

GenMut.pl and GenMutCfg.pl

Aim

MBSS_MutBnd.pl and MBSS_MutBndCfg.pl are perl scripts that generate a new bnd file (and a new cfg file for MBSS_MutBndCfg.pl). These new files contain external variables that control under/overexpression of nodes. The script MBSS_MutBndCfg.pl prepares both bnd and cfg files for simulating mutants. By default the external variables controlling mutations are set to 0. To simulate the mutant of a particular gene, the corresponding external variable needs to be set to 1.

How to run the script

The scripts are run as a command line. The list of nodes on which mutations can be applied needs to be provided. The bnd file (and the cfg) needs to be provided as well.

```
MBSS_MutBnd.pl <file.bnd> "<node_list>"
```

```
MBSS_MutBndCfg.pl <file.bnd> <file.cfg> "<node_list>"
```

Outputs

The generated files have the “_mut” suffix, e.g., “file_mut.bnd” and “file_mut.cfg”. The name of an external variable that controls overexpression of a node has the prefix “\$High_” (e.g., node “N” has “\$High_N” associated external variable). For underexpression, the prefix is “\$Low_N”.

The list of nodes, for which external variables are added, is given in the command line. The standard output provides the nodes that has been found in the bnd file. If these variables are set to 1 (in the cfg file), the state of the node is forced, no matter the initial condition. This is realized by an almost instantaneous transition, with a maximum rate given by the “max_rate” node variable in the new bnd file. The underexpression variable is prioritized against the overexpression variable.

NB: the “max_rate” is the maximum rate possible in c++, divided by the number of nodes. If the user is using such a large rate in another place, MaBoSS run could produce wrong results because addition of large rate may overflow c++ maximum number.