

---

**Algorithm 1** Takes a graphml file of all nodes and edges. Loads the graphml into networkx to obtain adjacency matrix. Plots scatterplots of each region with edges. Two main graphs are produced: graph produced by input graphml and other is graph with the regions presented by average points and connected by nearest Euclidean distance. Saves output as a new graphml file.

---

**Input:** A graphml of datapoints and edges.

**Output:** Plotly plot of nodes with connectivity paths.

```

1: function CONNECTIVITY(dictionaryregions)
2:   Read in graphml           ▷ read graphml and prepare to load into plotly scatter plot objects
3:   Initialize  $A = nx.adjacency$            ▷ generate adjacency matrix from networkx
4:   obtain  $D, V = LA.eig(A)$            ▷ eigen decomposition
5:    $D = np.diagflat(D)$            ▷ flatten diagonal matrix D
6:    $b = np.matrix(V) * np.matrix(D) * np.matrix(LA.inv(V))$            ▷ compare original A with
   decomposed
7:   Initialize dictionary of regions           ▷ looping through points and save to dictionary
8:   Initialize  $xe = []$            ▷ create empty list of x-coordinates for edges
9:   Initialize  $ye = []$            ▷ create empty list of y-coordinates for edges
10:  Initialize  $ze = []$            ▷ create empty list of z-coordinates for edges
11:  Initialize  $data = []$            ▷ create empty list of data points for plotly
12:  Initialize  $avgpt = OrderedDict()$            ▷ create empty dictionary of region:average point pairs
13:  for  $key$  in  $dictionaryregions$  do           ▷ loop through all of the regions in the input dictionary
14:    Initialize  $x = []$            ▷ create empty list of x-coordinates for nodes
15:    Initialize  $y = []$            ▷ create empty list of y-coordinates for nodes
16:    Initialize  $z = []$            ▷ create empty list of z-coordinates for nodes
17:    for  $coordinate$  in  $dictionaryregions[key]$  do           ▷ for each coordinate in region
18:      Append x,y,z values to respective lists           ▷ add x,y,z to lists
19:    end for
20:     $avgdict[key] = avgpt$            ▷ take average of x,y,z coordinate values and append to
   dictionary
21:    Create  $tracescatter = Scatter3d$            ▷ create plotly 3d scatterplot object
22:    Create  $avgscatter = Scatter3d$            ▷ create plotly 3d scatterplot object for avg points
23:    for  $key$  in  $enumerate(avgdict)$  do           ▷ for each region
24:      for  $j$  in  $range(len(avgdict))$  do           ▷ for each other region
25:        if  $i == j$  then           ▷ ignore comparisons of same region
26:          continue
27:        end if
28:         $dist = DISTANCE(ptA, ptB)$            ▷ compute distance
29:         $distlist.append(dist)$            ▷ append distances to a list
30:      end for
31:    end for
32:     $min(dist)$            ▷ Find closest region
33:  end for
34:   $AppendedgedatatoplotlyScatterobject$            ▷ Create Plotly Scatter plot object with edges
35:  plot
36: end function

```

---