tugHall version 2.0: USER-GUIDE-tugHall

Requirements for tugHall simulation:

R version **3.3** or later

libraries: **stringr**

Note that the program has two different procedures in general: the first is the simulation and the second is the analysis of the simulation results. Please, pay attention that the requirements for these procedures are **different**. This User-Guide pertains to the **simulation procedure** alone.

Table of Contents

1. Quick start guide

The simplest way to run tugHall:

- Save the **/tugHall_2_clones/** directory to the working folder;
- Run **tugHall_2_clone.R**.

The code has its initial input parameters and input files in the **/Input/** folder. After the simulation the user can see results of the simulation (please, see **User-Guide-Analysis_2** for details) in the dialogue box, which will save to the **/Output/** and **/Figures/** folders. Note that the analysis procedure requires additional libraries and a higher version of R - 3.6.0.

2. Structure of directories

Root directory:

User-Guide-tugHall_2.Rmd - user guide for simulation in the Rmd format.

User-Guide-tugHall_2.html - user guide for simulation in the html format.

User-Guide-tugHall_2.pdf - user guide for simulation in the pdf format.

User-Guide-Analysis_2.Rmd - user guide for analysis and report generation in the Rmd format.

User-Guide-Analysis_2.html - user guide analysis and report generation in the html format.

User-Guide-Analysis_2.pdf - user guide analysis and report generation in the pdf format.

dir **/tugHall_2_clones/** - directory that contains the program.

/tugHall_2_clones/ directory:

tugHall_2_clone.R - program to run a simulation and define the parameters.

dir **/Code/** - folder with the code and the function library.

dir **/Input/** - folder with the input files.

dir **/Output/** - folder with the output files.

dir **/Figures/** - folder with the plot figures.

/Code/ directory:

CanSim.bib, pic_lic.jpg - files necessary files for the user guide.

tugHall_clone_functions.R - file that contains the functions for the simulation / core of program.

Analysis_clones.R - file to analyze the results of a simulation and plot figures.

Functions_clones.R - file with the functions for the analysis of results.

/Input/ directory:

cloneinit.txt - file with a list of initial cells with/without destroyed genes.

gene_cds2.txt - file with hallmark variables and weights.

/Output/ directory:

cloneout.txt - file with simulation output.

geneout.txt - file with information about hallmark variables and the weights.

log.txt - file with information about all parameters.

Weights.txt - file with information about weights between hallmarks and genes.

Order_of_dysfunction.txt - see **USER-GUIDE-Analysis**.

VAF.txt - see **USER-GUIDE-Analysis**.

/Figures/ directory

In the **/Figures/** directory there are figures in *.jpg format, which appear after the analysis of the simulation results. See **USER-GUIDE-Analysis_2**. :

3. Inputs

Input of hallmark variables and gene weights

The file **tugHall/Input/gene_cds2.txt** defines the hallmark variables and weights (only first 10 lines are presented here):

Table 1. Input file for genes. Example of input file for hallmarks and weights in the file ***tugHall_2_clones/Input/gene_cds2.txt***.

Genes	length CDS	Hallmark	Suppressor or Oncogene	Weights
APC	8532	apoptosis	s	0.2616483
APC	8532	growth	s	0.3285351
APC	8532	invasion	s	0.3746081
KRAS	567	apoptosis	o	0.2099736
KRAS	567	growth	o	0.2881968
KRAS	567	immortalization	o	0.4735684
KRAS	567	angiogenesis	o	0.3525394
KRAS	567	invasion	o	0.0446472
TP53	1182	apoptosis	s	0.2543523
TP53	1182	growth	s	0.3076387

1. **Genes** - name of gene, e.g., TP53, KRAS. The names must be typed carefully. The program detects all the unique gene names.
2. **length CDS** - length of CDS for each gene, e.g., 2724, 10804.
3. **Hallmark** - hallmark name, e.g., “apoptosis”. Available names:
   - apoptosis
   - immortalization
   - growth
   - anti-growth
   - angiogenesis
   - invasion

Note that “growth” and “anti-growth” are related to the single hallmark “growth/anti-growth”. Note that “invasion” is related to “invasion/metastasis” hallmark.

