Graphical Representation of Stochastic Clustering

Vahid Partovi Nia and David Stephens

September 20, 2010

Summary

Labels are used to show grouping of subjects and a dendrogram is a tree providing visual guide to data grouping. The `labeltodendro` package links these two concepts and is made to achieve two goals: the first goal is to provide a flexible environment for plotting any arbitrary dendrogram, and the second is to summarise a matrix of integer labels using a dendrogram.

Keywords: Agglomerative clustering, Bayesian clustering, Dendrogram, Hierarchical clustering, Markov chain Monte Carlo, `R`

1 Introduction

This paper explains how the package `labeltodendro` can be used to reconstruct the dendrograms underpinning statistical clustering from label and height information only. Such data arise, for example, when Markov chain Monte Carlo (MCMC) is used to perform Bayesian inference in unsupervised learning problems, finite mixture modelling, or non-parametric modelling using the Dirichlet process. The package can be used to visualize and to summarize such data. We give details of the package functionality, and illustrate its use in a real data example.

Data partitioning or clustering is a fundamental exploratory tool for data analysis. In *hierarchical clustering* the tree (or dendrogram) representation of possible groupings is commonly used. Hierarchical clustering has two variants, agglomerative or divisive, each requiring a dissimilarity measure between a pair of groups. An *agglomerative* method constructs a tree bottom-up: at the very first step every subject is considered to be a singleton and the closest clusters are merged according to the defined dissimilarity measure consecutively until all subjects are in one cluster. This is available in `R` via the `hclust` function. *Divisive* clustering refers to top-down construction of the tree, starting with all subjects in one group, and splitting the clusters sequentially until each subject is in a single

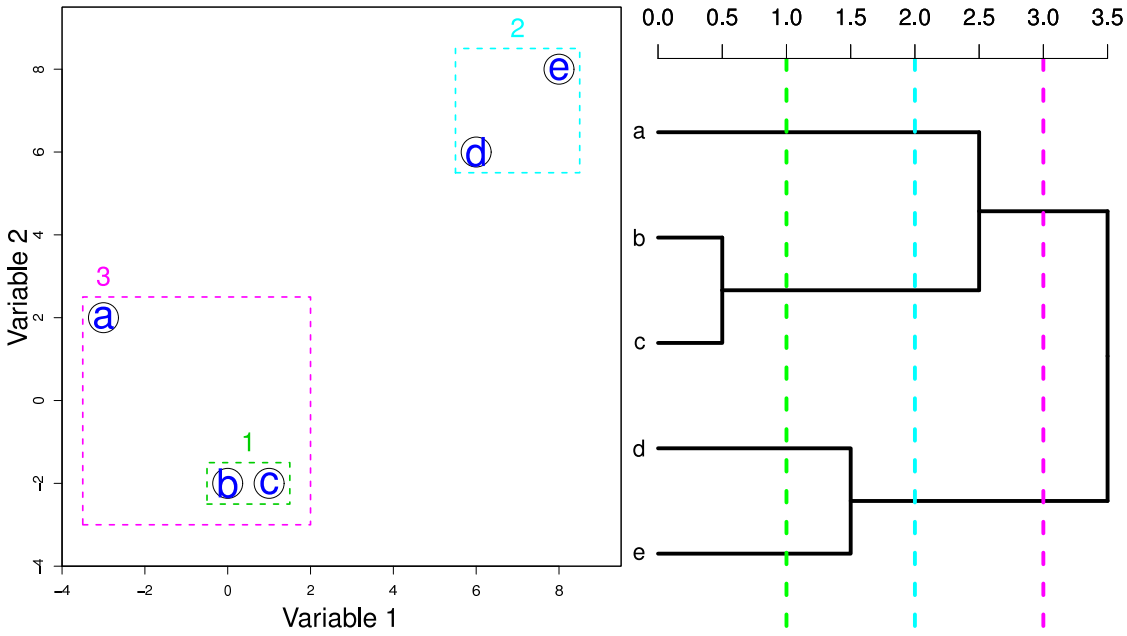


Figure 1: Five two-dimensional individuals (left panel) and its corresponding dendrogram (right panel).

