Consensus inference - Simulation design

Stephen Coleman

11/03/2020

Introdcution

We wish to design a number of simulations defined by different generating models. These are intended to showcase how useful consensus inference will be in real life. This means the data should be *realistic*. Each generating model will be a mixture of Gaussian distributions. Consider the marginal likelihood:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k p(x_i | z_i = k, \mu_k, \Sigma_k).$$

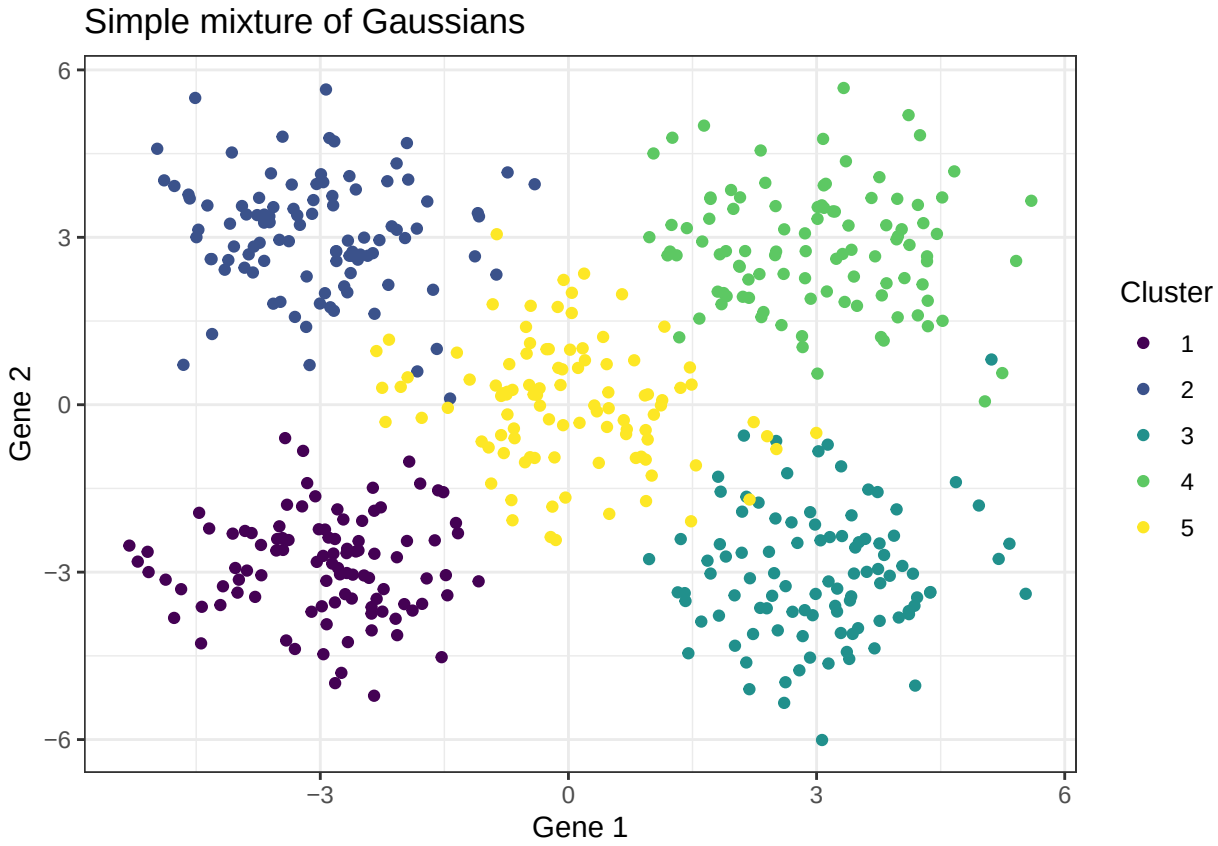
One can see that to generate the data from the model, 6 parameters must be considered:

- n: the number of samples;
- p: the number of features (possibly divided into p_s features containing signal and p_n containing only noise);
- K: the true number of clusters present;
- μ_k: the mean vector defining the k^{th} distribution;
- Σ_k: the covariance matrix associated with the k^{th} distribution; and
- π_k: the proportion of samples to be generated from the k^{th} distribution.

