clusterBMA: Combine insights from multiple clustering algorithms with Bayesian model averaging

Owen Forbes

Distinguished Professor Kerrie Mengersen
Dr Edgar Santos-Fernandez
Dr Paul Wu

Queensland University of Technology

Which clustering algorithm?

k-means, K = 5

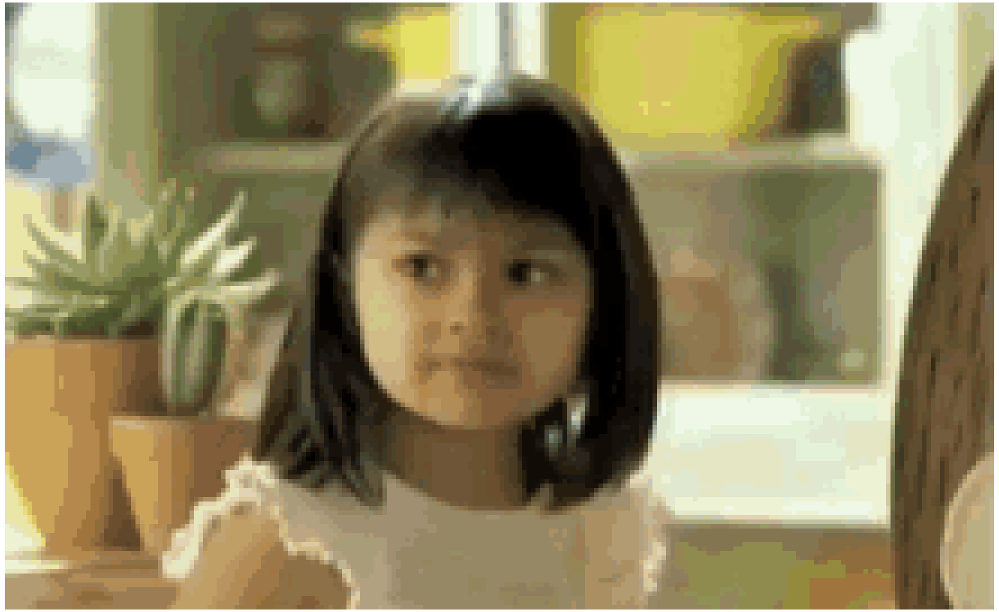
Hierarchical clustering (Ward), K = 5

Gaussian mixture model, K = 5

Inconsistent clustering across algorithms

Which clustering algorithm?

- Different algorithms will emphasise different aspects of clustering structure
- Choosing one 'best' model often arbitrary, unclear choice
  - ➡️ **Inference not calibrated for model-based uncertainty**
- Locking into one method loses insights offered by other methods about plausible clustering structure



BMA offers a nice framework for combining clustering solutions

- simple
- flexible
- intuitive

Combining Clustering Results with Bayesian Model Averaging

- Limited development for Clustering
    - Finite mixture models (Russell et al., 2015)
    - Naive Bayes classifiers (Santafe & Lozano, 2006)
    - Lacks implementation across multiple clustering algorithms

Advantages

- 🆒 Weighted averaging of results incorporating model quality / goodness of fit

- 💯 Intuitive framework for **probabilistic** inferences combining results from **different clustering algorithms**

- 😮 Each input solution can have a different number of clusters K

- 😎 **Quantify model-based uncertainty and enable more robust inferences** calibrated accordingly

Bayesian Model Averaging: Basics

$$P(\Delta|Y) = \sum_{l=1}^{L} (\Delta|Y, M_l) P(M_l|Y)$$

$$P(M_l|Y) = \frac{P(Y|M_l)P(M_l)}{\sum_{l=1}^{L} P(Y|M_l)P(M_l)}$$

BMA for Mixture Models - BIC weighting

- $P(Y|M_l)$ typically involves a difficult/intractable integral and is often approximated for many applications (Fragoso et al., 2018)
- **Russell et al. (2015)** weight results from multiple GMMs according to BIC

$$P(M_l|Y) \approx \frac{exp(\frac{1}{2}BIC_l)}{\sum_{l=1}^{L} exp(\frac{1}{2}BIC_l)}$$

- BIC definition for GMM

$$BIC_l = 2\log(\mathcal{L}) - \kappa_m \log(N)$$

- GMM likelihood

$$\mathcal{L}(\Theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

- Multivariate Gaussian density

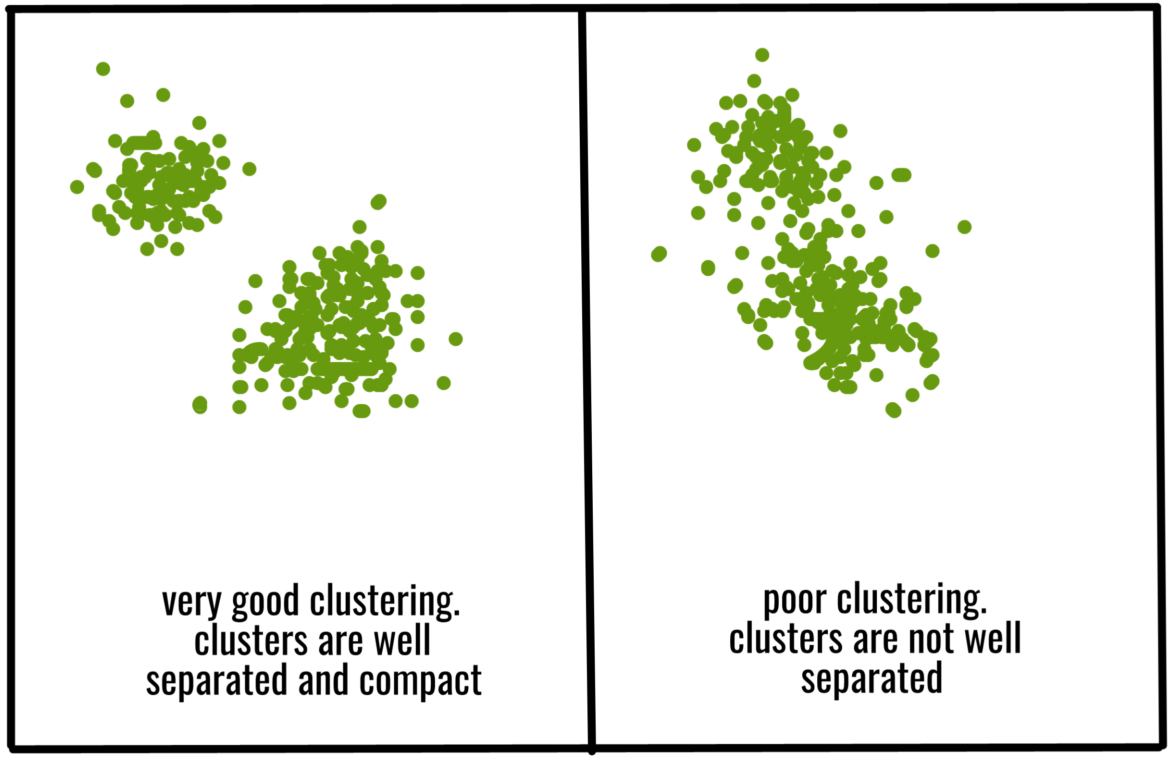
$$\mathcal{N}(x|\mu, \Sigma) = \frac{\exp\{-\frac{1}{2}(y-\mu)^T \Sigma^{-1}(y-\mu)\}}{|\Sigma|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}}$$

