Phylogenetic Tree Analysis using Clustering

Cadran Cowansage

Department of Computer Science

Texas A&M University

July 2007

Goal: To better understand phylogenetic search heuristics by using clustering techniques to look at the relationships between trees

Phylogenetic Clustering

- Partition data into subsets (clusters) that share
  - a common characteristic
  - proximity according to a distance measurement (distance can be used to measure the similarity of two elements)

Why Clustering?

- Difficult to analyze large sets of phylogenetic trees
  - Currently consensus tree methods used to summarize trees
  - **Problem:** Information about individual trees lost

Previous Work:

- Maddison proposed “tree island” classification technique
  - Trees differing by only 1 branch re-arrangement grouped as islands
  - Examined candidate trees
  - **Goal:** trees in different island express different information
- Stockham, Wang and Warnow clustered candidate trees
  - Made sub consensus trees to characterize each cluster
  - Compared 3 categories of clustering algorithms
  - **Goal**: To reduce information loss associated with having one consensus tree
- Hillis, Heath and St. John used visualization techniques to cluster optimal and near optimal trees
  - Plotted trees in 2-D space and examined cluster patterns
  - Compared trees collected with different search techniques
  - **Goal:** To find new ways to examine large sets of trees and avoid information loss

Our Methodology

- 44 taxa dataset (Murphy et al., 2001)
- Hyun Jung's search start trees -- 5 RSA trees created in Phylip
- Collected 5 sets of search path trees for NNI, SPR and TBR using SLS software
- Collected candidate trees trees using Paup*
- Computed RF distance matrices for sets of trees using Hash-RF
- Analyzed trees using R packages (cclust, clusterGeneration, cluster, etc.)

Partitioning Around Medoids (PAM) algorithm

- Part of Cluster package in R
- More robust version of K-means
- Partitions data into *k* clusters, “medoids”
- Uses dissimilarity matrix to minimize a sum of dissimilarities
- Algorithm selects *k* best representative data points as medoids
- Then assigns rest of data points to medoids
- Each cluster characterized by sum of dissimilarity of all points in cluster to their medoid

Silhouettes

- Purpose:
  - Evaluate clustering
  - Select best value of *k*
- Part of R Cluster package
- Measure of dissimilarity of each data point to its nearest neighbor outside cluster s(*i*)
- Average over all clusters defines quality of cluster