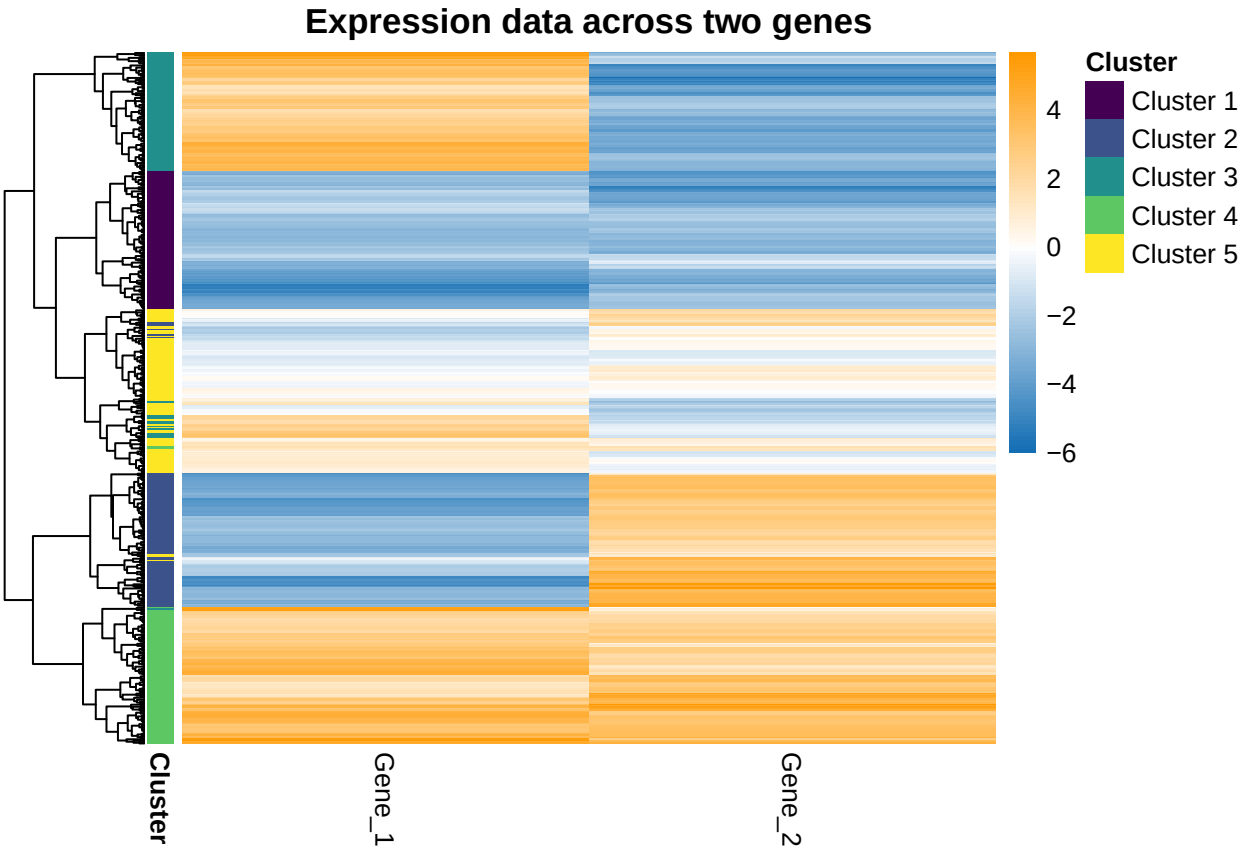
We are interested in a range of scenarios. Some should describe:

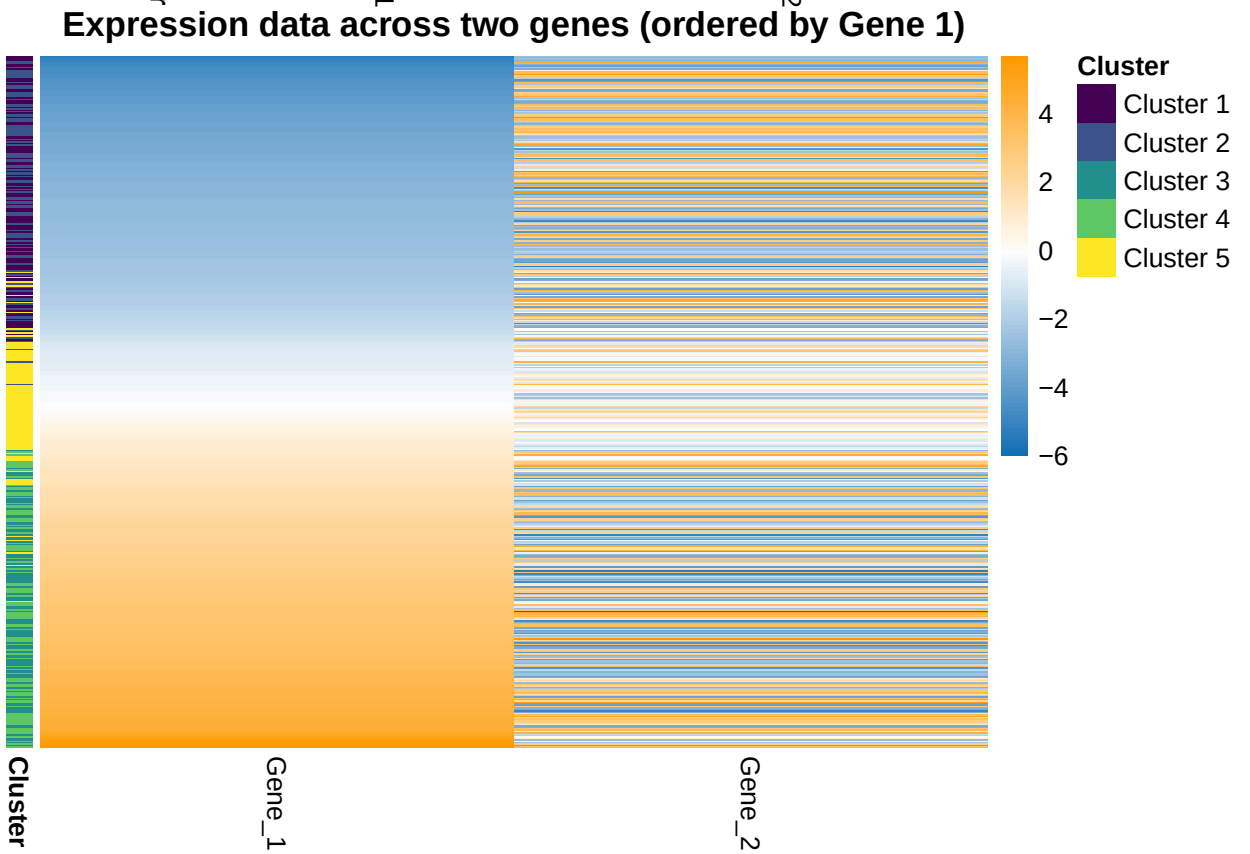
- Small n, large p (typical of 'omics data);
- Large n, small p (such as in flow cytometry);
- Varying π such that cluster sizes are quite varied (with some being very small compared to n).

We will normally hold the number of features containing signal at a relatively small portion of columns in the dataset (say $\max(0.2 \times p, \min(p, 100))$). A simple case of interest is 2D Gaussian data:



Inspecting a heatmap shows that each column is needed to define the clustering - a single feature is insufficient to find any clustering structure.

