Holmes 1.1

User Manual

The following document is both Holmes manual and an introduction to the Petri net theory. All modules have been described with the corresponding Petri net theory elements.

Poznan University of Technology

Institute of Computing Science

Marcin Radom Ph.D.

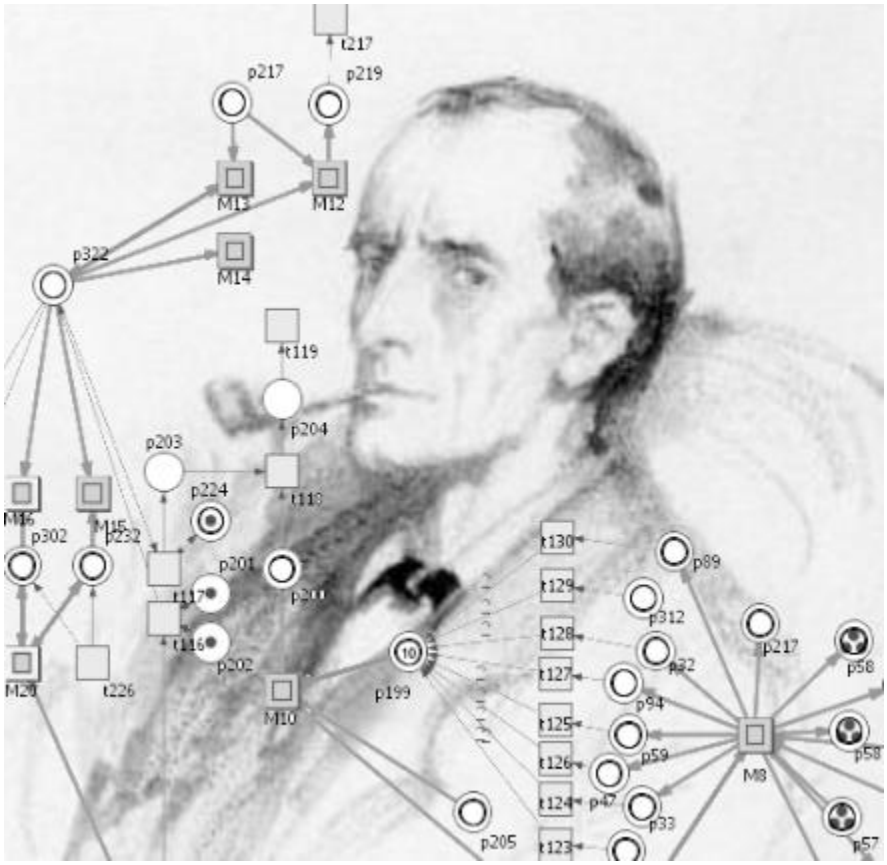
Bartłomiej Szawulak M.Sc.

01.03.2022

HOLMES

Integrated Petri Nets Environment

Version: 1.1 (March 2022)



Index

1 Introduction

1.1 Holmes - history

In its first version the program was a Bachelor Thesis (Andrzejewski, Chabelski i Szawulak, 2013) realized in 2012-2013 in the Faculty of Computing Science, Poznań University of Technology. Three students were responsible for the development in those years: : Hubert Andrzejewski, Piotr Chabelski and Bartłomiej Szawulak, the supervisor has been prof. Piotr Formanowicz. The program has been further developed as a Master Thesis (2013-2014) (Szawulak, 2014), with supervisor Marcin Radom Ph.D. Since December 2014 it has extended with many other analytical modules by Marcin Radom.

Program is written in Java (1.7, 1.8). Additionally it can use independent tool INA (*Integrated Net Analyzer*) (Starke, 1992) as t/p-invariants generator and the scripts in R language for the cluster analysis. The R environment is necessary for Holmes cluster computations, while INA is not, due to existence of Holmes own invariants generator.

The following free libraries are used as part of the program:

Sanaware JavaDocking (GNU GPL, http://www.javadocking.com/)

jXLS library (GNU LGPL, http://jxls.sourceforge.net/)

XStream (BSD License http://xstream.codehaus.org/license.html)

Simple-xml (Apache Licence, http://simple.sourceforge.net/home.php)

RCaller (GNU LGPL, https://code.google.com/p/rcaller/)

jFreeChart (GNU LGPL - Lesser General Public Licence)

JCommon (GNU LGPL - Lesser General Public Licence)

exp4J (Apache Licence, http://www.objecthunter.net/exp4j/license.html)

1.2 Manual

The program still undergoes changes and modifications. When new modules are being added, some existing interface elements may change. The parts of the program described in this manual which are known to undergo changes in the near future will be marked with the following icon.

For the moment there is no option to Undo changes and modifications to the net structure in the program. The modification of the net / analytical computations results of which are irreversible (i.e., they can only be reversed by loading previous project file) will be marked with the following icon. The program usually warns the user about such changes.

This icon represents sections with good advices about using some features of the program.

A reference to other sections of the manual.

This manual covers all of the program features, but may not be 100% accurate due to recent changes and future additions. In case of some serious problems, crashes or other issues with the tool the email is as follows: marcin.radom@put.poznan.pl

1.3 Common abbreviations

- **PN** *Petri Net*
- **P**, **T** *Place*, *Transition*
- **TPN** *Time Petri Net*
- **DPN** *Duration Petri Net* (so called ***Timed*** *Petri Net*)
- **FPN** *Functional Petri Net*
- **LMB** *Left Mouse Button*
- **RMB** *Right Mouse Button*

1.4 Requirements

- Processor **Intel i5**, **2Ghz**, min. 2 cores. Recommended: **Intel i7**,
- **4 GB RAM minimum**. RAM memory is more important than the CPU requirements. Some analytical modules may require 1-2GB of memory. Cluster computations, specifically Calinski-Harabasz metric calculations may require many GB of RAM depending on the number of t-invariants (e.g., in a case when there are more than few thousand of them). **These requirements are mostly for the analytical modules only** and the algorithms performing some complex computations. Drawing or modifying Petri net requires minimum memory and CPU times.
- Java Runtime Environment version 1.7 / 1.8 or higher is required.
- For cluster analysis R Language is required with some additional libraries (details in further manual chapters).
- Minimum resolution for the screen is 1376x768. Recommended: 1680x1024 or higher.

2 Interface

In this chapter main windows and its sub-windows will be described. It has been divided into 6 parts called **Sections**. **Section** 1 is menu, **Section** 2 – toolbar, etc. In later parts of the manual there will be references to some subwindows/tabs of the main windows, such as e.g., "tab Cluster is Section 6". **Such reference always refer to the main window and its six parts**.

2.1 Program main window

The main window of Holmes is given in the following Figure 2.1.