cluster. This technique is available in `R` through the `diana` function in the `cluster` package. The dendrogram represents how individuals group together and the tree representation often is lost when a stochastic search is adopted for finding the optimal grouping in Bayesian mixture modelling approach to clustering (Booth *et al.*, 2008). The `labeltodendro` package makes the dendrogram representation feasible in such cases.

2 Direct Creation of a Dendrogram

The `labeltodendro` package does not implement hierarchical clustering, but helps to create a dendrogram object directly from a set of labels and heights. Suppose one has a dendrogram in mind and wants to draw it without making fake data and defining fake dissimilarity measure, then our package is needed to do so. Such a tool is needed in MCMC-based clustering or other stochastic search clustering techniques.

Consider a simple dendrogram of Figure 1 (right panel) created by two dimensional data points of Figure 1 (left panel). Different groupings of the subjects are defined by cutting the tree at different heights. A grouping is represented by a vector of positive integers (*labels*), say $\mathbf{d}$. Subjects in the same group have the same label, such that the number of clusters is $\max(\mathbf{d})$. For instance cutting the dendrogram of Figure 1 at heights 0, 1, 2, 3, and 4 gives $\mathbf{d} = (1, 2, 3, 4, 5)$, $\mathbf{d} = (1, 2, 2, 3, 4)$, $\mathbf{d} = (1, 2, 2, 3, 3)$, $\mathbf{d} = (1, 1, 1, 2, 2)$, and $\mathbf{d} = (1, 1, 1, 1, 1)$, respectively. A dendrogram can be characterised by the different

groupings that it creates and the heights that the pairs of clusters join at.

label					height
1	1	1	1	1	3.5
1	1	1	2	2	2.5
1	2	2	3	3	1.5
1	2	2	3	4	0.5
1	2	3	4	5	0

Table 1: The label matrix and the height vector that characterises the dendrogram of Figure 1.

Now suppose the data of Figure 1 are not available. By giving the whose labelling $\mathbf{d}$ and the corresponding heights, see Table 1, we require to be able to redraw the dendrogram of Figure 1. This is feasible through the `tabletodendro` function. The `R` code that produces the tree is as follows:

```
> label.mat<-matrix(c(
+ 1,1,1,1,1,
+ 1,1,1,2,2,
+ 1,2,2,3,3,
+ 1,2,2,3,4,
+ 1,2,3,4,5
+ ),ncol=5,byrow=TRUE)
> height.vec<-c(3.5,2.5,1.5,0.5,0)
>  plot(tabletodendro(label.mat,height.vec,
+  labels=c("a","b","c","d","e")))
```

3 Counting and Sorting Labels

In Bayesian clustering, a stochastic search, like Markov chain Monte Carlo (Booth *et al.*, 2008) or the reversible jump algorithm (Green and Richardson, 1997) is applied to sample from the posterior distribution of the data groupings. Therefore, vectors of labels are produced and summary of the sampled labels is of interest. The most frequent labelling is usually reported (Kim *et al.*, 2006), but using our package we can go further and produce a dendrogram based on sampled labels.

