Quasi Steady State Petri Net (QSSPN) software manual.

This document describes QSSPN model building and running of simulation in MUFINS. We will describe how to build QSSPN model graphically in Snoopy Petri Net editor and run simulations in command line interface. Subsequently, we will describe integration of Snoopy and JyMet GUIs, which is major **new feature** of MUFINS. The Snoopy file with connectivity of the Petri Net part of the model can be uploaded to JyMet. Transition parameters as well as objective and constraint places connecting Petri Net to metabolic model can now be edited within JyMet. Simulations can be then run from JyMet and trajectories can be examined with JyMet plotting functions.

Material is organised as follows. In Chapters 1 and 2 we will describe QSSPN input and control files. This will introduce parameters used to describe models and simulations. In Chapter 3 we will describe QSSPN software distribution within MUFINS. Model building with Snoopy will be described in Chapter 4 and command line interface to qsspn simulation engine in chapter 5. The JyMet GUI to QSSPN will be described in Chapter 6.

1. QSSPN file format

The QSSPN file specifies the Dynamic Transition part of the model as well as constraint and objective places connecting Dynamic Transitions to Quasi Steady State Fluxes. First, the list of PN places is defined in the following format:

```
SUBSTANCES
  Name init max type
  …
END
```

Each line between `SUBSTANCES` and `END` specifies one PN place. The fields have the following meaning:

`Name` – node name.
`init` – initial number of tokens associated with the node.
`max` – the maximal number of tokens.
`type` – place type, 1 – common place, 2 – objective place, 3 – constraint place.

Subsequently, the text between tags `REACTIONS` and `END` specifies PN transitions. Each transition is specified by the text between `INTERACTION` and `END` tags:

```
INTERACTION
   SUBSTRATE Name rate type delay
      ACTIVITY n
          t1    a1
          …
          tn    an
      END
   END
   PRODUCT Name
   CONSUMED Name
END
```

Each `SUBSTRATE` and `PRODUCT` tag specifies a PN pre-place or post-place, respectively. The `CONSUMED` tag specifies which of the pre-places change state upon firing of the transition. When

transition fires it removes tokens from pre-places specified as `CONSUMED` and adds tokens to post-places. Thus, activators and inhibitors feature as transition pre-places, but are not specified as `CONSUMED`. The `ACTIVITY` section is associated with particular pre-place and provides a lookup table with contribution of the pre-place to transition propensity function defined in chapter 2.1. The `ACTIVITY` table with 0 rows:

```
ACTIVITY 0
END
```

indicates that a particular pre-place has mass-action activity function in the transition (chapter 2.1). This is a **new feature** implemented to support integration of quantitative models.

The variables have the following meaning:

`Name` – the name of PN place.
`rate` – transition rate.
`type` – transition type, `SLOW` (stochastic), `INSTANT` (immediate), `FAST` (continuous), `FLUX` (flux, **new feature**), `RESET` (reset, **new feature**).
`delay` – Applies to SLOW (stochastic) transitions only. If delay is greater than 0 transition is treated as delayed with delay time set by this parameter.
`t, a` – Threshold and propensity function contribution. If the state of the pre-place node is greater or equal to t in particular row of the table and smaller than t in next row than the pre-place contribution to propensity function is set to a.

The text between `GSMN` and `END` describes connectivity between Dynamic Transitions (regulation) and Quasi Steady State Fluxes (metabolism). `MODEL` tag can be used to specify the path to MUFINS reaction table file with the QSSF model. Settings of `MODEL` tag are overridden by the value of MODEL tag in the control file. The `EXT_TAG` specifies externality tag in MUFINS file. This is the string at the end of metabolite name, which indicates external (unbalanced) metabolites in FBA model.

The mapping of flux transition to a particular QSSF flux is defined in the following way

```
FLUX transition flux
```

where, transition is a transition name in the `INTERACTION` section of qsspn file and flux is the reaction name in MUFINS reaction table file. The transition type must be set as `FLUX` in the `INTERACTION` section. The flux transition must have objective place as the only pre-place and must be connected to only one post-place.

The objective place is specified by `METABOLITE` tag:

```
METABOLITE Name Objective
      ACTIVITY k
           t1    n1
           …
           tn    nk
      END
END
```

where `Name` is the name of PN objective place and `Objective` is the name of flux or metabolite in the QSSF. The activity table maps real values of linear programming optimisation of QSSF objective to integer number of tokens.