- s(*i*) = 1 well clustered
- s(*i*) = -1 poorly clustered

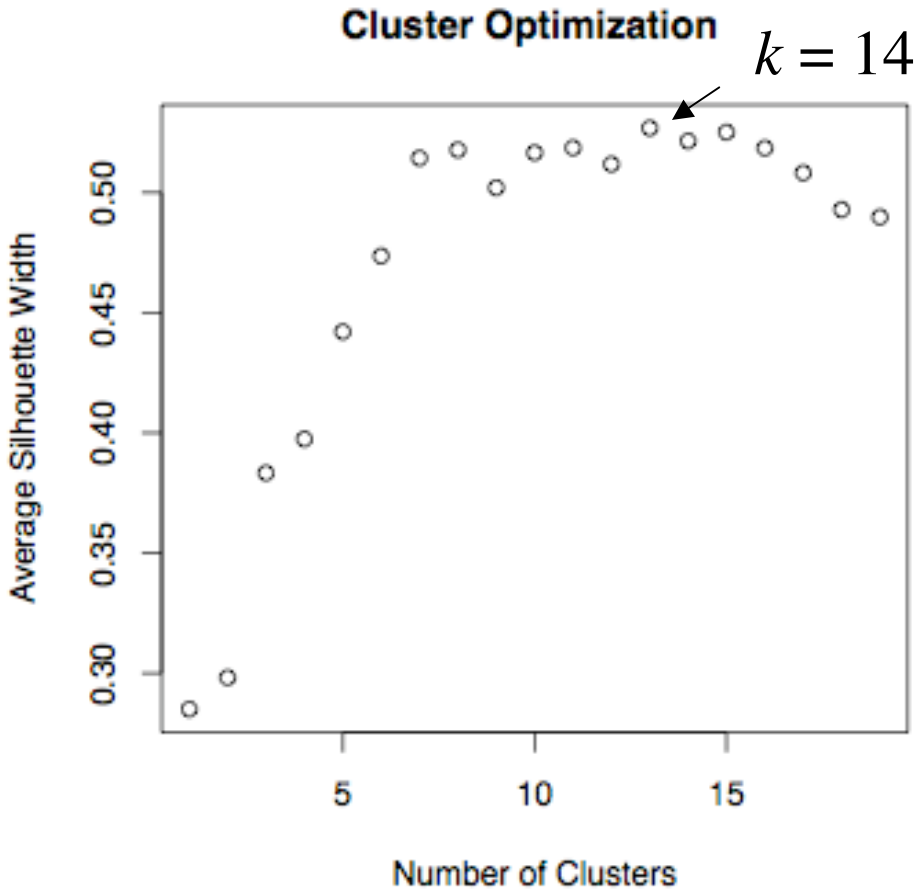
NNI Clustering

PAM input:

- RF distance matrix of search-path trees for 5 runs (166 trees)

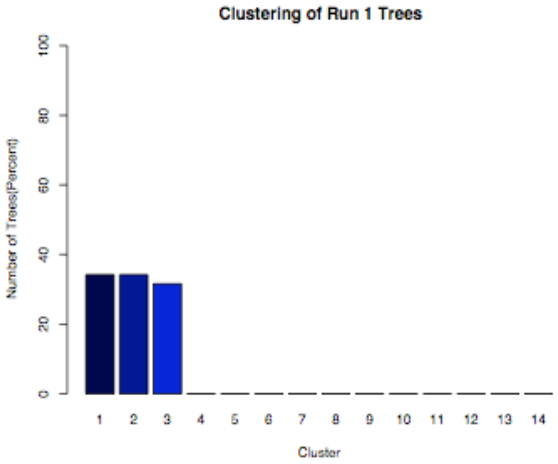
Step 1: Determine best value of k

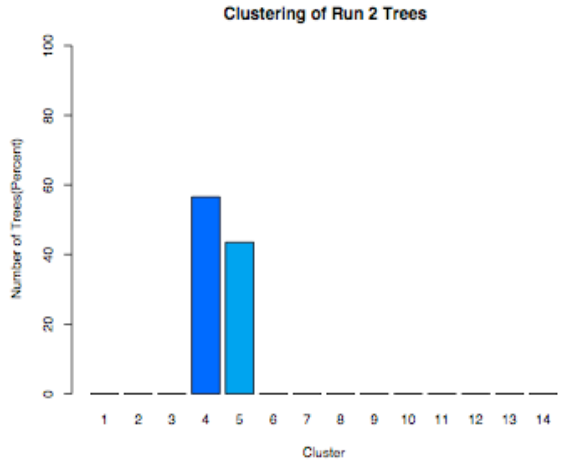
- Measure Avg. Silhouette width of PAM clusterings

