decision and survival [2, 97, 152]. Similarly, Krupel-like factor 2 (*Klf2*) protects cells against TNF-alpha induced apoptosis [86]. Furthermore, *Icam-2* and *Rhoh*, whose encoding genes are two other members of cluster 20, regulate the adhesiveness of primary B cells depending on their activation state and protect them from apoptosis [158, 164].

MicroRNA Target Discovery. LympMIR contains a set of 17 microRNAs that are potentially involved in lymphocyte cell development. It has been proposed that microRNAs bind target mRNAs specifically via base pairing. This, subsequently, leads to interference of the translational machinery or mRNA degradation, and thus can control whole groups of genes simultaneously [17]. Recent microarray studies have demonstrated that the microRNA expression negatively correlates with mRNA target expression in a tissue specific manner [129, 133, 200].

Having identified clusters of co-expressed genes with `MixDTrees-Dev` for the B cell and T cell data sets, we ask whether a certain microRNA could be a potential regulator of one of these clusters (see Figure 5.8). For this task, we first obtain lists of potential target genes for each microRNA from the miRBase Targets database [91], which contains predictions made by sequence based methods. Given our clustering results, we use an enrichment analysis to obtain a list of microRNAs, whose potential targets are over-represented in a cluster. This is an approach similar to finding Gene Ontology terms over-represented in a cluster of genes, as described in Appendix A. A lower p-value indicates a high count of microRNA targets in a particular cluster, i.e., higher "microRNA enrichment". By choosing a p-value cut-off, we can construct a list of enriched microRNAs for each cluster as well as a list of target genes related to the enriched microRNAs.

Table 5.1: *List of* `LympMIR` *enriched in the clusters from* `MixDTrees-Dev` *for data sets* `TCell` *and* `BCell`*. We display the cluster and data set id, the list of microRNA and list of target genes, with p-values* < 0.05 *and at least four target genes per cluster. Genes involved in cell proliferation or DNA repair are depicted in bold. The indices indicate to which microRNA a gene is related to, when there is more than one enriched microRNA in a cluster.*

Cluster ID	MicroRNA	Target Genes
`TCell` 3	miR-222	*Elovl6, Nme1, Rcn1, Rps3*
`TCell` 5	miR-15a[1],miR-181a[2],miR-221[3], miR-24[4],miR-26a[5]	*2410015N17Rik*[4], *Alad*[1,4], *Atpif1*[1,5], ***Aurkb***[2], ***Cdc25a***[1], ***Chek1***[1] ***Cks1b***[2,4], ***Cks2***[5], *Eed*[2], ***H2afx***[4], *Kpnb1*[3], ***Mcm5***[3], *Nasp*[3,5], *Pex7*[2], *Psmd12*[2], *Ranbp5*[2], *Rars*[1], ***Tk1***[3], *Trip13*[1], *Uchl5*[5]
`TCell` 10	miR-142-3p[6],miR-150[7]	*Gfi1*[6], *Marcks*[6], *Msh6*[6], *Pp11r*[7], *Psmc1*[6,7]
`TCell` 11	miR-146[8],miR-16[9],miR-181b[10]	*Atp1b3*[10], *Ipo4*[9], *Klhdc2*[10], *Mrpl30*[8], *Orc5l*[8], *Tuba4*[9]
`BCell` 3	miR-181b[1],miR-181c[2],miR-26a[3]	*Atpif1*[3], ***Aurkb***[1,2], *Cbx1*[3], ***Cdc45l***[2], ***Cks1b***[1,2], ***Cks2***[3], *Cox5a*[3], *Hmgb2*[1,2], *Melk*[1,2], ***Ttk***[1,2], *Uchl5*[3]
`BCell` 5	miR-15a[4],miR-15b[5],miR-221[6], miR-223[7]	***Cdca4***[4,5], ***Chek1***[4,5], ***Mcm4***[7], *Nasp*[6], *Nfyb*[6], ***Smc4l1***[7] *Tuba2*[4,5,7]
`BCell` 6	miR-155[8],miR-191[9]	***Ctps***[9], *Ddx1*[8], *Hint1*[9], ***Mcm2***[8], *Phf17*[8], *Prdx4*[9], *SNrpd1*[9]
`BCell` 19	miR-142-3p[14],miR-342[15]	*2410002F23Rik*[14], *H2-Eb1*[14], *Ltb*[15], *Tap2*[14,15]

For TCell, our target prediction scheme identified, in four out of the 20 clusters, significant enrichment for eleven out of the 17 initial microRNAs (Table 5.1). In these four clusters, we detect 35 candidate target genes in total, which is a considerable reduction of the set of 229 targets that had been previously predicted by sequence based methods alone [91]. For BCell these numbers are respectively, eleven out of the 17 microRNAs, four out of the 20 clusters, and 29 out of the 273 predicted targets (Table 5.1). In particular, we find the five microRNA families miR-15, miR-181, miR-221, miR-26, and miR-142-3p to be enriched in both TCell and BCell. See Table B.3 and Table B.4 for p-values of microRNA enrichment of all data sets.