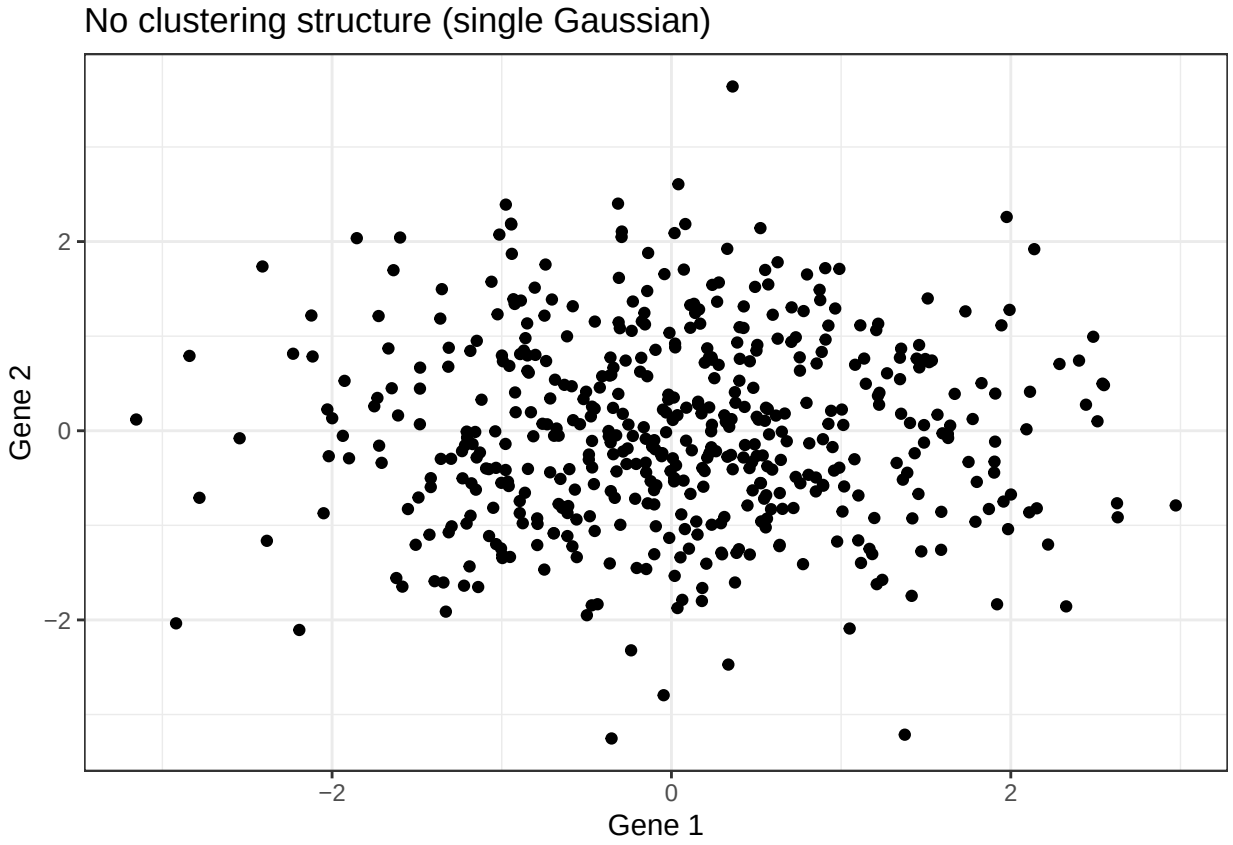

Expression data across two genes
Cluster
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
4
2
0
−2
−4
−6
Cluster
Gene_1
Gene_2