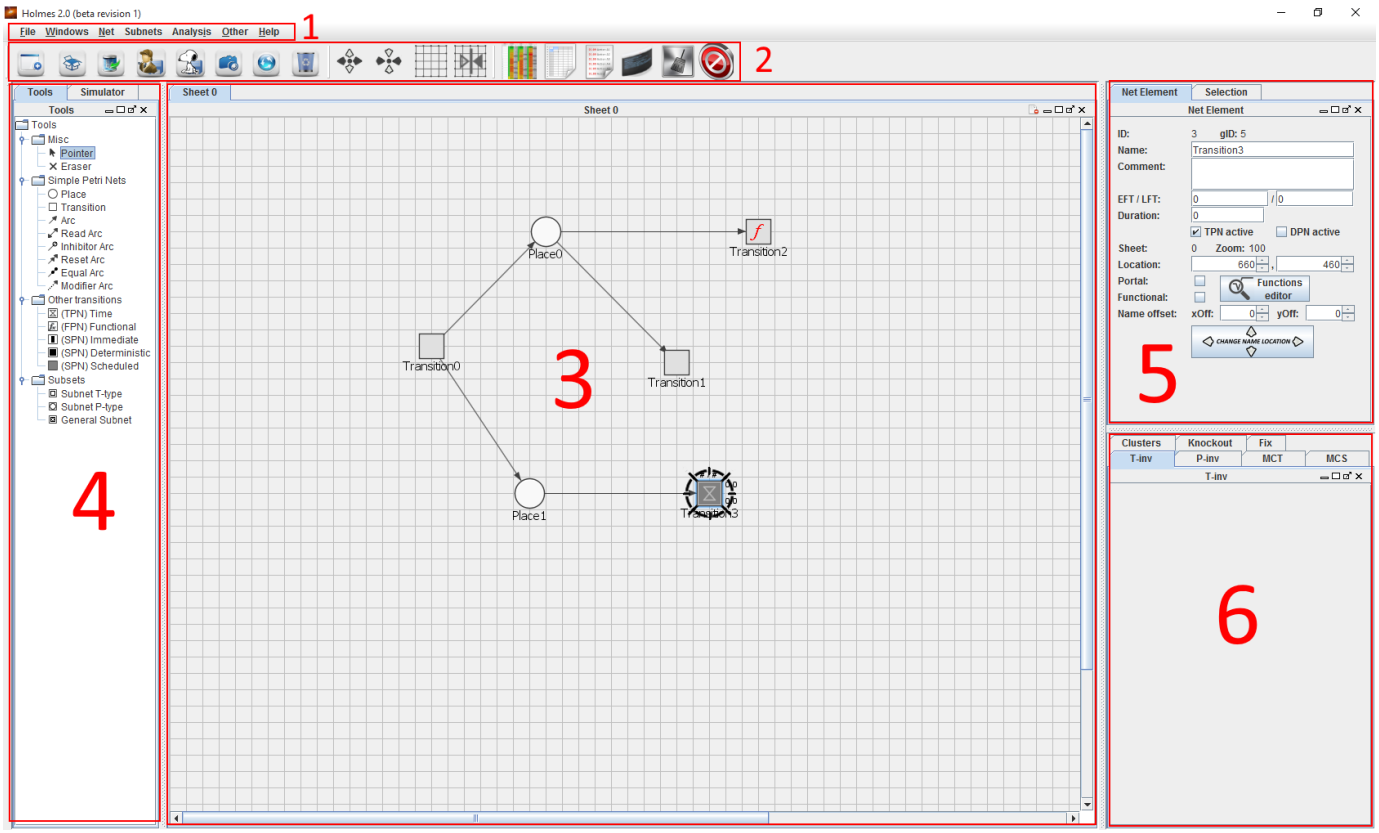


Figure 2.1. Program window

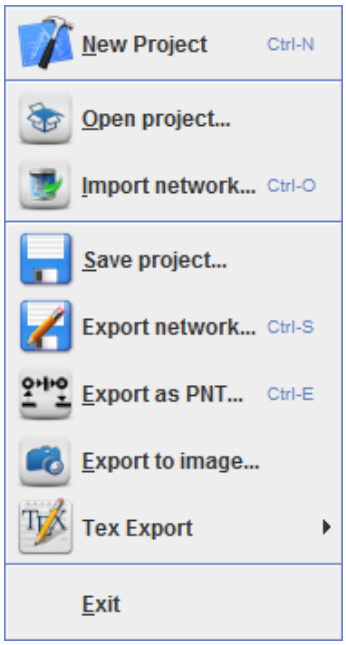
In general main view is divided into 6 sections:

1 Menu bar
2 Toolbar
3 Main drawing window
4 Petri net components panel with simulation tab
5 Selected elements view
6 Panels for different analytical modules results

Most of the times where there is reference to some section or subwindow, the text concerns Section 6 of the main view and its different analytical modules tabs.

2.1.1 Menu bar

File – file operations.



New Project (**Ctrl+N**) – selecting this option will clear the current project from the program memory. In a case of some (detected) changes, a question window will pop up, asking if the project saving should be performed.

Open project… - main option for opening new project within Holmes (**.project** file). Detailed description of the project files will be given later, here it should be stated that this is the safest and recommended option for working with Petri net models within the tool. Other computed data like invariants, MCT sets, etc. will be saved in one project file as well.

Import network… (Ctrl+O) – Petri net from other sources, mainly from the Snoopy program, can be imported by using this option.

Save project… - this option will save the project files. There are three different project extensions (apart from **.project** there are also **.apf** and **.abyss** extensions). **It is strongly recommended to use .project** (the default one) format for saving given Petri net model.

Detailed description of the project components will be given in Chapter 3.

Export Network… (Ctrl+S) – this allows exporting the net into other format, mainly Snoopy files.

Export as PNT… (Ctrl+E) – fastest way of creating .pnt net file, used e.g., by INA program.

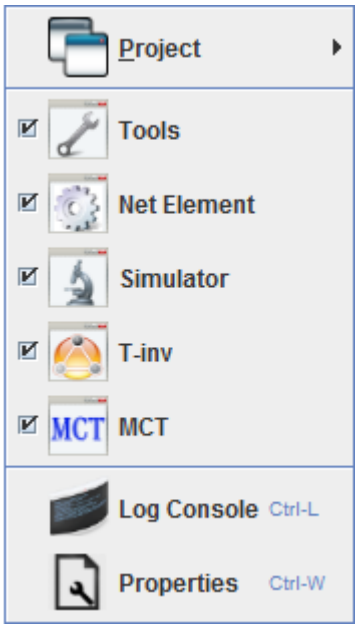
Export to image… - this will create a graphic .png file of the given net.

Tex Export – This will open sub-menu for various export modes into Tex files:

- **Places and transitions table…** - this will export tables of places and transition into Tex files, also a graphical image of the net will be saved as well.
- **Invariants table…** - invariants table export.
- **MCT table…** - MCT sets export.

Exit – this will close the program. In case of some detected changes performed on the net, a small windows will appear with a possibility of saving the project.

Menu ***Windows*** – options for adjusting Holmes subwindows and the Properties window for the whole program. **At its current version only real important option here is the Properties window.** Using different items in this menu is currently not advised (nor necessary).



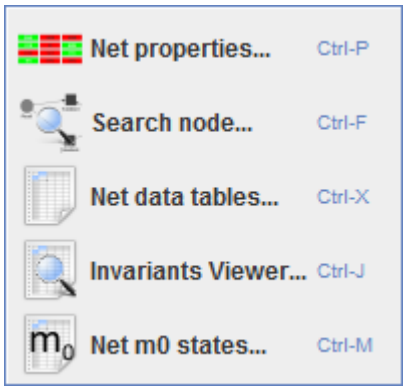
Project – a submenu with potential new sheets in which Petri net can be drawn.

Tools, Net Element, Simulator, T-inv, MCT – can be used to hide subwindows in Section 6 of the main window. It is advised not to use these options in the current version of Holmes.

Log Console (**Ctrl+L**) – window with various messages concerning analytical modules work. It can be also opened from the toolbar (Section 2 of the main window).

Properties (**Ctrl+W**) – The only important window in this menu, described in details in Chapter 7.

Net menu – from it a few data windows with information about the net can be opened. Their detailed descriptions will be given in Chapter 5 of the manual.



Net properties (Ctrl+P) – Petri net properties window.

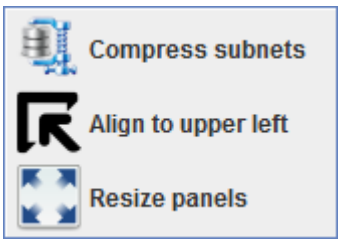
Search node… (Ctrl+F) – allows searching places / transition.

Net data tables… (Ctrl+X) – data window with tables concerning places, transitions and invariants.

Invariants Viewer… (Ctrl+J) – subwindow with detailed data about computed invariants.

Net m0 states… (Ctrl+M) – window for editing initial marking of the net.

Subnet – menu with a few options for adjusting graphical representations of nets which are drawn in different sheets of Section 4 of the main window.



Compress subnets – after importing coarse nets from Snoopy, some empty subnets can appear. This option will clear them.

Align to upper left – the net in the given sheet (Section 4 main window) will be aligned to the upper left corner of the screen.

Resize panels – it will refresh the default size of the sheets displaying

Petri net parts (if there is more than one net sheet).

Analysis – various analytical modules with their separate windows can be opened here.

Invariants generator… (**Ctrl+I**) – it will open window with invariant generator.

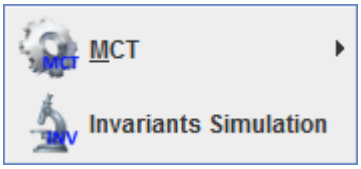
Minimal Cutting Sets… (**Ctrl+G**) – *Minimal Cutting Sets (MCS)* window.

Knockout analysis… (**Ctrl+K**) – knockout analysis window on the basis of t-invariants.

Cluster analysis… (**Ctrl+C**) – main window for working with the cluster analysis.

State Simulator… (**Ctrl+Q**) – main window of the simulator module.

Others – there are some development modules present in this menu, it is advised not to use them in the current version of Holmes.



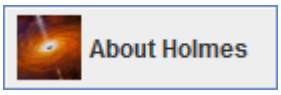
Some items may not function – a message box in such a case will appear.

MCT tab – contains basic options of exporting MCT data for further analysis:

- **Generate MCT groups** – MCT sets are generated automatically as soon as the t-invariants set becomes available in the project. Clicking this can only refresh the MCT sets, but it is no longer necessary.
- **Create Simple MCT file** - simple txt file with MCT sets.
- **MCT files** tab – inactive, to be developed with option in 2017.

Invariants Simulation - temporary inactive.

Menu **Help** – the only available option now is the information window about the program.



About Holmes – window containing data about the program, its development versions, authors and used libraries.

2.2 Toolbar (Section 2)

Section 2 of the main window is a toolbar. On it there are button for the most common functions and tools connected with Petri net drawing and edition.

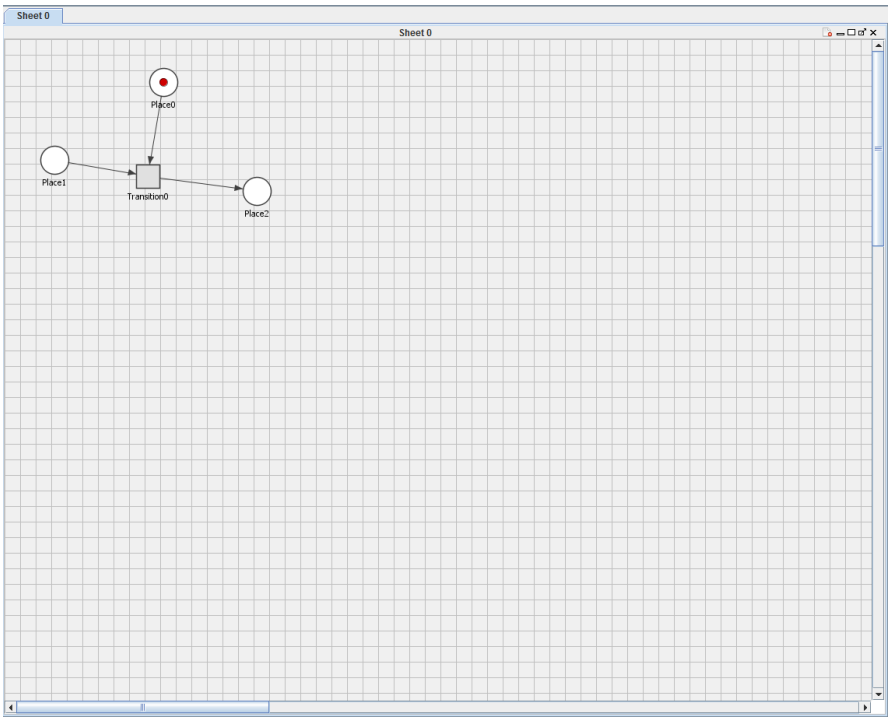
Default toolbar is given below:

In general button can be divided into 3 sets: A – common operations, B – editor grid options, C – auxiliary tools.