Aside: Cluster internal validation indices

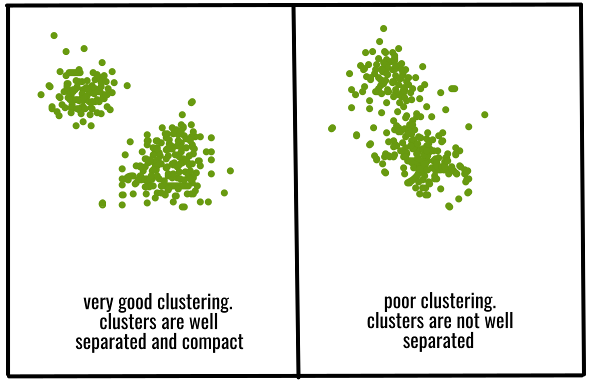
- Often used as a proxy for model quality in clustering
- Choose between candidate models with different numbers of clusters k
- Interpreted similarly to marginal likelihood/model evidence
- Typically measure compactness and/or separation of clusters

Compared to BIC...

- Agnostic to clustering algorithm
- Typically do not require likelihood term



New proposed weighting / approximation for posterior model probability



BIC for GMM driven by **Multivariate Gaussian density:**

$$\mathcal{N}(x|\mu,\Sigma) = \frac{\exp\{-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\}}{|\Sigma|^{\frac{1}{2}}(2\pi)^{\frac{D}{2}}}$$

Xie-Beni index

- Ratio of compactness to separation (maximise)

$$XB = \frac{\sum_i \sum_{x\in C_i} d^2(x,c_i)}{n(\min_{i,j\neq i} d^2(c_i,c_j))}$$

Calinski-Harabasz Index

- Ratio of separation to compactness (minimise)

$$CH = \frac{\sum_i n_i d^2(c_i,c)/(NC-1)}{\sum_i \sum_{x\in C_i} d^2(x,c_i)/(n-NC)}$$

- XB and CH have **complementary strengths** (Liu et al., 2010)

New proposed weighting / approximation for posterior model probability

XB and CH indices

- conceptually and mathematically similar to BIC
- Unlike BIC, can be calculated + directly compared across different clustering algorithms

New proposed weight:

$$\mathcal{W}_m = \frac{\frac{1}{CH_m}}{\sum_{m=1}^{M} \frac{1}{CH_m}} + \frac{XB_m}{\sum_{m=1}^{M} XB_m}$$

Approximate posterior model probability for weighted averaging:

$$P(Y|\mathcal{M}_m) \approx \hat{\mathcal{W}}_m = \frac{\mathcal{W}_m}{\sum_{m'=1}^{M} \mathcal{W}_{m'}}$$

Consistent quantity Δ - Similarity matrices

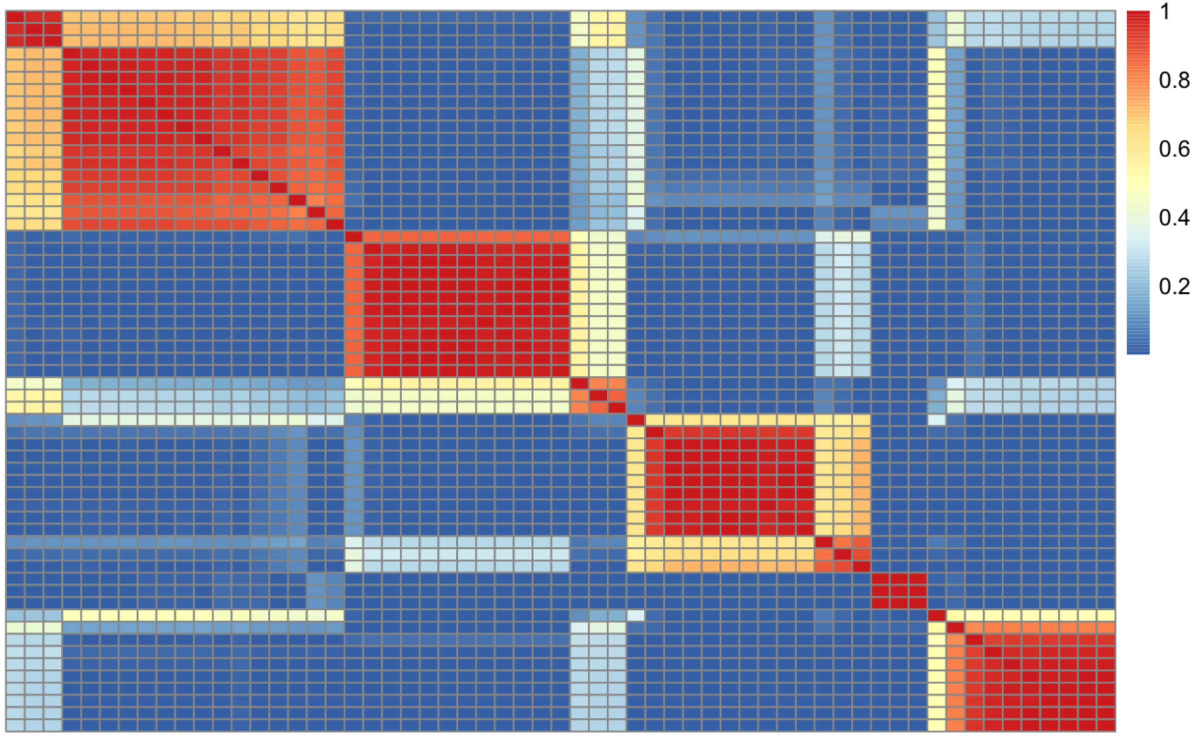
Previous work (Russell et al., 2015) has used pairwise similarity matrices as Δ for each model

