Journal of Statistical Software

April 2016, Volume 70, Issue 2. doi: 10.18637/jss.v070.i02

GMCM: Unsupervised Clustering and Meta-Analysis Using Gaussian Mixture Copula Models

Anders E. Bilgrau
Aalborg University

Poul S. Eriksen
Aalborg University

Jakob G. Rasmussen
Aalborg University

Hans E. Johnsen
Aalborg University Hospital

Karen Dybkær
Aalborg University Hospital

Martin Bøgsted
Aalborg University Hospital

Abstract

Methods for clustering in unsupervised learning are an important part of the statistical toolbox in numerous scientific disciplines. Tewari, Giering, and Raghunathan (2011) proposed to use so-called Gaussian mixture copula models (GMCM) for general unsupervised learning based on clustering. Li, Brown, Huang, and Bickel (2011) independently discussed a special case of these GMCMs as a novel approach to meta-analysis in high-dimensional settings. GMCMs have attractive properties which make them highly flexible and therefore interesting alternatives to other well-established methods. However, parameter estimation is hard because of intrinsic identifiability issues and intractable likelihood functions. Both aforementioned papers discuss similar expectation-maximization-like algorithms as their pseudo maximum likelihood estimation procedure. We present and discuss an improved implementation in R of both classes of GMCMs along with various alternative optimization routines to the EM algorithm. The software is freely available in the R package **GMCM**. The implementation is fast, general, and optimized for very large numbers of observations. We demonstrate the use of package **GMCM** through different applications.

Keywords: **GMCM**, unsupervised learning, clustering, high-dimensional experiments, meta-analysis, reproducibility, evidence aggregation, copulas, p value combination, **idr**, **Rcpp**, R, C++.

1. Introduction

Unsupervised learning based on cluster analysis is an important discipline in many fields of science and engineering to detect groups of data with similar properties. Gaussian mixture

models (GMMs) are perhaps the most widely used method for model based clustering of continuous data. However, the assumption of jointly normally distributed clusters in GMMs is often violated. Tewari *et al.* (2011) presented the semi-parametric class of Gaussian mixture copula models (GMCMs) for general clustering in unsupervised learning and highlighted them as a flexible alternative to GMMs when obviously non-normally distributed clusters are present. The attractiveness of the GMCMs is predominantly due to an invariance under all monotone increasing marginal transformations of the variables. This scale invariance of the variables stems from the rank-based nature of copula models and makes the GMCMs highly versatile.

The GMCMs have found some success in applications after Li *et al.* (2011) independently from Tewari *et al.* (2011) proposed using a special-case for a non-standard meta-analysis methodology named reproducibility analysis. Their method has been adopted by the ENCODE project (Bernstein, Birney, Dunham, Green, Gunter, and Snyder 2012, p. 58; The ENCODE Consortium 2011, p. 15) and applied on ChIP (chromatin immunoprecipitation) sequencing data. The meta-analysis approach with GMCMs consists of clustering genes or features that agree on statistical evidence and those that do not. In other words, the features are clustered into a reproducible and an irreproducible group. The flexibility of the GMCMs makes them suitable for meta-analysis of multiple similar experiments.

The work of Li *et al.* (2011) is especially important in genomics as both data and results are subject to substantial variability due to limited samples sizes, high-dimensional feature spaces, dependence between genes, and confounding technological factors. This high variability has brought into question the reliability and reproducibility of many genomic results (Ioannidis, Ntzani, Trikalinos, and Contopoulos-Ioannidis 2001; Ein-Dor, Zuk, and Domany 2006; Tan, Downey, and Spitznagel 2003). Others, however, argue that the lack of reproducibility is only superficial (Zhang *et al.* 2008). Together with a rapid evolution of many different high-throughput technologies and vast online repositories of publicly available data, this motivates the need for a robust and flexible meta-analysis toolbox, which can evaluate or aggregate results of multiple experiments even across confounding factors such as differing technologies.

The high flexibility of the GMCMs comes at a cost, however. The likelihood is difficult to evaluate and maximize, partly because of intrinsic identifiability problems as we describe in detail later. We have solved some of the issues and implemented them in the R (R Core Team 2016) package **GMCM** (Bilgrau, Boegsted, and Eriksen 2016).

Although copula theory is an elegant way of approaching rank-based methods, we present the GMCMs in a more traditional fashion. We refer to the general model of Tewari *et al.* (2011) simply as the *general model* or *general GMCM* and the special case model of Li *et al.* (2011) is referred to as the *special model* or *special GMCM*.

In the following, we present the general GMCM followed by the special GMCM and the derivation of the likelihood function. Subsequently, the key features of package **GMCM** are presented and compared to the **idr** package (Li 2014). The technical details of the problematic maximization of the likelihood are then discussed. Finally, our package is evaluated based on different applications before concluding with a discussion of GMCMs.

This document was prepared and generated using **knitr** (Xie 2013), a dynamic report generation tool inspired by `Sweave` (Leisch 2002), and the R packages **Hmisc** (Harrell, Jr 2016) and **RColorBrewer** (Neuwirth 2014). The simulation study was carried out using parallel computing with **doMC** and **foreach** (Kane, Emerson, and Weston 2013; Revolution Analytics

and Weston 2015; Revolution Analytics 2015).

2. Gaussian mixture copula models

2.1. The general GMCM for clustering

We consider a large $p \times d$ matrix $[x_{gk}]$ of observed values where the rows are to be clustered into m groups. The general GMCM assumes an m-component Gaussian mixture model (GMM) as a latent process, $\boldsymbol{Z} = (Z_1, \ldots, Z_d)^\top$, with the following distribution

$$\text{GMM:} \qquad \begin{cases} H \sim \text{Categorical}(\alpha_1, \ldots, \alpha_m), \\ \boldsymbol{Z} | H = h \sim \mathcal{N}_d(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h), \end{cases} \tag{1}$$

where $H \in \{1, 2, \ldots, m\}$ corresponds to the class and $\alpha_1, \ldots, \alpha_m$ are the mixture proportions satisfying $\alpha_h > 0$ for $h = 1, \ldots, m$ and $\sum_{h=1}^m \alpha_h = 1$. Thus, the latent GMM is parameterized by

$$\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_m, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m).$$

We denote the joint and kth marginal cumulative distribution functions (CDF) of the GMM by

$$\Gamma(\boldsymbol{z}; \boldsymbol{\theta}) = \sum_{h=1}^m \alpha_h \Phi(\boldsymbol{z}; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h) \quad \text{and} \quad \Gamma_k(z; \boldsymbol{\theta}) = \sum_{h=1}^m \alpha_h \Phi_k(z; \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h),$$

respectively, where Φ and Φ_k are the joint and kth marginal CDFs of the multivariate normal distributions, respectively. Analogous equations hold for the joint and marginal probability density functions (PDF) which we denote by lower-case γ and γ_k.

Let $\boldsymbol{X} = (X_1, \ldots, X_d)^\top$ be an observation with *known* marginal CDFs $F_1, \ldots, F_d$ and assume the relationship

$$X_k = F_k^{-1}(\Gamma_k(Z_k; \boldsymbol{\theta})), \quad \forall k \in \{1, \ldots, d\} \tag{2}$$

between the observed and the latent variables. By Equation 2 and the probability integral transform the vector $\boldsymbol{U} = (U_1, \ldots, U_d)^\top$ where $U_k = \Gamma_k(Z_k) = F_k(X_k)$ have uniformly distributed marginals.

When $F_1, \ldots, F_d$ are known we can derive an expression for the likelihood of this model. For later use we simplify the notation by introducing the vector functions $\Gamma_\circ : \mathbb{R}^d \times \Theta \to \mathbb{R}^d$ and $F_\circ : \mathbb{R}^d \to \mathbb{R}^d$ defined by

$$\Gamma_\circ(\boldsymbol{Z}; \boldsymbol{\theta}) = \big(\Gamma_1(Z_1; \boldsymbol{\theta}), \ldots, \Gamma_d(Z_d; \boldsymbol{\theta})\big)^\top \quad \text{and} \quad F_\circ(\boldsymbol{X}) = \big(F_1(X_1), \ldots, F_d(X_d)\big)^\top,$$

where Θ is the parameter space. The vector function $\Gamma_\circ$ applies the kth marginal transformation Γ_k on the kth entry of the observation and similarly does $F_\circ$. Again by the probability integral transform, $\boldsymbol{Z}$ is transformed by $\Gamma_\circ$ into the marginally uniformly distributed random vector $\boldsymbol{U}$ with CDF

$$C(\boldsymbol{u}; \boldsymbol{\theta}) = \Gamma\big(\Gamma_\circ^{-1}(\boldsymbol{u}; \boldsymbol{\theta}); \boldsymbol{\theta}\big).$$

The PDF c of $\boldsymbol{U}$ is computed by the change of variables theorem or by differentiation of C using the multivariable chain rule. If we abbreviate the notation by not explicitly stating the dependence on the parameters $\boldsymbol{\theta}$, the PDF is given by

$$c(\boldsymbol{u};\boldsymbol{\theta}) = \gamma(\Gamma_\circ^{-1}(\boldsymbol{u}))\left|J_{\Gamma_\circ^{-1}}(\boldsymbol{u})\right| = \frac{\gamma(\Gamma_\circ^{-1}(\boldsymbol{u}))}{\prod_{k=1}^{d}\gamma_k(\Gamma_k^{-1}(u_k))}, \tag{3}$$

since the Jacobian matrix $J_{\Gamma_\circ^{-1}}(\boldsymbol{u})$ is diagonal. The CDF C and PDF c are the so-called *copula* and *copula density* of the GMM model, respectively (Nelsen 2006). Hence $\boldsymbol{U}$ is distributed according to the Gaussian mixture copula density c, and the observation $\boldsymbol{X}$ is some marginal transformation of $\boldsymbol{U}$. The model is thus completely specified by

$$\text{GMCM:} \quad \begin{cases} H \sim \text{Categorical}(\alpha_1,\ldots,\alpha_m), \\ \boldsymbol{Z}|H = h \sim \mathcal{N}_d(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h), \\ \boldsymbol{U} = \Gamma_\circ(\boldsymbol{Z};\boldsymbol{\theta}), \\ \boldsymbol{X} = F_\circ^{-1}(\boldsymbol{U}). \end{cases} \tag{4}$$

From this, we see that the GMCM operates on three levels: a latent level $\boldsymbol{Z}$, a copula level $\boldsymbol{U}$, and an observed level $\boldsymbol{X}$. Figure 1 (A–C) illustrates the three levels of a 2-dimensional 3-component GMCM. Here, $F_\circ$ and $F_\circ^{-1}$ map panel A to B and B to A, respectively. Likewise, $\Gamma_\circ$ defines the mapping between panels C and B.