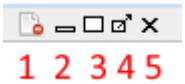
- Set A – main operations, mostly concerning files and described already in Chapter 2.1
  - **A1** - opening of new drawing sheet, in current version on Holmes this button is inactive, cf. hierarchical networks (later in the manual)
  - **A2** - **opening of the project file (.project)**
  - **A3** - importing net file from other tools (Snoopy, INA)
  - **A4** - **saving the project (.project)**
  - **A5** - net export to other file formats (np. Snoopy)
  - **A6** - creation of current net picture (png, jpg, bmp)
  - **A7** - refreshing of the current net sheets (currently does nothing)
  - **A8** - new project / clearing current project data
- Set B – editor grid
  - **B1** - extending dimension in the current net, i.e., for every net component (place, transition) its coordinates (x,y) are multiplied by 1.1 (10%).
  - **B2** - shrinking dimension, similar as before, each (x,y) coordinate is multiplied by 0.9 **ATENTION: THIS IS NOT ZOOM** . These buttons make permanent changes to the net elements coordinates (however, due to the fact that this is performed to the whole net, the relations between elements remain the same).
  - **B3** - showing / hiding grid lines in the net editor (Section 3)
  - **B4** - alignment of (x,y) coordinates of the net elements to the grid. In simple words, every x and y coordinate is aligned to the nearest multiplicity of 20. E.g., x=17 is aligned to 20, y=45 to 40, etc. for every net element.
- Set C – additional icons
  - **C1** - cluster analysis window.
  - **C2** - net tables for places, transitions and invariants.
  - **C3** - simulator log window
  - **C4** - Holmes log/console window
  - **C5** - removal of temporary color patterns assigned to the net elements, due to, e.g., coloring invariants, MCT sets, clusters, etc.

Button **C6** – **IMPORTANT:** Due to the fact that the tool is still being developed, in this or further version there may be some „development/debug" buttons marked by this icon. If such an icon is visible, it should be ignored, because depending on the algorithm assigned to it can influence the project data.

2.3 Main editor area (Section 4)



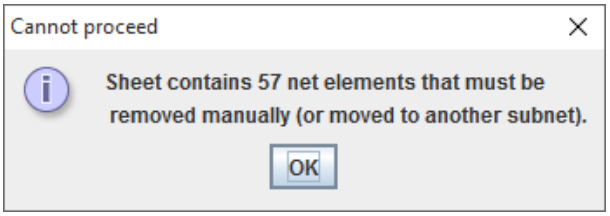
This is the most commonly used area in Holmes, in which the net is drawn, edited, etc. Default state for this area is that, in which only Sheet 0 (as in the picture) is present and active. Other sheets may appear if hierarchical, structured net is being created (they can be selected, if present, from the menu **Windows** -> **Project** -> …). Elements placed in this area create the net and they are chosen from net tools subwindow (Section 3 of the main Holmes window). In the right top corner there are five icons:



When a cursor is pointing at any of these icons, a short description will appear.

These icons 1-5 are as follows:

- **1** - **Delete** – removing sheet from Section 4 of the main windows. It is not possible for Sheet 0. **ATTENTION**: in a structured net, this button in theory should force Holmes to REMOVE all elements from the net that have their graphical representations in the removed sheet. In the current version this feature is disabled, due to the fact, that hierarchical / structured net creation and analysis still undergoes development in Holmes. Pressing this button will results in similar window like:



In other words, in the current version of Holmes, the use must manually remove elements from the sheet (which can be done quite fast, as it will be explained later), and only for such an empty sheet this button will remove it (again: if it is NOT Sheet 0).

- **2** - **Minimize** - sheet minimization to the bottom panel which will appear below Sections 3, 4 and 6 of the main window. In such a state only described five icons will mark the minimized sheet, with button 2 replaced by small icon named **Restore**. Clicking it will undo the clicking of Minimize.
- **3** - **Maximize** – pressing this button will maximize the sheet to the whole screen. After that, second button will be replaced with **Restore** button to undo the maximization.
- **4** - **Externalize** – this button will create separate window for the sheet, outside of the Holmes Section 4 scope. Again, after that, **Restore** button will appear among five icons described here.
- **5** - **Close** - closing/**hiding** (**but no deletion!**) of a sheet. Using is currently not recommended. Closed sheet can be restored from the menu Windows -> Project -> … from menu bar.

2.4 Moving across the net sheet

Sheet (0 or others) contains the whole net or its elements. In a typical case not a whole net can be seen at once, given the size of the elements and their coordinates. There are vertical and horizontal bars for moving across the net sheet.

However, it is much easier to move across the net elements using mouse wheel and keyboard special buttons: **Shift, Control** and **Alt**. They will be explained below this paragraph.

2.4.1 Moving vertically and horizontally

The fastest way of moving the net is using mouse wheel with or without pressing the **Shift** button. **IF THE CURSOR IS ANYWHERE WITHIN THE SECTION 4 ARA (EDITOR)** using mouse wheel will move the net vertically. With the **Shift** button pressed, the mouse wheel will move the net horizontally.

2.4.2 Zoom in / out

This feature is connected with the **Ctrl** button and mouse wheel. Simply speaking pressing **Ctrl** button and moving mouse wheel up / down while the cursor is anywhere on the net subwindow will zoom in and out the net picture.

There is an option to reset zoom to neutral 100% size, by right clicking on the net subwindow and choosing **Fast Zoom Reset** from the context menu.

2.4.3 Fast moving into selected area

Pressing **Alt** button and clicking on the empty space in the net sheet will center the screen on this net region.

2.5 Additional information

Other sections and subwindow will be described later, in the chapter corresponding to their function.

2.6 Problems with windows

Holmes interface is created using Java library *JavaDocking* from *Sanaware*. Not all its functions are fully implemented and secured, which may lead to problems when trying to “redrawn” the main windows by, e.g., changing the subwindows position. It is strongly advised to leave the interface of the current version of Holmes in their default positions.

In case of problems, the program can be closed and opened again what will restore it to its default form. **It should be noted that it is impossible to remove or broke something on the net by changing something in the Holmes windows.** Even if the net sheets disappear, the net and the project will remain intact so Holmes can be then closed and opened in order to restore default view.

3 Project and supported files.

3.1 Supported net types

Shortcut	Name of the net	Files	Other data
PN	Petri Net	Snoopy (.spped), INA (.pnt), Holmes (.project)	Read arc from extPN can be used
extPN	Extended Petri Net	Snoopy (.spept), Holmes (.project)	Read arcs (double arcs), blocking arcs, reset arcs and equal arcs.
TPN	Time(d) Petri Net	Snoopy (.sptpt), Holmes (.project)	Two types: *Time Petri Net* oraz *Timed Petri Net* zwany też *Duration Petri Net* (**DPN**)
FPN	Functional Petri Net	Holmes (.project)	To every arc a function can be assigned.
SPN	Stochastic Petri Net	Holmes (.project) *	* feature still under construction. For the moment stochastic transitions can be created and stochastic simulations performed.
extFuncPN	Extended Functional Petri Net	Holmes (.project)	
extFuncTPN	Extended Functional Time(d) Petri Net	Holmes (.project)	
extSPN	Extended Stochastic Petri Net	Holmes (.project)	*more functions will be added later in 2017

3.2 Data types in project / program

This chapter can be used as reference after other chapters explaining analytical modules have been read. **Holmes Project** means the data is saved in the project file.

Reading/writing	Data and files
Petri net structure	
Table from chapter 3.1	Saving as project file is the safest way to store the studied model.
Invariants: places and transitions	
Holmes Project, .inv, .csv	Reading and writing file from programs: INA (Integrated Net Analyzer), Mona Lisa, Charlie and as a CSV file. Every invariant can be named, names will be stored in project file.
Cluster analysis data	
.hcl – Holmes CLuster file	File separated from project file.
T-invariant knockout analysis	
Reading/writing data is not necessary, import from MonaLisa is possible	Due to the fact that such an analysis is very fast, storing the data is not necessary. It is possible to import MonaLisa knockout result files.
Minimal Cutting Sets	
.objr – Objective Reaction single MCS file	Separate data file for MCS.
.mcs – MCS full data	Separate data file for MCS.

Initial states/markings	
Holmes Project	Many states can be stored in the project.
Stochastic net (*firing rates*)	
Holmes Project	*firing rates* for transitions can be stored (multiple vectors) in project file
SSA	
Holmes Project	Data for the Gillespie Stochastic Simulation Algorithm (when implemented in 2017) can be stored in project file
Knockout analysis by simulation	
.sim – Simulation Data	Separate file for computed data.
Others	
.txt (TeX table)	Export of various tables is possible in Tex tables format.
.txt (inne)	Many results from Holmes modules can be exported as text files.

4 Editor and Petri net creation

This chapter will begin with the theoretical introduction to the Petri nets theory. This theory will be mixed with the description of specific features of our program concerning the usage of given Petri net theory elements in the graphical environment Holmes provides for the user.