- To get similarity matrix, multiply allocation matrix by its transpose:

$$S_m = A_m A_m^T$$

- **invariant to number and labelling of clusters across solutions**

Consensus matrix

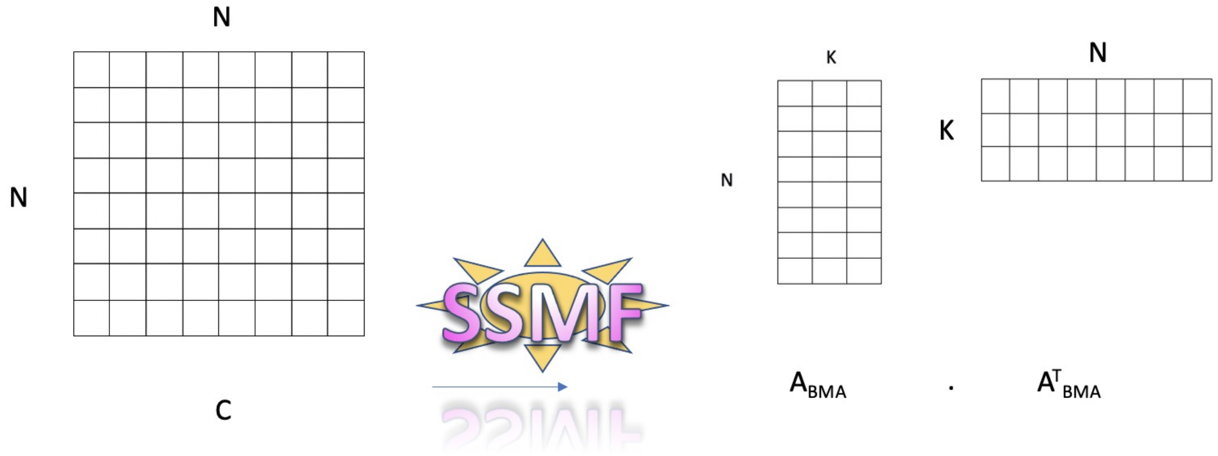


$$C = \sum_{m=1}^{M} \hat{\mathcal{W}}_m S_m.$$

Consensus matrix ➡️ Matrix factorisation ➡️ Cluster allocation probabilities

- Symmetric Simplex Matrix Factorisation (SSMF; Duan, 2020) to get $N \times K$ allocation matrix A_m from $N \times N$ consensus matrix C

- Generates probabilistic cluster allocations from pairwise probabilities

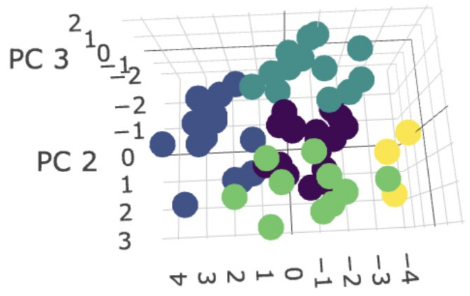


- Includes L2 regularisation step to reduce overfitting & redundant clusters

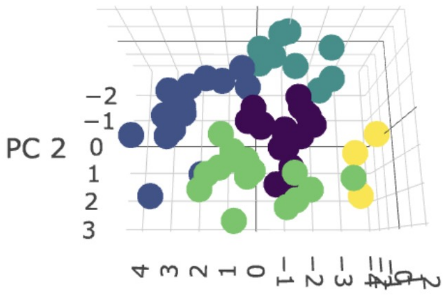
Case study: Clustering adolescents based on resting state EEG recordings

