**Main()**

{

readinSGAMatrix():

read in SGA matrix and save into intervMatrix

readinCompleteMatrix ():

Two interfaces are provided in getting complete matrix

1) read in driver activation state and DEG matrix separately

readinDriverActState ():

read in activation state of drivers and save into combinedMatrix

readinDEGMatrix ():

read in DEGMatrix and append to combinedMatrix

combinedMatrix = driverActState + DEGmatrix

2) read in complete matrix all at one time

read in complete matrix and save to combinedMatrix

readInEdges():

read in edge information, save to edgeList

buildNetwork():

create nodeList

add parent and children to each node

calculateCPTforEachNode()

while true:

calculateJointProbForAllNodes () --> JointProbAll

if not converged (change of JointProbAll > 0.1%)

getInferStates()

else

output inferActStateMatrix

output combinedMatrix

output jointProbAll

output CPT

break

}

**Data Structure:**

**Class Node:**

int index

string name

char type

int[] children

int[] parents

double[] CPT1 : only save P(A=1|Pa), P(A=0|Pa)=1-P(A=1|Pa)

calculateCPT(combinedMatrix)

**Class Data:**

int[] intervMatrix

int[] combinedMatrix ( inferredActState + DEGmatrix )

string[] edgeList

Node[] nodeList

calculateCPTforEachNode()

getInferStates()

calculateJointProbForAllNodes()

**Functions in Class Data:**

**Data::calculateCPTforEachNode():**

{ foreach node

Node.calculateCPT(combinedMatrix)

}

**Data::getInferStates()**

{

Foreach tumor t

Foreach inferredAct A

{ inferA = inferActivation (t,A)

sample a random number r

if r > inferA set action of A to 0

else set A to 1

update A’s CPT and A children’s CPT

}

}

**inferActivation (t, A)**

{

check SGA state:

if ( SGA == 1 )

set inferA to 1

else ( SGA == 0 )

inferA = calculateProbA (t, A)

return inferA

}

**calculateProbA (t, A)**

{

get the parent of A

find the value of A and A’s parents in combinedMatrix

lookup the CPT value of A : CPT\_A for A=1, A=0

get the children of A

foreach children Ac

get the parent of Ac

find the value of Ac and Ac’s parent in combinedMatrix

lookup the CPT value of Ac : CPT\_Ac for A=0, A=1

inferA = 1 / { 1 + exp [ log ( P(A=0)/P(A=1) ) ] }

while log ( P(A=0)/P(A=1) ) = log[P(A=0)] - log[P(A=1) ]

log [ P(A=0)] = log [CPT\_A(A=0)] + ∑ log [CPT\_Ac (A=0) ]

log [ P(A=1)] = log [CPT\_A(A=1)] + ∑ log [CPT\_Ac (A=1) ]

return inferA

}

**Data::calculateJointProbForAllNodes()**

{

In combinedMatrix

foreach tumor

foreach Node

P(A|Pa(A) = CPT(A)

lg [ JointProbAll ] += lg [ P(A|Pa(A) )

}

**Functions in Class Node:**

**Node::calculateCPT (combinedMatrix)**

{

foreach case

get the node and its parent value in combinedMatrix

count the combination of those column values:

N = numOfparents

If NodeValue == 0

Count0 [ ] ++

else

count1 [ ] ++

CPT1[ ] = count1[ ] / ( count0[ ] + count1[ ] )

}

Note: In code implementation, readinMatrices, readinEdges and buildNetwork are implemented in the constructor of Data class. In the main() code, it only creates an object of Data class.