4.1 Petri nets in Holmes

The following list describes different type of Petri nets and the level of support for them in the program:

- **Petri Net** (classical PN) - For the moment the most supported type: invariants, MCT, MCS, cluster, knockout analysis, various simulation modes, etc.
- **PN with time** (TPN, DPN) – classical approaches works, additional time calculations, full simulation support.
- **Functional PN** – assigning functions to arcs, simulation support.
- **Stochastic PN** - basic simulation algorithms, firing rates manager, this type will be extended later in 2017.
- **Continuous PN** – not supported, development time: 2017-2018.
- **Hybrid (PN + continuous)** – same as above.
- **Hybrid (mixed types of supported PN)** – the ability of Holmes to draw and connect different transitions from supported Petri net into a single, working model.

4.1.1 Petri nets and their creation in Holmes

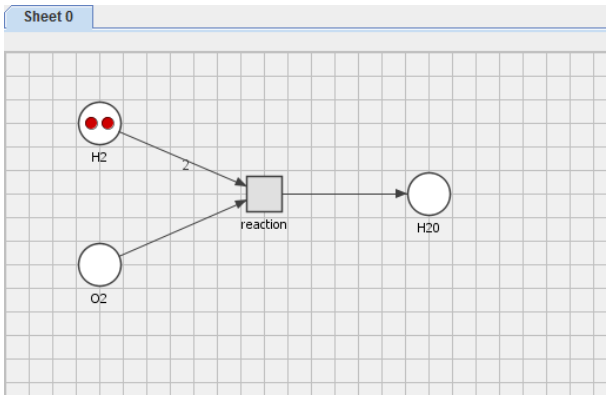
Petri net theory started with a Ph.D. thesis of Carl Adam Petri in 1962 „*Kommunikation mit automaten*" (Petri, 1962). The definitions in this chapter are later and they are adjusted in a form suitable for the examples presented in this manual.

Petri net, PN, is a 5-elements set $N = \{P, T, F, W, m_0\}$, where:

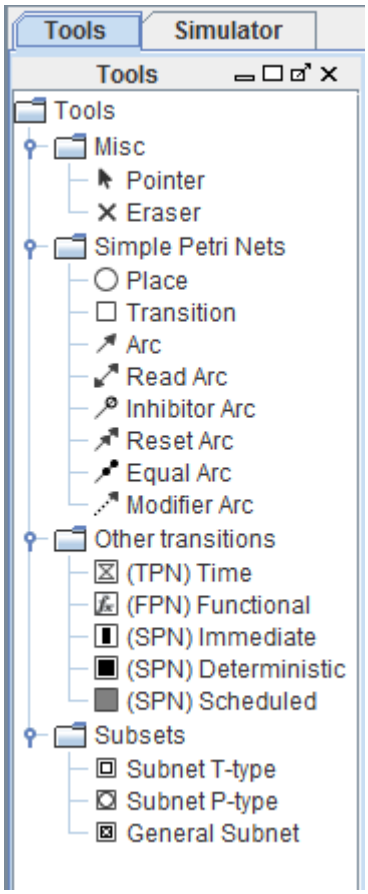
- P and T are disjunctive, finite sets of respectively places and transitions: $P \cap T = \varnothing$,
- $F \subseteq (P \times T) \cup (T \times P)$ defines a set of arcs,
- $W : F \rightarrow N$ assigns an integer value to every arc,
- $m_0 : P \rightarrow N_0$ is an initial state of the net.

Places and transitions are connected with directed arcs having specific weights, in such a way that **to** and **from** a place (transition) there can be arcs going *only* **from** and **to** a transition (place). In other words two places or two transitions can never be connected directly. In places there are so called *tokens* which are graphical representations of a value telling how much substance a given place represents is present in the net in a given moment (state). In the biological models places often correspond to chemical compounds while transitions correspond to the system reactions.

In the program net elements are presented in the form given in the following picture. In this example the net have 3 places and 1 transition. Place named H2 contains 2 tokens, representing two hydrogen compounds - H_2. There are no tokens in places O2 and H2O. Transition name is '*reaction*'.



Arc connecting place H2 with transition have weight equal to 2. Other weights are equal to 1, therefore they are not displayed.



In order to create any element of a Petri net, at first it must be chosen from Petri net tool subwindow (Section 4) and then placed by LMB clicking on any free area in Section 3 (editor).

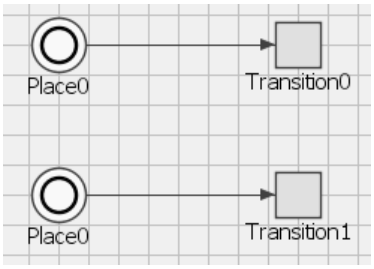
In order to return to a neutral cursor the user can click Pointer from tools subwindow, but much faster way is to simply click RMB on any free area in the editor sheet (Section 3). If by doing so a context menu appears (described later) it only means that the cursor has been already set to the neutral (Pointer) status.

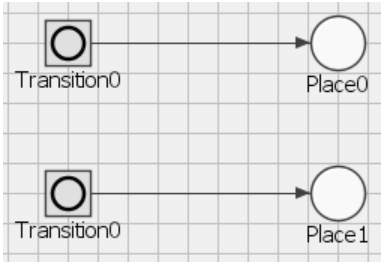
After selecting any drawing mode from tools section the cursor will change its icon. As long as the icon is present, LMB clicking will add another same element to the editor area.

Different net elements from tools section will be described later in this chapter.

In a case of a classical Petri net, standard elements like places, transitions and arcs suffice to create any net of this type. Here we will describe a different graphical elements, outside of a Petri net theory, however very useful in drawing net where arcs do not cross each other too often. These are

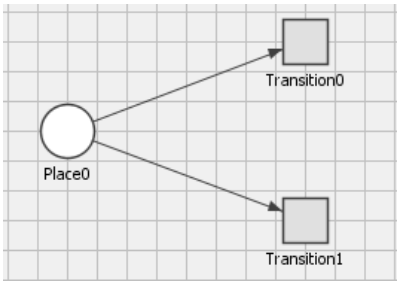
the so called *logical nodes* – *logical places* and *logical transitions* (Snoopy naming) or, as they are called in Holmes – *portals*, both for places and for transitions. Simply speaking, they are a **multiple graphical representations** of a **single** place or transition.

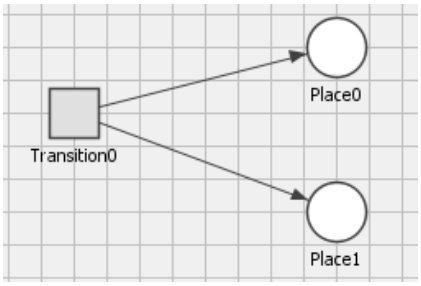




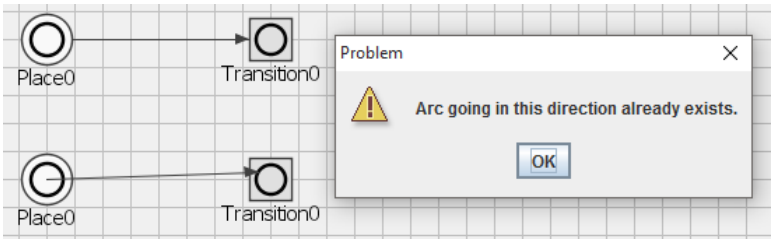
On the left picture Place0 is a portal (a logical place) – **single place** in the net **by definition**, but having two separate graphical representations in the editor. For example, if Place0 will have two tokens assigned, two dots will appear in both graphical representations – again: because it is in fact a single place from the Petri net definition perspective. The same situation is given on the right picture for the Transition0 – it is one transition having two separate graphical representations.

Two pictures below show **the same** nets as two picture above, but without portals.





The sole purpose of portals is to reduce the number of crossing arcs when drawing a complex net. In hierarchical nets they have additional purpose which will be explained later.



It should be noted that only one arc can go to/from a place from/to a transition. Using portals does not change this rule. Trying to create multiple arc, even using portals, will result in displaying a small window warning that such an action is impossible. In the picture such a situation is presented. The bottom arc is drawn towards portal of Transition0, however an arc from Place0 to Transition0 already exists (upper part of the picture). Because it is easier to miss when using portals, such safety feature has been implemented in our tool.

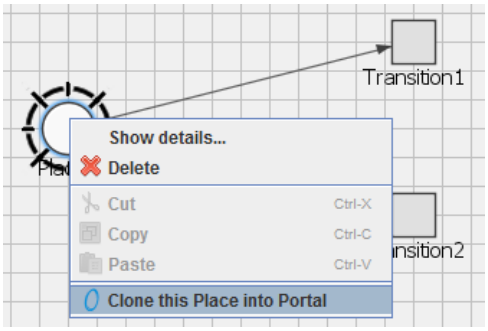
One way of creating a portal is possible by using a context menu. When clicking RMB on a place/transition, such a menu will appear. The option in question is:

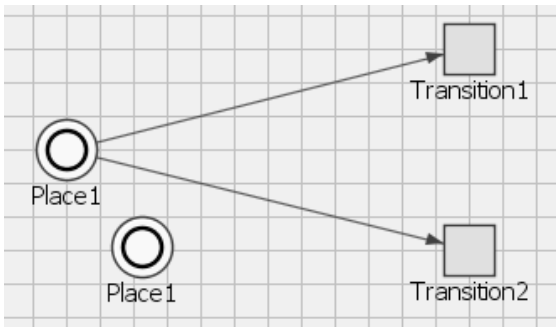
- **Clone this Place into Portal**

or in a case of a transition:

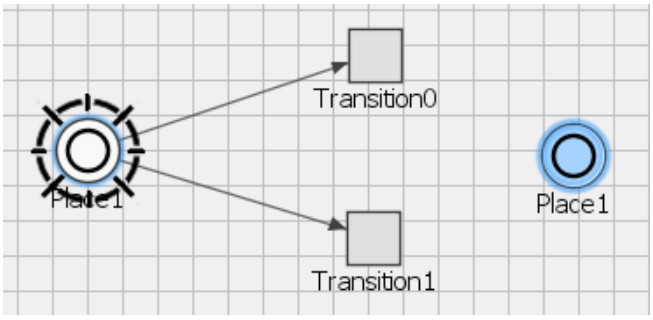
- **Clone this Transition into Portal**

The following pictures show the result of such action.