Model results

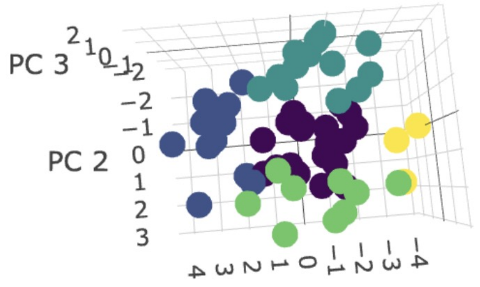
k-means, K = 5



Hierarchical clustering (Ward), K = 5

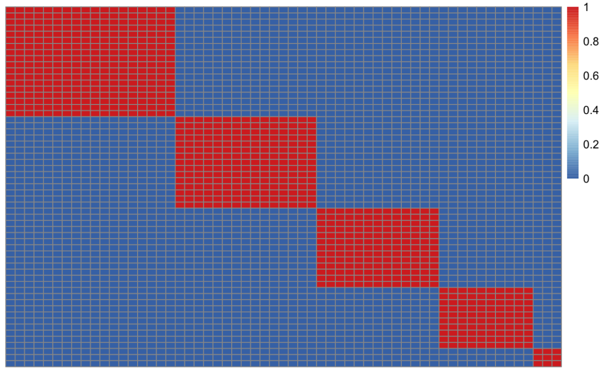


Gaussian mixture model, K = 5

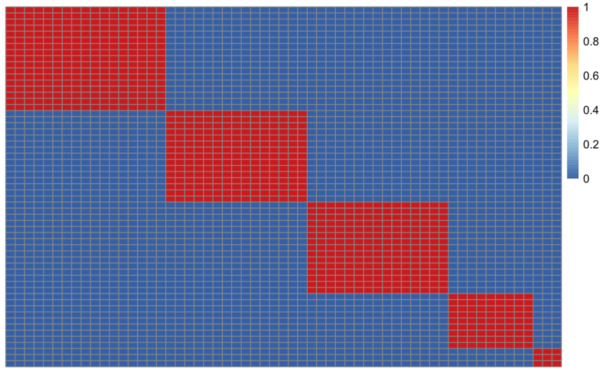


Model results ➡️ Similarity matrices

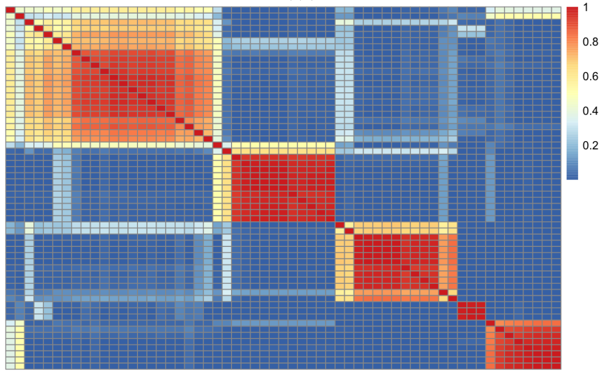
k-means $\hat{\mathcal{W}}_m = 0.36$



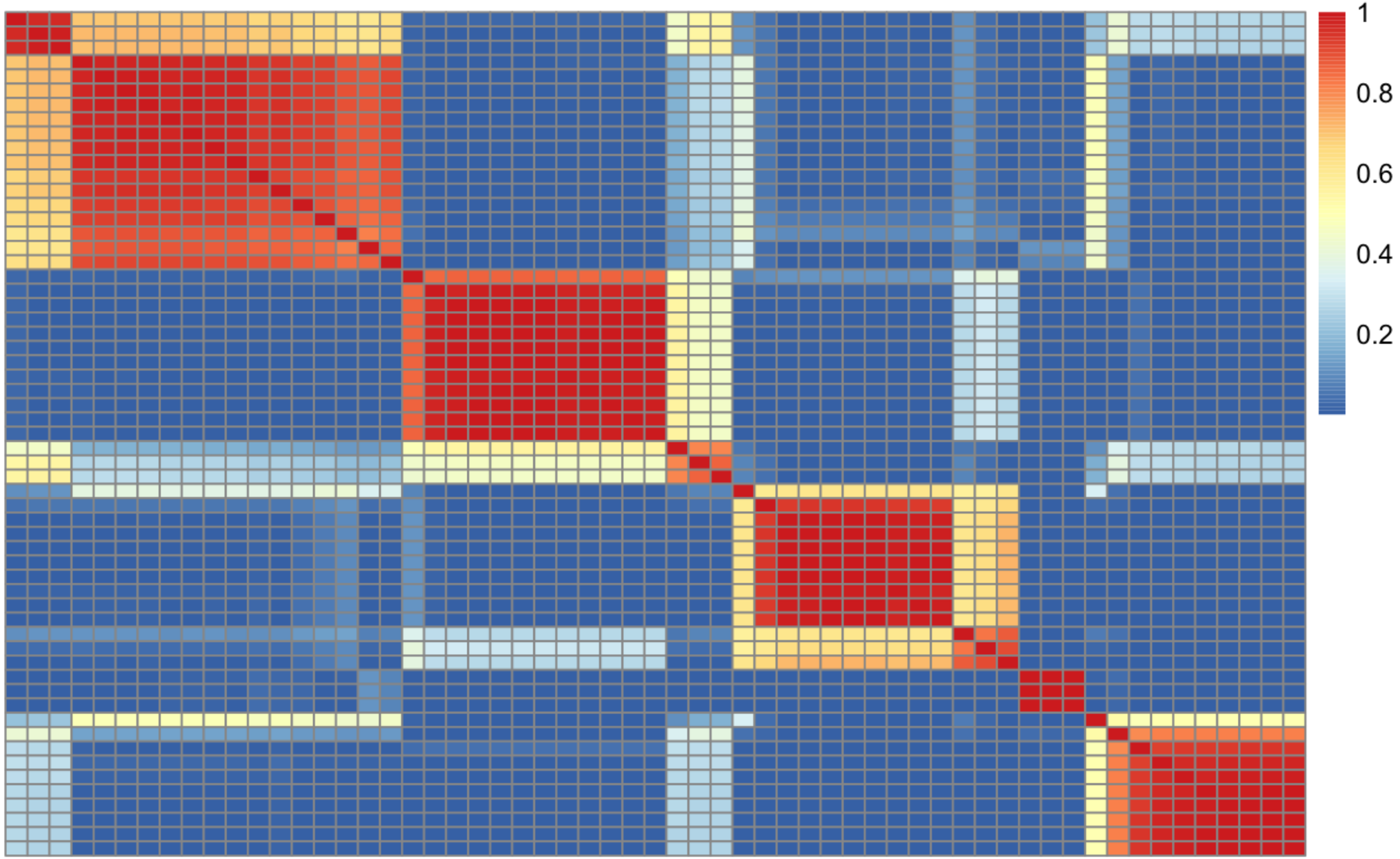