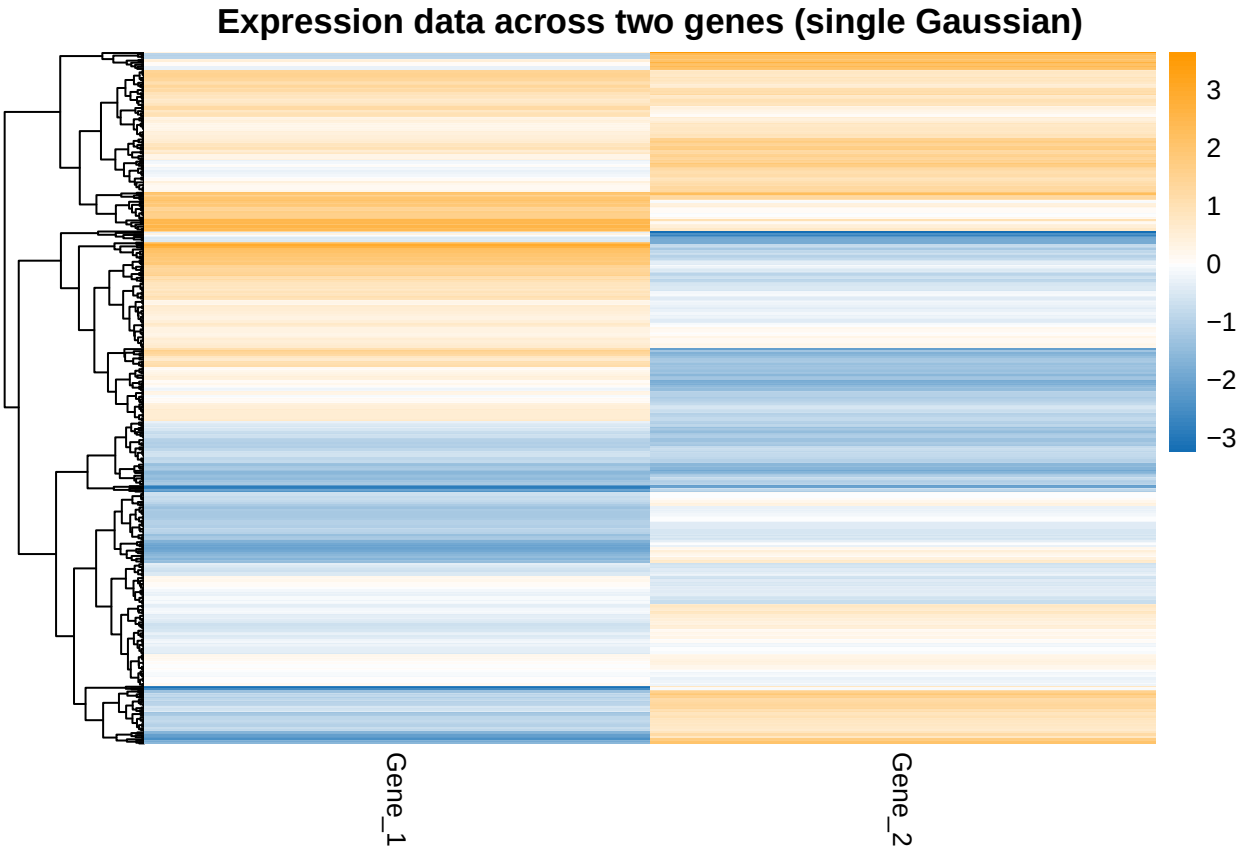

Expression data across two genes (ordered by Gene 1)
Cluster
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
4
2
0
−2
−4
−6
Cluster
Gene_1
Gene_2

This is characteristic of many cases we are interested in; within a single feature there might be no clear clustering structure, but across a combination of a subset of features (possibly all features) there emerges clear clustering structure.

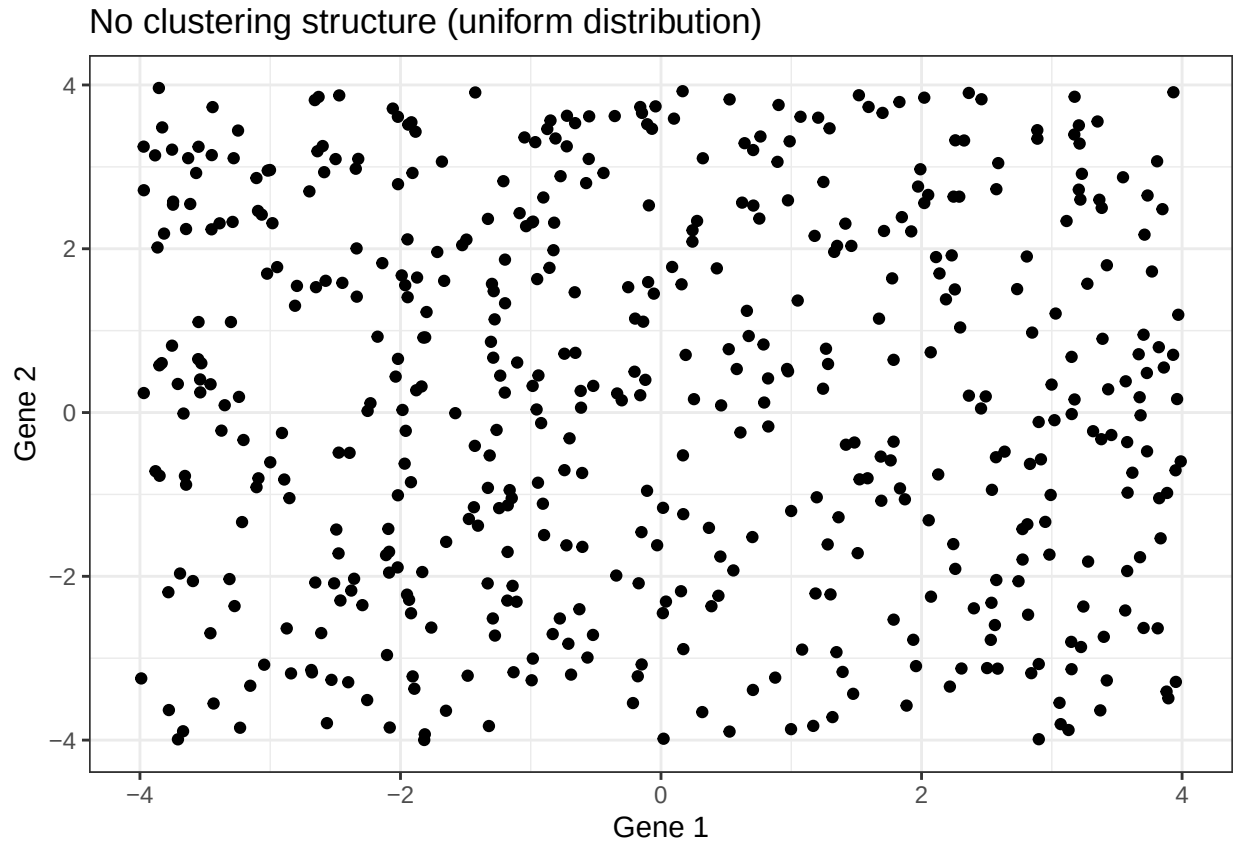
Another case we would be interested in is the case where there is no clustering structure (i.e. all points are drawn from the same distirbution).