The first difficulty in summarising the label samples arises in counting them, because the labels are exchangeable, for instance $\mathbf{d}_1 = (1, 1, 1, 2, 2)$ and $\mathbf{d}_2 = (2, 2, 2, 1, 1)$ impose the same grouping and hence are equivalent. Counting groupings is easily performed after

uniquely relabelling the sampled labels. One way of unique labelling is insisting that the labels appear in increasing order, so individual j can only have one of the labels $1, \ldots, j$, for $j = 1, \ldots, n$, which n is the number of individuals. In other words the first individual always has label 1, the second individual has label 1 if belongs to the same group as individual one, and has label 2 otherwise, and so forth. Therefore, $\mathbf{d}_2$ never occurs after relabelling. The unique relabelling is implemented in the `relabel.matrix` function. The following code shows how this function works.

```
>  label.mat<-matrix(c(
+  3,2,2,1,4,
+  4,2,2,1,3,
+  1,3,3,4,2,
+  4,5,3,1,2
+  ),ncol=5,byrow=TRUE)
>  relabel.matrix(label.mat)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    2    3    4
[2,]    1    2    2    3    4
[3,]    1    2    2    3    4
[4,]    1    2    3    4    5
```

After relabelling, counting groupings is straightforward, and we may store the counted labels in a matrix and their respective frequencies in a vector. However we are still far from dendrogram representation because the space of partitions of a set is partially ordered, so all counted labels cannot be represented by a single dendrogram. We take the most frequent label as a reference partition and search for its agglomerations and divisions in the counted labels to provide a fully ordered matrix of labels. A vector of labels, say $\mathbf{d}_1$, is an agglomeration of another, say $\mathbf{d}_2$, if $\mathbf{d}_2$ defines a refinement of $\mathbf{d}_1$. It is not necessary to define another function to check divisions, swapping the arguments of a function that checks agglomeration, which is available through `is.aggvec` in our package, is enough to do so:

```
> d1<-c(2,2,2,1,1)
> d2<-c(2,4,3,1,1)
> d3<-c(2,2,3,3,1)
> is.aggvec(d2,d1)
[1] FALSE
> is.aggvec(d1,d2)
```

```
[1] TRUE
> is.aggvec(d1,d3)
[1] FALSE
> is.aggvec(d3,d1)
[1] FALSE
```

In R, the package `clue` (Hornik, 2005) can also be used.

Making a matrix of labels sorted in agglomerative order is feasible through the `selectlabels` function which uses `is.aggvec` as an internal function. The following code shows how this function can be used

```
> label.mat<-matrix(c(
+ 1,1,1,1,1, #agglomeration
+ 1,1,1,2,2, #reference partition
+ 1,2,2,2,3, #not agglomeration nor division
+ 1,3,3,3,4, #not agglomeration nor division
+ 1,2,2,3,4, #division
+ 1,2,3,4,5  #division
+ ),ncol=5,byrow=TRUE)
> label.freq<-c(10,60,50,40,30,20)
> selectlabels(label.mat,label.freq)
$labmat
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    1    1
[2,]    1    1    1    2    2
[3,]    1    2    2    3    4
[4,]    1    2    3    4    5

$freq
[1] 10 60 30 20
```

The function `selectlabels` takes the most frequent labelling, defined by the frequency vector, as the reference grouping and searches for its agglomerations and divisions. This usually yields a label matrix with smaller number of rows compared with the original, because often there are a lot of groupings that are not agglomerations nor divisions of the reference partition.

Having a fully ordered labelling matrix provides the possibility of dendrogram representation. However, taking frequencies as the height of dendrogram is inappropriate because

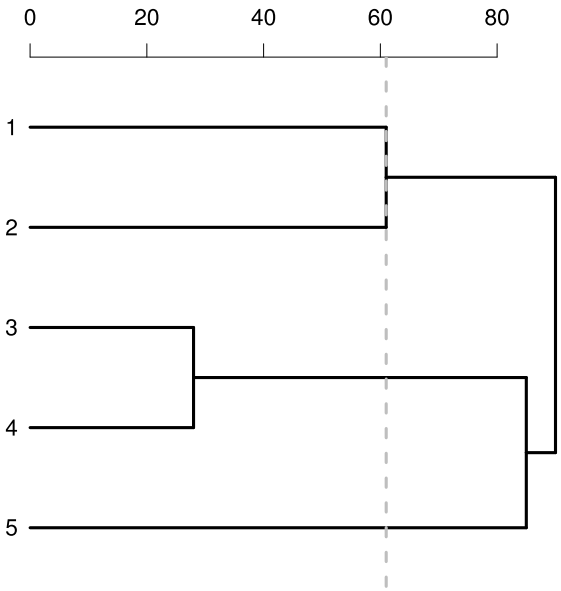
a vector of frequencies is not necessarily monotone. The cumulative absolute increment are associated to the height of the dendrogram yielding a monotone height vector. For instance `freq=(10,60,30,20)` is transformed to `height=(70,60,30,20)`. Therefore the frequency of a grouping appeared in the dendrogram can be obtained by subtracting the height of that grouping from the immediate grouping with the next smallest height.