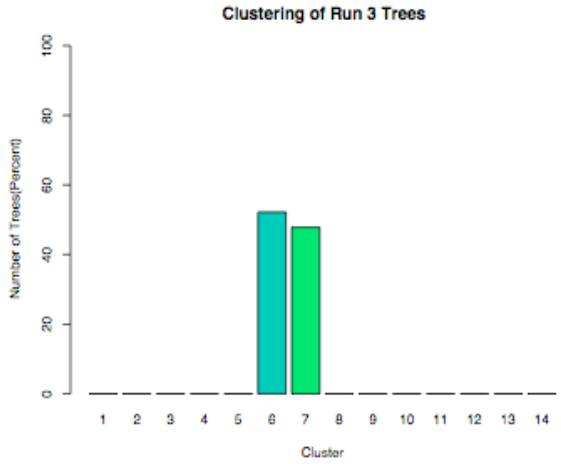


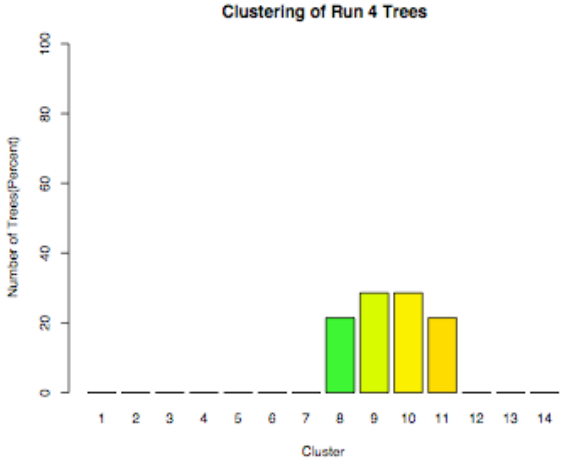
NNI Clustering

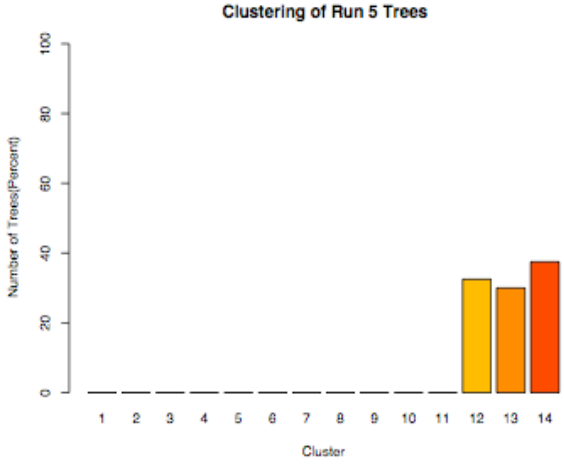
Step 2: Cluster NNI trees into 14 clusters