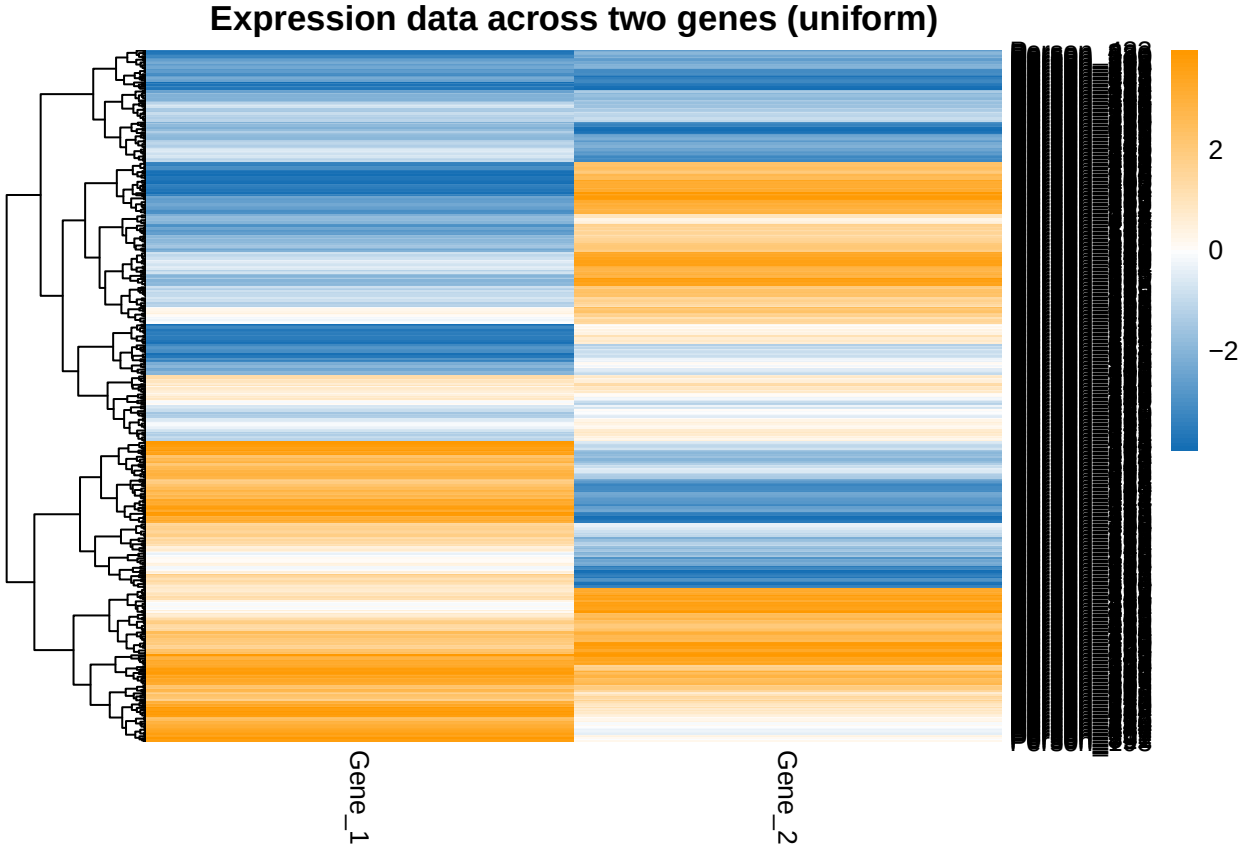
In the first case we consider a 2D Gaussian distribution. In this case by combining many chains we hope that the final consensus matrix consists of a single cluster.



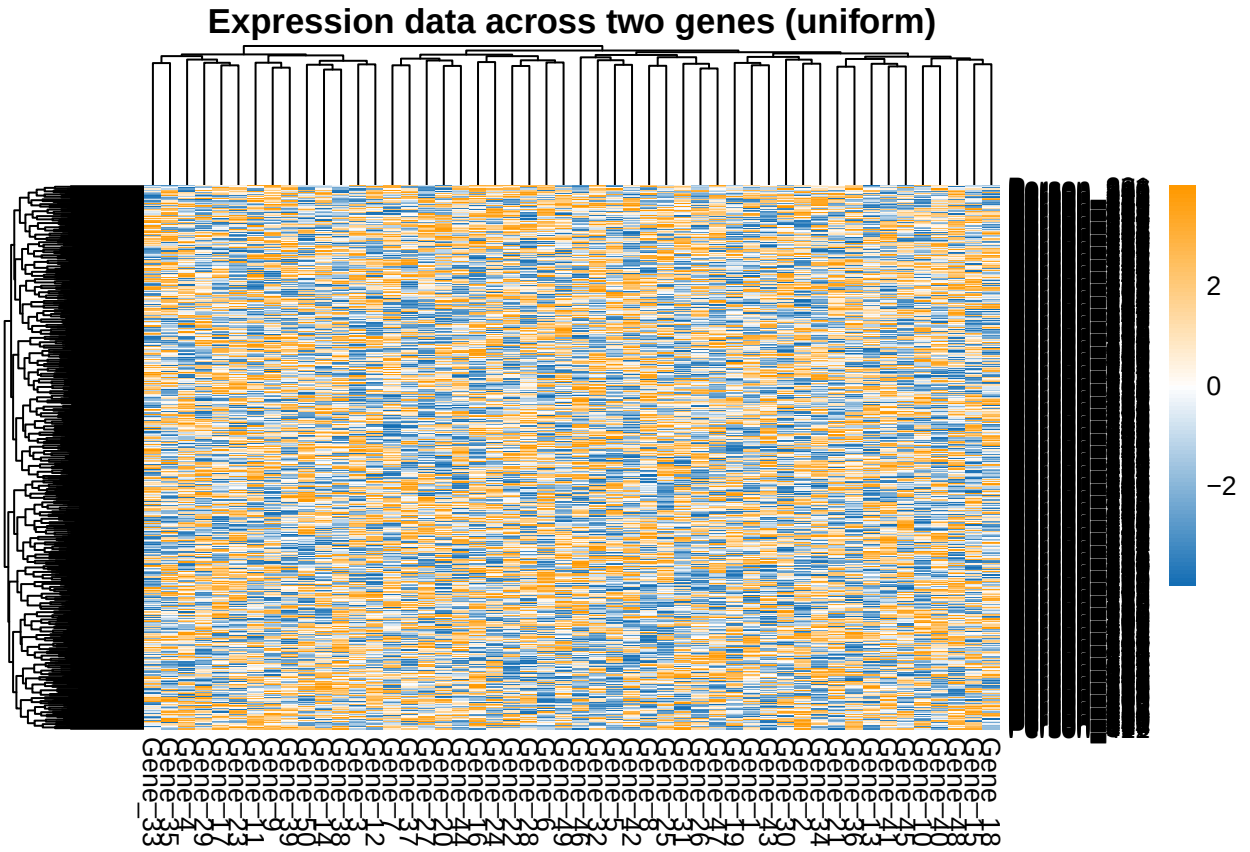


Secondly we consider data generated from a uniform distribution.