The maximum frequent partition is lost after making the height monotone. For example `height=(70,60,30,20)` can be generated by the frequency vector `freq=(70,60,30,20)`, or `freq=(10,60,30,20)`, or `freq=(10,30,30,20)`. Hence we define a new `R` class, namely `labclust` which is quite similar to the already defined `R` class `hclust`, but also contains the height that was produced by the maximum frequent grouping. The end result of `tabletodendro` that we already discussed and `labeltodendro` which we will discuss later is *not* a `dendrogram`, but a `labclust` object. Any `labclust` object can be converted to a `dendrogram` object using `as.dendrogram` generic function.

4 Stochastic Clustering

Before introducing Markov chain Monte Carlo clustering, in order to demonstrate the usefulness of our package on a toy example, we consider a trivial random sampling of labels. We generate random grouping of five individuals by randomly sampling with replacement from five integers, namely 1 up to 5. This gives a probability of $\left(\frac{1}{5}\right)^5$ of all individuals being in one cluster. Applying 1000 Monte Carlo replications provides enough samples of groupings that even after relabelling the sampled labels and counting them, it is hard to create a representative dendrogram. The result of the analysis using our package is a `labclust` object, and the information of the most frequent labelling is additionally available and can be used to represent the most frequent grouping on the tree, see below

```
> random.sample<-matrix(sample(1:5000,replace=TRUE)%%5,1000,5)
> labclust.object<-labeltodendro(random.sample)
> plot(labclust.object)
> abline(h=labclust.object$hcut,lty=2,col="gray")
```



The dendrogram above represents the following table of labels and frequencies.

label					frequency	height
1	1	1	1	1	1	79
1	1	1	2	2	6	78
1	2	2	3	3	26	72
1	2	2	3	4	46	46

We also have provided a specific plot function, `colorplot`, which works like `plot` but provides a better visual representation of a `labclust` object. The `colorplot` function provides both horizontal and vertical dendrogram trees. This function colours descending leaves of dendrogram according to the most frequent labelling and is useful to attach to other plot frames like the `image` plot for a better visualisation.

Now suppose that instead of a randomly sampled labels, a Markov chain Monte Carlo algorithm samples the labels. Assume that y is the available data grouped into C clusters, the data in cluster $c, c = 1, \ldots, C$, say y_c, consist of data having the same label in $\mathbf{d}$. Also suppose that y_c is distributed according to the parametric density $f(y_c \mid \theta_c)$.

In Bayesian modelling, we consider that the parameter θ_c itself follows the distribution $f(\theta_c)$ that probably involves some hyper-parameters in itself. Therefore the marginal density for data in cluster c is

$$f(y \mid \mathbf{d}) = \int \prod_{c=1}^{C} f(y_c \mid \theta_c) f(\theta_c) d\theta_c. \tag{1}$$

With clustering prior $f(\mathbf{d})$, the posterior distribution is the natural measure imposed by the model to compare groupings. The higher the posterior, the more appropriate the grouping will be; we have

$$f(\mathbf{d} \mid y) = k^{-1} f(y \mid \mathbf{d}) f(\mathbf{d}), \tag{2}$$

in which $k > 0$ is the normalising constant and plays no role in the analysis.