4. **Suppressor or oncogene.** - Distinction of oncogene/suppressor:

   - o: oncogene
   - s: suppressor
   - ?: unknown (will be randomly assigned)

5. **Weights** - Hallmark weights for genes, e.g., 0.333 and 0.5. For each hallmark, the program checks the summation of all the weights. If it is not equal to 1, then the program normalizes it to reach unity. Note that, if the gene belongs to more than one hallmark type, it must be separated into separate lines.

After that, the program defines all the weights, and all the **unknown weights** are set equal to 0. Program performs normalization so that the sum of all weights should be equal to 1 for each column. The **tugHall/Output/Weights.txt** file saves these final input weights for the simulation. Only the first 10 lines are presented here:

Table 2. Weights for hallmarks. Example of weights for hallmarks and genes from ***tugHall/Output/Weights.txt*** file. Unknown values equal 0.

Genes	Apoptosis, H_a	Angiogenesis, H_b	Growth / Anti-growth, H_d	Immortalization, H_i	Invasion / Metastasis, H_{im}
APC	0.2565501	0.0000000	0.2709912	0.0000000	0.4445540
KRAS	0.2058822	0.3264502	0.2377183	0.4735684	0.0529836
TP53	0.2493962	0.3715365	0.2537549	0.5264316	0.0765560
PIK3CA	0.2881715	0.3020133	0.2375356	0.0000000	0.4259064

1. **Genes** - name of genes.
2. **Apoptosis,** H_a - weights of hallmark “Apoptosis”.
3. **Angiogenesis,** H_b - weights of hallmark “Angiogenesis”.
4. **Growth / Anti-growth,** H_d - weights of hallmark “Growth / Anti-growth”.
5. **Immortalization,** H_i - weights of hallmark “Immortalization”.
6. **Invasion / Metastasis,** H_{im} - weights of hallmark “Invasion / Metastasis”.

Input the probabilities

The input of the probabilities used in the model is possible in the code for parameter value settings, **“tugHall_2_clones.R”**:

Probability variable and value	Description
E0 <- 2E-4	Parameter $E0$ in the division probability
F0 <- 1E0	Parameter $F0$ in the division probability
m <- 1E-6	Mutation probability m'
uo <- 0.5	Oncogene mutation probability u_o
us <- 0.5	Suppressor mutation probability u_s
s <- 10	Parameter in the sigmoid function s
k <- 0.1	Environmental death probability k'

Filename input

Also in the code **“tugHall_2_clones.R”** user can define names of input and output files, and additional parameters of simulation:

Variables and file names	Description

genefile <- 'gene_cds2.txt'	File with information about weights
clonefile <- 'cloneinit.txt'	Initial Cells
geneoutfile <- 'geneout.txt'	Gene Out file with hallmarks
cloneoutfile <- 'cloneout.txt'	Output information of simulation
logoutfile <- 'log.txt'	Log file to save the input information of simulation
censore_n <- 30000	Max cell number where the program forcibly stops
censore_t <- 200	Max time where the program forcibly stops

Input of the initial clones

The initial states of cells are defined in **"tugHall_2_clones/Input/cloneinit.txt"** file:

Clone ID	List of mutated genes	Number of cells
1	""	1000
2	"APC"	10
3	"APC, KRAS"	100
4	"KRAS"	1
5	"TP53, KRAS"	1
…	…	100
1000	""	10

1. **Clone ID** - ID of cell, e.g., 1, 324.
2. **List of mutated genes** - list of mutated genes for each cell, e.g. "", "KRAS, APC". The values are comma separated. The double quotes ("") indicate a cell without mutations.
3. **Number of cells** - number of cells in each clone, e.g., 1, 1000.