HC $\hat{\mathcal{W}}_m = 0.27$



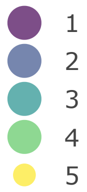
GMM $\hat{\mathcal{W}}_m = 0.37$



Model results ➡️ Similarity matrices ➡️ Consensus matrix

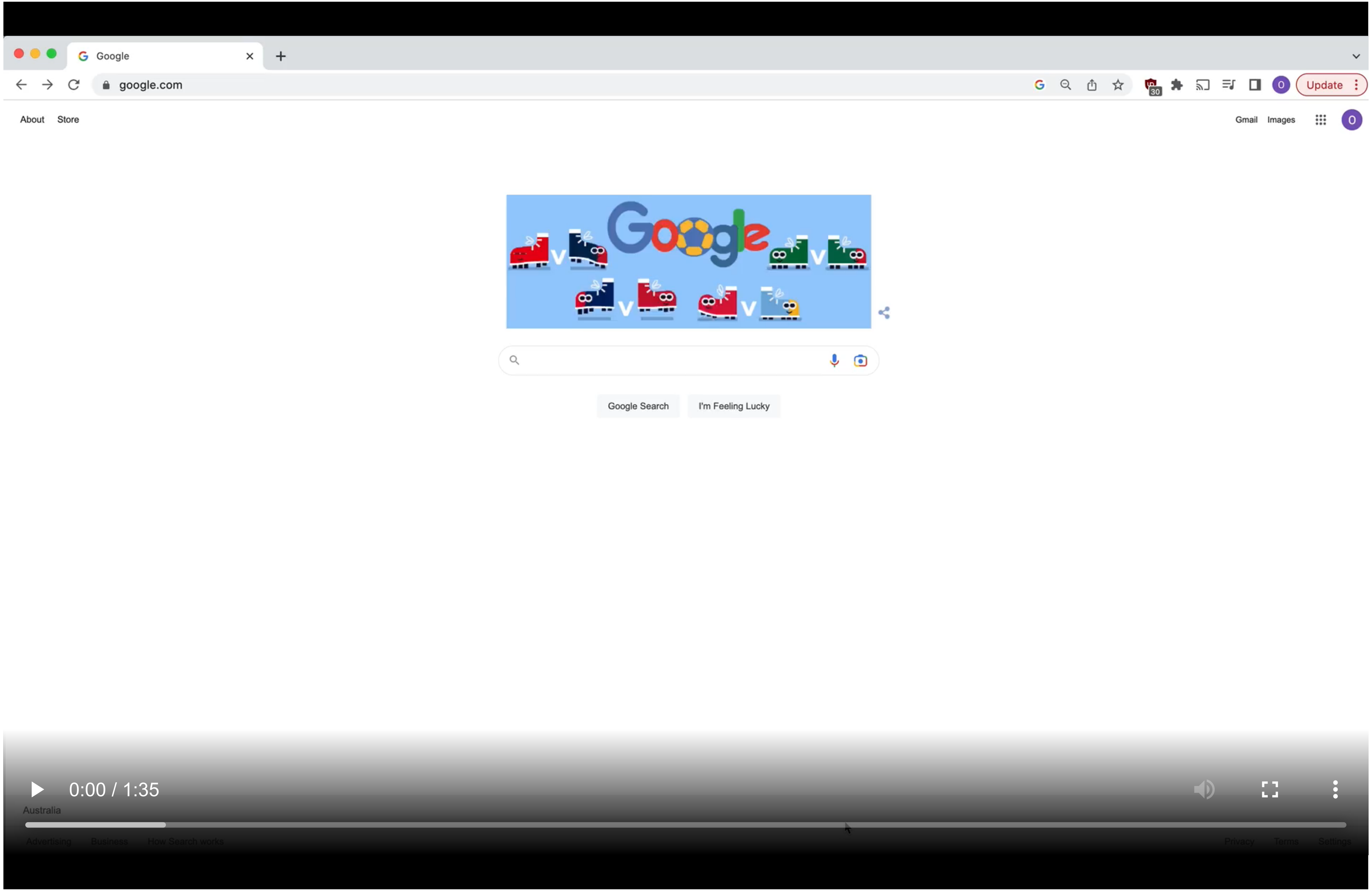


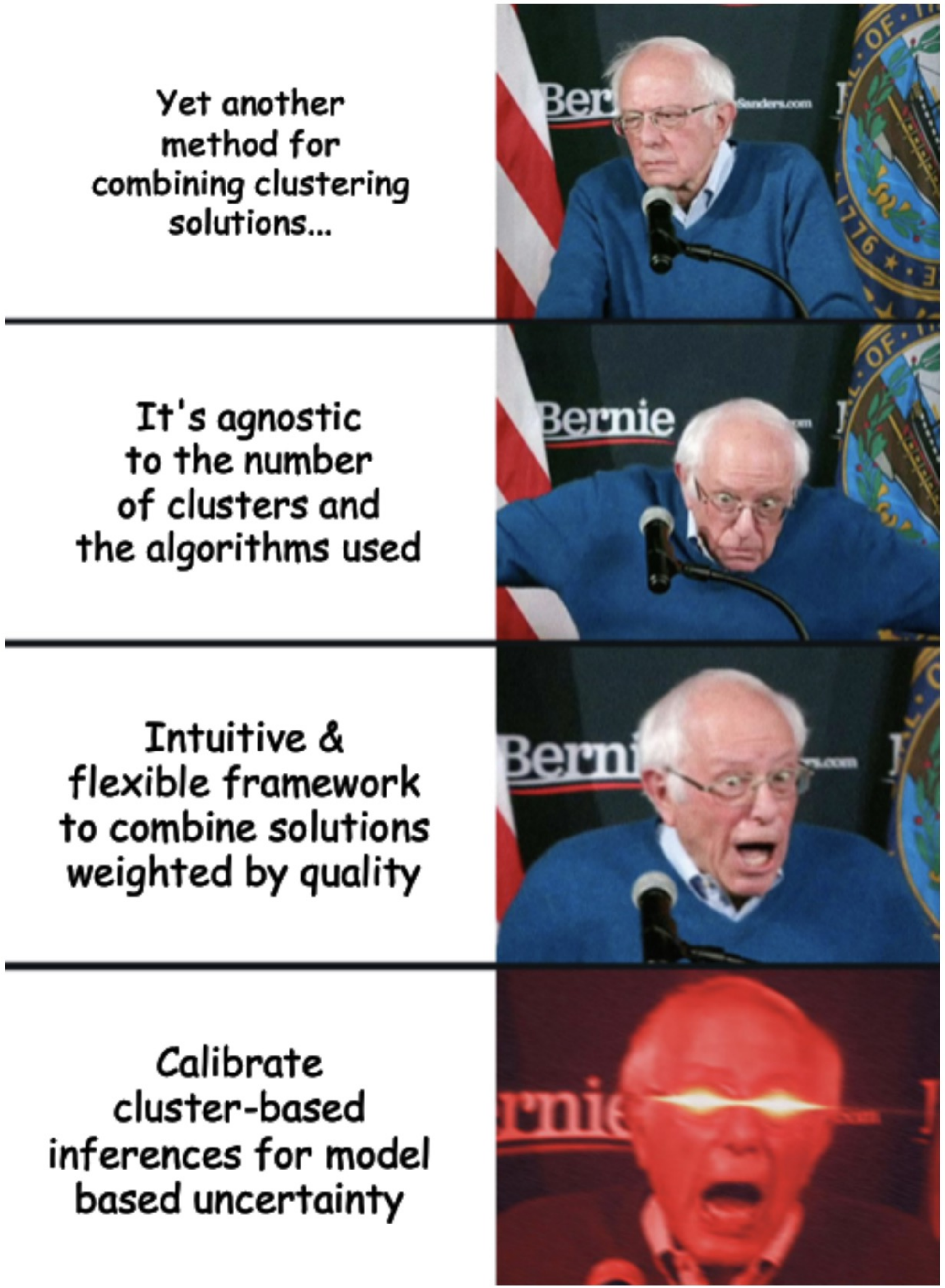
BMA Clusters with allocation uncertainty



- **Uncertainty can be propagated forward for further analysis in a Bayesian framework**

