Centroid Algorithm

- Find a centroid, a tree minimizing the sum of squared distances, for a set of trees
- Start at a tree and check if any neighbour has a better objective function
- Repeat until a local optimum is reached

Conjecture:

- We tested for up to 7 taxa treespace and a variation of different tree set sizes
- This algorithm always returned a gloabal optimal solution

Variation for application

Problems :

- Number of trees
- Unknown number of local and global optimal solution to the problem
- Hard to prove that it finds a global optimal solution

Variation:

- Greedy choice, only following the path with the most improvement in each step
- Start with a sample of the tree set and add more trees until the tree set is found
- Choice of the starting tree is important
- Return value will be a local optimum

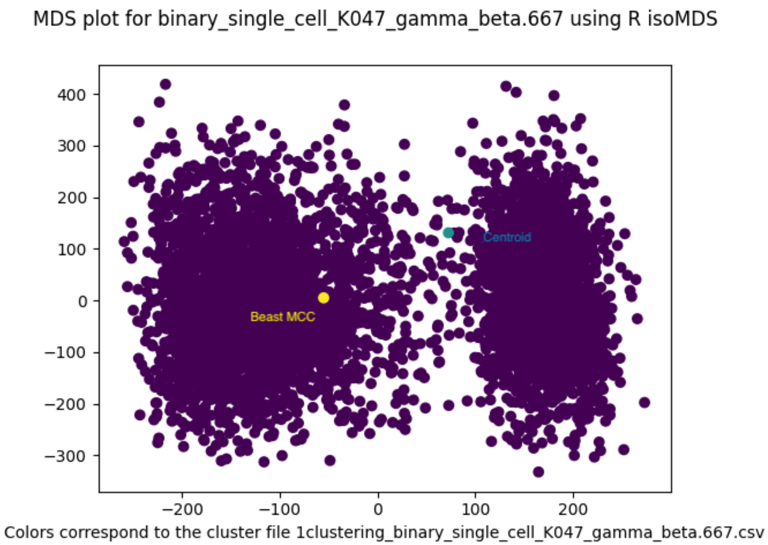

Figure: Comparing the MCC(yellow) vs Centroid tree (blueish), visual result is also present in the tree distances!

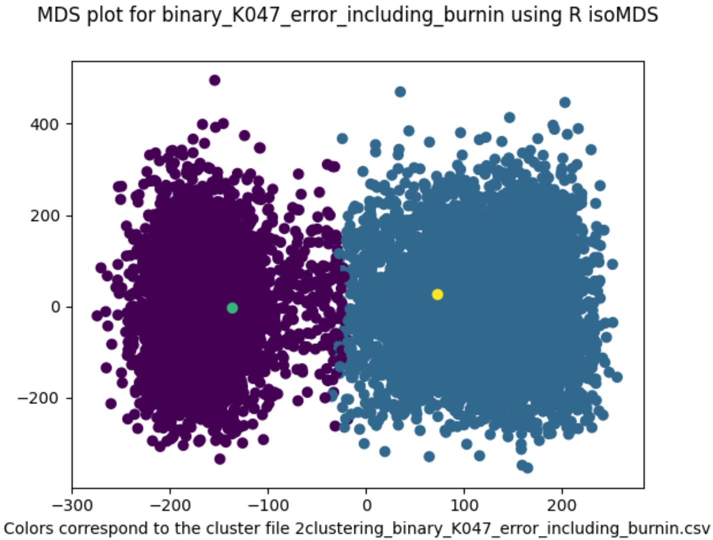

Figure: Summarize identified clusters seperately

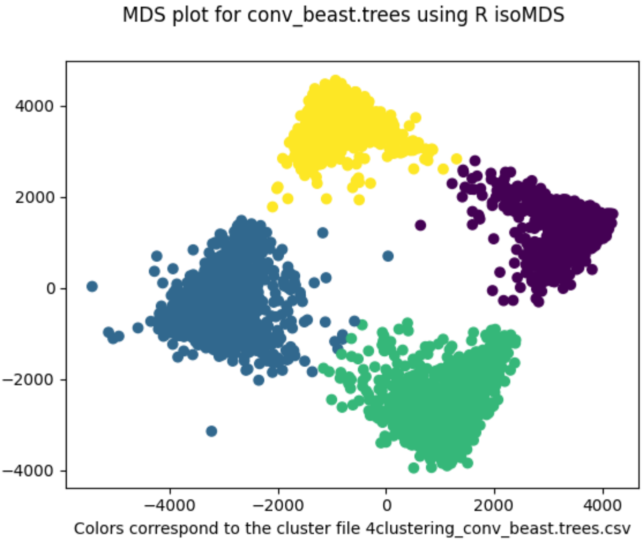

Figure: Able to identify clusters via the true tree-distance Matrix

Choosing the number of clusters

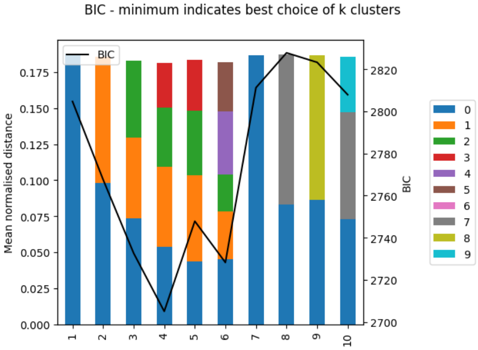

Figure: Choosing the number of clusters with Bayesian inference criterion

- Clustering is not using the MDS, only for visualization
- MDS is not a perfect visualisation