As mentioned earlier, the BCell clusters 3, 5, and 6 show a similar expression profile. We find that cluster 5 from TCell overlaps substantially with clusters 3 and 5 from BCell (Table 5.1). In TCell cluster 5, we find miR-15a, miR-181a, miR-26a, miR-24, and miR-221 as potential regulators and 20 potential target genes, seven of which are also present among the 18 BCell candidate genes of clusters 3 and 5. The developmental profiles of the clusters of both lineages show similar phenotypical features, namely up-regulation in the proliferating large cell populations (DN4, DPL, large pre-BII) and from then on strict down-regulation. In TCell cluster 5 there are eight genes and in the BCell clusters 3 and 5 there are nine target genes that are known to be involved in DNA metabolism, cell-cycle and mitosis (Table 5.1). This suggests a regulatory role for the identified microRNAs in reducing the transcript levels of genes that are important for cell proliferation. This is supported by the fact that a similar role for microRNA was found in Drosophila germline stem cells [94].

At the individual gene level, we identify some candidate microRNA targets for further detailed analysis: the three known genes (*H2-Eb1*, *Ltb*, *Tap2*) of BCell cluster 19 are all involved in the antigen presentation by MHC class II molecules [158, 166]. In the context of the cell cycle, *Chek1* (clusters TCell 5 and BCell 5) and *Cdc25a* (cluster TCell 5) are important for the transition between G1/S and G2/M phases [32]. Furthermore, both genes are candidate targets of the same microRNA, miR-15a, which is related to apoptosis in chronic lymphoid leukemia cells [44]. Another interesting gene codes for the nuclear factor Y (*Nfyb*; cluster BCell 5), which regulates *Hoxb4* [85], *Cdc34* [170] and the major histocompatibility complex in mice [237]. These are all important genes for lymphoid development. The mRNA of the growth factor independence-1 transcription factor (*Gfi1*; cluster TCell 10) is a potential target of miR-142-3p. *Gfi1* has as function the restriction of cell proliferation and maintenance of the functional integrity of lymphocyte cells [116]. Moreover, *Gfi1* is implicated in the transition from CD4/CD8 double negative to double positive T cells [188].

Simulated Data (SIM). We used MixDTrees-Dev with MAP and MLE estimates, mixture of Gaussians (MoG), k-means and SOM to compute clusters. We can compare to the classes used in data generation with cluster results to compute specificity (Eq. 3.14) and sensitivity (Eq. 3.13) of the clustering solutions. To compare the significance of differ-

