Itgrag Quick Function Reference

PLEASE ALSO TAKE A LOOK AT main.m FOR USAGE EXAMPLES !

Note: If you execute Octave from a different directory, use *addpath("/your/path/to/Itgrag_v0.3").*

Notation

graph index: idx
adjacency list: adjL
array: [ ](dim1, dim2)
cell array: {}(dim1,dim2)
function handle: **@**

The adjacency list is stored as a (n,1)-cell array. Each cell holds the indices of adjacent nodes. The dimensions of arrays or cell arrays are omitted, if they are clear form the context.

Graph Generation Functions

- idx = generateEmptyGraph()
- idx = generateRandomGraph(@pointDistributionFunction, {}, @nodeConnectionFunction, {})

  Point Distribution Functions:

  @uniformRandom: {numNodes, dimensions}

  @fixedCoordinates: {[coordinates]}

  Node Connection Functions:

  @connectThreshold: {threshold}

  @connectThresholdGrid: {threshold}
- idx = generateGraph(coordinates, adjL)

Elementary Distortion Functions

- idx = distortGraph(idx, @distortionFunction, {})

  Distortion Functions:

  @addNodes: {[node coordinates](n,d)}

  @deleteNodes: {[node indexes](n,1)}

  @addEdges: {[source nodes, target nodes](n,2)}

  @deleteEdges: {[source nodes, target nodes](n,2)}

  @displaceNodes: {[node indexes](n,1), [displacements](n,d)}

Composed Distortion Functions

- idx = addEdgesRandomly(idx, partition, numEdges)

- idx = deleteEdgesRandomly(idx, partition, numEdges)
- idx = deleteNodesRandomly(idx, partition, numNodes)
- idx = applyNoise(idx, partition, minAmplitude, maxAmplitude)

Partitioning Functions

- idx = partitionGraph(idx, @partitioningFunctions, {})

  Partitioning Functions:

  @partitionMETIS: {number of partitions} [1]

Graph Edit Distance Functions

- SET_EDIT_COST(addDeleteNode, addDeleteEdge, @nodeMoveCostFunction, {}, @edgeStretchCostFunction, {})

  Node Move Cost Functions:

  @linearThresholded: {gradient, threshold}

  Edge Stretch Cost Functions:

  @linearThresholded: {gradient, threshold}

- [cost, nodeAttributedCost] = getEditDistance(idx1, idx2, @editDistanceAlgorithm, {})

  Edit Distance Algorithms:

  @editDistanceViaItgragDeltas: {}

Utility Functions

- [numNodes, numEdges, numPartitions, numDimensions] = getGraphStats(idx)
- [numNodes, numEdges] = getPartitionStats(idx, partition)
- isolatedNodes = getIsolatedNodes(idx)

I/O-Functions

- coordinates = readNodeCoordinates(filename)
- adjL = readAdjacencyList(filename)
- adjL = readAdjacencyMatrix(filename)
- [coordinates, adjL] = readDot(filename)
- saveNodeCoordinates(idx, filename)
- saveAdjacencyList(idx, filename)
- saveAdjacencyMatrix(idx, filename)
- saveSimple(idx, filename)

[1] Requires METIS to be installed. Refer to: http://glaros.dtc.umn.edu/gkhome/views/metis

- saveDot(idx, filename)
- savePNG(idx, filename, color) [2]
- saveEditDistancePNG(idx, filename, localEditDistance, maxDistance) [2]

System Functions

- INIT
- CLEAR

[2] Requires graphviz (neato) to be installed. Refer to: http://www.graphviz.org