Bayesian inference criterion

set of trees $\mathcal{T}$, clustering σ, $\mathcal{R}$ set of summary trees
$m = |\mathcal{T}|$, k clusters

$$\tilde{d}(\mathcal{T}, \mathcal{R}, \sigma) = \frac{\sum_{i=1}^{m} d(\mathcal{T}_i, \mathcal{R}_{\sigma(i)})}{m * \frac{(n-1)(n-2)}{2}}$$

$$h(\mathcal{T}, \mathcal{R}, \sigma) = 1 - \tilde{d}(\mathcal{T}, \mathcal{R}, \sigma)$$

$$BIC = \frac{k}{2} * ln(m) - 2 * ln(h((\mathcal{T}, \mathcal{R}, \sigma))^m)$$

- Defining a likelihood for data $\mathcal{T}$ given the model $\mathcal{R}$, k
- Depending highly on the clustering **and** the summary method

MDS Distortion

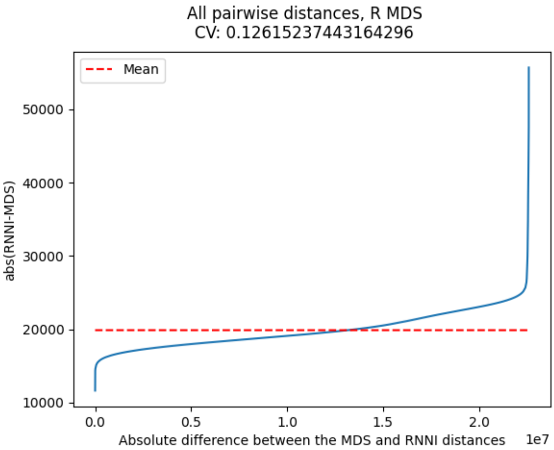

Figure: A constant distortion of the distances would be ideal

- Distortion $= |D_{MDS} - D_{RNNI}|$ for all trees
- $CV = \frac{\sigma}{\mu}$, coefficient of variation for distortion

Another Application of the SoS

- Given a summary tree and a treeset
- compute the relative sum of squared distance for the summary
- relative meaning to divide by the number of trees
- Do this for different burnin percentages
- Indication for the quality of the summary tree
- Also indicates whether the posterior set has converged

Converged data

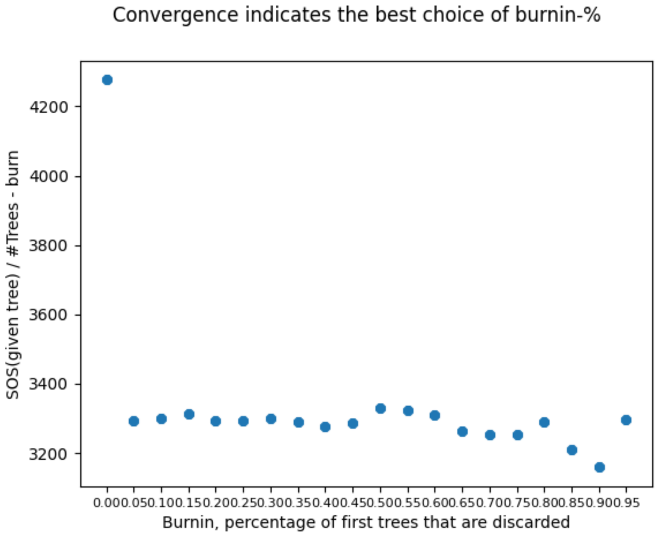

Figure: Good summary tree for a converged chain

Not so converged data

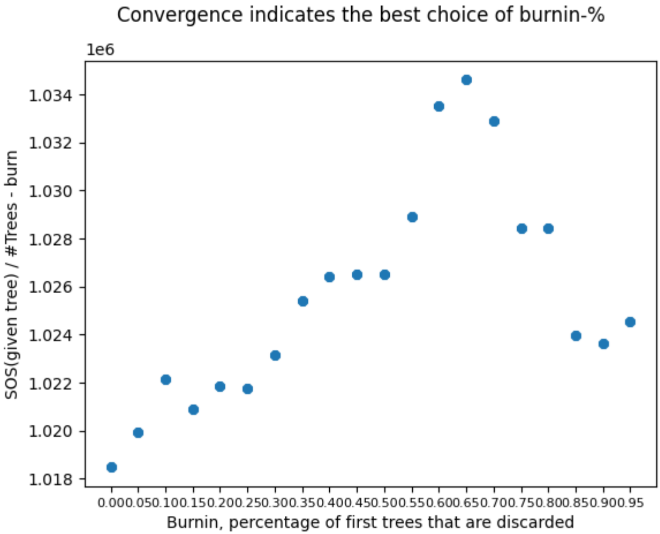

Figure: Increasing the rel. SoS value indicates that the chain has not converged