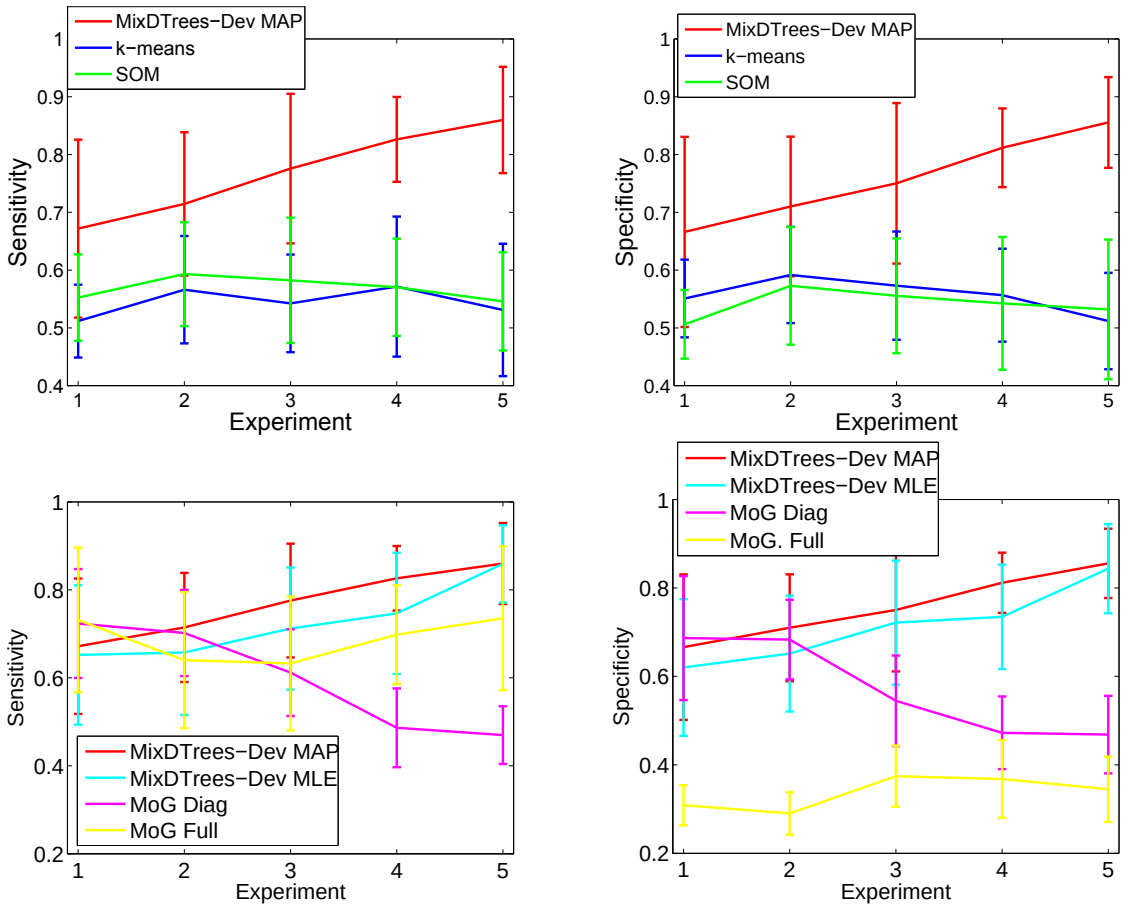


Figure 5.9: *We display the mean sensitivity (left plots) and mean specificity (right plots) against five experimental settings: (1)* $w_{u|v,k} \in [-\epsilon, \epsilon]$ *(independent data), (2)* $w_{u|v,k} \in [-0.5, 0.5]$*, (3)* $w_{u|v,k} \in [-1, 1]$*, (4)* $w_{u|v,k} \in [-1.0, -0.5] \cup [0.5, 1]$ *and (5)* $w_{u|v,k} \in [-1, -1+\epsilon] \cup [1-\epsilon, 1]$*. The dependence increases with experiment number.*

ences, we apply an one tailed paired t-test to evaluate the null hypothesis that two methods have the same mean specificity (or sensitivity) in a given experimental setting. Hereafter, for short, we simply state—method M_1 has a higher sensitivity than method M_2 (p-value below 0.05)—when the null hypothesis is rejected.

We observe that the `MixDTrees-Dev` with MAP estimates (`MixDTrees-Dev MAP`) have a higher specificity and sensitivity than k-means and SOM in all experimental settings (Figure 5.9 top) (p-value < 0.005). In the independent case ($w_{u|v,k} \in [-\epsilon, \epsilon]$), this is not expected, since the data agrees well with the assumptions of k-means and SOM. This also explains the large standard deviations of `MixDTrees-Dev MAP` in that case. As expected, the `MixDTrees-Dev MAP` clearly improves the cluster recovery in settings with noticeable dependence structure, while the performance of k-means and SOM deteriorates slightly.

In comparison to others mixture model methods (Figure 5.9 bottom), `MixDTrees-Dev MAP` also obtains a significantly higher specificity and sensitivity in almost all experimental

settings. The mixture of Gaussians with diagonal covariance matrices performs well in the independent case (experimental setting 1), which meets the model assumptions, but it has poor results in experiments with higher dependence (p-values < 0.05 for settings 3, 4 and 5). The mixture of Gaussians with full covariance matrix (`MoG-Full`) has a reasonable sensitivity in all settings, but very poor specificity (p-value < 0.05 in settings 3, 4 and 5 for sensitivity and in all settings for specificity). The reason for these results is that `MoG-Full` tends to have some clusters with few data points, as a reflection of over-fitting [143]. Note that we use a MAP estimate for `MoG-Full` to minimize this problem. `MixDTrees-Dev` with MLE estimates (`MixDTrees-Dev MLE`) has good overall performance, but it is outperformed by `MixDTrees-Dev MAP` in all cases, except for experimental settings 1 and 5 (p-value < 0.05 for settings 2, 3 and 4). In experimental setting 5, where data are highly dependent, by definition, both methods work similarly.

These results demonstrate that the `MixDTrees-Dev` is a better alternative than SOM and k-means in all cases. In relation to other mixture models, `MixDTrees-Dev` represents a good trade-off between a complex model class, such as multivariate Gaussian with full covariance matrices, and the simple Gaussian with diagonal covariance matrices. Furthermore, MAP estimates of the `MixDTrees-Dev` represent a more robust alternative to the MLE counterpart.