In Bayesian clustering, one difficult task is to find the label vector $\mathbf{d}$ that maximises the posterior $f(\mathbf{d} \mid y)$. Heller and Ghahramani (2005) suggest to take an agglomerative path and Chen *et al.* (2006) suggest sampling from the posterior distribution $f(\mathbf{d} \mid y)$ using Gibbs sampling. An agglomerative construction provides a useful visual tool to recognise other possible groupings via the dendrogram, but the agglomerative paths may give a poor approximation to the posterior mode. On the contrary the Gibbs sampler gives no visualisation and often suffers from the poor mixing especially for large number of clustering subjects.

Reversible Gibbs sampling is not difficult to implement when full conditional densities are available and a reversible Markov chain satisfies better theoretical properties and often a better mixing. In the reversible Gibbs sampling an individual, say i, is randomly chosen and its label, say d_i, is the ith element of $\mathbf{d}$. Suppose that the vector $\mathbf{d}$ at the current step imposes C clusters on data. If the label represents a singleton cluster, d_i is altered from 1 up to C, and otherwise up to $C + 1$. One of the resulting C or $C + 1$ label vectors are sampled according to their posterior probability up to a normalising constant $f(y \mid \mathbf{d}) f(\mathbf{d})$ and the sampled label vector will be used for the next Gibbs sampling iteration.

One may propose a combined sampler to widen the search space and solve the mixing problem of the Gibbs sample i.e. a Gibbs sampler combined with split-merge moves. In the split step a cluster is chosen at random and is splitted randomly into two clusters. This gives a Gibbs move if one of the two clusters is a singleton. The merge step consists of choosing two clusters at random and merging them together. At last one of all created label vectors created by the split-merge moves are sampled according to their Metropolis-Hastings acceptance probability being a function of $f(y \mid \mathbf{d}) f(\mathbf{d})$ (Booth *et al.*, 2008; Green and Richardson, 1997; Jain and Neal, 2004).

Adding the split-merge moves provides a better mixing of the Gibbs sampler because often some refinements or agglomerations of the maximum *a posteriori* (MAP) grouping take a considerable posterior probability, but are hardly ever proposed by the Gibbs sampler - see for example Table 2.

For a large number of Markov chain Monte Carlo iterations, the most frequent sampled grouping is an approximation to the MAP partition. Furthermore, the ratio of frequencies for two partitions converges to the posterior ratio for these two partitions as the number of iteration increases. Therefore, taking the frequencies to build the arm of the dendrogram

Gibbs Moves	Split Moves	Merge Moves
(1,1,1,1,2,2,2,2,3,3,3)	(1,1,2,2,3,3,3,3,4,4,4)	(1,1,1,1,1,1,1,1,2,2,2)
(2,1,1,1,2,2,2,2,3,3,3)	(1,2,1,2,3,3,3,3,4,4,4)	(1,1,1,1,2,2,2,2,1,1,1)
(3,1,1,1,2,2,2,2,3,3,3)	(1,2,2,1,3,3,3,3,4,4,4)	(1,1,1,1,2,2,2,2,2,2,2)
(4,1,1,1,2,2,2,2,3,3,3)	(2,1,1,1,2,2,3,3,4,4,4)	
⋮	⋮	
(1,1,1,1,4,2,2,2,3,3,3)	(1,1,1,1,2,3,3,2,4,4,4)	
⋮	⋮	
(1,1,1,1,2,2,2,2,3,3,4)		

Table 2: The Gibbs moves compared with the split-merge moves for data in an iteration that 10 observations clustered into three groups, two groups including four observations and one containing three observations, that is $\mathbf{d} = (1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3)$.

gives dendrogram heights approximately proportional to the posterior probabilities for the represented groupings.