---

4. Outputs

The output data consists of several files after the simulation. The "log.txt" and "geneout.txt" files contain the input information about variables and gene names. "Weights.txt" has information about the weights of genes for hallmarks (Please refer the section "Inputs"). "Cellout.txt" has information about the dynamics of cell evolution and all variables.

"log.txt" file

The file **“log.txt”** contains information about probabilities and file names. These variables are explained in the “Inputs”.

Table 3. log.txt file. Example of log.txt file.

Variable	Value
genefile	Input/gene_cds2.txt
clonefile	Input/cloneinit.txt
geneoutfile	Output/geneout.txt
cloneoutfile	Output/cloneout.txt
logoutfile	Output/log.txt
E	0.001
F	10
m	1e-05
uo	0.5
us	0.5
s	10
k	0.2
censore_n	30000
censore_t	100

“geneout.txt” file

The file **“geneout.txt”** contains input information about the weights that connect the hallmarks and genes, which are defined by the user. These variables also are explained in the “Inputs”.

Table 4. geneout.txt file. Given below is an example of the geneout.txt file.

Gene_name	Hallmark_name	Weight	Suppressor_or_oncogene
APC	apoptosis	0.2565501	s
KRAS	apoptosis	0.2058822	o
TP53	apoptosis	0.2493962	s
PIK3CA	apoptosis	0.2881715	o
KRAS	immortalization	0.4735684	o
TP53	immortalization	0.5264316	s
APC	growth|anti-growth	0.2709912	s
KRAS	growth|anti-growth	0.2377183	o
TP53	growth|anti-growth	0.2537549	s
PIK3CA	growth|anti-growth	0.2375356	o

“cloneout.txt” file

The file **“cloneout.txt”** contains the results of the simulation and includes the evolution data: all the output data for each clone at each time step (only the first 10 lines are presented):

Table 5. Output data. Example of output data for all clones. The names of columns are related to the description in the Tables 1,2 and *USER-GUIDE-Analysis_2*’s figures. Columns are from 1 to 15.

Time	N_cells	AvgOrIndx	ID	ParentID.Birthday	c.	d.	i.	im.	a.	k.	E.	N	Nmax.	M
0	-	avg	-	-	0	0	1	0	0.0066929	0.2	0.001	1000	1000	0
0	1000	1	1	0:0	0	0	1	0	0.0066929	0.2	0.001	1000	1000	0
1	-	avg	-	-	0	0	1	0	0.0066929	0.2	0.001	794	1000	0
1	794	1	1	0:0	0	0	1	0	0.0066929	0.2	0.001	794	1000	0
2	-	avg	-	-	0	0	1	0	0.0066929	0.2	0.001	626	1000	0
2	626	1	1	0:0	0	0	1	0	0.0066929	0.2	0.001	626	1000	0
3	-	avg	-	-	0	0	1	0	0.0066929	0.2	0.001	483	1000	0
3	483	1	1	0:0	0	0	1	0	0.0066929	0.2	0.001	483	1000	0
4	-	avg	-	-	0	0	1	0	0.0066929	0.2	0.001	388	1000	0
4	388	1	1	0:0	0	0	1	0	0.0066929	0.2	0.001	388	1000	0