5.4.2 `MixDTrees` with Estimated Structure

To evaluate the application of our method in real biological data, we make use of gene expression from lymphoid cell development. First, we compare a `DTree` inferred from the whole data with the lymphoid developmental tree. Then, we apply `MixDTrees-Str` to find modules of co-regulated genes, and evaluate the results with GO and KEGG enrichment analysis (Section 5.4.2). Finally, we compare our method with other unsupervised learning methods. Additionally, to investigate characteristics of `MixDTrees-Str` and compare it with other methods, we use simulated data from mixture models with different degrees of variable dependence.

Data

Lymphoid Tree (`LymphoidTree`). We produce an expression compendium of mouse lymphoid cell development by combining measurements of wild-type control cells from several studies [3, 156, 165, 220, 229] based on the Affymetrix U74 platform. Our data contain four stages of early development hematopoietic cells [3] (hematopoietic stem cell (HSC), pluripotent progenitor (PPP), common lymphoid progenitor (CLP), common myeloid progenitor (CMP)); three B cell lineage stages [220] (pro-B cells (Bpro), pre-B cells (Bpre) and immature B cells (Bimm)); one Natural Killer (NK) stage [165]; and four T cell lineage stages (double negative T cells (TDN) [156], cd4 T cells (TCD4), cd8 T cells (TCD8) and natural killer T cells (TNK) [229]). The developmental tree describing the

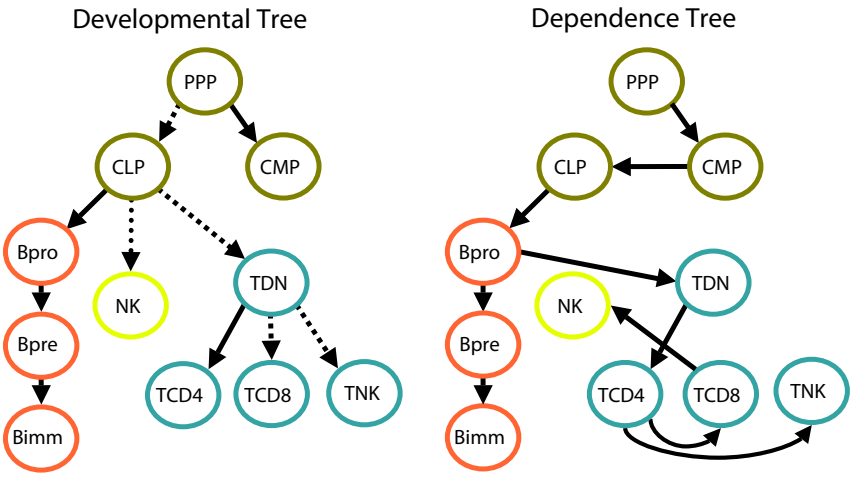


Figure 5.10: *We depict in the left the developmental tree with the stages contained in the Lymphoid data set. The dashed edges represent edges "wrongly" assigned in the* `DTree` *estimated from the Lymphoid data. Such edges connect pairs of vertices where the path length between these vertices in the developmental tree and estimated tree differs by one, while the dotted edge represents the case with path length differs by three. In the right, we have the* `DTree` *estimated from the Lymphoid data.*

order of differentiation of the cells is depicted in Figure 5.10 left. We pre-process the data as follows: we apply variance stabilization [104] on all chips, take median values of stages with technical replicates, use HSC values as reference values and transform all expression profiles to log-ratios. We keep genes showing at least a 2-fold change in one developmental stage. The final data set consists of 11 developmental stages and 3697 genes.

Simulated Data. We generate data from mixtures with four types of variable dependence ranging from: Gaussians with diagonal covariance matrix (Σ^{diag}), `DTree` with low variate dependence (Σ^{DTree^-}), `DTree` with high variate dependence (Σ^{DTree^+}) and Gaussians with full covariance matrix (Σ^{full}). These choices range from the independent case (Σ^{diag}) to the complete dependent case (Σ^{full}). For each setting, we generate ten such mixtures, and sample 500 development profiles from each. In all cases, we chose the μ from the range $[-1.5, 1.5]$, $L = 4$, $K = 5$ and mixture coefficients equal to $\alpha = (0.1, 0.15, 0.2, 0.2, 0.35)$. For Σ^{diag} , diagonal entries are sampled from $[0.01, 1.0]$, and non-diagonal entries are set to zero. For Σ^{DTree}, we randomly generate tree structures, one for each mixture component, and then chose $\sigma^2_{u|v,k}$ from $[0.01, 1.0]$ and $w_{u|v,k}$ from $[0.0, 0.5]$ for Σ^{DTree^-} and $w_{u|v,k}$ from $[0.0, 1.0]$ for Σ^{DTree^+}. The generation of Σ^{full} is based on the eigenvalue decomposition of the covariance matrix ($\Sigma = Q\Lambda Q^T$) as in [168], where Λ is drawn from $[0.01, 0.5]$. The orthogonal matrix Q is obtained by sampling values from a lower triangular matrix M from the range $[20, 40]$, followed by the Gram-Schmidt Orthogonalization procedure.