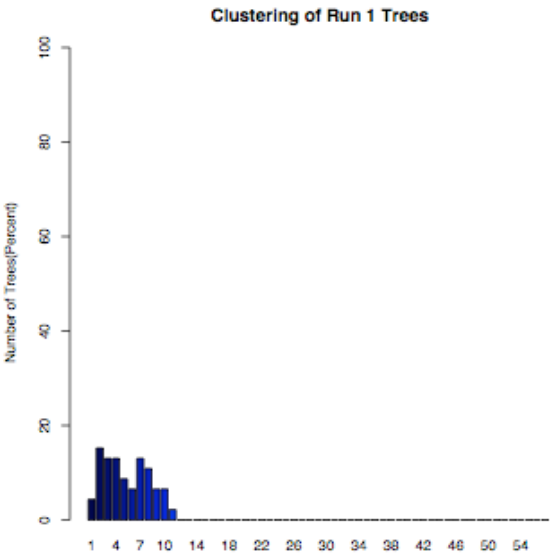


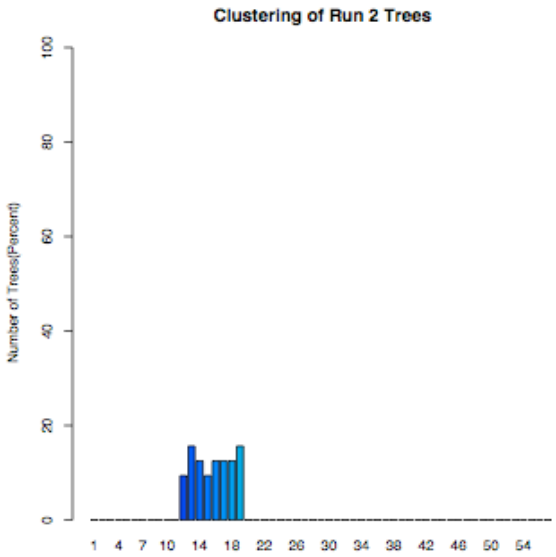


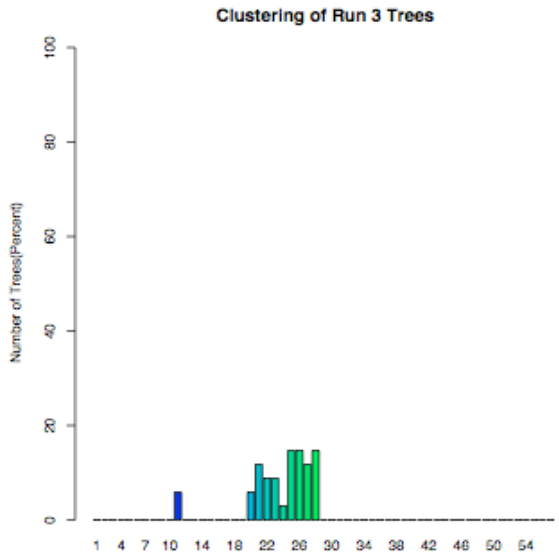


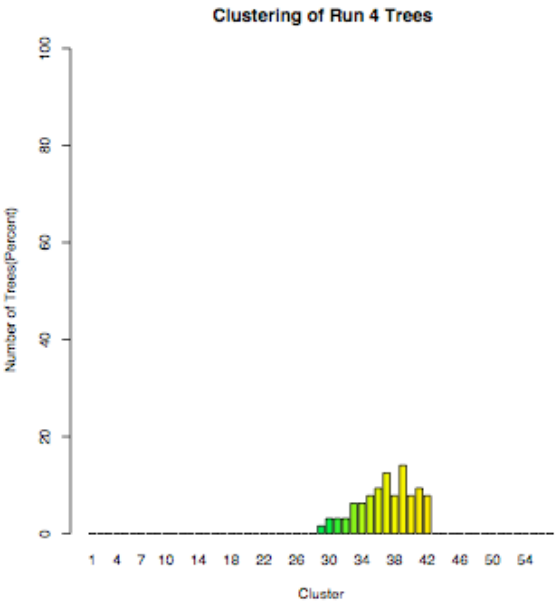
…Trees cluster by run

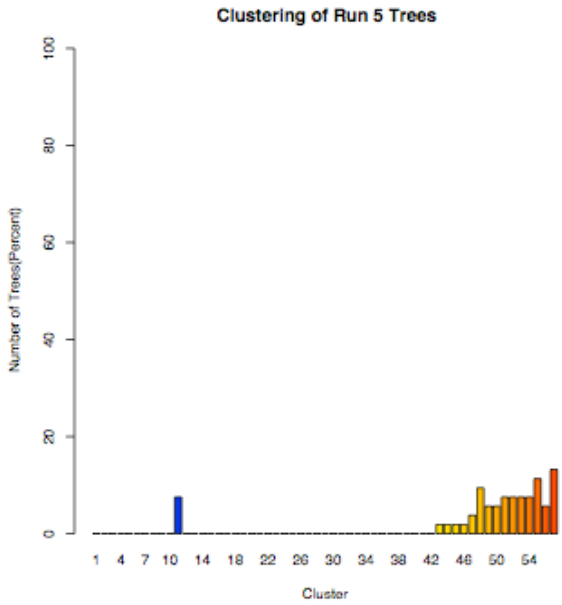