To assess the class of an observation, Tewari *et al.* (2011) proposed using

$$\kappa_h = \mathsf{P}(H = h \mid \boldsymbol{u}, \boldsymbol{\theta}), \tag{5}$$

which is the a posteriori probability that the observation was generated from component h. To decide the class for the observation, the maximum a posteriori (MAP) estimate can be used. That is, the h corresponding to $\max_h(\kappa_h)$.

2.2. The special-case GMCM for meta-analysis

In the Li *et al.* (2011) reproducibility analysis, the $p \times d$ matrix $[x_{gk}]$ consists of test statistics or p values testing the same null hypothesis for a large number p of, e.g., genes for each of $d \geq 2$ studies. Rows correspond to genes, indexed by g, and columns to experiments, indexed by k. Without loss of generality, *large* values are considered to be indicative of the alternative hypothesis. A prototypical example in genomics is a matrix of transformed p values for the hypothesis of no differential expression of genes between treatment and control groups for two or more experiments. The task is here to determine which genes g are jointly significant in all experiments. Ordinary meta-analysis methodologies involve combining confidence intervals of effect sizes, test statistics, or p values in a row-wise manner and assessing the significance whilst controlling for the number of false positives (Owen 2009).

Li *et al.* (2011) proposed a special case of Equation 4 with $m = 2$ components corresponding to whether the null or alternative hypothesis is true, where $h = 1$ corresponds to spurious signals and $h = 2$ to genuine ones. Hence α_1 and $\alpha_2 = 1 - \alpha_1$ are the fraction of spurious and genuine signals, respectively. Li *et al.* (2011) further imposed the following constraints on the parameters:

$$\begin{aligned} \boldsymbol{\mu}_1 &= \mathbf{0}_{d\times 1} \quad = (0, 0, \ldots, 0)^\top, \\ \boldsymbol{\mu}_2 &= \mathbf{1}_{d\times 1}\mu = (\mu, \mu, \ldots, \mu)^\top, \quad \mu > 0, \end{aligned} \tag{6}$$

and

$$\boldsymbol{\Sigma}_1 = \boldsymbol{I}_{d\times d} = \begin{bmatrix} 1 & 0 & \cdots \\ 0 & 1 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \qquad \boldsymbol{\Sigma}_2 = \begin{bmatrix} \sigma^2 & \rho\sigma^2 & \cdots \\ \rho\sigma^2 & \sigma^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \tag{7}$$

where $\rho \in [-(d-1)^{-1}, 1]$ and $\sigma^2 > 0$. The lower bound on ρ is a requirement for $\boldsymbol{\Sigma}_2$ to be positive semi-definite. In other words, if the null hypothesis is true, the latent variable is a d-dimensional standard multivariate normal distribution. If not, it is a latent d-dimensional multivariate normal distribution with equal means and a compound symmetry covariance structure. Figure 1 (D–F) shows an example of the observed, copula, and latent levels of the special GMCM where $d = 2$.

With the above constraints the special model contains only the parameters $\boldsymbol{\theta} = (\alpha_1, \mu, \sigma^2, \rho)$, i.e., the dimensionality of the parameter space is substantially reduced. Furthermore, all marginal CDFs are equal, $\Gamma_1 = \cdots = \Gamma_d$, and similarly are all PDFs equal, $\gamma_1 = \cdots = \gamma_d$.

Li *et al.* (2011) defined the *local irreproducibility discovery rate* of an observation as

$$\operatorname{idr}(\boldsymbol{u}) = \kappa_1 = \mathsf{P}(H = 1 \mid \boldsymbol{u}, \boldsymbol{\theta}), \tag{8}$$

analogously to the local false discovery rate (lFDR) of Efron (2004, 2005, 2007). Notice, that Equation 5 coincides with Equation 8 for the special model. As the multiple testing problem is present when more observations are obtained, an adjusted *irreproducibility discovery rate* was also defined by Li *et al.* (2011):

$$\operatorname{IDR}(\alpha) = \mathsf{P}(H = 1 \mid \boldsymbol{u} \in I_\alpha, \boldsymbol{\theta}), \tag{9}$$

where $I_\alpha = \{\boldsymbol{u} \mid \operatorname{idr}(\boldsymbol{u}) < \alpha\}$, i.e., the probability of a gene being non-reproducible while in the rejection region. The adjusted $\operatorname{IDR}(\alpha)$ relates to idr in the same manner as the marginal false discovery rate (mFDR) relates to the lFDR.

2.3. The GMCM likelihood function

Suppose we have observed p i.i.d. samples $\boldsymbol{x}_1 = (x_{11}, \ldots, x_{1d}), \ldots, \boldsymbol{x}_p = (x_{p1}, \ldots, u_{pd})$ from Equation 4 which can be arranged into the observation matrix introduced in Section 2.1. From these, the marginal uniform variables $\boldsymbol{u}_1 = F_\circ(\boldsymbol{x}_1) = (u_{11}, \ldots, u_{1d}), \ldots, \boldsymbol{u}_p = F_\circ(\boldsymbol{x}_p) = (u_{p1}, \ldots, u_{pd})$ are computed and are independent and identically distributed according to the copula density of Equation 3. The log-likelihood is thus given by

$$\begin{aligned} \ell(\boldsymbol{\theta}; \{\boldsymbol{x}_g\}_{g=1}^p) \propto \ell(\boldsymbol{\theta}; \{\boldsymbol{u}_g\}_{g=1}^p) &= \sum_{g=1}^p \log c(\boldsymbol{u}_g; \boldsymbol{\theta}) \\ &= \sum_{g=1}^p \log \sum_{h=1}^m \frac{\alpha_h}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_h|}} \exp\left(-\frac{1}{2}(\Gamma_\circ^{-1}(\boldsymbol{u}_g) - \boldsymbol{\mu}_h)^\top \boldsymbol{\Sigma}_h^{-1} (\Gamma_\circ^{-1}(\boldsymbol{u}_g) - \boldsymbol{\mu}_h)\right) \\ &\quad - \sum_{g=1}^p \sum_{k=1}^d \log \sum_{h=1}^m \frac{\alpha_h}{\sqrt{2\pi\Sigma_{hkk}}} \exp\left(-\frac{1}{2\Sigma_{hkk}} (\Gamma_k^{-1}(u_{gk}) - \mu_{hk})^2\right), \end{aligned} \tag{10}$$

since the Jacobian arising from transformation $F_\circ$ is not dependent on $\boldsymbol{\theta}$ (and thus constant when optimizing with respect to $\boldsymbol{\theta}$).

In practice, $F_1, \ldots, F_d$ are unknown and estimated by the empirical CDF

$$\hat{F}_k^{(p)}(x) = \frac{1}{p} \sum_{g=1}^{p} \mathbb{1}[x_{gk} \leq x].$$

Hence the pseudo-observations

$$\hat{u}_{gk} = \hat{F}_k^{(p)}(x_{gk}) = \frac{1}{p} \operatorname{rank}(x_{gk}) \tag{11}$$

of u_{gk} are plugged into the log-likelihood and the maximizing parameters are determined. However, since p is large, $\hat{F}_k^{(p)}$ is a good estimate of F_k and thus $\hat{u}_{gk} = \hat{F}_k^{(p)}(x_{gk}) \approx F_k(x_{gk}) = u_{gk}$. The GMCM is rank-based since plugging a variable into its empirical CDF corresponds to a particular ranking scheme in which the lowest value is awarded rank 1 and ties are given their largest available rank. To avoid infinities in the computations $\hat{u}_{gk}$ is rescaled by the factor $p/(p+1)$.

The usage of $\hat{u}_{gk}$ violates the assumption of independent observations as the ranking introduces dependency between the observations. The introduced dependency is arguably negligible when p is large. We ignore this problem and refer to Chen, Fan, and Tsyrennikov (2006) and the references therein for a more detailed discussion of this problem which is common to all copula model estimation procedures.