1. **Time** - the time step, e.g., 1, 50.
2. **N_cells** - the number of cells in this clone, e.g. 1000, 2.
3. **AvgOrIndx** - “avg” or “index”: “avg” is for a line with averaged values across different (index) lines at the same time step; “index” shows the cell’s index at the current time step, e.g., avg, 4,7.
4. **ID** - the unique ID of clone, e.g., 1, 50.
5. **ParentID.Birthday** - the first number is the parent ID, the second number is the birthday time step, e.g., 0:0, 45:5.
6. **c** - the counter of cell divisions for the clone.
7. **d** - the probability of division for the cell, e.g., 0.1, 0.8.
8. **i** - the probability of immortalization for the cell, e.g., 0.1, 0.8.
9. **im** - the probability of invasion/metastasis for the cell, e.g., 0.1, 0.8.
10. **a** - the probability of apoptosis for the cell, e.g., 0.1, 0.8.
11. **k** - the probability of death due to the environment, e.g., 0.1, 0.8.
12. **E** - the E coefficient for the function of the division probability, e.g., 10^4, 10^5.
13. **N** - the number of primary tumor cells at this time step, e.g., 134, 5432.
14. **Nmax** - the theoretically maximal number of primary tumor cells, e.g., 10000, 5000.
15. **M** - the number of metastasis cells at this time step, e.g., 16, 15439.

Continuation of Table 5. Columns are from 16 to 22.

Time	N_cells	AvgOrIndx	Ha	Him	Hi	Hd	Hb	type	mut_den
0	-	avg	0	0	0	0	0	0	0
0	1000	1	0	0	0	0	0	0	0
1	-	avg	0	0	0	0	0	0	0
1	794	1	0	0	0	0	0	0	0
2	-	avg	0	0	0	0	0	0	0

2	626	1	0	0	0	0	0	0	0
3	-	avg	0	0	0	0	0	0	0
3	483	1	0	0	0	0	0	0	0
4	-	avg	0	0	0	0	0	0	0
4	388	1	0	0	0	0	0	0	0

16. **Ha** - the value of the hallmark “Apoptosis” for the cell, e.g., 0.1, 0.4444.
17. **Him** - the value of the hallmark “Invasion / Metastasis” for the cell, e.g., 0.1, 0.4444.
18. **Hi** - the value of the hallmark “Immortalization” for the cell, e.g., 0.1, 0.4444.
19. **Hd** - the value of the hallmark “Growth / Anti-growth” for the cell, e.g., 0.1, 0.4444 .
20. **Hb** - the value of the hallmark “Angiogenesis” for the cell, e.g., 0.1, 0.4444 .
21. **type** - the type of the cell: “0” is primary tumor cell, “1” is the metastatic cell, e.g., 0, 1.
22. **mut_den** - the density of mutations (tumor mutation burden) for the cell, e.g., 0, 0.32.

The columns from 23 to 26 are related to names in the form **PosDriver. *gene name***, where ***gene name*** is related to user defined genes. The number of columns equals the number of the genes. These columns show the position(s) of driver mutation(s) in a gene: the first number is the mutational site on the gene and the second number is the time step of the mutation, e.g., 3493:4, 4531:34.

Continuation of Table 5. Columns are from 23 to 26.

Time	N_cells	AvgOrIndx	PosDriver.APC	PosDriver.KRAS	PosDriver.TP53	PosDriver.PIK3CA
0	-	avg				
0	1000	1				
1	-	avg				
1	794	1				
2	-	avg				
2	626	1				
3	-	avg				
3	483	1				
4	-	avg				
4	388	1				

23. **PosDriver.(Gene_1=“APC”)** - for the first gene.
24. **PosDriver.(Gene_2=“KRAS”)** - for the second gene.
25. **PosDriver.(Gene_…)** - …
26. **PosDriver.(Gene_last=“PIK3CA”)** - for the last gene.

The columns from 27 to 30 are related to names in the form **PosPassngr. *gene name***, where ***gene name*** is related to user defined genes. The number of columns equals the number of the genes. These columns show the position(s) of **passenger** mutation(s) in a gene: the first number is the mutational site on the gene and the second number is the time step of the mutation, e.g., 8952:43, 531:4.

Continuation of Table 5. Columns are from 27 to 30.