SPR Clustering





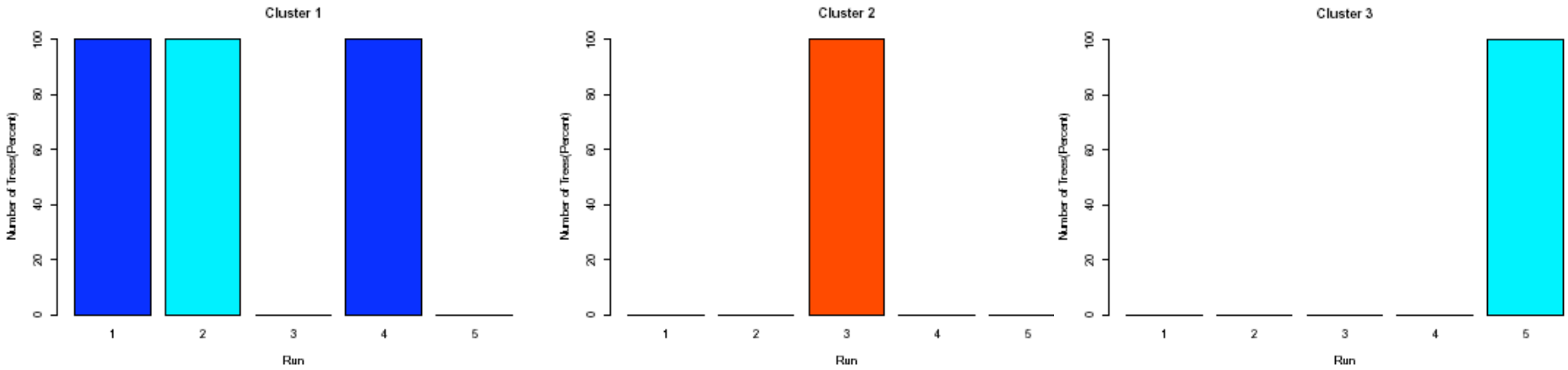




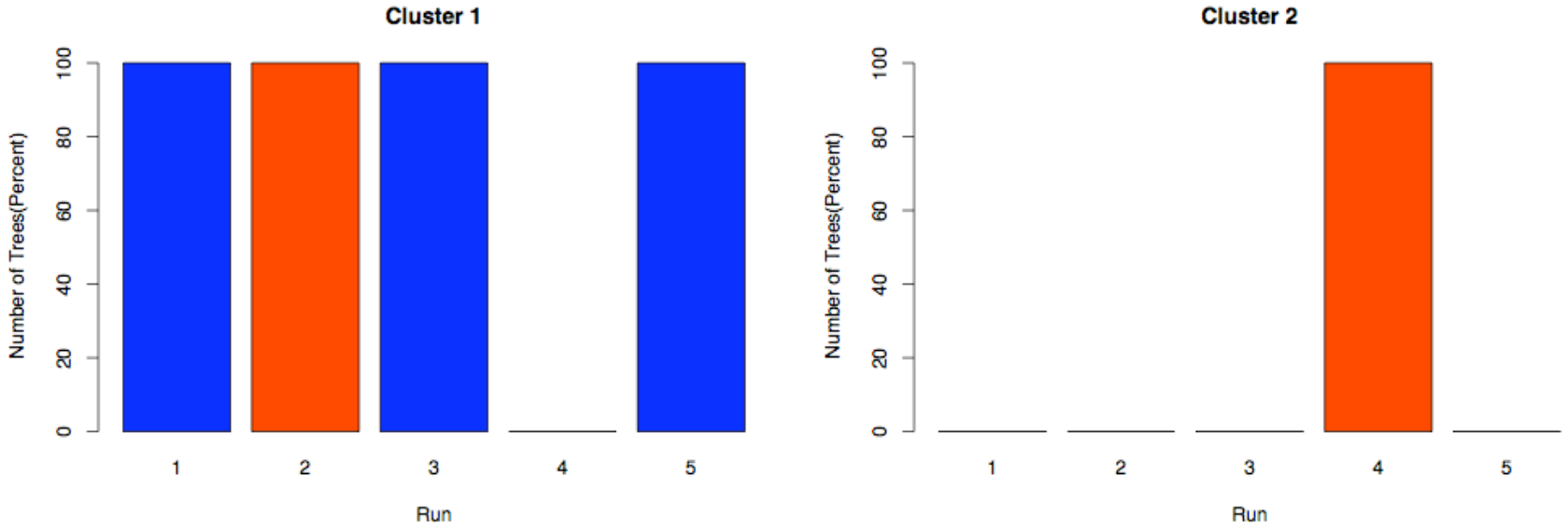


…SPR and TBR trees do too

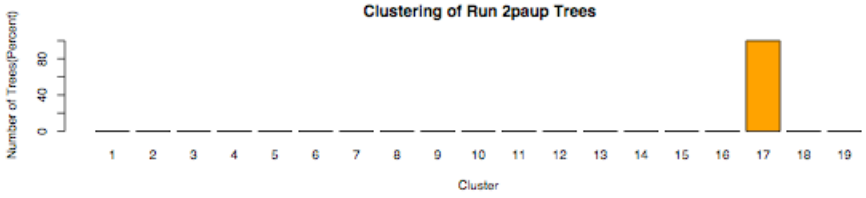
Clustering of Paup Candidate Trees for SPR