We apply MoG with full and diagonal covariance matrices and `MixDTrees-Str` with

MLE and MAP estimates to all data sets. The mixture estimation method is initialized with $K = 5$ random `DTrees` (or Multivariate Gaussians) as described in Section 2.3.2. Next, we train the mixture model using the EM-algorithm. We also performed clustering with k-means [146], self-organizing maps (SOM) [221] and spectral clustering [154]. We compare the class information from the data generation to compute the corrected Rand index [103] and evaluate the clustering solutions.

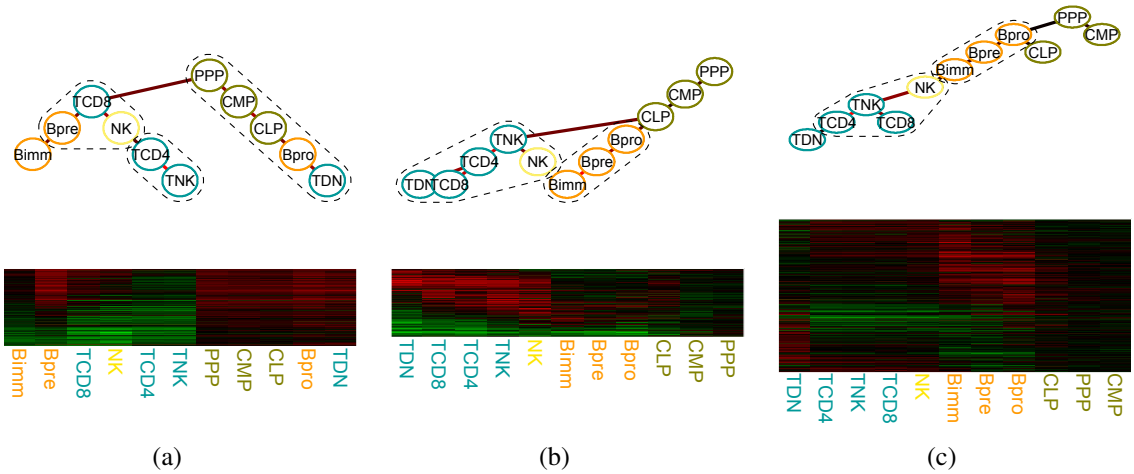
Enrichment of Gene Ontology and KEGG Pathways. Gene group validity is assessed by the results of Gene Ontology (GO) enrichment analysis [24], which helps the indication of functional roles of genes in a particular group. A more reliable and smaller alternative is the Kyoto Encyclopedia of Genes and Genomes (KEGG) [114], which has manually annotated gene pathways. In particular, several pathways related to lymphoid development such as signal transduction, immune system and cell cycle pathways, are described by KEGG. For the GO (or KEGG) enrichment analysis, we use the statistic of the Fisher-exact Test to obtain a list of GO terms (or KEGG pathways), whose participating genes are over-represented in a group as described in Appendix A.

Results

Inferring the `DTree` Structure. An initial question is how well we can recover the original developmental tree, as agreed upon by developmental biologists (Figure 5.10 left), if we apply the structure estimation method described in Sec 5.2.3 to the complete gene expression data (see Figure 5.10 right for the estimated `DTree`). To quantify the difference between these trees, we compute the path distance between all pairs of vertices, and calculate the Euclidean distance between the resulting distance matrices [202], which indicates a distance of 15.74. To assess the statistical significance of this distance, we generate 1000 random trees with the same distribution of out-degrees per vertex as the developmental tree. For each random tree, we compute the distance with the developmental tree. This test indicates a p-value of 0.002 of finding a distance as low as 15.74. Looking at these differences in detail, we can observe that 5 out of the 10 edges are correctly assigned, 4 edges connects vertices pairs with a path distance equal to 1, i.e., PPP and CLP, CLP and TDN, TDN and TCD8, and TDN and TNK, and one edge connect vertices with a path distance of 3 (NK is connected to TCD8 instead the CLP). Furthermore, "wrong" edges have a tendency to be connected to vertices in the same level of the developmental tree (e.g. TCD8 and TNK both connected with the TCD4).

