

Brief summary of the function files

`adjusted_rand_index.m` Computes the Adjusted Rand Index (ARI).

`spatial_d.m` Computes spatial depth.

`rspatial_dp.m` Constructs the spatial-depth-based similarity matrix.

`Maha_d.m` Computes Mahalanobis depth.

`Maha_dmcreator.m` Constructs the Mahalanobis-depth-based similarity matrix.

`if_corr.m` Performs a simple test for correlations among variables and determines the covariance matrix strategy for constructing the Mahalanobis-depth-based similarity matrix.

`EM_EEV.m` Implements the Expectation-Maximization (EM) algorithm for the EEV model in the Gaussian Parsimonious Clustering Model (GPCM) family.

`llh_calculator.m` Computes the log-likelihood in the EM algorithm.

`getlocalcenter.m` Identifies the deepest point for each subset and labels them as local centers.

`DLCC.m` Main function implementing the DLCC algorithm.

`filter_center.m` Implements the local centers filtering procedure.

`sim_mat.m` Constructs the similarity matrix for neighborhoods of filtered centers.

`get_temp_cl.m` Groups the filtered centers.

`get_temp_cl_WK.m` Groups filtered centers with a given k value, flexible min strategy.

`spec_clus_withsim.m` Performs spectral clustering given an adjacency matrix.

`group_adjust.m` Examines the initial grouping and makes it satisfy the flexible min strategy.

`get_temp_cluster.m` Provides temporary clusters based on the grouping results of filtered centers.

`left_class.m` Classifies remaining observations based on temporary clusters. Built-in classification methods include Maximal Depth Classifier, Random Forest (RF), and K-Nearest Neighbor (KNN). The “maxdepth” procedure is also included in this function.

`KNNdep.m` Implements the KNN algorithm based on the depth-based similarity matrix.

`depth_by_cluster.m` Computes data depth with respect to each cluster.

`cluster2cv.m` Converts clustering labels to a class vector.

`loop_rfdlcc.m` Executes the Random Forests classification 100 times with different seeds.

`DCscore.m` Computes the DC metric for evaluating clustering results.

`DBCA.m` Performs DBCA algorithm.

`DBCAmerge.m` Finds the merge thresholds of DBCA-clusters to define the maximum η value which connects all points in a certain cluster.