For Bayesian models with analytically tractable posteriors $f(\mathbf{d} \mid y)$, such as Heard *et al.* (2006) or conjugate Dirichlet mixtures (Jain and Neal, 2004; Heller and Ghahramani, 2005), it is possible to draw a full dendrogram. One may build the arms of dendrogram using the (log) posterior probability rather than using the frequency of sampled partitions. However, the MAP grouping derived by taking agglomerative or divisive paths is poor and a better solution usually is derived when an MCMC algorithm is applied. However, the dendrogram representation is lost when MCMC algorithms is applied, and the `labelodendro` package may be able to retrieve a part of the lost dendrogram.

Sometimes it is difficult to take the agglomerative path because there are many Bayesian models that the marginal posterior (1) is difficult to handle analytically or even sometimes is intractable (Jain and Neal, 2004; Kim *et al.*, 2006; Tadesse *et al.*, 2005). Therefore, the marginal posterior (2) cannot be analytically calculated, but still it is possible to sample from the full posterior. The Metropolis-Hasting algorithm (sometimes combined with a Gibbs sampler) is used to sample from the full posterior

$$f(\mathbf{d}, \theta_1, \cdots, \theta_C \mid y) \propto f(\mathbf{d}) \prod_{c=1}^{C} f(y_c \mid \theta_c) f(\theta_c) \tag{3}$$

and then the sampled parameters $\theta_1, \cdots, \theta_C$ are omitted, yielding samples from the marginal $f(\mathbf{d} \mid y)$. After the MCMC sampling, the `labeltodendro` package can be applied to visualise the samples by a dendrogram. All possible agglomerations and divisions of the most frequent grouping may not appear in the sampled labels, especially for moderate number of clustering individuals, therefore a full dendrogram in such cases is often unachievable,

although possibly a partial dendrogram can be produced.

5 Example

We demonstrate application of the package on the spike-and-slab Bayesian clustering of Partovi Nia and Davison (2010) using the metabolite data of Messerli *et al.* (2007) consisting of 14 plants measured over 43 metabolites, all but one having four replicates. The data are available in the R package `bclust`. There are two mutants defective in starch biosynthesis, *pgm* and *isa2*; four defective in starch degradation, *sex1*, *sex4*, *mex1*, and *dpe2*; a mutant that accumulates starch as a pleiotropic effect, *tpt*; four unknown mutants, *deg172*, *deg263*, *ke103* and *sex3*; and three wildtype plants, *WsWT*, *RLDWT*, and *ColWT*. The idea was to find out what avenues should be explored when seeking to characterise them using their metabolite profiles.