Place1 will change its graphical form and additionally another portal node for such a place appears. When selecting this option for already existing portal icon, another graphical node for a clicked portal will appear (in such a case for the described example it would be the third “Place1” graphical node).



When clicking / selecting a portal node, the clicked node will be mark with a crosshair, and other portal nodes **belonging to the same place/transition** will be drawn with a light blue color.

When selecting many portal at once (possible by pressing LMB and drawing a rectangle over some area of the net) will mark in blue all the selected portals – however in this case only their common names will allow the distinction between multiple portal-places and portal-transitions.

In the near future it will be possible to select **multiple different places / transitions** and change them into a common portal. This option is inactive at the moment (Holmes 1.0)

4.1.1.1 Sets of places and transitions

Petri net dynamic involves tokens, which in general are the object determining a state of a net. The components described so far are static ones. Details about the Petri net theory can be found, e.g., in (Murata, 1989). Before transitions activation and firing will be described, some preliminary definitions must be introduced.

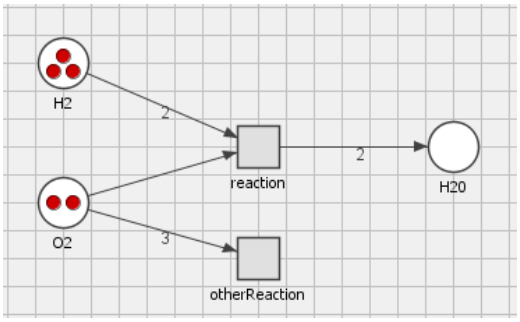
For a given transition *t* its set of input places (pre-places) is such a subset of set P, in which every place have at least one arc going to (directed into) *t.* Set of pre-places will be called •t .

Set of output places of transition *t* (set of post-places) is such a subset of set P, in which every place have at least one arc directed into them from the transition *t.* Set of post-places will be marked as t•

Analogously, sets of pre-transitions and post-transitions for a given place *p* can be defined, marked respectively as •p and p•

4.1.1.2 Transition enabling / activation

Transition *t* is called enabled (active) if in each place from set •t there are at least as many tokens as the weight of an arc connecting a given place with *t*.



Transition *reaction* is enabled, because in place H2 there are 3 tokens, more than the weight of an arc connecting it with *reaction* (which is equal to 2) and there are 2 tokens in place O2, which is also more than the weight of an arc connecting O2 with *reaction* (weight=1, not written by assumed convention).

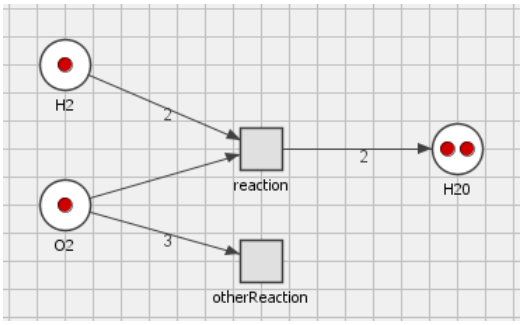
Transition *otherReaction* in not enabled. It requires **at least** 3 tokens in O2 to be enabled, while there are only 2 of them.

4.1.1.3 Transition firing

Enabled transition may (but do not have to) fire. In a simulation of a classical (non-stochastic) PN, the chance of firing for an enabled transition is assumed to be 50% in every analyzed simulation step. Firing of the transition takes the tokens from its pre-places (•t) in the number defined by weights of the arcs, and in the same moment produces tokens in all post-places (t•) in the number defined by the weights of the proper connecting arcs.

Firing the transition changes the state of the net, i.e., distribution of tokens in places. For the last example, assuming H2 is place p0, O2 – p1, H2O – p2, the initial state for the example is the 3-value vector (3, 2, 0) assigning number of tokens for every place.

For example, firing the transition *reaction* from the last example will change the state of the net as follows: $m_0 \rightarrow$[*reaction*] $\rightarrow m_1$. State m_1 is a vector (1, 1, 2), a net for this state after transition firing is given in the picture below.



Transition takes 2 tokens from H2, 1 from O2, and produce 2 tokens in H2O, all according to the rules that govern transition firing in a classical Petri net.

Both transitions in m_1 are inactive (i.e., they are not enabled) due to too small number of tokens in their pre-places.

4.1.1.4 Conflict of transitions

Two or more transitions are in a conflict if they compete for tokens in same shared pre-places. In the last example both transitions are in a conflict because they both share place O2 as their pre-place.

If there are less than 3 tokens in O2, only *reaction* can be enabled by this place (assuming there are enough tokens in H2). However, if there are 3 or more tokens in O2, both *reaction* and *otherReaction* become enabled. If there are exactly 3 tokens and *reaction* fires first, the *otherReaction* stops being enabled. On the other hand if in the simulation *otherReaction* fires first, *reaction* stops being enabled in the same moment. Therefore, for every simulator there is a need to shuffle enabled transitions in each step (i.e., in some internal list of transitions, from which enabled transitions are selected to be fired in a given step), in order to provide equal chances of firing for the whole simulation.

In the example, with 3 tokens in O2 it is not possible to fire both transitions – it would be possible if there were at least 4 tokens in O2. For the latter case, there are 4 possible scenarios in a simulation step:

- neither transition fires, O2 = 4 (4 – 0 tokens taken = 4)
- *reaction* fires, tokens in O2 = 3 (4 – 1 taken = 3)
- *otherReaction* fires, tokens in O2 = 1 (4 – 3 taken = 1)
- both transitions fire, tokens in O2 = 0 (4 – 1 – 3 taken = 0)

Such scenarios are possible in the default mode for the classical Petri net simulation, when there are 50% chances for firing. Later in this manual we will talk about other modes, e.g., ***Maximum mode*** in which every enabled transition must fire immediately (with respect to the number of tokens in pre-places). In such a mode only fourth explained scenario will be possible from the example.

Another simulation mode is such in which only one transition per step can fire. Such a mode is often typical in the stochastic simulations, it will also be explained later in greater details. For the classical PN such mode is called ***Single mode*** in Holmes (Chapter 6.5 in this manual).

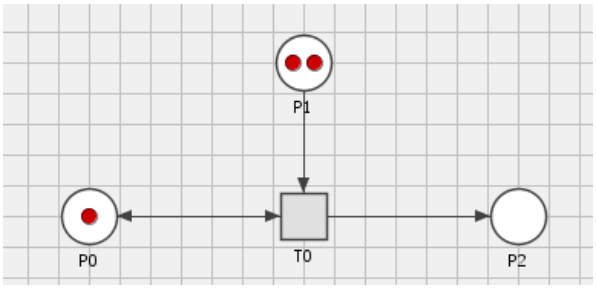
4.1.2 Extended Petri net

There are four additional types of arcs in this type of Petri net. Such nets can be created, e.g., in a Snoopy (Heiner, Richter i Schwarick, 2008). It should be noted that using them can sometimes significantly influence the possibilities of further net analysis, e.g., analysis based on t-invariants. Holmes support all these types of arcs.

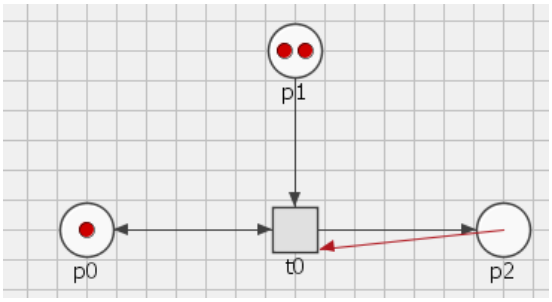
4.1.2.1 Read arc

This is the most common and popular type of an extended arc, which, in theory can be „created" using only arcs from a classical Petri net definition. In simple words such an arc is bidirectional – it is directed both into place and into transition it connects. Such an arc works in the following way: in order to enable a transition, a place connected by the read arc must have enough tokens (equal of more than the weight). When the transition fires, **it does not consume tokens** from a place connected by the read arc, nor produces them in this particular place.

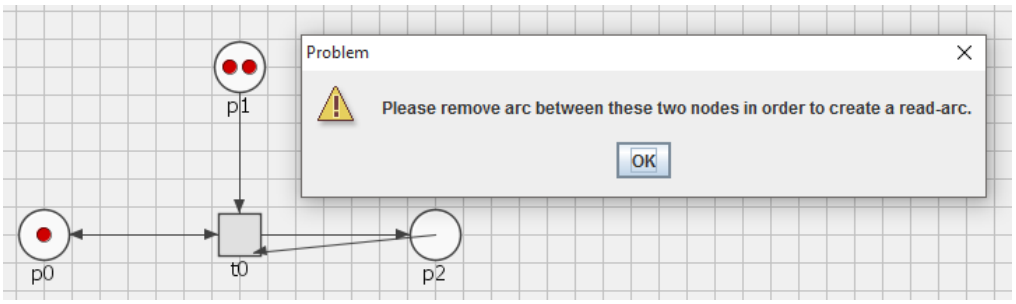
In other words however, one can also say that in the same moment a transition consumes tokens and produces the same amount of them in a place connected by a read arc – the final result is the same – the number of tokens in such a place does not change after transition firing.