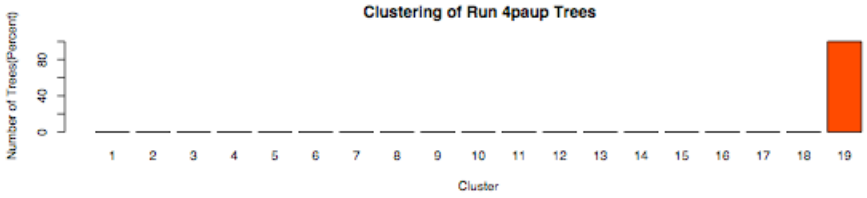


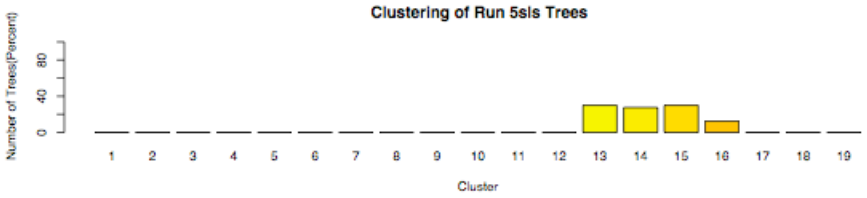
and TBR

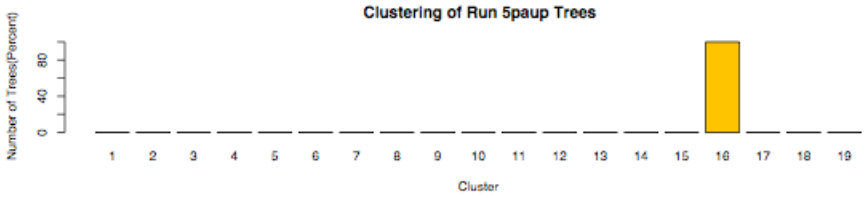


Clustering of path trees & candidate trees for NNI

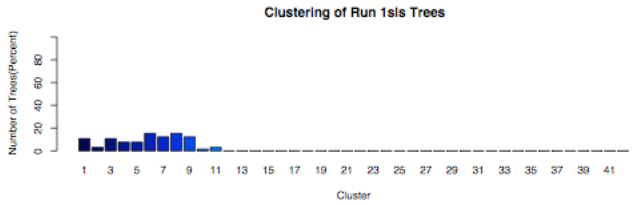


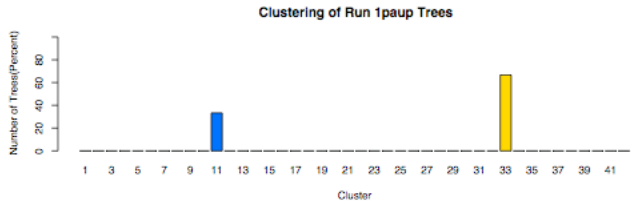


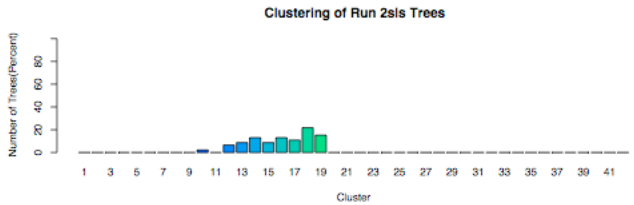


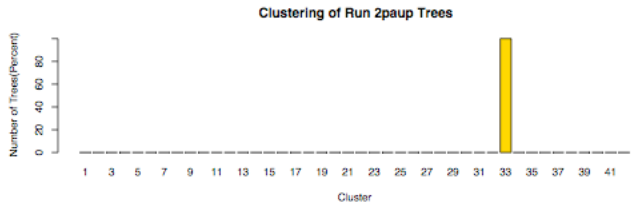


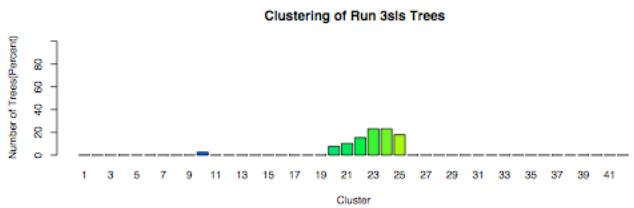
Clustering of path trees & candidate trees for TBR