3. The GMCM package

3.1. Package overview

The **GMCM** package currently has 14 user visible functions of which the majority are for convenience. The functions are presented in Table 1 and the **GMCM** reference manual. Two different parameter formats are used depending on if the special or general model are used. In the general model a specially formatted list of parameters is used, which has the name `theta` in the function arguments. The function `rtheta` generates such a prototypical list with random parameters and `is.theta` conveniently tests if the argument is properly formatted. If the special model is to be used, the required parameters are simply given in a numeric vector $(\alpha_1, \mu, \sigma, \rho)$ of length 4, with name `par` in the arguments. The functions `meta2full` and `full2meta` provide the easy conversion between the general `theta` and the special `par` format.

The most important functions are `fit.full.GMCM` and `fit.meta.GMCM`. They allow to fit the general and special GMCMs, respectively. The `method` argument of these functions specifies the optimization routine to be used. If the general model is used `get.prob` returns a matrix of posterior probabilities κ_{gk} as defined in Equation 5. In the special model, the function `get.IDR` is used to compute local idr (i.e., the posterior probability of belonging to the irreproducible component) and adjusted IDR values.

The functions `SimulateGMMData` and `SimulateGMCMData` allow to simulate observations from the models specified in Equations 1 and 4, respectively.

Beside the following tutorial, a small usage example of the special model can also be run using `help("GMCM")`. All simulations and computations were carried out on a regular laptop (1.7 GHz Intel Core i5, 4GB DDR3 RAM).

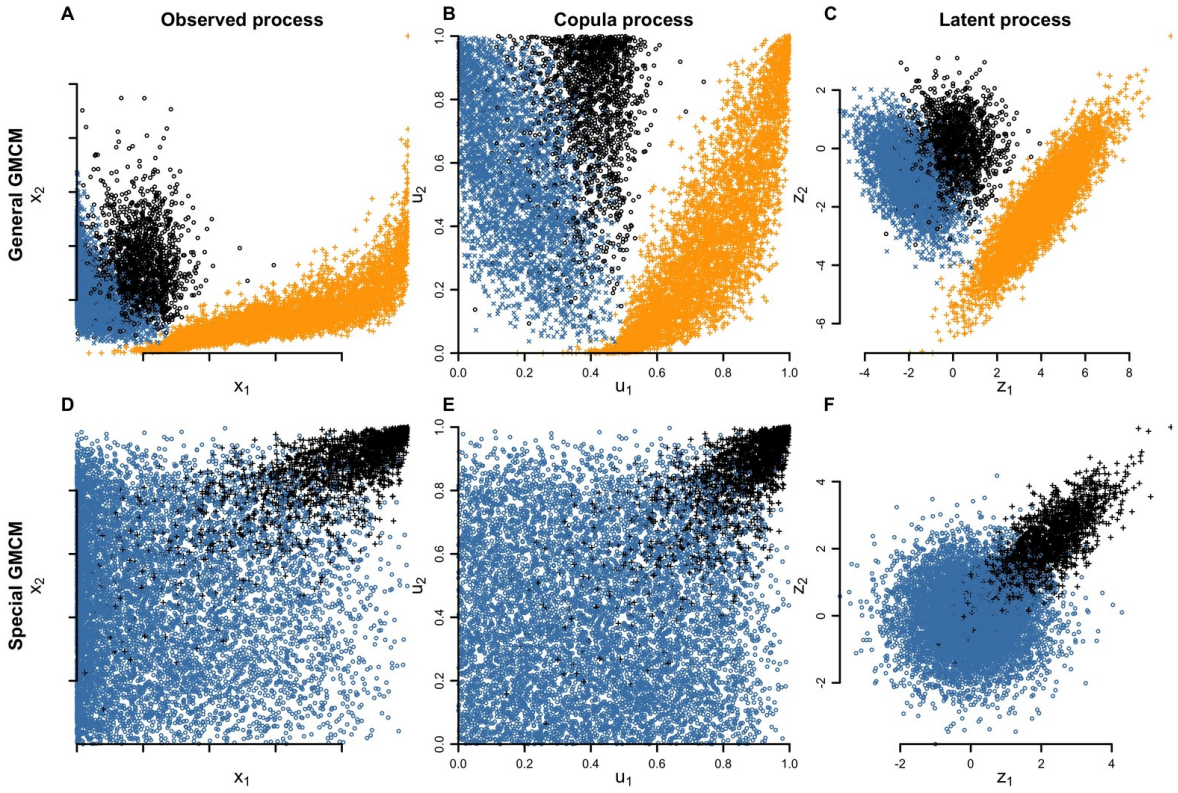

Figure 1: From left to right the observed, copula (or rank), and latent process are shown. The first and second row of panels illustrate 10,000 realizations from the general and special model, respectively. The component from which the realizations come is indicated by color and point-type. Each dimension in the special model corresponds to an experiment where simultaneously high values in both experiments are indicative of good reproducibility.

3.2. Using the package

We proceed with a small tutorial to present the use of the package. As an illustration, we load the package and simulate 10,000 observations from a 2-dimensional 3-component GMCM with randomly chosen parameters:

```
R> library("GMCM")
R> set.seed(100)
R> n <- 10000
R> sim <- SimulateGMCMData(n = n, theta = rtheta(m = 3, d = 2))
```

The `sim` object is a `list` containing the `matrix` of the realized latent process (`sim$z`), the `matrix` of true realizations from the GMCM density (`sim$u`), the formatted parameters (`sim$theta`), and the component from which each observation is realized (`sim$K`). Figure 2 shows the realized data.

Subsequently, we select a starting estimate from the data, fit the ranked observed data using Nelder-Mead (`"NM"`), and compute the posterior probabilities of each observation belonging to each component:

Function	Description	
`fit.full.GMCM`	Fit the general model.	(4)
`fit.meta.GMCM`	Fit the special model.	(4)(6)(7)
`get.prob`	Get class probabilities for the general model.	(5)
`get.IDR`	Get class probabilities (idr and IDR) for the special model.	(8)(9)
`SimulateGMCMData`	Generate samples from a GMCM.	(4)
`SimulateGMMData`	Generate samples from a GMM.	(1)
`Uhat`	Rank and scale the columns of the argument.	(11)
`choose.theta`	Choose starting parameters in the general GMCM.	
`full2meta`	Convert from `theta` format to `par`.	
`meta2full`	Convert from `par` format to `theta`.	
`rtheta`	Generate random `theta`.	
`is.theta`	Test if `theta` is correctly formatted.	
`rmvnormal`	Generate multivariate Gaussian observations.	
`dmvnormal`	Fast evaluation of the multivariate Gaussian PDF.	

Table 1: Overview of the user visible functions and their purpose in approximate order of importance. Please consult the documentation (e.g., `help("Uhat")`) for function arguments and return types. The relevant equations are indicated right-justified.

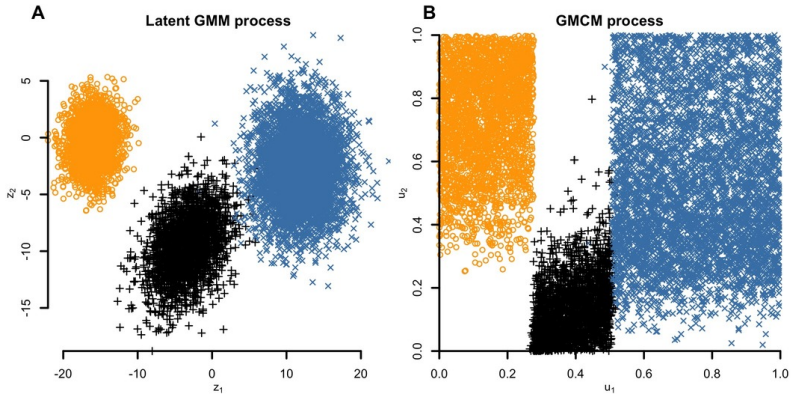

Figure 2: Panel A shows realizations from the latent process and panel B the corresponding marginally uniformly distributed process. Note, that while B shows the true realizations from the GMCM $\boldsymbol{u}_g$, the ranked observed values $\hat{\boldsymbol{u}}_g$ are almost visually identical because of the relative large number of observations.

```
R> ranked.data <- Uhat(sim$u)
R> start.theta <- choose.theta(ranked.data, m = 3)
R> mle.theta <- fit.full.GMCM(u = ranked.data, theta = start.theta,
+    method = "NM", max.ite = 10000, reltol = 1e-4)
R> kappa <- get.prob(ranked.data, theta = mle.theta)
R> Khat <- apply(kappa, 1, which.max)
```

The function `Uhat` ranks and rescales as described in Section 2.3. The `choose.theta` function uses the k-means algorithm on the rank level to find an initial set of parameters. From the k-means clustering results, crude estimates of the mixture proportions, mean values, and

	$\hat{H}$ (GMCM)			$\hat{H}$ (k-means)		
H	1	2	3	1	2	3
1	2747	0	0	2693	54	0
2	0	2276	6	5	2270	7
3	0	57	4914	26	882	4063

Table 2: Confusion matrices of the GMCM and k-means clustering results.