No clustering structure (uniform distribution)
Gene 2
Gene 1
4
2
0
−2
−4
−4
−2
0
2
4


Expression data across two genes (uniform)
Gene_1
Gene_2
2
0
−2

In these scenarios, it becomes more and more obvious that there is no clustering structure as p grows:



Thus if a consensus inference approach can recognise the lack of structure in the 2D case then it is likely to be able to do so in higher dimensions. This means that for $p = 2$ we have the following scenarios:

n	p	Distribution	K	pi
500	2	Gaussian	5	vec(1 / K)
500	2	Gaussian	1	vec(1 / K)
500	2	Uniform	1	vec(1 / K)

For the Gaussian cases we use 5 different mean vectors:

$$\mu_1 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \mu_2 = \begin{bmatrix} -3 \\ 3 \end{bmatrix}, \mu_3 = \begin{bmatrix} 3 \\ -3 \end{bmatrix}, \mu_4 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \mu_5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and a common diagonal covariance matrix

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

to define our 5 distributions we draw from.

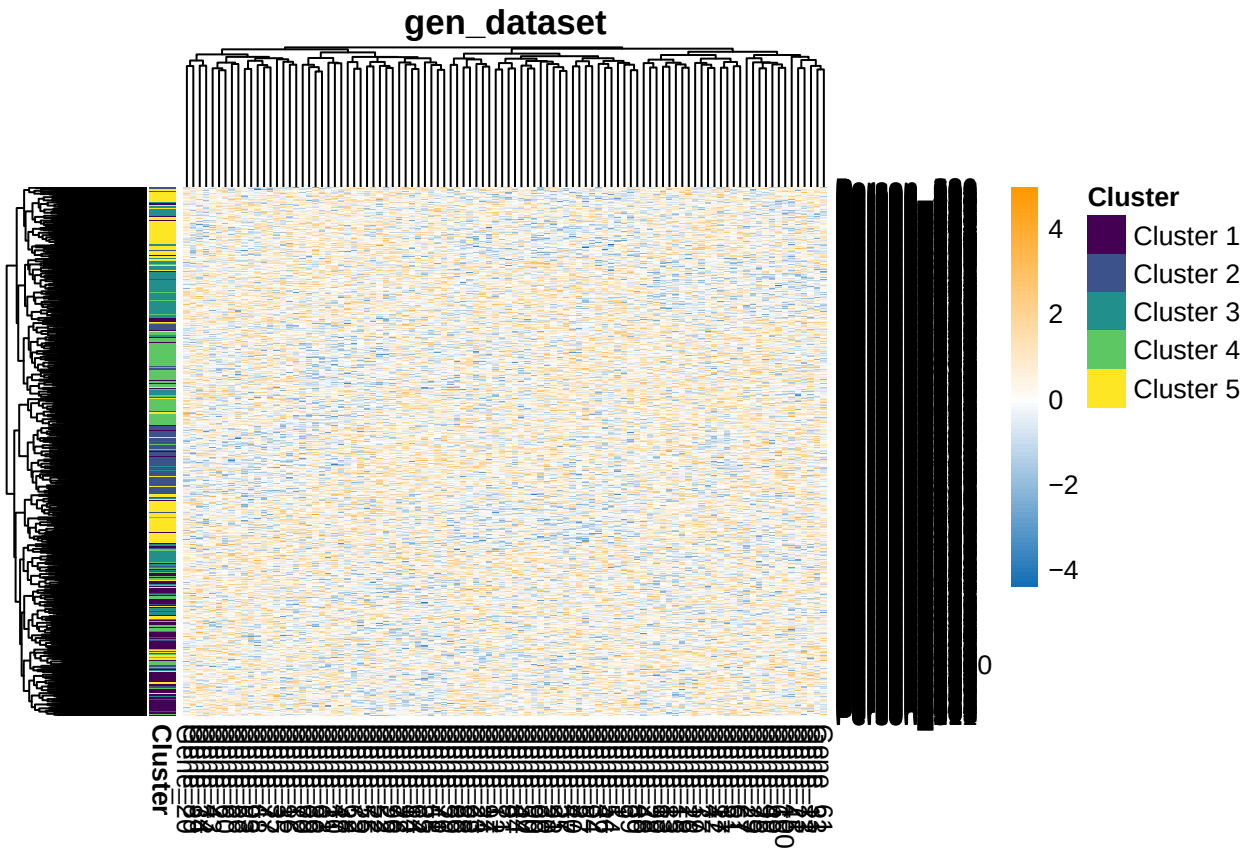
Number	n	p	K	pi	mu	sigma
1	100	30	3	constant	1:K	I
2	100	30	5	constant	1:K	I
3	100	500	3	constant	rnorm(K)	I
4	100	500	5	constant	rnorm(K)	I

Number	n	p	K	pi	mu	sigma
5	100	30	3	varying	1:K	I
6	100	30	5	varying	1:K	I
7	100	500	3	varying	rnorm(K)	I
8	100	500	5	varying	rnorm(K)	I
9	1000	30	5	constant	1:K	I
10	1000	30	7	constant	1:K	I
11	1000	500	5	constant	rnorm(K)	I
12	1000	500	7	constant	rnorm(K)	I
13	1000	30	5	varying	1:K	I
14	1000	30	7	varying	1:K	I
15	1000	500	5	varying	rnorm(K)	I
16	1000	500	7	varying	rnorm(K)	I