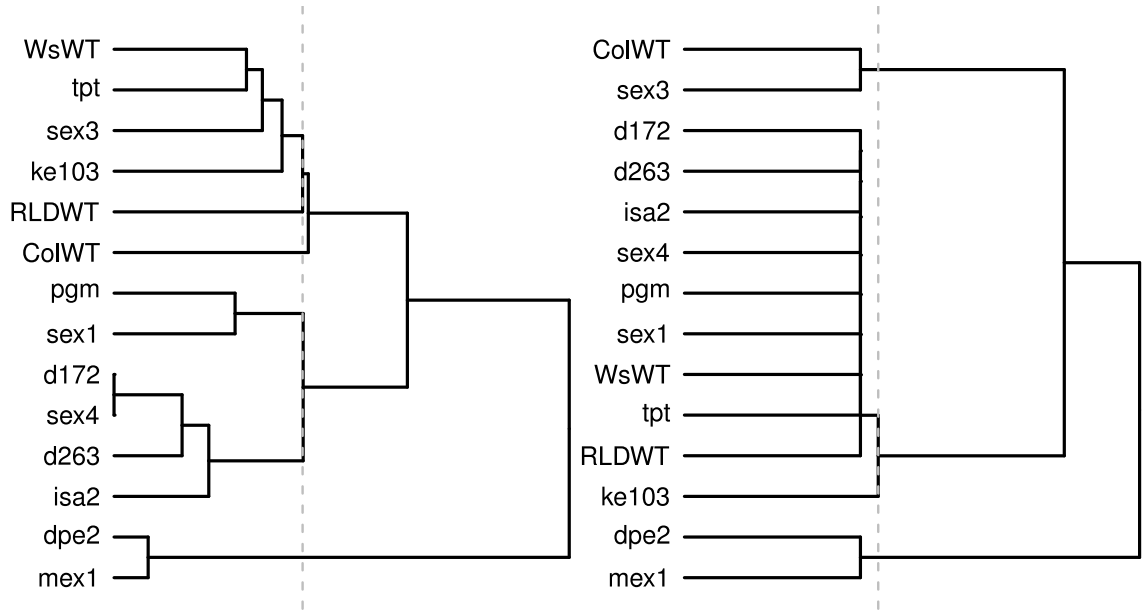


Figure 2: Dendrograms of the metabolite data using agglomerative clustering (left panel) and reversible Gibbs sampler with 1000 iterations (right panel). The vertical dashed gray line shows the MAP grouping approximation.

The Guassian mixture model proposed in Partovi Nia and Davison (2010) allows clustering of replicated data and has tractable marginal posteriors. They proposed using the agglomerative path to approximate the posterior mode and simultaneously gain dendrogram representation, see Figure 2 (left panel).

Figure 2 (right panel) shows the dendrogram extracted using the labeltodendro package from a reversible Gibbs sampler after 1000 iterations with the optimal agglomerative

grouping as the initial starting point.

This Gibbs sampler improves the the MAP grouping 16.7 on the log posterior scale. Since the heights of the labellings i.e. the log marginal posteriors are known, these heights instead of counts are used as the arms of dendrogram.

Acknowledgement

This work is supported by the Swiss National Science Foundation through the prospective researchers fellowship no PBELP2-125531, and also funded in the context of the Swiss National Centre for Competence in Research in Plant Survival (`www.unine.ch/nccr`). We particularly thank Anthony C. Davison and Arpit Chaudhary.

References

Booth, J. G., Casella, G. and Hobert, J. P. (2008) Clustering using objective functions and stochastic search. *Journal of the Royal Statistical Society, Series B* **70**, 119–139.

Chen, T., Morris, J. and Martin, E. (2006) Probability density estimation via an infinite Gaussian mixture model: application to statistical process monitoring. *Applied Statistics* **55**, 699–715.

Green, P. and Richardson, S. (1997) On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, Series B* **59**, 731–792.

Heard, N. A., Holmes, C. C. and Stephens, D. A. (2006) A quantitative study of gene regulation involved in the immune response of *Anopheline* mosquitoes: An application of Bayesian hierarchical clustering of curves. *Journal of the American Statistical Association* **101**, 18–29.

Heller, K. A. and Ghahramani, Z. (2005) Bayesian hierarchical clustering. In *Twenty-second International Conference on Machine Learning*.

Hornik, K. (2005) A CLUE for CLUster Ensembles. *Journal of Statistical Software* **14**, 12.

Jain, S. and Neal, R. M. (2004) A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics* **13**, 158–82.

Kim, S., Tadesse, M. G. and Vannucci, M. (2006) Variable selection in clustering via Dirichlet process mixture models. *Biometrika* **93**, 877–893.

Messerli, G., Partovi Nia, V., Trevisan, M., Kolbe, A., Schauer, N., Geigenberger, P., Chen, J., Davison, A. C., Fernie, A. R. and Zeeman, S. C. (2007) Rapid classification of phenotypic mutants of Arabidopsis via metabolite fingerprinting. *Plant Physiology* **143**, 1481–1492.

Partovi Nia, V. and Davison, A. C. (2010) High-dimensional Bayesian clustering with variable selection: The R package bclust. Submitted.

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