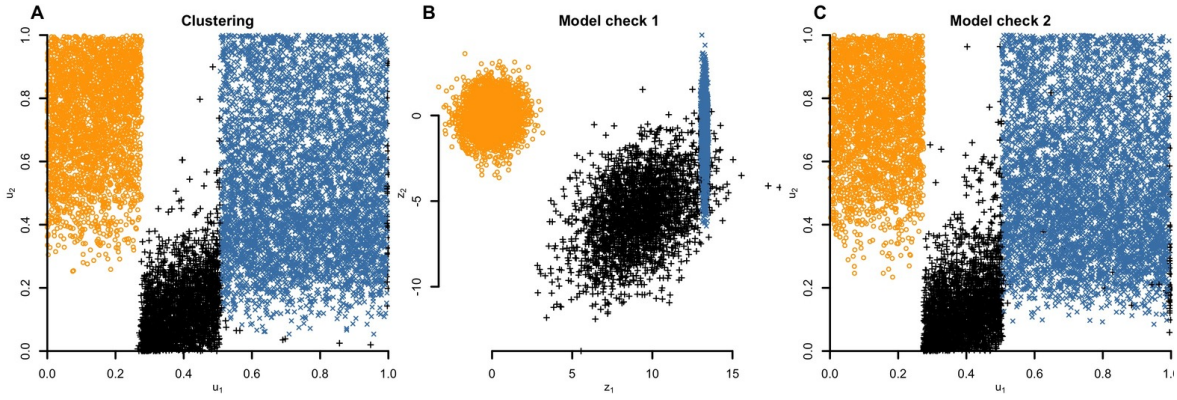

Figure 3: In panel A the estimated class labels of the observations are indicated by color and point-type. To allow to check the model panels B and C show 10,000 realizations from the GMM and GMCM using the fitted parameters.

variances can be computed. The correlations in all components are taken to be zero. This usually provides reasonable initial parameters. Objections could be raised against using such a procedure on the rank and not the latent level. However, as we are only interested in the relative position of the components, this often provides as reasonable starting parameters. The function `fit.full.GMCM` does the actual optimization of the likelihood to arrive at the maximum likelihood estimate (MLE). The default Nelder-Mead (`NM`) procedure converged in 499 iterations in about 6 seconds.

In serious applications the starting values should be chosen carefully and the algorithm ought to be started from different positions of the parameter space to investigate the stability and uniqueness of the MLE. The estimate with the largest likelihood should then be returned.

The confusion matrix obtained when comparing the GMCM clustering results to the true class labels, see Table 2, indicates an accuracy of 99.4%. In (unfair) comparison, the k-means algorithm has an accuracy of 90.3%. Figure 3 shows the clustering results and simple model checks by simulating from the fitted parameters. Though a high clustering accuracy is achieved, we see from the model check in Figure 3B compared to Figure 2A that the underlying parameters are not really identifiable. However, we see from panel C, that the fitted parameters model the observed ranks closely and thus provide a high predictive accuracy.

3.3. Runtime and technical comparison

For the special model, the **GMCM** package allows for an arbitrary number of dimensions (or experiments) d to be included whereas the **idr** package only supports $d = 2$. The **GMCM** package considerably decreases the per iteration runtime of the pseudo expectation-maximization

p	Package	Algorithm	Runtime (s)	Iterations (n)	s/n	Rel. speed
1,000	**idr**	PEM	3.03	22	0.138	50.4
	GMCM	PEM	1.27	125	0.010	3.7
	GMCM	NM	0.75	275	0.003	1.0
10,000	**idr**	PEM	17.64	15	1.176	143.7
	GMCM	PEM	4.16	163	0.025	3.1
	GMCM	NM	1.94	237	0.008	1.0
100,000	**idr**	PEM	257.63	17	15.155	304.2
	GMCM	PEM	40.79	258	0.158	3.2
	GMCM	NM	10.71	215	0.050	1.0

Table 3: Runtime comparisons of the **idr** and **GMCM** packages with increasing number of observations p. The benchmarked optimization procedures are the pseudo EM algorithm (PEM) and the Nelder-Mead (NM) method. The runtime is given in seconds. The last column shows the relative speed per iteration compared to the fastest procedure.

(PEM) algorithm compared to the **idr** package. The optimization procedures such as Nelder-Mead (NM), simulated annealing (SANN), and others which only rely on evaluations of the likelihood further reduce the runtime compared to the PEM.

Run and iteration times for an increasing number of observations are see Table 3 for a simulated dataset with parameters $(\alpha_1, \mu, \sigma, \rho) = (0.7, 2, 1, 0.9)$. The algorithms were all run with the starting values $(0.5, 2.5, 0.5, 0.8)$. The parameters were chosen such that the **idr** package does not converge prematurely.

To assess the optimization routines in the **idr** and **GMCM** packages, $1,000$ datasets with 10,000 observations were simulated from the special model with parameters $\boldsymbol{\theta} = (0.9, 3, 2, 0.5)$. The special model was fitted to each of the datasets using each of the available routines with random initial parameter values. Figure 4 shows the results from the fitting procedures. The maximum number of iterations was set to 2,000. The SANN procedure was given 3,000 iterations.

The clusters of parameter estimates away from the true values seen in Figure 4 presumably correspond to local maxima of the likelihood. Hence many of the procedures are fairly often caught in such local maxima. Interestingly, while the estimates of the standard deviation $\hat{\sigma}$ and correlation $\hat{\rho}$ for the PEM algorithm seem to be biased, the algorithm achieved a high clustering accuracy. We also see that the PEM algorithms in **GMCM** and **idr** behave quite differently. The maximal number of iterations, 2000, was hit only by the PEM algorithm 274 and 18 times for the **idr** and **GMCM** packages, respectively. Also notable is the factor 555 reduction in total runtime between the fastest and slowest fitting procedures.

All warnings produced by the PEM algorithm in **idr** were equal to `"NaNs produced"`. PEM in **GMCM** only warned that the maximum number of iterations was reached. The errors produced by SANN and L-BFGS-B seem to arise when the estimates of the covariance matrix became singular. The vast majority of the errors for L-BFGS-B were divergence to non-finite likelihood values. The unique error thrown by PEM (**idr**), `"missing value where TRUE/FALSE needed"`, seems to stem from a simple bug.

Considering computational efficiency and robustness, accuracy, and precision of parameter estimates, we chose the Nelder-Mead as the default optimization procedure.

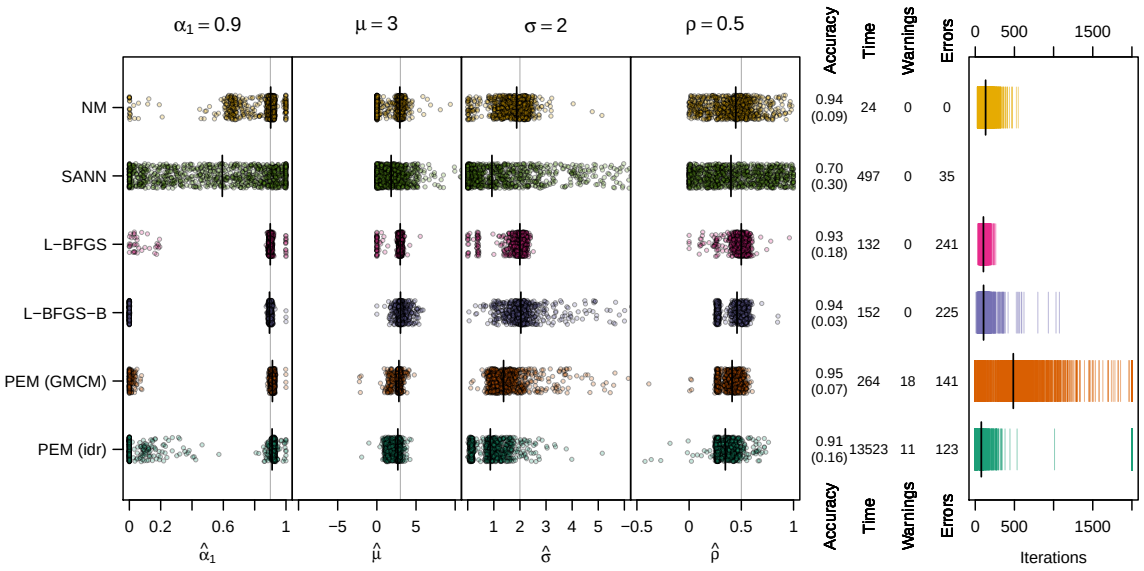

Figure 4: Parameter fitting results for the different optimization procedures. From left to right, the first four panels show plots of the fitted parameter estimates. The true parameter values are plotted as vertical lines. Next, the mean clustering accuracy (and standard deviation), total runtime in minutes for all 1000 fits, and the number of warnings and errors are shown. The last panel shows the number of iterations for each fit. The black vertical lines indicate the median.

3.4. Availability of the package

The **GMCM** package is open-source and available both from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=GMCM and from the GitHub repository at https://github.com/AEBilgrau/GMCM.git for bug reports as well as easy forking and editing.

4. Maximum likelihood estimation

4.1. Maximizing the likelihood

The optimization of the likelihood function in Equation 10 is non-trivial. There exists no closed form expression for Γ_k^{-1}. Furthermore there are intrinsic problems of identifiability of the GMCM parameters. These problems will greatly affect any estimation procedure.