Transition T0 is enabled, because both in P0 and P1 there are enough tokens (more than 1 in both cases). If T0 fires, it will only produces token in place P2, because P0 is connected with T0 by read arcs. On the other hand, 1 token will be taken from P1 in case of T0 firing.

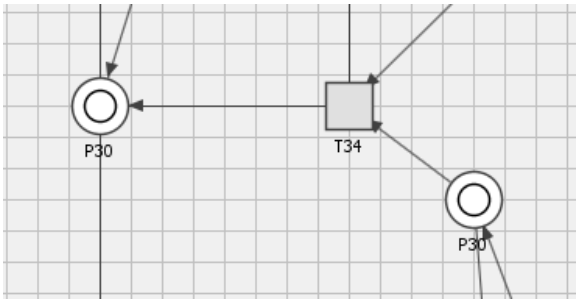


There are two ways for drawing read arc. One is to simply select it from Petri net tool subwindow, the second is to draw two normal arcs in opposite directions. There is a safety measure in Holmes for the case, when there is already a non-read-arc arc between place and transition, and the user choses read arc from net tools subwindow and try to draw it. In such a case a warning message will appear and the operation will not succeed:



In the above example there is already a normal arc from t0 to p2. The user **selected read-arc** from PN tools subwindow and try to draw it from p2 to t0. The message says that this cannot be performed (such a window would not appear if the user **kept the selection of a normal arc** – in that case two arcs leading in both direction would be automatically converted by Holmes into read arc).

ATTENTION! In literature there are two names for such arcs: read arc and double arc. They are not the same, i.e. read arc has a common single weight assigned to it, while double arc can have two separate weights for both ingoing and outgoing connection. Holmes allows this, cf. example below.

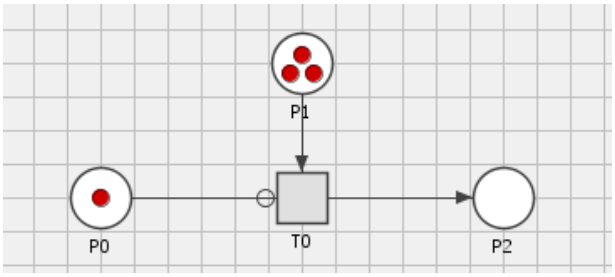


In this example, T34 is connected with P30 with a double arcs – P30 is given here as a portal. The left P30 node has arc going from T34, the bottom right node – into the T34. For these arcs two weights can be assigned – and by doing so this will create a double arc. The difference between double arc and read-arc is a little blurred however, therefore a caution is advised.

Read arcs impact on the t-invariants based analysis will be discussed later.

4.1.2.2 Inhibitor arc

This is a blocking arc, which sole purpose is to block the firing of a transition, when there is enough tokens in a place with such an arc.



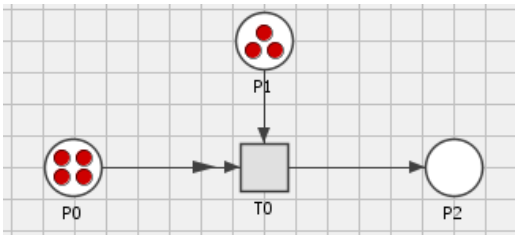
In the example, T0 is **not** enabled thus it cannot fire. There is of course more than enough tokens in P1 to enable T0, however, there is also 1 token in P0. This is enough to initiate the disabling ability of the inhibitor arc (we assume, that is weight is 1, therefore it is not drawn on the net picture). As long as there is **at least** one token in P0, it inhibits firing ability of T0 by its inhibitor arc.

NOTE: this kind of arc can only lead from place to the transition, never the opposite.

This type of arc is not 'visible' for the invariants generator. All the simulation modules however behave correctly in accordance to the inhibitor arc functions.

4.1.2.3 Reset arc

This type of arc **does not have weight**. In case of firing of a transition that has this type of arc going into it, this transition will consume all the tokens in the places connected with it by the reset arc.

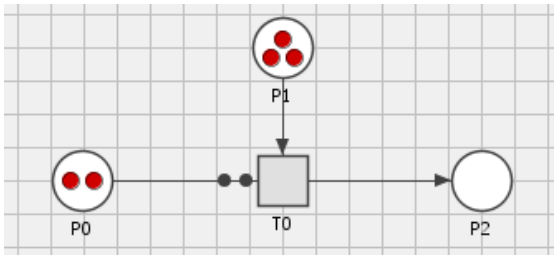


In the following example, T0 is active, reset arc leads from P0 to T0. If T0 fires, it will take 1 token from P1, and **all the tokens** from P0, then it will produce 1 token in P2.

NOTE: this kind of arc can only lead from place to the transition, never the opposite.

4.1.2.4 Equal arc

Last type of arc is an equal arc. Its function is as follows: transition connected by it with a place is considered enabled (with respect to that place) if and only if there is exactly as many tokens in such place as is the weight of an equal arc.



In the example T0 is not enabled. There are enough tokens in P1, however there are 2 tokens in P0, connected with T0 by an equal arc. The weight of this arc is 1, therefore T0 is not enabled. It would be only in a state, where in P0 there is exactly one token.

NOTE: this kind of arc can only lead from place to the transition, never the opposite.

4.1.3 Petri nets with time

A brief theory will be introduced in this section, details concerning Petri nets with time can be found e.g., in(Popova-Zeugmann, 2013).

For the beginning it should be noted, that there are two distinct types of Petri nets with time supported by Holmes at this time: *Time Petri Nets* (**TPN**) and *Timed Petri Nets* (*Duration Petri Nets*, **DPN**).

Holmes allows the „configuration" of the classical transitions to work in one of the two ways defined by TPN and DPN nets, but also **in both ways at once**. This especially concern the simulators: Holmes can simulate both TPN and DPN models, but also hybrid ones (later referred as TDPN). This connection is possible only to some degree, as it will be explained later.

4.1.3.1 Time Petri Nets (TPN)

The definition of such a net is as follows.

Time Petri net (TPN) is a 6-elements set $\mathcal{Z} = \{P, T, F, W, m_0, I\}$, where:

- P and T are disjunctive, finite sets of respectively places and transitions: $P \cap T = \varnothing$,
- $F \subseteq (P \times T) \cup (T \times P)$ defines a set of arcs,
- $W : F \rightarrow N$ assigns an integer value to every arc,
- $m_0 : P \rightarrow N_0$ is an initial state of the net,
- $I : T \rightarrow Q_0^+ \times (Q_0^+ \cup \{\infty\})$ and for every transition $t \in T$, where $I(t) = (I_1(t), I_2(t))$ the inequality $I_1(t) \leq I_2(t)$ holds.

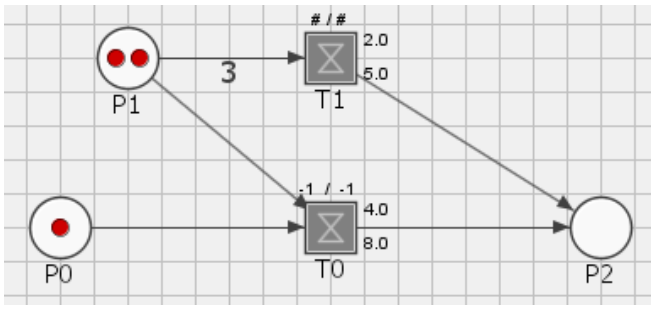
First five elements of set $\mathcal{Z}$ are the same as in the definition of a classical Petri net. Later, this subset of set $\mathcal{Z}$ will be called as a **skeleton** (of a Time Petri net).

Last line can be read as follows: there is a function I, which for every transition $t \in T$ assigns two positive rational number (or zero), where the second number may be in theory replaced by an infinity symbol, and (while being the numbers) the following inequality must hold: $I_1(t) \leq I_2(t)$.

For the moment (Holmes 1.0) only integer numbers are allowed as time parameters for TPN, what in fact does not seriously impact the analytical approaches (cf. (Popova-Zeugmann, 2013).

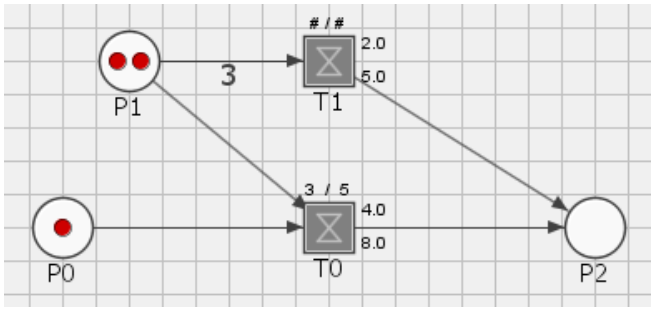
First time ($I_1(t)$) is the *Earliest Firing Time (EFT)*, i.e., that much time must pass **after transition becomes enabled**, before it **can** fire. The second time ($I_2(t)$) – is the *Late Firing Time* (*LFT*) – it is the maximum time an enabled transition can wait until it fires. Therefore, with every transition a counter is combined, counting time from zero to some value between EFT and LFT after transition becomes enabled. The time is determined randomly when the transition becomes enabled (determined every time this happens, i.e., when a transition (before firing) stops being enabled and after some time becomes enabled again, the time is generated anew). For the inactive (not enabled) transition its time value for its internal counter is often marked with # in the literature.

In Holmes it is assumed that time counts the simulation steps. By such a step we understand all the sequences of task in the algorithms, that determine transitions activation and firing. In the simulator, every time a transition becomes enabled, a value T_x is generated for it ($EFT \leq T_x \leq LFT$) and the transition clock starts counting time from 0. When it reaches generated time T_x, transition fires.