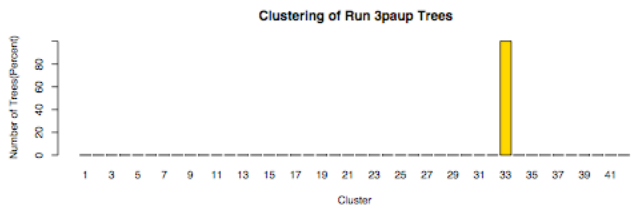


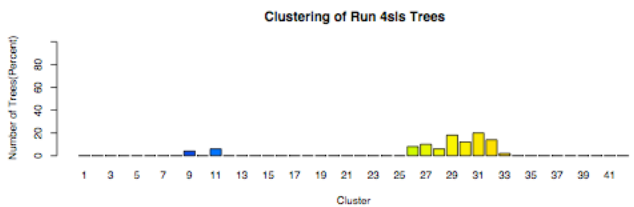


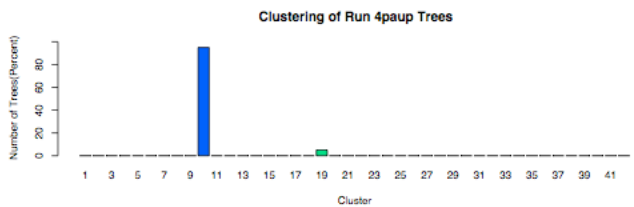


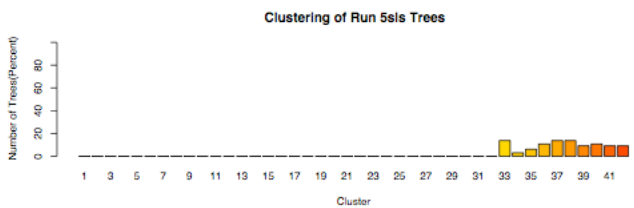


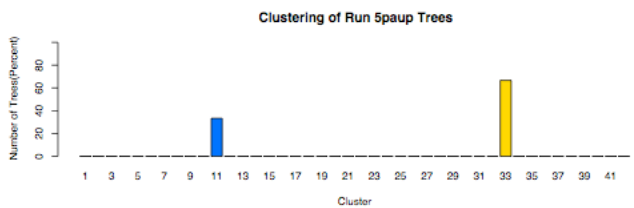












Results:

- Trees on the same search path cluster together
- NNI candidate trees from each run cluster together
- Some TBR and SPR candidate trees are clustered by run
- Some candidate trees and near optimal search path trees cluster