Another important question is how well does the `DTree` capture dependence in the data? One simple way to assess this is to measure the proportion of the mutual information represented in the tree edges, in comparison to the total mutual information of all pairs of variables with the "treeness" index (Eq. 5.17).

For example, the score for the developmental tree (Figure 5.10 left) is 0.22, whereas for the estimated `DTree` (Figure 5.10 right), the "treeness" index is 0.42. For measuring the



(a) (b) (c)

Figure 5.11: *We depict the* `DTree` *and expression profiles of groups 1 (a), 4 (b) and 5 (c) from* `MixDTrees-Str MAP` *for the Lymphoid data. Dashed shapes around developmental stages represent the strongly connected components. See Section 5.3.3 for complete description of the plotting procedure.*

statistical significance of this, we generate random data by shuffling values of gene expression profiles x_i and estimating a `DTree` from this random data, which indicates a p-value of 0.0001.

Inferring Gene Modules with `MixDTrees-Str`**.** We estimate `MixDTrees-Str MAP` from the Lymphoid data following the protocol used for the simulated data. The Bayesian information criteria [145] indicates 13 groups as optimal.

First, we measure the average treeness of the `MixDTrees-Str` (we calculate Eq. 5.17 and take the sum weighted by α). For the `MixDTrees-Str MAP` this value is 0.54, which indicates an increase of 28% over the treeness index for the single `DTree`. This supports our claim that mixture of Dependence Trees with estimated structures is more successful in modeling dependencies in the data.

In relation to the groups of co-expressed genes found by `MixDTrees-Str`, in general, stages from the same developmental lineage are at same branches of the estimated `DTree` structure. Furthermore, groups present prototypical expression patterns such as over-expression in cells from a particular lineage, but not in other lineages (e.g., groups 2 and 5 for B cells, groups 4 and 6 for T cells and group 11 for Natural Killer cells) or groups displaying under-expression in particular lineages (e.g., groups 7 and 12 for T cells and groups 10 and 12 for B cells).

In Figure 5.11, we display some of these groups, which we discuss in more details. Group 1 (Figure 5.11 (a)) is an interesting case, where the `DTree` structure differs considerably from the developmental tree. On the right branch, we found a SCC (stages PPP, CLP, CMP, TDN, Bpro) with only early developmental stages, and all of them display high over-expression patterns. On the other hand, the majority of stages in the SCC on the

left branch (Bimm, Bpre, TCD8, NK, TCD4, TNK) are immature developmental stages (leaves in the developmental tree depicted in Figure 5.10 left). Enrichment analysis using GO and KEGG shows that group 1 is over-represented for *cell cycle* and *dna repair* (p-values < 0.001). This matches the biological knowledge that earlier differentiation stages of development are cycling cells, while immature cells are resting [140, 177]. Group 4 (Figure 5.11 (b)) contains a SCC (left branch) with all T cell stages plus the closely related NK cell. At these stages, genes display an over-expression pattern. Enrichment analysis indicates over-representation for Gene Ontology terms as *T cell activation, differentiation and receptor signaling*; and KEGG pathways such as *T cell signaling* and *NK cell mediated cytotoxicity* (p-values < 0.001). Similarly, group 5 (Figure 5.11 (c)) has a SCC with all B cell stages. Furthermore, for B cell stages, genes are preferentially over-expressed. GO analysis indicates enrichment for terms such as *B cell activation* (p-values < 0.001), while KEGG analysis indicates enrichment in pathways such as *Hematopoietic cell lineage* and *B Cell receptor signaling* (p-values < 0.05). These results show how `MixDTrees-Str` can be used to find groups of biologically related genes, as the associated `DTree` structure adds relevant information regarding expression similarity of developmental stages.

Comparison with other Clustering Methods . For comparison purposes, we also perform clustering of the Lymphoid data with other methods: k-means, self-organizing maps (SOM), MoG with full covariance matrix, MoG with diagonal matrix and the bi-clustering methods Samba [210] and non-negative matrix factorization [31]. Additionally, we evaluate distinct variations of `MixDTrees`: `MixDTrees-Str` with MAP and MLE estimates, and `MixDTrees-Dev` with the developmental tree from Figure 5.10 (left) as structure.