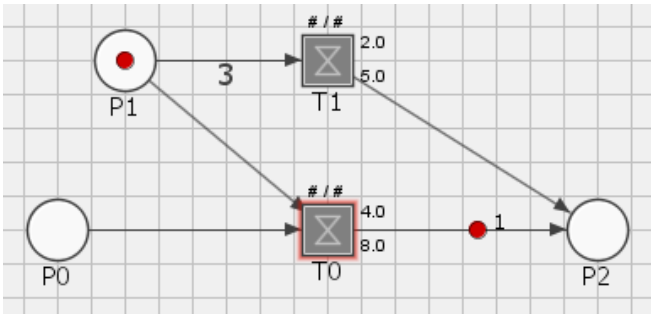


Example is given in the picture. T0 is enabled, its time parameters are: **EFT** = 4.0 and **LFT** = 8.0.

T1 is not enabled. Its parameters are: **EFT** = 2.0 and **LFT** = 5.0. It is not enabled because in P1 there are 2 tokens, 1 less than required by the weight of the proper arc. Its 'inactivity' is marked in Holmes by # / # symbols. -1 / -1 for T0 in Holmes shows the enabled transition, for which time towards which a clock will count haven't been determined (i.e., because the simulation in Holmes in this example has not started yet). Still, it can help in the distinction between enabled and not enabled transitions.



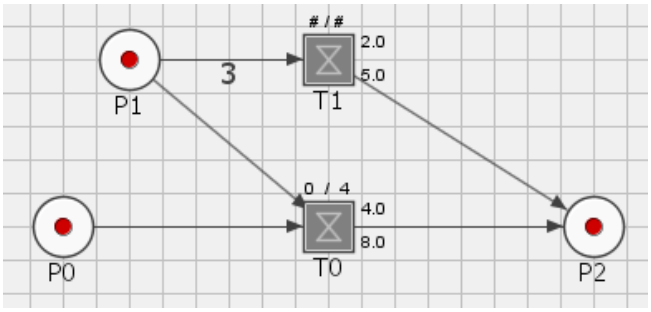
In the picture there is a situation in the simulator 3 steps later. The state understood as tokens distribution did not change. However, Petri nets with time have their state described by more complex structure. Among marking, which determines number of tokens in places, there is a second vector for the transitions, which hold the value of internal counters for the transitions and the time value the counters try to reach. For the transition T0, three steps later there is now 3 / 5 values assigned, meaning that the clock value is equal to 3, and the clock counts toward 5 (value between 4 and 8).



Two additional steps later, the situation is as presented in the picture. Internal clock of T0 counted from 0 to 5, therefore T0 fires. It takes 1 token from P1, 1 from P0, and produces 1 token in P2.

It stops being enabled after firing, because there are no more tokens in P0.

IF there were e.g., 2 tokens in P0 **BEFORE** firing, the situation **AFTER** firing would be different – as presented in the following picture below:



The difference is as follows: P0 had 2 tokens, 1 has been taken when T0 fired after counting to 5 (as described in the above pictures). Because after firing T0 is still active, its clock is reset to 0, **and new value between EFT and LFT is being generated**. In the example this value is 4 this time. Four steps later, if T0 will still be enabled, it can fire again.

It should be noted, that when transition stops being enabled after firing (or also before that), its internal clock and value toward which the clock counts are being reset and marked as # / #. If transition is still enabled after firing, the clock is reset to 0, and a new value between EFT and LFT is being generated.

4.1.3.2 Timed Petri Nets (DPN: Duration Petri Nets)

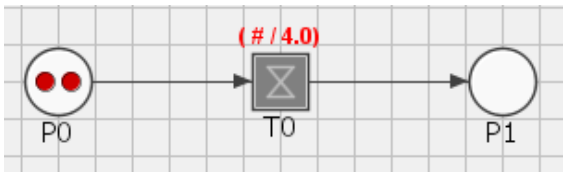
The second supported Petri net with time is *Timed/Duration Petri Nets*. Its definition is as follows:

Timed Petri net (DPN) is a 6-elemens set $\mathcal{D} = \{P, T, F, W, m_0, D\}$, where:

- P and T are disjunctive, finite sets of respectively places and transitions: $P \cap T = \varnothing$,
- $F \subseteq (P \times T) \cup (T \times P)$ defines a set of arcs,
- $W : F \rightarrow N$ assigns an integer value to every arc,
- $m_0 : P \rightarrow N_0$ is an initial state of the net,
- $D : T \rightarrow Q_0^+$.

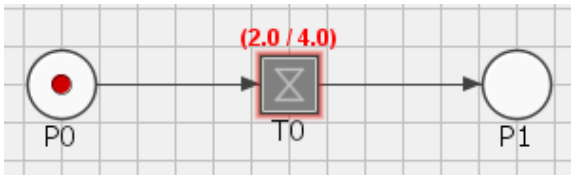
Again, first 5 elements makes the skeleton of Timed Petri net – it is in fact a classical Petri net. The last element is a function which assigns to every transition a positive rational number (or zero).

Therefore, to every transition a value d_x is assigned. This value determines **how long tokens production will last if the transition fires**. The firing of a transition is changed in this type of net and it is divided into two phases. **An enabled transition MUST fire immediately** (such a behavior is always assumed in the Timed Petri nets). In the first phase, when enabled transition fires, it will immediately consume tokens from its pre-places from its set •t. Then, the second phase of firing begins. In this phase a transition counts from 0 towards d_x. When its clock reaches d_x then the transition produces tokens in its post places from t•.



Representation of a DPN transition is similar as before. There are no EFT and LFT values, but there is a value for the transition internal clock (first one before /) and the value d_x (second one, after /).

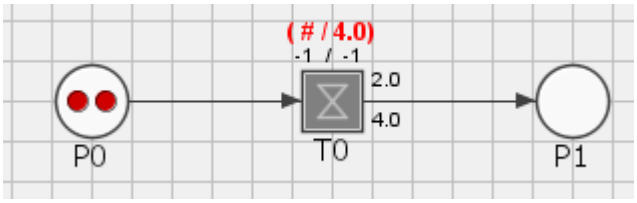
In the example, # / 4.0 means the transition is enabled, but it has not fired yet. It will of course fire immediately when the simulation starts, the red value will then change into 0 / 4.0. After two steps if the simulation, it will looks like in the picture below.



Red frame marks the enabled and 'firing' transition, meaning that it already consumed to tokens from pre-places, but it has not yet produced tokens in post-places. When simulator reaches 4th step, T0 will produce one token in P1 (and immediately 'fire' again, i.e., consume token from P0 and it will start counting (again) towards $d_x = 4$.

4.1.3.3 Time-Duration Petri Nets (TDPN)

This special type of net can be created in Holmes. It connects TPN and DPN features into one transition (to some degree).



There is however one problem with such merging of two net types. In DPN enabled transition must fire immediately. In TPN – no sooner than when the clock reaches EFT value (and no later than LFT). In Holmes, in the

Properties window and options concerning simulator (see Chapter 8.1 – Properties) two distinct and separate scenarios can be chosen for the simulator behavior.

- Enabled transition TDPN will take tokens from •t and start counting towards d_x after its internal clock reaches T_x: $EFT \leq T_x \leq LFT$. It is default mode for the simulator. In the example above it could be e.g., $T_x = 3$ ($EFT=2 < T_x=3 < LFT=4$), meaning that after 3 steps, if the transition fires, it will consume tokens from its pre-places, and then will start counting towards d_x. When this value is reached, transition will produce tokens in post-places. The transition can stop being enabled BEFORE T_x happens. When it happens however, and the transition starts counting to d_x, it cannot be disabled.
- The second mode (options „**TDPN transition acts like DPN when TPN internal clock = EFT**" in **Simulator** tab of the **Properties** window) assumes, that T_x will not be randomly generated. It means, that such a transition counts toward EFT, but as soon as this value is reached, it will fire immediately in a way described for the DPN.

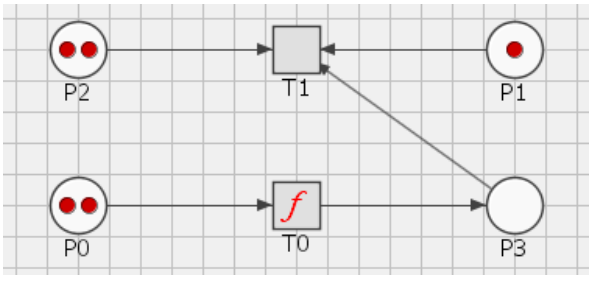
4.1.4 Functional nets

It is somehow difficult to provide a single definition for such nets from the literature. Such a net can be found in (Valk, 1978) or (Hofestädt i Thelen, 1998).

Functional Petri net is 5-elements set $F = \{P, T, F, V_F, m_0\}$, where:

- P and T are disjunctive, finite sets of respectively places and transitions: $P \cap T = \varnothing$,
- $F \subseteq (P \times T) \cup (T \times P)$ defines a set of arcs,
- $V_F: F \rightarrow N^+$, $V_F \in \{ \{ g\{x_1, \ldots, x_n\} \mid x_i \in P, g \rightarrow Q \rightarrow N^+ \} \vee N^+ \}$
- $m_0 : P \rightarrow N_0$ is an initial state of the net.

Functional net has the same structure as a classical net, i.e., places, transitions and arcs. The difference lies in arcs, more precisely: weights are replaced by functions, which are constantly evaluated depending on the net state. Set V_F can have two distinct „objects" : the first one is the evaluated function, the second (alternatively) is the normal weight as in the classical net. In this way, not all arcs must have functions assigned to them.