A quick demo

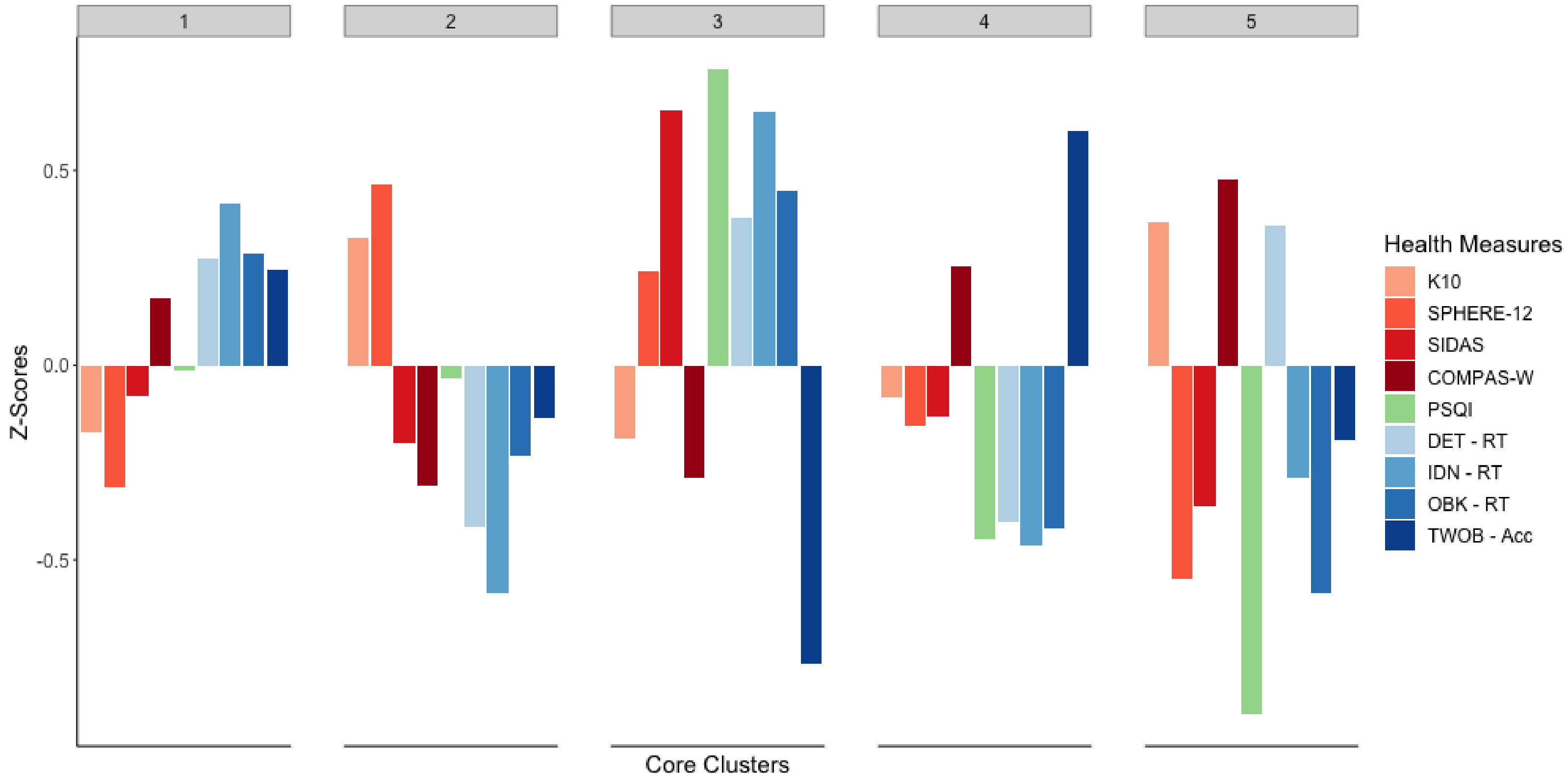

Google
google.com
About
Store
Gmail
Images
Google Search
I'm Feeling Lucky
0:00 / 1:35
Australia


Yet another method for combining clustering solutions...
It's agnostic to the number of clusters and the algorithms used
Intuitive & flexible framework to combine solutions weighted by quality
Calibrate cluster-based inferences for model based uncertainty

Next steps

- Benchmark against other ensemble clustering methods
- Compare weighting with BIC vs $\hat{\mathcal{W}}_m$ for GMMs
- Consider alternative internal validation metrics to approximate $P(\mathcal{M}_m|Y)$
- 🐦 Twitter: **@oforbes22**
- Preprint: **bit.ly/clusterBMA_preprint**

Cluster uncertainty for applied inference



Uncertainty in cluster allocation could be used to **moderate risk prediction** based on data-driven brain phenotypes

Equal Prior Weights

XB and CH indices

- conceptually and mathematically similar to BIC
- Unlike BIC, can be calculated + directly compared across different clustering algorithms

New proposed weight:

$$\mathcal{W}_m = \frac{\frac{1}{CH_m}}{\sum_{m=1}^{M} \frac{1}{CH_m}} + \frac{XB_m}{\sum_{m=1}^{M} XB_m}$$

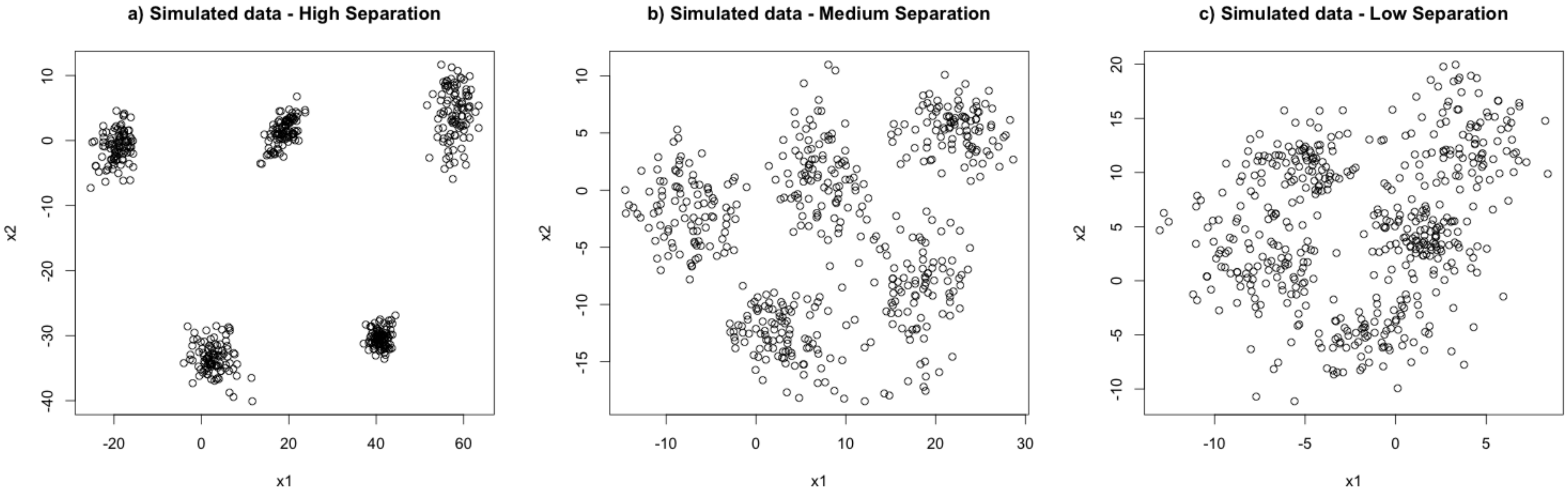
Approximate posterior model probability for weighted averaging:

$$P(Y|\mathcal{M}_m) \approx \hat{\mathcal{W}}_m = \frac{\mathcal{W}_m}{\sum_{m'=1}^{M} \mathcal{W}_{m'}}$$

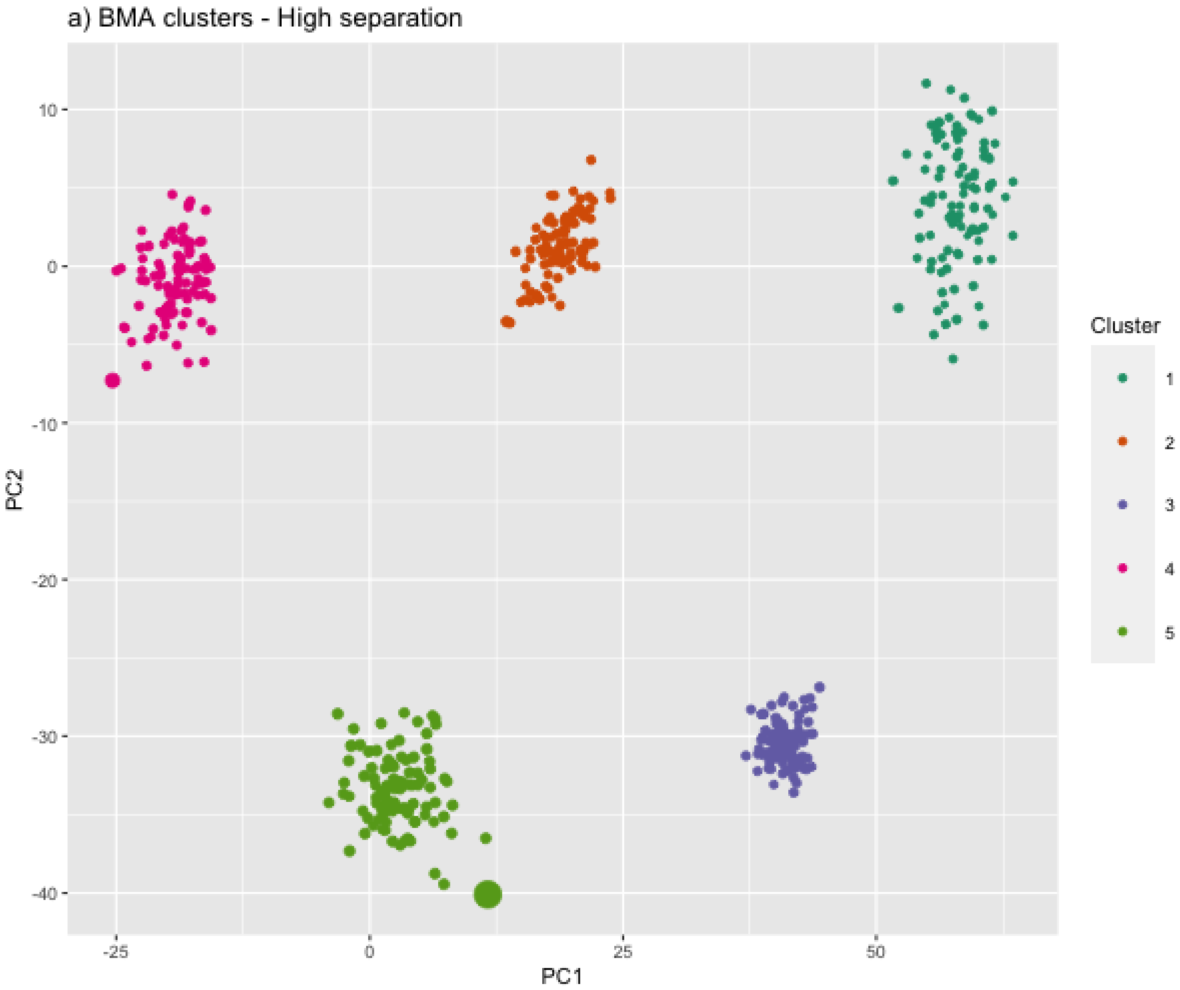
Substituting in for model evidence and prior in BMA posterior model probability:

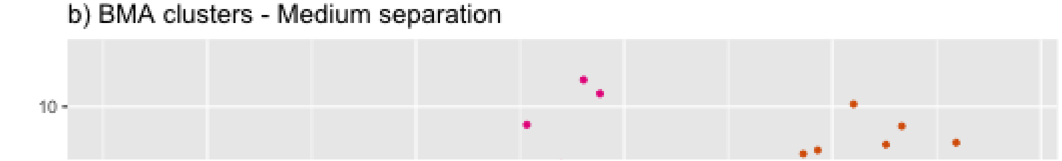
Simulation study (A): Cluster separation

Simulated datasets - R package "clusterGeneration"