Both Li *et al.* (2011) and Tewari *et al.* (2011) make use of a pseudo EM (PEM) algorithm to find the maximizing parameters. Tewari *et al.* (2011) use the PEM as a "burn-in" and then switch to a gradient descent algorithm. In both papers the likelihood function of the GMM, ℓ_{GMM}, specified by Equation 1, and the estimators for the corresponding EM algorithm are derived. The PEM algorithm then iteratively alternates between estimating pseudo-observations $\hat{z}_{gk} = \Gamma_k^{-1}(\hat{u}_{gk}; \boldsymbol{\theta})$ and subsequently updating $\boldsymbol{\theta}$ by an E and M step. While this intuitively is a viable approach, it effectively ignores the Jacobian of Equation 3 as the

transformation Γ^{-1} depends on the parameters $\boldsymbol{\theta}$. In short, the wrong likelihood is thus optimized and a pseudo (or quasi) maximum likelihood estimate is found. This may yield an inefficient optimization routine and biased parameter estimates. This problem of the PEM is appreciated by Tewari *et al.* (2011).

A fundamental problem with the PEM algorithm is the alternating use of pseudo-observations and parameter updates. The pseudo data is not constant in the ℓ_{GMM} which implies no guarantee of convergence nor convergence to the correct parameters.

To clarify, let $\boldsymbol{\theta}^{(m)}$ denote the mth estimate of $\boldsymbol{\theta}$. From $\boldsymbol{\theta}^{(m)}$, pseudo data is estimated by

$$\hat{z}_{gk}^{(m)} = \Gamma_k^{-1}(\hat{u}_{gk}; \boldsymbol{\theta}^{(m)}), \quad g \in \{1, \dots, p\},\ k \in \{1, \dots, d\}.$$

The PEM algorithm alternates between updating parameter estimates and pseudo data which results in the following log-likelihood values,

$$\begin{aligned} \dots, \ell_{\text{GMM}}(\boldsymbol{\theta}^{(m)} | \{\hat{\boldsymbol{z}}_g^{(m)}\}_g),\ \ell_{\text{GMM}}(\boldsymbol{\theta}^{(m+1)} | \{\hat{\boldsymbol{z}}_g^{(m)}\}_g), \\ \ell_{\text{GMM}}(\boldsymbol{\theta}^{(m+1)} | \{\hat{\boldsymbol{z}}_g^{(m+1)}\}_g),\ \ell_{\text{GMM}}(\boldsymbol{\theta}^{(m+2)} | \{\hat{\boldsymbol{z}}_g^{(m+1)}\}_g),\ \dots, \end{aligned}$$

given in the order of computation. Conventionally, convergence is established when the difference of successive likelihoods is smaller than some $\epsilon > 0$. The implementation of Li *et al.* (2011) through the R package **idr** determines convergence if

$$\ell_{\text{GMM}}(\boldsymbol{\theta}^{(m+1)} | \{\hat{\boldsymbol{z}}_g^{(m+1)}\}_{g=1}^p) - \ell_{\text{GMM}}(\boldsymbol{\theta}^{(m)} | \{\hat{\boldsymbol{z}}_g^{(m)}\}_{g=1}^p) < \epsilon,$$

where $\epsilon > 0$ is pre-specified. However, an increase in successive likelihoods is only guaranteed by the EM algorithm when the (pseudo) data are constant. Since both, the pseudo data and parameter estimates, have changed the above difference can be, and often is to our experience, negative. In the **idr** package this sometimes happens in the first iteration without warning. Such cases arguably stop the procedure prematurely since any negative difference obviously is smaller than some positive ϵ. The EM algorithm only guarantees that the difference

$$\ell_{\text{GMM}}(\boldsymbol{\theta}^{(m+1)} | \{\hat{\boldsymbol{z}}_g^{(m)}\}_{g=1}^p) - \ell_{\text{GMM}}(\boldsymbol{\theta}^{(m)} | \{\hat{\boldsymbol{z}}_g^{(m)}\}_{g=1}^p)$$

is non-negative and thus might be more suitable for determining convergence.

The PEM convergence criterion used by Tewari *et al.* (2011) corresponds to determining if the difference in successive parameter estimates is sufficiently small while recording the highest observed likelihood estimate. This partly remedies the problem. However, the PEM still inherits the conventional problems of the EM algorithm. It often exhibits slow convergence and offers no guarantee of finding the global optimum.

Our software package **GMCM** offers fast optimization of both the general and special model. Our implementation of the PEM algorithm supports various convergence conditions. By default, it determines on convergence if

$$\left| \ell_{\text{GMCM}}(\boldsymbol{\theta}^{(m+1)} | \{\hat{\boldsymbol{u}}_g^{(m)}\}_{g=1}^p) - \ell_{\text{GMCM}}(\boldsymbol{\theta}^{(m)} | \{\hat{\boldsymbol{u}}_g^{(m)}\}_{g=1}^p) \right| < \epsilon.$$

and returns the parameters which yield the largest likelihood value. These are not necessarily the parameters obtained in the last iteration. The internal function `PseudoEMAlgorithm` is called when `fit.full.GMCM` or `fit.meta.GMCM` are run with `method = "PEM"`.

Instead of the EM approach, however, we propose optimizing the GMCM likelihood function in Equation 10 using procedures which rely only on likelihood evaluations. To make this a feasible approach considerable effort has been put into evaluating the log-likelihood function of Equation 10 in a fast manner by implementing core functions in C++ using **Rcpp** and **RcppArmadillo** (Eddelbuettel 2013; Eddelbuettel and François 2011; François, Eddelbuettel, and Bates 2016). With fast likelihood evaluations the standard optimization procedure `optim` in R is used with various optimization procedures, such as Nelder-Mead (also known as the amoeba method), simulated annealing, and BFGS quasi-Newton methods.

When the parameters are passed to `optim` we use various transformations to reformulate the optimization problem as an unconstrained one. We logit-transform the mixture proportions. In the general model, a Cholesky decomposition combined with a log-transformation is used to ensure positive definiteness of the covariance matrices. In the special model, the variance σ^2 is ensured to be positive due to a log-transform. The restriction on the correlation ρ to be in the interval $[-(d-1)^{-1}, 1]$ is guaranteed by an affine and logit function composition.

Additional speed has also been gained by implementing a faster inversion of the marginals Γ_k. Similarly to Li *et al.* (2011), we linearly interpolate between function evaluations. However, we distribute the default 1,000 function evaluations to the components according to the current estimate of the mixture proportions. The determined number of function evaluations for component h within the kth dimension is then sampled equidistantly in the interval $\mu_{hk} \pm a\sqrt{\Sigma_{hkk}}$ where $a = 5$ by default. Lastly, the monotonicity of Γ_k is used to quickly invert the function by reflection around the identity line. Furthermore, we approximate the mixture CDF Γ_k by using the approximation of the error function $\text{erf}(x) \approx 1 - (a_1 t + a_2 t^2 + a_3 t^3)\exp(-x^2)$ where $t = 1/(1+bx)$ and a_1, a_2, a_3, and b are constants (Abramowitz and Stegun 1970, p. 299; Hastings, Hayward, and Wong 1955).

4.2. Identifiability of parameters

The model suffers from unidentifiable parameter configurations. As a consequence of the GMCM invariance to translations only relative distances between the location parameters $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m$ can be inferred. We arbitrarily anchor the first component at $\boldsymbol{\mu}_1 = \mathbf{0}$ as a partial solution. To account for scaling invariance, the first component is required to have unit variance in each dimension, that is $\Sigma_{1kk} = 1$ for all k. However, problems of identifiability persist in a number of scenarios. In cases where two or more components in the latent GMM are well-separated from each other the relative distances and component variances are not identifiable for all practical purposes. For example in the special GMCM, the parameter configuration $\boldsymbol{\theta} = (0.5, 10, 1, 0)$ will be indistinguishable from $(0.5, 100, 1, 0)$. The ranking destroys all information about the relative variances and distances between the well-separated components.

The clustering might also easily fail when the location and variance parameters for two or more components are similar along the same dimension. Suppose for example that $\boldsymbol{\mu}_1 = (0, 0)$, $\boldsymbol{\mu}_2 = (4, 0)$, and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_{2\times 2}$, i.e., the location and variance parameters are equal along the ordinate axis. In such cases, the ranking will create a homogeneous cluster which cannot be easily separated.

Even though the parameters may not be fully uniquely estimable in all cases, the general model can still be an effective clustering algorithm as measured by clustering accuracy.

Table 4 describes three situations in the special model where the parameter estimates and

Situation	α_1	μ	σ	ρ
1	1	·	·	·
2	0	·	·	0
3	·	0	1	0

Table 4: Equivalent optima for data corresponding to pure noise. A dot (·) denotes an arbitrary value. The given values need only to be approximate.

thus the following clustering should be carefully interpreted. If the parameter estimates approach any of the given numbers then the remaining parameters, represented by dots, are effectively non-identifiable. For example in Situation 1, if the mixture proportion α_1 approaches 1 then the remaining parameters can easily diverge as they no longer contribute to the likelihood. In Situation 2 where $\boldsymbol{\theta} = (0, \cdot, \cdot, \rho)$ extra caution should be displayed if ρ becomes substantially different from zero as all observations will be deemed reproducible. While the above corrections somewhat remedy these issues, the three situations can still be observed, especially when data consisting of nearly pure noise is supplied.

5. Applications

5.1. Reproducibility of microarray results

In molecular biology, microarrays are often used to screen large numbers of candidate markers for significant differences between case and control groups. Microarrays simultaneously probe the DNA composition or transcribed RNA activity of multiple genes in a biological sample. The number of probes ranges in the orders of 10,000 to 6,000,000, depending on the specific microarray.