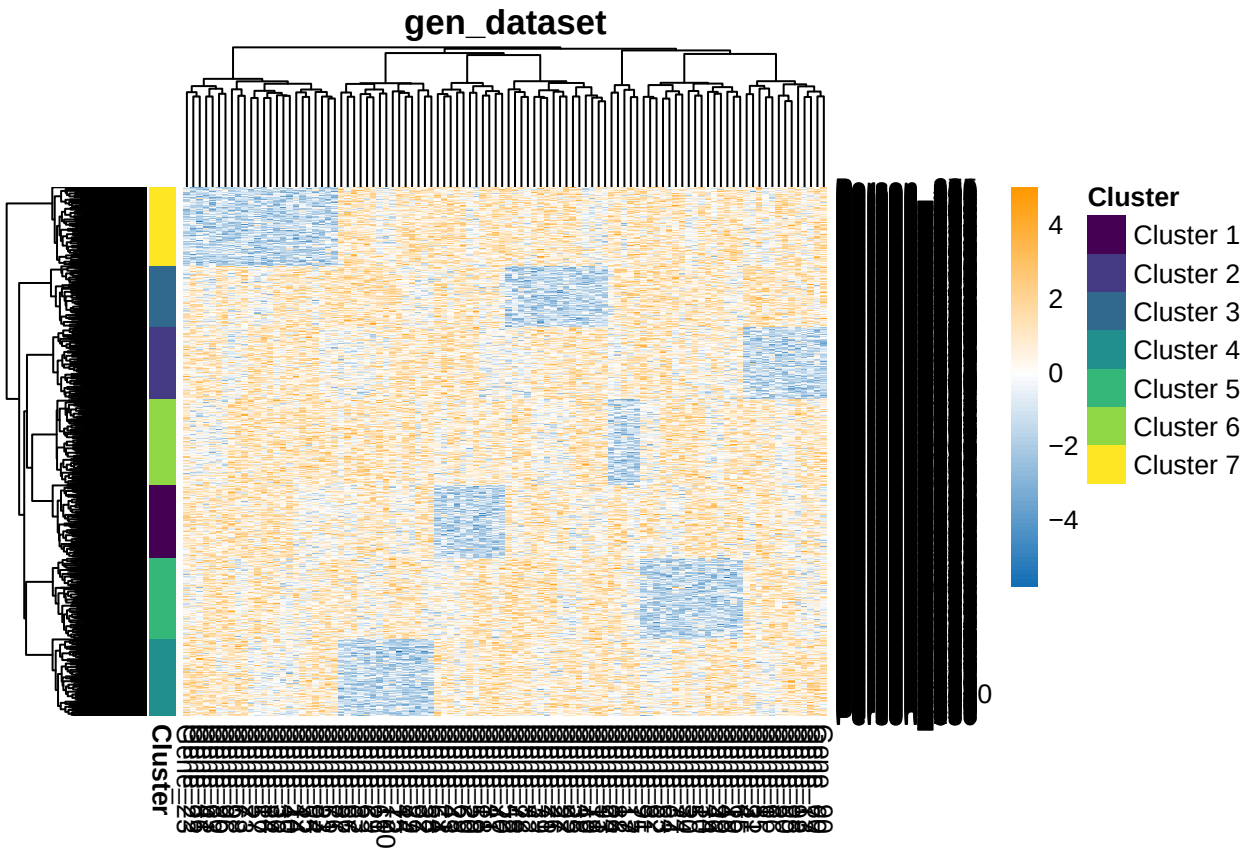
Simulations

We wish to generate data.

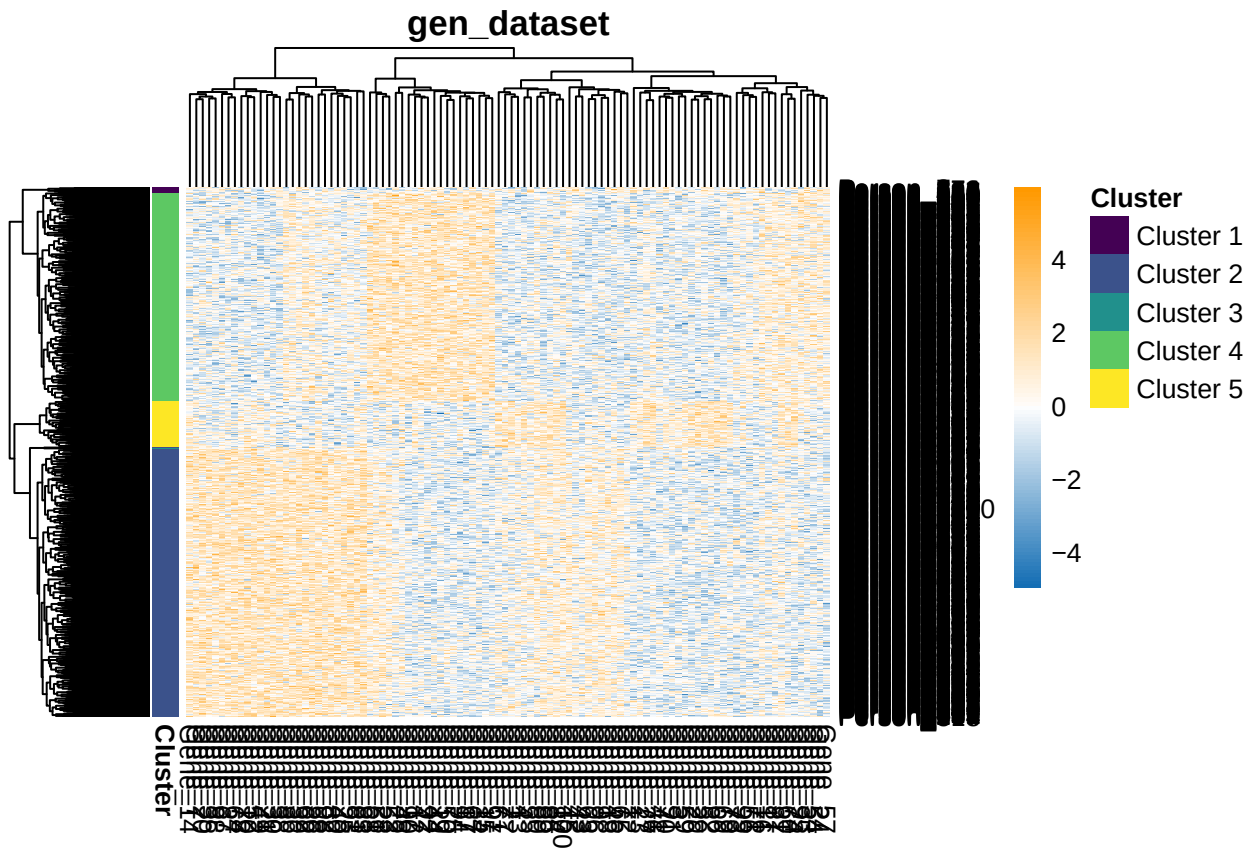
```
data_lst <- genDataFromTable(my_table)

# my_data <- generateDataset(cluster_means, n, p, pi)
# plotData(my_data$data, my_data$cluster_IDs, main = "Simple dataset")

plotData(data_lst[[11]]$data[, 1:100], data_lst[[11]]$cluster_IDs, cluster_rows = T, cluster_cols = T)
```