To evaluate the performance of the methods, we use a heuristic of comparing p-values of KEGG enrichment analysis in a similar way as in [73]. The results of the comparison of `MixDTrees-Str MAP` and MoG diag can be see in Figure 5.12. In short, the best method should present a higher enrichment for a higher number of KEGG pathways. As illustrated in Figure 5.12, `MixDTrees-Str MAP` is superior to MoG diag in 9 out of 11 pathways. Furthermore, most of the 11 KEGG pathways enriched with a p-value < 0.05 in one of the methods (points depicted in Figure 5.12) are directly involved in immune system and developmental processes. We apply the same procedure for all pairs of methods and count the events $\{p\text{-value } \mathrm{m}_1 < p\text{-value } \mathrm{m}_2\}$, where m_1 and m_2 are the two methods in comparison. As can be seen in Figure 5.13 (left), `MixDTrees-Str MAP` outperforms all methods, while `MixDTrees-Str MLE` and k-means also obtained higher enrichment than other methods. Overall, SOM, MoG Full and Samba obtain poor enrichment results. In fact, these methods are outperformed by all other methods. We repeat the same analysis for GO enrichment (see Figure 5.13 right). The result are in agreement with the KEGG enrichment analysis, that is, `MixDTrees-Str MAP` has higher enrichment than all other methods, while SOM and MoG Full obtain poor results.

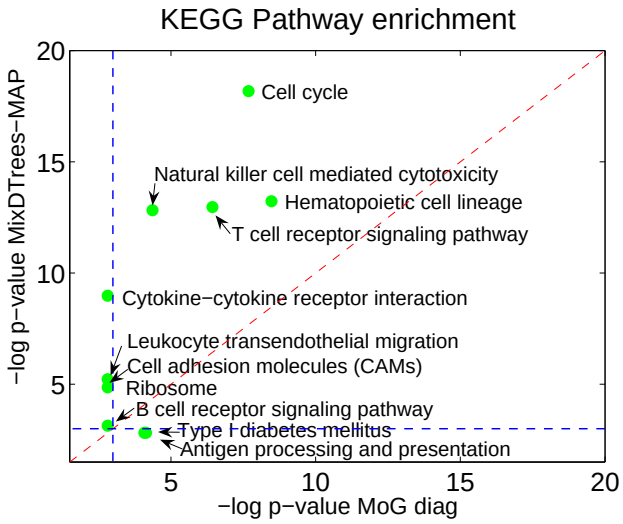


Figure 5.12: *We depict the scatter plot comparing the KEGG pathway enrichment of* `MoG diag` *(x-axis) and* `MixDTrees-Str-MAP` *(y-axis). We use $-log(p)$-values, where higher values indicate a higher enrichment. The blue lines correspond to $-log(p)$-value cut-off used (p-value of 0.05). Only KEGG pathways with a $-log(p)$-value higher than (2.99) in one of the results are included.* `MixDTrees-Str-MAP` *has a higher enrichment for 9 out of the 11 KEGG pathways.*

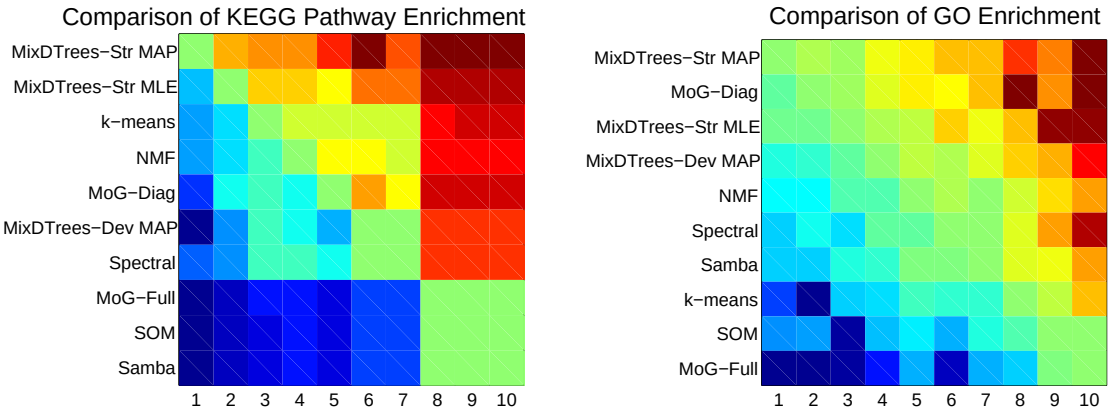


Figure 5.13: *Heat-maps plot displaying the comparison of KEGG (left) and GO (right) enrichment for 10 distinct clustering methods. Red (or blue) values indicate that the method in the y-axis has a higher (or lower) count of enriched KEGG pathways (GO terms) than the method on the x-axis. The numbers on x-axis correspond to the methods in the y-axis.*