Time	N_cells	AvgOrIndx	PosPasngr.APC	PosPasngr.KRAS	PosPasngr.TP53	PosPasngr.PIK3CA
0	-	avg	-	-	-	
0	1000	1				

1	-	avg	-	-	-
1	794	1			
2	-	avg	-	-	-
2	626	1			
3	-	avg	-	-	-
3	483	1			
4	-	avg	-	-	-
4	388	1			

27. **PosPassngr.(Gene_1="APC")** - for the first gene.
28. **PosPassngr.(Gene_2="KRAS")** - for the second gene.
29. **PosPassngr.(Gene_...)** - …
30. **PosPassngr.(Gene_last="PIK3CA")** - for the last gene.

---

5. How to run

In order to make the simulation, please follow the procedure:

1. Copy **/tugHall_2_clones/** directory into the working directory.
2. CD to the **/tugHall_2_clones/** directory.
3. Run the **tugHall_2_clone.R** file, using the command line like

```
R --vanilla < tugHall.R
```

or using the line by line procedure in **R Studio**. In this case we have:

- `load library(stringr)` and `source(file = "Code/tugHall_clone_functions.R")`;
- create the Output and Figures directories, if needed;
- define the simulation parameters;
- make the input file for initial cells, if needed;
- run the *model()* function to simulate;
- run `source("Code/Analysis_clone.R")` in order to analyze the results and plot the figures in the dialogue box (see **User-Guide-Analysis_2**).

4. To obtain analysis reports of the simulation, please refer to **User-Guide-Analysis_2.RMD**. In **User-Guide-Analysis_2.RMD**, commands are embedded to include files under **Output/** and **Figure/**. So, after analysis with tugHall, you can generate analysis reports automatically from **User-Guide-Analysis_2.RMD**. For more details, please refer to "Writing reproducible reports in R" on the github (https://nicercode.github.io/guides/reports/).

6. Difference with cell-based code

6.1. Reason to develop clone-based code

- Clone-based code was designed to accelerate calculation and increase number of cell. Advantage

of clone-based algorithm is making trial for all cells at 1 clone with one application of **trial()** function. In cell-based algorithm **trial()** apples to each cell. But if number of cells equal number of clones, then speed up is 1. That's why clone-based code works faster for any cases.

- Another reason is a case, when we need to simulate huge number of cells like 10^7 or 10^9, but mutation rate is very low. Cell-based algorithm takes a huge computational cost, and vice verse clone-based algorithm will work very fast, if mutated cells will appear slowly.

6.2. Usage of *trial()* function

- In **trial()** function program apples several trials like enveronmental death, apoptosis death, division process, etc. We changed the trials with probability p (for some death process) for each cell in the clone with for 1 trial with procedure:

$$N_{cells} = N_{cells} - Binom(p, N_{cells}),$$

where $Binom(p, N_{cells})$ is random generation for the binominal distribution, N_{cells} is a number of cells in a clone. Probability p is one of probability of death process, for example $p = a'$ or $p = k$ etc.

- For cell division with probability d' the new number of cells will be:

$$N_{cells} = N_{cells} + Binom(d', N_{cells})$$

- Check at the end of **trial()** function: if $N_{cells} = 0$, then the clone has died.

6.3. Usage of mutation function

- In mutation function we have changed the mutation to birth of a new clone (one mutation is a birth of one clone):

$$N_{new\_clones} = Binom(m, N_{new\_cells}),$$
$$N_{new\_cells} = Binom(d', N_{cells}).$$

- Passenger or Driver mutations do not matter for new clone's generation. Only during analysis, we will distinguish Passengers or Drivers clones.

6.4. Average function

- The average values $\bar{x}$ of probabilities or hallmarks are found by summation on the clones x_i with multiplication by cells number $N_{cells,i}$ of this clone:

$$\bar{x} = \frac{\sum_i x_i \cdot N_{cells,i}}{\sum_i N_{cells,i}},$$

where summation applies for all clones $i = 1..N_{clones}$.

- For this purpose, we added the calculation of cells number (primary and metastasis) before average and hallmarks update.