In the study of haematological malignancies it is of biological interest to know how normal B-lymphocytes develop (Lenz and Staudt 2010; Rui, Schmitz, Ceribelli, and Staudt 2011; Küppers 2005). Hence, B-cells from removed tonsil tissue of six healthy donors were sorted and isolated using fluorescence-activated cell sorting (FACS) into five subtypes of B-cells: Naïve (N) B-cells, Centrocytes (CC), Centroblasts (CB), Memory (M) B-cells, and Plasmablasts (PB). As part of the immune response to an infection, the CBs proliferate rapidly and become CCs within the so-called germinal centers (GC). The 6×5 samples were profiled with *Affymetrix GeneChip HG-U133 plus 2.0* (U133) microarrays (see Bergkvist *et al.* 2014, for further details).

It is, e.g., of interest to identify which gene expressions have been altered within the GCs from which the CCs and CBs come. We therefore tested the hypothesis of no difference in genetic expression between CC and CB samples against N, M, and PB samples for all the gene expressions present on the U133 array. Since gene profiling technologies are rapidly evolving the experiment was later repeated with new donors and on the newer *GeneChip Human Exon 1.0 ST* (Exon) microarray.

The 30 samples on the U133 arrays and the 30 samples on Exon arrays were pre-processed and summarized to gene level separately and independently using the RMA algorithm with the **Bioconductor** (Gentleman *et al.* 2004) package **affy** (Gautier, Cope, Bolstad, and Irizarry 2004) using custom CDF files (Dai *et al.* 2005). This pre-processing resulted in the genetic

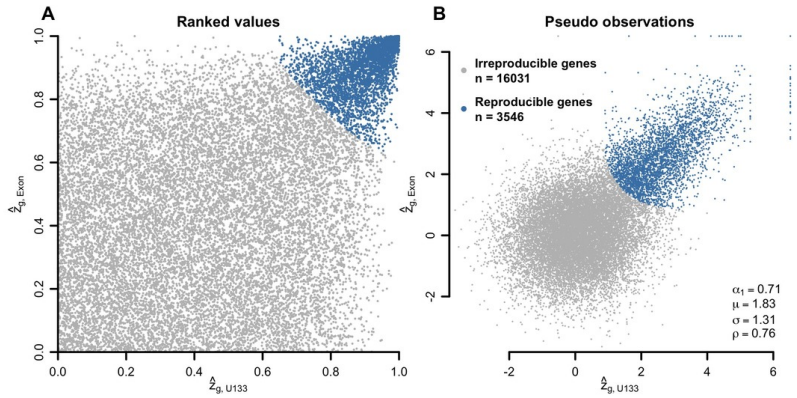

Figure 5: Panel A shows a plot of the scaled ranks of p values for the Exon experiment against the scaled ranks of the p values for the U133 experiment. Presumably, genes located in the upper left or lower right corner of the plots are false positive results in either experiment. Panel B shows the estimated latent GMM process. The fitted parameters shown are used to marginally transform panel A into the B.

expression levels of 37,923 probe-sets for the U133 arrays and 19,750 probe-sets for the Exon arrays both annotated with Ensembl gene identifiers (ENSG identifiers).

Each experiment was analyzed separately using a mixed linear model and the empirical Bayes approach using the **limma** package (Smyth 2004) to test the hypothesis of no differential expression for each gene between the CC + CB and the N + M + PB groups. The tests yield two lists of p values for the U133 and Exon arrays.

The p value lists were reduced to the 19,577 common genes present on both array types and combined into a matrix $[x_{gk}]_{19577\times 2}$ where x_{gk} is one minus the p value for varying gene expression for gene g in experiment $k \in \{\text{U133}, \text{Exon}\}$.

To determine the genes which are differentially expressed in a reproducible way, the special GMCM was fitted with the Nelder-Mead optimization procedure using `fit.meta.GMCM`. The procedure was started in 3 different starting values and the estimate with the largest log-likelihood was chosen. The best estimate converged in 311 iterations. Subsequently, the local and adjusted IDR values were computed with `get.IDR`. A total of 3,546 genes (18.1%) were found to have an adjusted IDR value below 0.05 and deemed reproducible. The results are illustrated in Figure 5 along with the parameter estimates. The algorithm successfully picks p values which are high-ranking in both experiments.

If the MAP estimate is used – which corresponds to a local idr value less thanӑ0.5 – then 4, 510 (23%) genes are deemed reproducible.ă This percentage isăconsistentăwith the estimate of the mixture proportion $\alpha_1 = 0.71$ of the null component.

Note, since no biological ground truth is available, the accuracy cannot be determined. However, since genes which are not differentially expressed are expected to be irreproducible the accuracy may be high.

For comparison, the number of genes marginally significant at 5% significance level after Benjamini-Hochberg (BH) correction (Benjamini and Hochberg 1995) is 3,968 and 6,713 for the U133 and Exon experiments, respectively. The number of commonly significant genes (i.e., simultaneously significant in both experiments) is 3,140 or 16%. This corresponds to

the common approach of using Venn diagrams.

The list of reproducible genes, which can be ranked by their idr values, provides a more accessible list of genes for further biological down-stream analyses than the unordered list of genes obtained by the Venn diagram approach.

The p values from the experiments are available in **GMCM** via `data("u133VsExon", package = "GMCM")`.

5.2. Effects of cryopreservation on reproducibility

Cryopreservation is a procedure for preserving and storing tissue samples by cooling them to sub-zero temperatures. It is convenient for researchers and a crucial component of biobanking. Cryopreservation is usually assumed by default to alter the biological sample since many cryopreserving substances are toxic, the freezing procedure may damage the sample due to ice crystallization, and it may induce cellular stress response. Fresh is therefore considered favorable to cryopreserved tissue. Few studies have analyzed the effect of the cryopreservation on phenotyping and gene expression. Recently, we studied cryopreservation to gauge the actual impact of the cryopreservation on global gene expression in a controlled comparison of cryopreserved and fresh B-lymphocytes. Similarly to the above, the B-cells were prepared from peripheral blood of 3 individual healthy donors and FACS sorted into 2×4 B-cell subtypes, Immature (Im), Naïve (N), Memory (M), and Plasmasblasts (PB). Half of the samples were cryopreserved and thawed prior to the gene expression profiling using the Exon array while the other half was profiled fresh. The resulting data was pre-processed using RMA (see Rasmussen *et al.* 2014, for further details). As a supplement to **?**, we performed a reproducibility analysis using the special model. The analysis nowăpresented hereăwas originally omittedăfrom **?** due to our concerns about complexity and the length added to the manuscript.

If cryopreservation has relatively negligible effects on global screenings, then a high reproducibility should be expected for differential expression analyses within the fresh and frozen samples – however only for the true differentially expressed genes. For each probe set, the samples were analyzed using linear mixed models as described in Rasmussen *et al.* (2014) and the hypothesis of no differential expression between pre (Im + N) and post germinal center (M + PB) cells was tested for both fresh and cryopreserved samples separately to mimic the situations where only fresh or frozen samples are available. The special GMCM was fitted using the resulting absolute value of the test statistics to determine the level of reproducibility of each probe set. Local and adjusted irreproducible discovery rates were computed for all probe sets and this level of reproducibility was discretized into three groups: highly reproducible ($\text{IDR}_g < 0.05$, cf. Equation 9), reproducible ($\text{idr}_g < 0.5$, cf. Equation 8), and irreproducible ($\text{idr}_g \geq 0.5$).

The best parameter estimate of 40 fits was $\boldsymbol{\theta} = (\alpha_1, \mu, \sigma, \rho) = (0.73, 1.08, 1.32, 0.86)$. The reproducibility analysis deemed 1,667 (8.9%), 1,402 (7.5%), and 15,639 (83.6%) genes highly reproducible, reproducible, and irreproducible, respectively. Figure 6 shows these classifications of the p values for differential expression between pre and post germinal cells for the fresh and frozen samples. The total of 3,069 (16.4%) reproducible probe sets seems quite high and agrees with the estimated mixture proportion of 0.73. Again, the model correctly captures the genes with simultaneously low p values. Recall also that non-differentially expressed genes are expected to be irreproducible and the actual accuracy is thus much higher

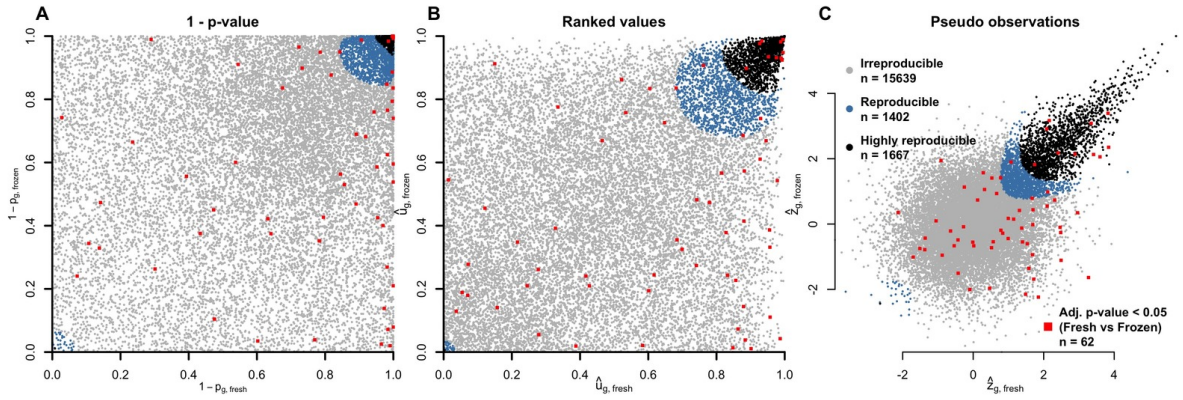

Figure 6: Results from the reproducibility analysis of cryopreserved samples. Panel A shows the p values p_g for the test of no differential expression between the pre- and post germinal center groups for fresh and frozen samples. Panel B shows the corresponding ranked p values $\hat{u}_g$, and panel C shows the estimated latent process $\hat{z}_g$. The estimated level of reproducibility for each probe set is color coded according to the legend in panel C. Genes significantly different across fresh and frozen samples are plotted as red squares regardless of the reproducibility level.

although it (again) cannot easily be estimated when no biological ground truth is available.