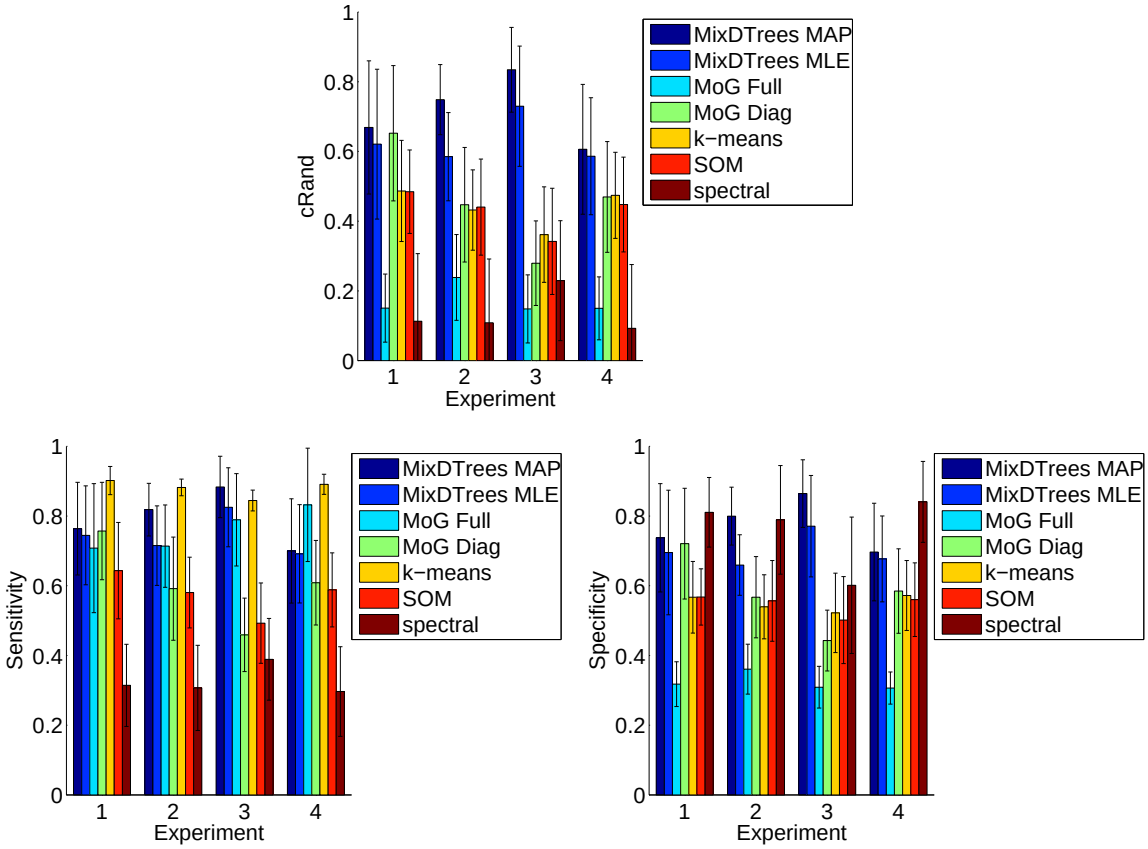


Figure 5.14: *We depict the mean corrected Rand (top), sensitivity (bottom left) and specificity (bottom right) of true label recovery for distinct clustering methods (y-axis) against data generated with distinct model assumptions (x-axis) (1 for Σ^{diag}, 2 for Σ^{DTree^-}, 3 for Σ^{DTree^+} and 4 for Σ^{full}). These choices range from the independent case Σ^{diag} to the complete dependent case Σ^{full}.*

Simulated Data. As expected, every method performs well on the data generated with the corresponding model assumptions (see Figure 5.14). An exception is the `MoG` with full covariance matrices, which has low corrected Rand for all data sets. An analysis of the specificity index indicates that the poor performance of MoG Full is caused by over-fitting, since it tends to merge real groups (see Figure 5.14 bottom right). Moreover, spectral clustering presents very low sensitivity values (see Figure 5.14 bottom left), which indicates a tendency to split real groups. In both data from Σ^{DTree}, `MixDTrees-Str MAP` has higher values than `MixDTrees-Str-MLE`, which indicates a higher robustness of the MAP estimates (a paired t-test indicated superiority of `MixDTrees-Str MAP` with p-value < 0.05 in both Σ^{DTree^-} and Σ^{DTree^+}). Also, `MixDTrees-Str MAP` obtains the highest values in all settings (p-value < 0.05), outperforming `MoG` Full, `MoG` Diagonal, k-means, SOM and spectral clustering, with the exception of MoG Diagonal in the Σ^{diag} data. These results show that `MixDTrees-Str-MAP` has a better performance than compared methods in data coming from distinct dependence structures, and it is robust against over-fitting.