Pre-cluster

……

Main C2G

Initiate\_Empty\_Queue;

Create\_Root\_Node;

Queue.push(Root Node);

While Queue is not empty:

Node\_id = Queue.pop();

Best\_Entropy = inf;

Best\_strategy = [];

For all marker pairs:

Compute\_Best\_Gate();

Compute\_Entropy();

If entropy < Best\_Entropy:

Best\_Entropy = entropy;

Best\_strategy = Current\_gates;

End

If Best\_strategy is not []:

Create\_Children\_Nodes;

Queue.push(All New Nodes);

End

End

Compute\_entropy

Compute\_Best\_Gate

Set: Grid\_Size;

Compute: x\_interval and y\_interval;

Grid\_X = (x-min(x))/x\_interval;

Grid\_Y = (y-min(y))/y\_interval;

Hash\_key = Grid\_X\*100+Grid\_Y

Grid = unique(Hash\_key);

Initiate Num\_of\_Cell\_Grid;

%% Above is a n-by-m matrix. n is total number of grid and m is total number of cell population. It’s used to record number of different cells in different grid.

For each cell:

Add 1 to right place in Num\_of\_Cell\_Grid;

End

Compute Adjacent Matrix;

%% overlap between 2 population is. Where pi is number of cells in population i out of all cell in population i.

MCL clustering.

%% Each MCL cluster is one gate we want to draw.

For each MCL cluster:

If it has no target population, skip;

Compute grid convex hull of the cluster;

%%The union of above convex hull is the gate;

End

Compute\_convex\_hull

%% Here, we want to ignore certain percentage of outlier cell. But the outlier level is unknown. We want to try different values.

Best\_f\_score = 0;

Best\_Boundary = [];

For different ignore percentage:

Boundary = [];

For Each population in the cluster:

Drop small grids;

Compute Convex hull;

Boundary = Union(Boundary, Convex\_hull);

End

Compute F\_score;

If F\_score > Best\_f\_score:

Best\_f\_score = F\_score;

Best\_Boundary = Boundary;

End

End