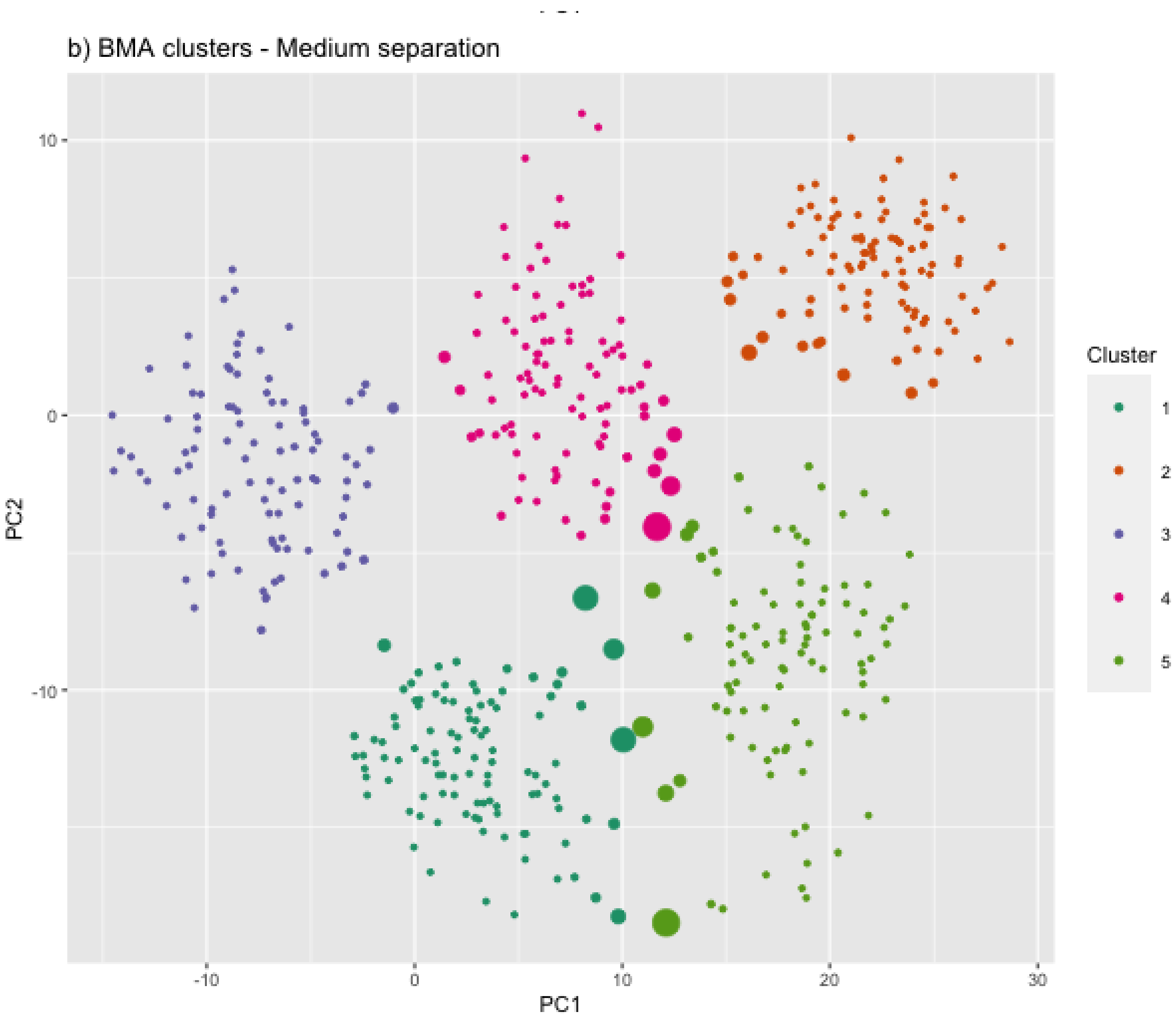


BMA solutions





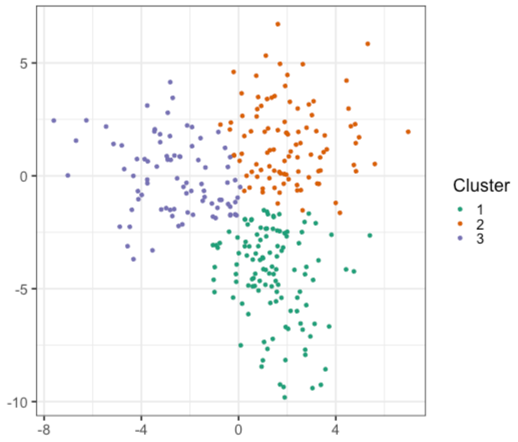
BMA solutions



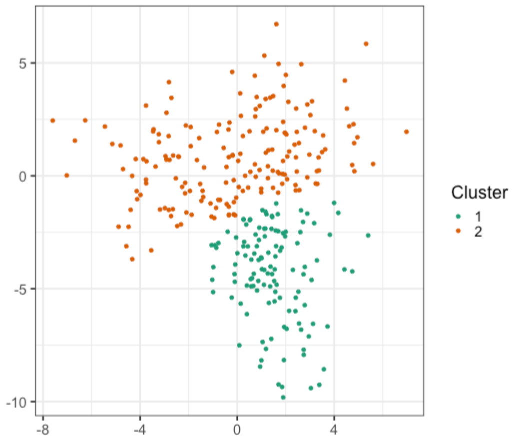
BMA solutions

Simulation study (B): Different numbers of clusters

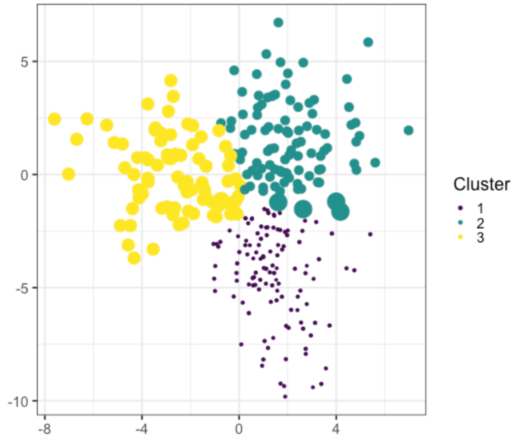
k-means - K = 3. Weight = 0.664



Hierarchical Clustering - K = 2. Weight = 0.336



BMA clusters - K = 3



k-means

- 'Hard' clustering
- Minimises within-cluster sums of squares

Hierarchical Clustering (Ward's Method)

- 'Hard' clustering
- Each observation starts out in its own cluster
- Repeated pairwise fusion of clusters that minimises change in within-cluster sums of squares (Ward)

Gaussian Mixture Model

- 'Soft' clustering
- Models data as coming from a mixture of Gaussian distributions

k-means objective function

$$J = \sum_{i=1}^{K} \left(\sum_{k} ||x_k - c_i||^2 \right)$$

Ward's objective function

$$D(c_1, c_2) = \delta^2(c_1, c_2) = \frac{|c_1||c_2|}{|c_1| + |c_2|} ||c_1 - c_2||^2$$

Mixture of multivariate Gaussians

$$p(x_n|\mu, \Sigma, \pi, K) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$