Naturally, one might wonder whether genes changed due to cryopreservation to a large extent are deemed irreproducible. The paired design allowed us to investigate this hypothesis. The hypothesis of no difference in expression between fresh and frozen samples for each gene was therefore tested and the significant BH-adjusted p values at the 5% level are highlighted in Figure 6. The expectation above was then tested using a test for non-zero Spearman correlation between the p values and idr values which yielded a non-significant correlation ($\rho = 0.009, p = 0.21$). In other words, high evidence for a change between fresh and frozen is not associated with greater irreproducibility (idr). Alternatively, a Fisher's exact test also did not yield a difference in odds (odds ratio $= 0.67, 95\%$ CI $= (0.36, 1.32), p$ value $= 0.23$) of having a BH-adjusted significant change due to cryopreservation in the reproducible group (odds $= 48/(15591-48)$) compared to the irreproducible (odds $= 14/(3055-14)$). Thus there is no evidence for an over-representation of the irreproducible genes among the significant ones. We might thus conclude that though some genes change due to cryopreservation, the differential analysis between subgroups to a great extent still yields the same results whether the samples are fresh or frozen.

Lastly, notice that some genes in the lower left corner of Figure 6 (A–C) near the origin are also being deemed reproducible. This is an artifact of the model due to the high correlation of $\rho = 0.86$ in the reproducible component.

The p values and test scores are available in **GMCM** via `data("freshVsFrozen", data = "GMCM")`.

5.3. Image segmentation using the general GMCM

In computer vision and graphics, image segmentation is useful to simplify and extract features of pictures. To illustrate the flexibility of the model and the computational capability of the

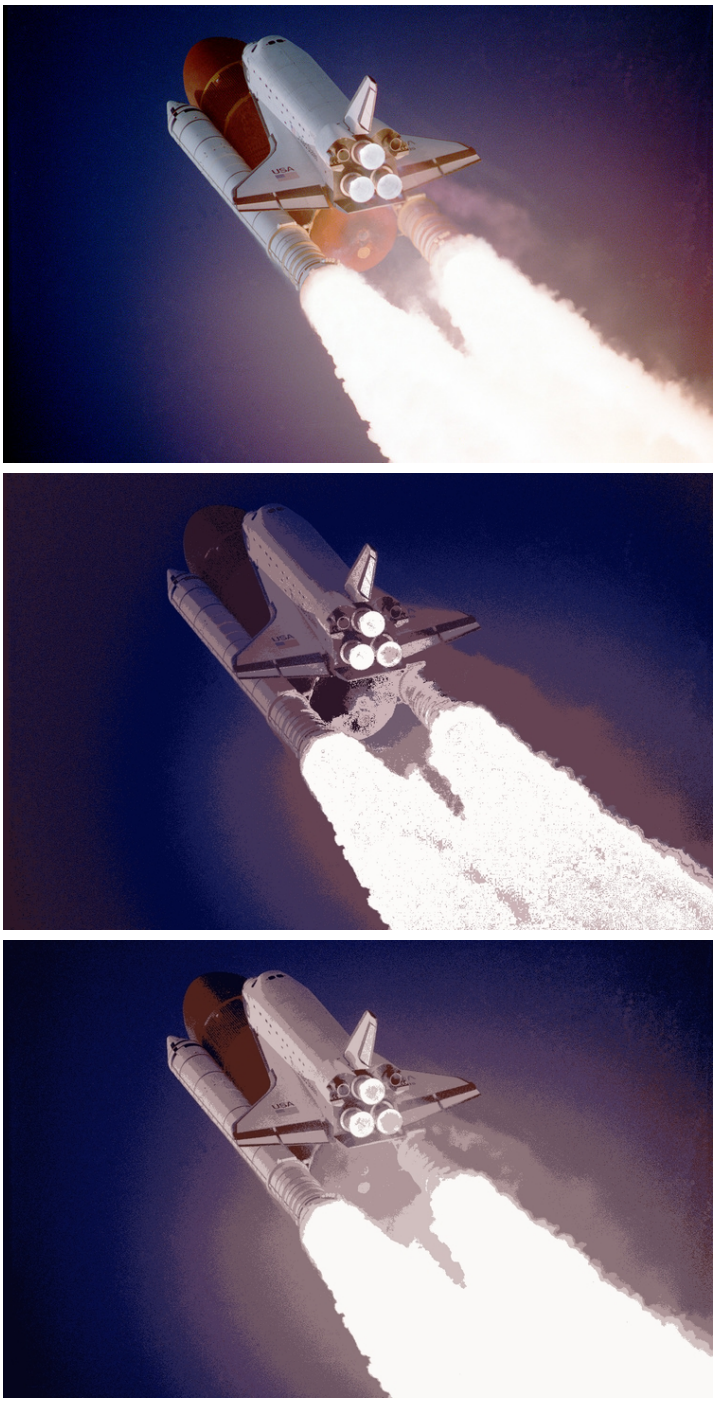

Figure 7: Top: the original 1.4 Mpx JPEG image of the space shuttle Atlantis' climb to orbit during mission STS-27 in December 1988. Middle: the image segmented into 10 colors by the GMCM. Bottom: the image segmented into 10 colors by the k-means algorithm. Image credit: NASA.

GMCM package a 1.4 Mpx (965×1500 px) image of the Space Shuttle Atlantis, seen at the top of Figure 7, was segmented into 10 colors.

The JPEG image can be represented as a $1{,}447{,}500 \times 3$ matrix where each column corresponds to a color channel in the RGB color space and each row corresponds to a pixel and observation in the GMCM. The values are in this case in the interval $[0, 1]$.

A 3-dimensional, 10-component GMCM was fitted using the PEM algorithm which resulted in the middle image of Figure 7. The segmented colors were chosen using the location estimates $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_{10}$. That is, the three dimensional vector $\hat{F}_\circ^{-1}(\Gamma_\circ(\hat{\boldsymbol{\mu}}_h; \boldsymbol{\theta})) \in [0, 1]^3$ in the RGB space was used as the color of cluster h. Alternatively, the average RGB value of each cluster could be used.

For comparison the 1.4 Mpx image was also segmented with the k-means algorithm. The results are seen at the bottom of Figure 7. The final color given to each cluster corresponded to the mean estimated by the algorithm.

As can be seen, the k-means algorithm and the GMCM yield quite different segmentations and different details of the image are captured. For example, the GMCM seems to capture more details of the bottom of the orange external tank. However perhaps erroneously, the GMCM also clusters the black left edge of the photo together with a light cluster. Which method is superior method depends on the application at hand. We acknowledge that disregarding spatial correlations between pixels is quite naïve. However, this example should illustrate the computational capability of the package to handle large datasets with a high number of clusters.

The package **jpeg** was used to read, manipulate, and write the JPEG image from R (Urbanek 2014).

6. Concluding remarks

The software for the gradient descent algorithm used by Tewari *et al.* (2011) to arrive at a maximum likelihood estimate is written in the proprietary language MATLAB (The MathWorks Inc. 2014) and also not provided as open source software. Hybrid procedures, similar to the one proposed by Tewari *et al.* (2011), can easily be constructed with the **GMCM** package. The **GMCM** package solves some of the previously described issues regarding the maximum likelihood estimation and provides a considerable speed-up in computation times. However, there seems to be no complete remedy for all of the challenges of the GMCMs. As stated, the transformation into uniform marginal distributions by ranking will result in a loss of information about the distance between components that are well separated.

The intrinsic identifiability problems of GMCMs may in practice often not be a big issue. Even though the parameters of the assumed underlying GMM can be difficult to estimate due to the flat likelihood function, the clustering accuracy can still be very high. Furthermore, the actual parameters, except perhaps the mixture proportions, do often not seem to be of particular interest in applications. Hence, the merit of the GMCMs should be measured by predictive accuracy which still remains to be explored. In this respect, we believe that the theoretical and practical properties of the special GMCM and IDR approach should be studied further and compared to common p value combining meta-analyses, such as the methods of Fisher, Stouffer, Wilkinson, Pearson, and others, see, e.g., Owen (2009). Interestingly, and perhaps also of slight concern, it can be seen that the IDR approach would be deemed unreasonable by Condition 1 in Birnbaum (1954) whenever $\rho \neq 0$. It is unclear whether the method fulfills properties such as admissibility (Birnbaum 1954) and relative optimality in Bahadur's sense

(Littell and Folks 1971).

The simulation study in Section 3.3 revealed relatively many errors thrown by the **GMCM** package. We are committed to pinpoint the exact sources of the errors and provide fixes in future versions. We suspect the errors encountered are due to divergence of the parameters and should therefore be treated as such. With this in mind we believe that software should fail loudly with an error or a warning when it indeed fails.