The constraint node is specified by `ENZYME` tag:

```
ENZYME Name Flux
      ACTIVITY k
         n1          lb1   ub1
         …
         nk          lbk   ubk
      END
END
```

where `Name` is the name of constraint node and `Flux` is the flux in the QSSF, which is constrained. The activity table maps an integer number of tokens n into lower (lb) and upper (ub) flux bound in QSSF.

Alternatively, the activity table can be applied to a list of fluxes rather than a single flux. This enables modelling of enzymes capable of catalysing multiple reactions in metabolic network. Instead of the flux name the keyword "list" is used. The list of m fluxes is provided between `LIST` and `END` tags. Each flux name is associated with weight w to further increase flexibility of expressing enzyme-reaction relationship. Each flux on the list is assigned bounds determined by activity table and multiplied by the weight.

```
ENZYME Name list
      ACTIVITY k
         n1          lb1   ub1
         …
         nk          lbk   ubk
      END
 LIST m
         Flux1 w1
         …
         Fluxm wm
      END
 END
```

2. QSSPN control file format

The HepatocyteQSSPN.ctrl.txt file can be edited with text editor. It specifies parameters of the simulation. The file has the following format:

```
MODEL path
NUMBER_OF_SAMPLES N
SEED seed
TIME_MAX tmax
MAXIMAL_TIMESTEP timestep
MAX_CHANGE change

OUTPUT out
LOG log
MONITOR m
```

```
PROGRESS progress

TARGET name t

INITIAL_STATE
name value
…
END

PROPENSITY_FUNCTION name
formula
END

RESET_FUNCTION name
formula
END

PETRI_NET_MONITORS
name
…
END

SIMULATION
```

The `path` specifies file containing metabolic network in MUFINS reaction table format. On some Linux distributions, the full path to the file may be required. The parameter `N` indicates the number of independent trajectories starting from the same initial conditions. The `seed` is the seed of random number generator. The `tmax` is the maximum simulation time. The trajectory stops after maximal time is reached. The maximal timestep parameter of the QSSPN algorithm is set to `timestep`. The change is a **new feature**. It is a maximal change of place marking in adaptive timestep Euler algorithm used in fireContinuousTransitions function. It is defined in chapter 2.8, here it suffices to say that value of 0.01 indicates 1% change of place marking.

The `INITIAL_STATE` and `END` tags enclose initial marking of PN places. The `name` is the place name and `value` is its marking at the start of simulation. This **new feature** has been introduced to allow specification of continuous place marking by real numbers rather than integers. The Snoopy editor interface supports only integer marking.

The major **new feature** of the QSSPN is use of algebraic formulas to define transition propensity functions and reset functions. The `PROPENSITY_FUNCTION name`, `END` lines enclose algebraic formula used to calculate propensity function of transition "`name`". The formula uses pre-place names as variables. Parameters must be introduced as values. Alternatively, parameters may be introduced as pre-places connected to the transition with read edges. The parameter values may be then specified in `INITIAL_STATE` section. The pre-place names can be linked with "+", "-", "*", "/" and "^" (power) operators. Brackets can be used. In current implementation we do not use functions. The `RESET_FUNCTION, END` tags enclose formula, which is used by `RESET` transition to calculate value to which marking of its post-places will be set.

The state of the system is recorded after each `m` iterations of QSSPN algorithm. Trajectory is written in tab separated text format to the file specified by out path. The names of PN nodes to be monitored in the trajectory file need to be listed between `PETRI_NET_MONITORS` and `END` tags. Each name should be written in a separate line. The log file reports completion of each trajectory and can be used to monitor progress of the simulation. The log file also contains statistics of

trajectory sampling. If the `PROGRESS` tag is specified the trajectory number and simulation time are printed to standard output after each `progress` iterations (**new feature**).

The QSSPN trajectory samples are analysed to calculate the number of trajectories that exhibit certain behaviour. If behaviour of interest can be expressed as a reachability problem (certain node reaching certain state) the TARGET tag can be used to facilitate calculations. In the log file the QSSPN software reports the number of trajectories, where target node specified by name reaches threshold of t. The trajectory is stopped after these conditions have been met. If the user wishes to calculate sample of trajectories which are not stopped before tmax and analyse them later, a dummy target that can never be reached needs to be specified (any PN node with threshold larger than its maximal number of tokens). For example in HepatocyteQSSPN.ctrl.txt we set:
TARGET Chenodiol_r 100000

The `SIMULATION` indicates the function implementing QSSPN algorithm. Currently only this function is available, but in future versions we may implement functions performing more complex simulation protocols (e.g. place inactivation scans).