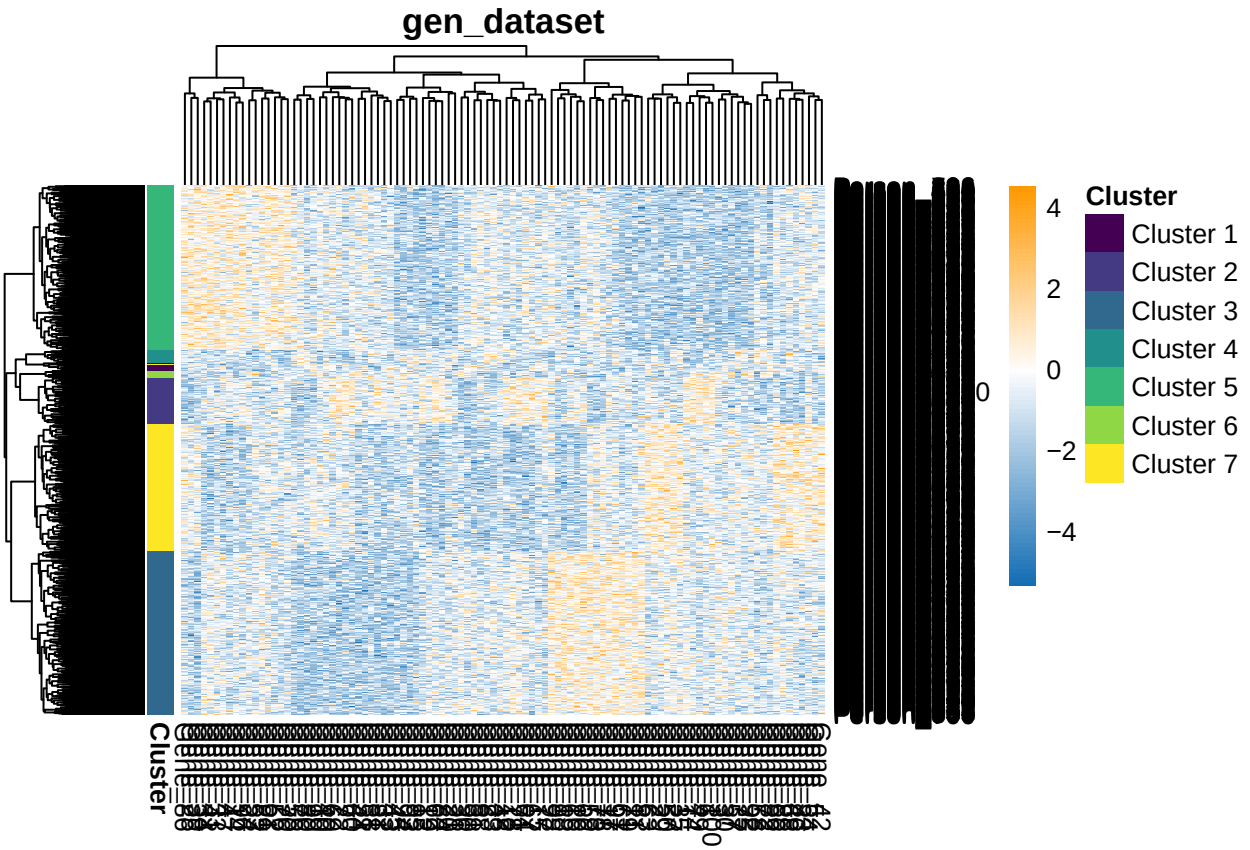
```
plotData(data_lst[[12]]$data[, 1:100], data_lst[[12]]$cluster_IDs, cluster_rows = T, cluster_cols = T)
```




```
plotData(data_lst[[15]]$data[,1:100], data_lst[[15]]$cluster_IDs, cluster_rows = T, cluster_cols = T)
```




```
plotData(data_lst[[16]]$data[,1:100], data_lst[[16]]$cluster_IDs, cluster_rows = T, cluster_cols = T)
```


gen_dataset
Cluster
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
4
2
0
−2
−4
0
0