**TDI iteration framework**

The main code runs in bashScript. It involves 5 modules with modification of TDIC, and 10 files.

**5 modules**: generateGlobalDriver.exe (c++)

TDICm.exe (c++)

extractTriplets (python)

initializeGlobalDriverMatrix (python)

updateGlobalDriveMatrix (python)

TDICm is modified version of TDIC.

10 files: 4 input files, 2 output files, and 4 intermediate files.

**4 input files**: SGAmatrix

DEGmatrix

globalPriors

tumorSpecificPriors

**4 intermediate files**: globalDriverList ( single SGA )

TDI results (number of tumors files)

Triplets

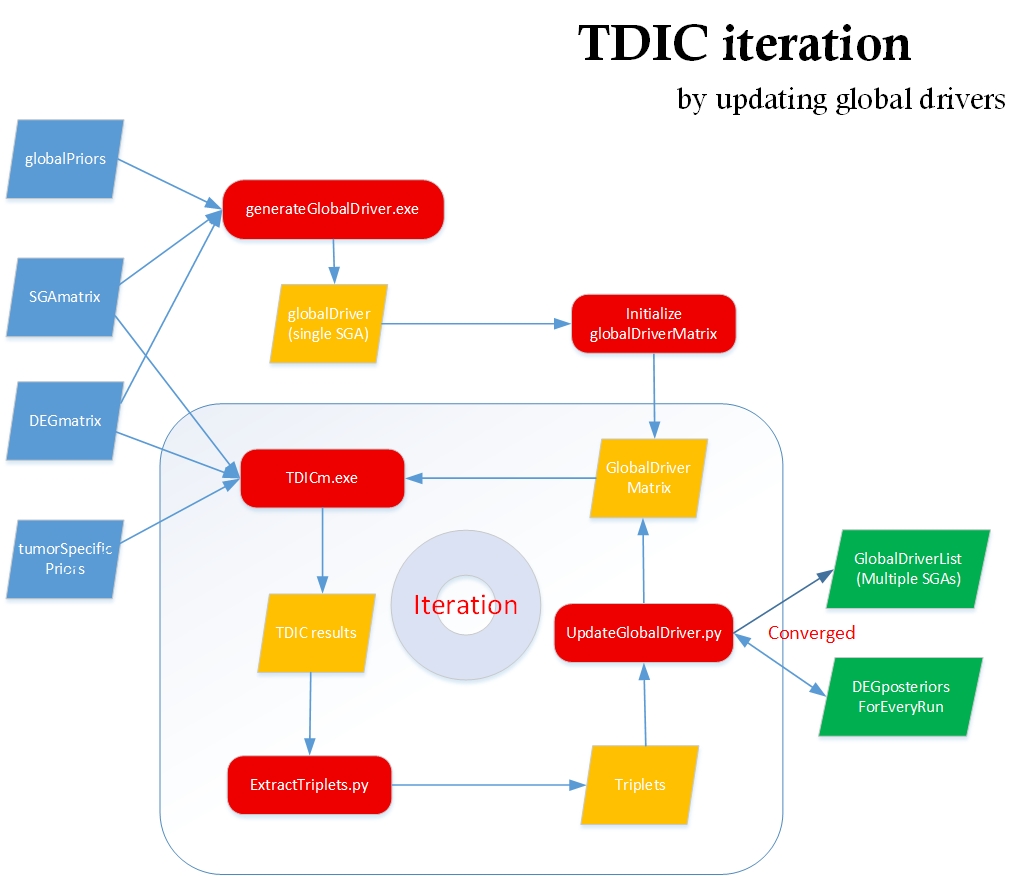
globalDriverMatrix

**2 output files**: gloabalDrverList (multiple SGAs)

DEGposteriors

.

See below for the flow chart



**Modification of TDIC:**

if SGA == 0

look up the global driver value in globalDriverMatrix

**Pseudo\_code**

**Main**

Run generateGlobalDriver.exe to get globalDriverList

Run initGlobalDriverMatrix.py to covert globalDriverList to globalDriverMatrix

Create a file DEGposteriors. The file will contain a map of DEG to posterior for every run

Create a file DEGtoSGAs. The file will contain a map of DEG to SGAs for the last run

count = 0

while true

count ++

Run TDICm.exe to get TDI results

Extract TDItriplets from TDI results

UpdateGlobalDriverMatrix ( count )

Determine if all DEGs are coverged, if so, break

**initializeGlobalDriverMatrix**:

foreachDEG e in DEGMatrix

Find the driver d of e in globalDriverList(singleSGA)

Add the column e into globalDriverMatrix with the content of the column d in SGAmatrix

**updateGlobalDriverMatrix ( count )**{

read in DEGposteriors, get the avePosterior of the last run for every DEGs

foreach DEG e in TDItriplets

if the avePosterior of the last run for this DEG < 1 (not converged)

get the average of the posterior of e in triplets -> avePosterior

get the top count(1..) frequency SGAs of e -> topSGAs

if avePosterior increase < 0.05 ( e is converged)

add 1 to this DEG’s avePosteriors in file DEGpostriors

add topSGAs to this DEG’s in file DEGtoSGAs

else

add avePosterior to this DEG’s avePostriors

updateGlobalDriverMatrixOfoneDEG ( e , topSGAs )

**}**

**updateGlobalDriverMatrixOfoneDEG (deg, topSGAs )** {

Combine topSGAs columns in SGAmatrix using 'or' operator to generate a new column c

Update column deg of globalDriverMatrix with c

}