Progress Report, Nathan Cunningham - August 3, 2016

Integration of MDI and SMC clustering code

- Continued working on knitting together SMC code and MDI code, shifted focus to look at a single dataset being clustered according to a finite mixture model.
- The single dataset aspect doesn't change the problem, as each of the datasets are clustered reasonably independendly.
- The SMC clustering method provided some problems:
  - The cluster container structure ('`clusterContainer`') contains one entry for each cluster which is updated `nGenes` times. The SMC clustering method for each gene will out a clustering allocation for each particle.
  - As such, the SMC algorithm is left to run through all the genes before updating `clusterContainer`.
  - Therefore, only one allocation can be output from the SMC algorithm. This is selected based on the maximum `logweight` across particles. Also considering doing this according to the particle which maximised the expected adjusted rand index across particles, but this wouldn't scale well as it would require $\binom{N}{k}$ calculations of the rand index for N particles.
  - The SMC algorithm, as is, constructs a variable `sumy` of the form

    ```
    sumy1, it = data(i);
    ```

    which only takes the first entry of each variable. This is later used for calculating the μ^* value and subsequently the `logprob` which decides which cluster to allocate an observation to. Have modified this code to take in all observations.
  - As such, the logprob is a vector of logprobs for each observation, and the logprob of assigning to a cluster is the product of these logprobs.
  - In original SMC the prior propability of assigning to a cluster `prob` is based on the normalised generalised gamma. Updated so that this is the `prob` calculated in the MDI. Which is the probability of assigning to a cluster and is upweighted according to concordance across datasets.
  - The μ values at each run of the SMC algorithm are the μ values from the previous cluster allocation. In the SMC algorithm they are initialised by a user-specified value. May lead to cluster allocations being rigid, as observations are biased towards staying in their current cluster.
- The algorithm seems a bit slower to run as a result of the changes.

Adjusted Rand Index between the two allocations (below) < 0. Original MDI code won't run with number of clusters restricted to just 5.

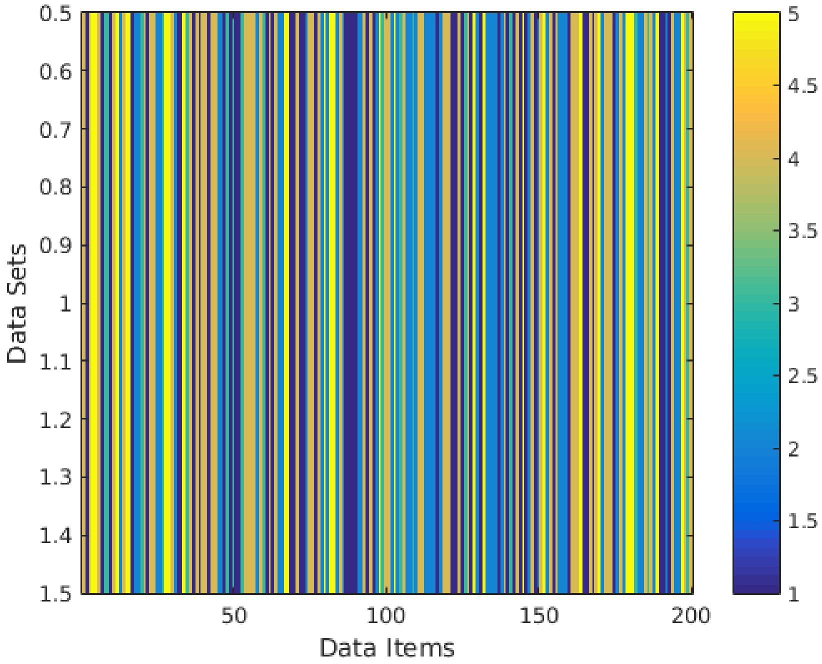

Figure 1: Clustering for Gaussian test data with 5 clusters using SMC&MDI (no. of clusters restricted to 5)

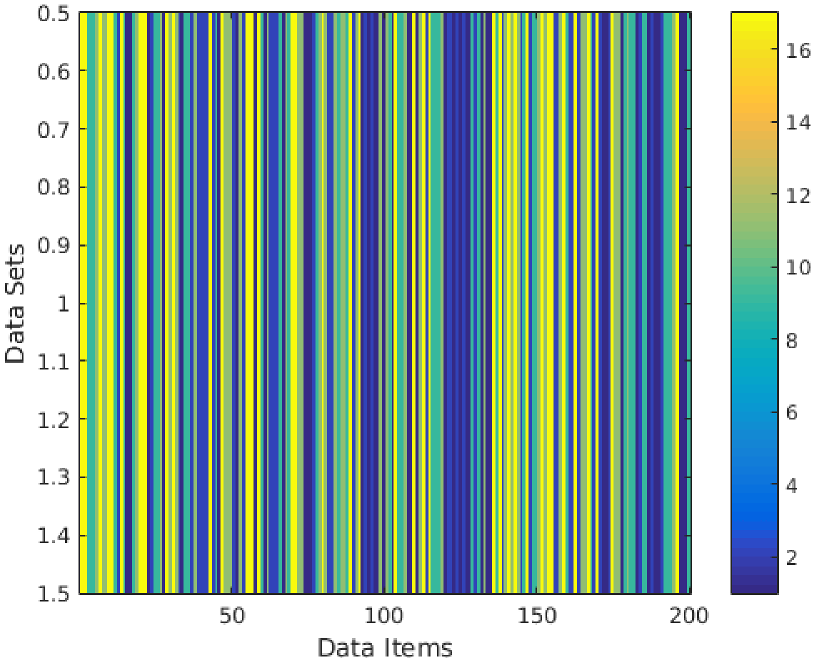

Figure 2: Clustering for Gaussian test data with 5 clusters using original MDI (no. of clusters restricted to 20)