In conclusion, the **GMCM** package provides a fast implementation of a flexible and widely applicable tool proposed for reproducibility analysis and unsupervised clustering. The flexibility and applicability are however gained at the cost of a complicated likelihood function.

Acknowledgments

We thank Andreas Petri for his help on the microarray pre-processing workflow. The technical assistance from Alexander Schmitz, Julie S. Bødker, Ann-Maria Jensen, Louise H. Madsen, and Helle Høholt is also greatly appreciated. As are the helpful statistical comments from Steffen Falgreen. This research is supported by MSCNET, EU FP6, CHEPRE, the Danish Agency for Science, Technology, and Innovation as well as Karen Elise Jensen Fonden.

References

Abramowitz M, Stegun I (1970). *Handbook of Mathematical Functions*. Dover Publishing Inc., New York.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, **51**(1), 289–300.

Bergkvist KS, Nyegaard M, Bøgsted M, Schmitz A, Bødker JS, Rasmussen SM, Perez-Andres M, Falgreen S, Bilgrau AE, Kjeldsen MK, *et al.* (2014). "Validation and Implementation of a Method for Microarray Gene Expression Profiling of Minor B-Cell Subpopulations In Man." *BMC Immunology*, **15**(3). doi:10.1186/1471-2172-15-3.

Bernstein BE, Birney E, Dunham I, Green ED, Gunter C, Snyder M (2012). "An Integrated Encyclopedia of DNA Elements in the Human Genome." *Nature*, **489**(7414), 57–74. doi:10.1038/nature11247.

Bilgrau AE, Boegsted M, Eriksen PS (2016). ***GMCM**: Fast Estimation of Gaussian Mixture Copula Models*. R package version 1.2.3, URL https://CRAN.R-project.org/package=GMCM.

Birnbaum A (1954). "Combining Independent Tests of Significance." *Journal of the American Statistical Association*, **49**(267), 559–574.

Chen X, Fan Y, Tsyrennikov V (2006). "Efficient Estimation of Semiparametric Multivariate Copula Models." *Journal of the American Statistical Association*, **101**(475), 1228–1240. doi:10.1198/016214506000000311.

Dai M, Wang P, Boyd AD, Kostov G, Athey B, Jones EG, Bunney WE, Myers RM, Speed TP, Akil H, Watson SJ, Meng F (2005). "Evolving Gene/Transcript Definitions Significantly Alter the Interpretation of GeneChip Data." *Nucleic Acids Research*, **33**(20), e175. doi:10.1093/nar/gni179.

Eddelbuettel D (2013). *Seamless R and C++ Integration with* ***Rcpp***. Springer-Verlag, New York.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Efron B (2004). "Large-Scale Simultaneous Hypothesis Testing: The Choice of a Null Hypothesis." *Journal of the American Statistical Association*, **99**(1), 96–104.

Efron B (2005). "Local False Discovery Rates." *Technical report*, Division of Biostatistics, Stanford University. URL http://statweb.stanford.edu/~ckirby/brad/papers/2005LocalFDR.pdf.

Efron B (2007). "Size, Power and False Discovery Rates." *The Annals of Statistics*, **35**(4), 1351–1377. doi:10.1214/009053606000001460.

Ein-Dor L, Zuk O, Domany E (2006). "Thousands of Samples are Needed to Generate a Robust Gene List for Predicting Outcome in Cancer." *Proceedings of the National Academy of Sciences of the United States of America*, **103**(15), 5923–5928. doi:10.1073/pnas.0601231103.

François R, Eddelbuettel D, Bates D (2016). ***RcppArmadillo*: *Rcpp* Integration for *Armadillo*** *Templated Linear Algebra Library*. R package version 0.6.600.4.0, URL https://CRAN.R-project.org/package=RcppArmadillo.

Gautier L, Cope L, Bolstad BM, Irizarry RA (2004). "**affy** – Analysis of Affymetrix GeneChip Data at the Probe Level." *Bioinformatics*, **20**(3), 307–315. doi:10.1093/bioinformatics/btg405.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). "**Bioconductor**: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology*, **5**, R80. doi:10.1186/gb-2004-5-10-r80.

Harrell, Jr FE (2016). ***Hmisc*: *Harrell Miscellaneous***. R package version 3.17-2, URL https://CRAN.R-project.org/package=Hmisc.

Hastings C, Hayward JT, Wong JP (1955). *Approximations for Digital Computers*, volume 170. Princeton University Press Princeton.

Ioannidis JP, Ntzani EE, Trikalinos TA, Contopoulos-Ioannidis DG (2001). "Replication Validity of Genetic Association Studies." *Nature Genetics*, **29**(3), 306–9. doi:10.1038/ng749.

Kane M, Emerson J, Weston S (2013). "Scalable Strategies for Computing with Massive Data." *Journal of Statistical Software*, **55**(1), 1–19. doi:10.18637/jss.v055.i14.

Küppers R (2005). "Mechanisms of B-Cell Lymphoma Pathogenesis." *Nature Reviews Cancer*, **5**(4), 251–62. doi:10.1038/nrc1589.

Leisch F (2002). "Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), *Compstat 2002 – Proceedings in Computational Statistics*, pp. 575–580. Physica Verlag, Heidelberg.

Lenz G, Staudt LM (2010). "Aggressive Lymphomas." *The New England Journal of Medicine*, **362**(15), 1417–29. doi:10.1056/NEJMra0807082.

Li Q (2014). ***idr**: Irreproducible Discovery Rate*. R package version 1.2, URL https://CRAN.R-project.org/package=idr.

Li Q, Brown JBJB, Huang H, Bickel PJ (2011). "Measuring Reproducibility of High-Throughput Experiments." *The Annals of Applied Statistics*, **5**(3), 1752–1779. doi:10.1214/11-AOAS466.

Littell RC, Folks JL (1971). "Asymptotic Optimality of Fisher's Method of Combining Independent Tests." *Journal of the American Statistical Association*, **66**(336), 802–806.

Nelsen RB (2006). *An Introduction to Copulas*. 2nd edition. Springer-Verlag.

Neuwirth E (2014). ***RColorBrewer**: **ColorBrewer** Palettes*. R package version 1.1-2, URL https://CRAN.R-project.org/package=RColorBrewer.

Owen AB (2009). "Karl Pearson's Meta-Analysis Revisited." *The Annals of Statistics*, **37**(6B), 3867–3892. doi:10.1214/09-AOS697.

Rasmussen SM, Bilgrau AE, Schmitz A, Falgreen S, Bergkvist KS, Tramm AM, Bæch J, Jacobsen CL, Gaihede M, Kjeldsen MK, Bødker JS, Dybkær K, Bøgsted M, Johnsen HE (2014). "Stable Phenotype Of B-Cell Subsets Following Cryopreservation and Thawing of Normal Human Lymphocytes Stored in a Tissue Biobank." *Cytometry Part B: Clinical Cytometry*. doi:10.1002/cytob.21192.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Revolution Analytics (2015). ***doMC**: Foreach Parallel Adaptor for the **multicore** Package*. R package version 1.3.4, URL https://CRAN.R-project.org/package=doMC.

Revolution Analytics, Weston S (2015). ***foreach**: Foreach Looping Construct for R*. R package version 1.4.3, URL https://CRAN.R-project.org/package=foreach.

Rui L, Schmitz R, Ceribelli M, Staudt LM (2011). "Malignant Pirates of the Immune System." *Nature Immunology*, **12**(10), 933–40. doi:10.1038/ni.2094.

Smyth GK (2004). "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology*, **3**(1), 1–25.

Tan PK, Downey TJ, Spitznagel Jr EL (2003). "Evaluation of Gene Expression Measurements from Commercial Microarray Platforms." *Nucleic Acids Research*, **31**(19), 5676–5684. doi:10.1093/nar/gkg763.

Tewari A, Giering MJ, Raghunathan A (2011). "Parametric Characterization of Multimodal Distributions with Non-Gaussian Modes." In *2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, pp. 286–292. IEEE. doi:10.1109/ICDMW.2011.135.

The ENCODE Consortium (2011). "A User's Guide to the Encyclopedia of DNA Elements (ENCODE)." *PLoS Biology*, **9**(4), e1001046. doi:10.1371/journal.pbio.1001046.

The MathWorks Inc (2014). *MATLAB – The Language of Technical Computing, Version R2014b*. Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Urbanek S (2014). ***jpeg**: Read and Write JPEG Images*. R package version 0.1-8, URL https://CRAN.R-project.org/package=jpeg.

Xie Y (2013). *Dynamic Documents with R and **knitr***. CRC Press.

Zhang M, Yao C, Guo Z, Zou J, Zhang L, Xiao H, Wang D, Yang D, Gong X, Zhu J, Li Y, Li X (2008). "Apparently Low Reproducibility of True Differential Expression Discoveries in Microarray Studies." *Bioinformatics*, **24**(18), 2057–63. doi:10.1093/bioinformatics/btn365.

Affiliation:

Anders Ellern Bilgrau
Department of Mathematical Sciences
The Faculty of Engineering and Science
Aalborg University
9220 Aalborg East, Denmark
E-mail: anders.ellern.bilgrau@gmail.com
and
Department of Haematology, Research Laboratory
Faculty of Medicine
Aalborg University Hospital
9000 Aalborg, Denmark

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
April 2016, Volume 70, Issue 2 *Submitted:* 2014-03-07
doi:10.18637/jss.v070.i02 *Accepted:* 2015-03-26