Simulation Data – TFunHDDC Test Results

TFunHDDC was run with random, vector, mini-em, and kmeans initializations, 3 clusters, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6 thresholds, and all models. Number of repetitions was set to 30, and all other settings were default, except for number of cores for parallel. The tests were run on one generation of the simulated data (the non-outlier case from the paper).

Each run of the function used one of each parameter. This was done to generate more classifications to calculate CCR and ARI on, as well as to see if the Python implementation was finicky/picky about the parameters in terms of correctly classifying the simulated data.

I was a bit greedy/excessive with the number of parameter combinations run (170 different parameter combinations), so I've displayed a result for each initialization with a CCR and ARI of 1

Initialization	Model	Threshold	CCR	ARI
Random	AKBKQKDK	0.01	1	1
Vector	AKJBKQKDK	0.01	1	1
Mini-em	AKKBKQKDK	0.01	1	1
Kmeans	AKBQKDK	0.01	1	1

Additionally, quite a few of the runs had resulted in 0 CCR but 1 ARI (except for init = vector)

Random	AKJBKQKDK	0.5	0	1
Mini-em	ABQKDK	0.5	0	1
Kmeans	ABKQKDK	0.3	0	1

The table below displays the number of parameter combinations that resulted in a perfect CCR, and perfect ARI (each initialization is run 42 times in total

Initialization	# Perfect CCR	# Perfect ARI
Random	2	25
Vector	34	34
Mini-em	4	23
Kmeans	2	41

Although the CCR is quite low for most runs, in many of cases with perfect ARI, the CCR is 0, 0.33, or 0.66, if not 1. This would indicate that the function is just mislabelling the data, as opposed to misclassifying.