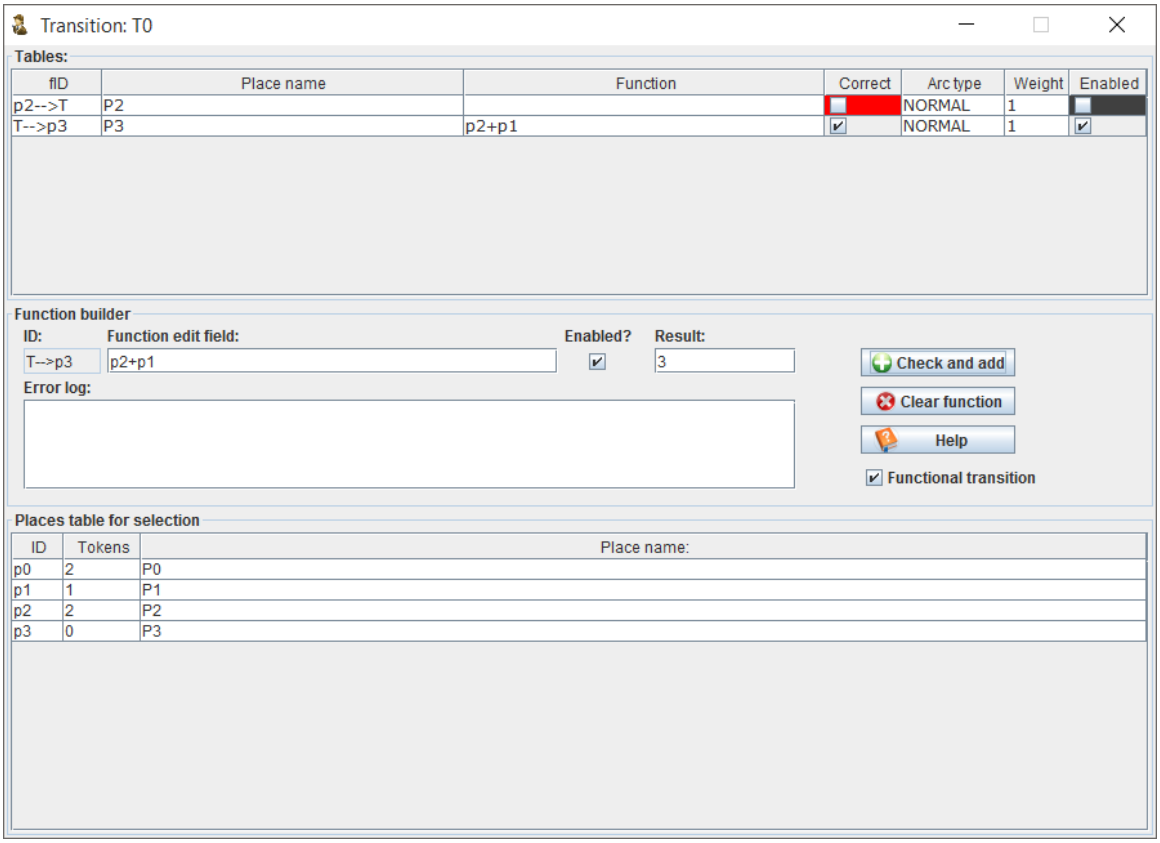
In Holmes, all the transitions which have **at least one** arc with assigned function, are marked with an "f" symbol. The example of such a transition in the picture is T0.

It should be noted, that it is not so obvious to tell from the picture alone, whether T0 is enabled or not, or how many tokens it will produce – it depends on arcs with functions. All we can say here, is that in this state T1 is not enabled (no tokens in P3).

Set $\bullet t_0$ = { P0 }, set $t_0\bullet$ = { P3 }. On the next picture a window with functions manager will be presented. Its main goal is the creation and management of functions that can be written and

assigned to the arcs of a given transition (therefore this window is activated in Holmes for the specific transition).

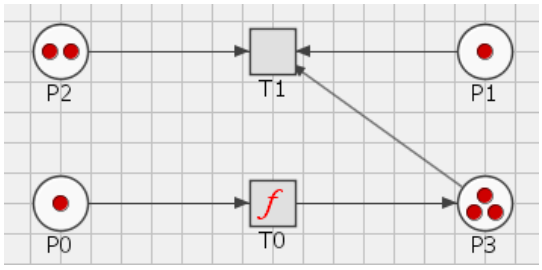
Not all things connected with the function editor will be described here. For the detailed description see 4.4.3.



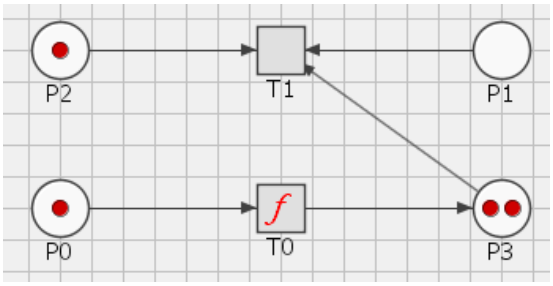
In the upper table there are two rows, for the arcs connected with T0. First column serves as the identifier for arcs, in this example p2-->T is the incoming arc (from place P2) while T-->p3 denotes an arc going from T0 to the place P3. The first row/arc does not have function assigned (empty field under *Function* column). Its weight is equal to 1 (*Weight* column). In theory, the second arc/row has the same weight, but since the column *Enabled* and *Correct* are both checked, the function takes precedence over the static weight value.

The very function is quite simple: p2 + p1. It means, that the weight depends on the state of the net, namely to value of tokens in places P1 and P2. It should be noted, that P1 is neither in $\bullet t_0$ nor $t_0\bullet$. It means, that function variables can be the places of the whole net, which allows the creation of more 'state-dependent' connections between transitions and places.

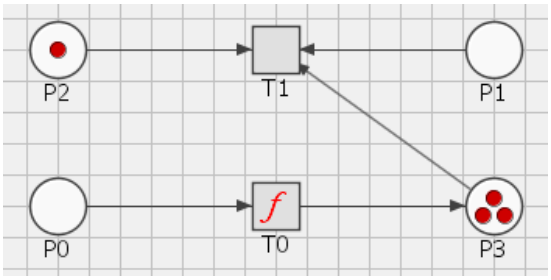
The bottom table has purely informative goal – it shows all the existing places of the net – their tokens (in the current state), names and what is important – IDs used as the names of variables in functions.



Knowing the function one can tell the weight of the arc (in the current state/simulation step), and thus answer the question whether T0 is enabled or not. It is enabled, because there is one token in P0, and the arc does not have function assigned. When the transition fires, it will produce p2+p1 tokens in p3. Since p2+p1 = 3, three tokens will be produced in P3, one taken from P0.



Let's assume T1 firing in the next step. It does not produce anything (no post-places, $t_1\bullet = \{ \varnothing \}$), but this so called output transition will take 1 token from P1, P2 and P3 ($\bullet t_1 = \{ P1, P2, P3 \}$). T1 is no longer enabled, however T0 still is. Effects of its firing are showed in the next picture.



T0 takes 1 token from P0 and produces only 1 token (this time) in P3, because p2+p1 in the last state was equal to 1.

It should be noted at the end, that net functions can be assigned to the classical, time and many other net types in Holmes. In the current version of Holmes, functions can be assigned only to the "classical" arcs. Read arc, inhibitor, reset and equal arc cannot have functions assigned. However, in future it is possible to implement such feature for the extended arcs, with the exception of the reset arc (cf. its description).

4.1.5 Stochastic nets

Nets of this type are still being developed in Holmes, however there are some functionalities already present in our program. Here the basic description will be given. Information about Holmes support for these nets in the current version will be described in details in the simulators chapter of the manual.

Stochastic Petri net (SPN) is a set $S = \{P, T, F, \Lambda, W, m_0\}$, where:

- P and T are disjunctive, finite sets of respectively places and transitions: $P \cap T = \varnothing$,
- $F \subseteq (P \times T) \cup (T \times P)$ defines a set of arcs,
- $W : F \rightarrow N$ assigns an integer value to every arc,
- $\Lambda : T \rightarrow R_0^+$ is a firing rates set for transitions,
- $m_0 : P \rightarrow N_0$ is an initial state of the net.

The main difference lies in a set Λ with *firing rates* for transitions. Function of such a set will be described later in the simulators chapter. In general the firing rates are modifiers for the stochastic functions deciding the probability of firing for the enabled transitions.

In the current version of Holmes, every transition can have firing rate assigned (1.0 by default). In future, extensions for SPN will be implemented, i.e., net types of transitions.

Name	Symbol	Meaning	Status
Stochastic Transition		Standard stochastic transition, identical to the classical one with the exception of firing rate assigned. Mass Action Kinetics mode is available in the simulator, where the tokens distributions in pre-places have an impact on the probability of transition firing.	Working
Deterministic Transition		Delay being a specific value. It can be interpreted as a time transition TPN where *EFT*>0 and *EFT*=*LFT*.	 Basic functionality working
Immediate Transition		Transition fires immediately when activated, DPN $d_x = 0$.	 Basic functionality working
Scheduled Transition		Transition fires at the specific time / schedule.	 Not yet implemented

4.1.6 Hierarchical net (multi-leveled nets)

It should be noted, that this is not a new type of a Petri net, e.g., with its own definition. It is an extension of the graphical representation of the net, similar in functionality to the already explained portals – and in fact such nets require portals in their structure to further extend it. In simple words such nets have some of their region drawn in different sheets than other, making the decomposition of the net structure easier to handle and analyse (in theory). There are some features of these nets, that must be clarified from the beginning:

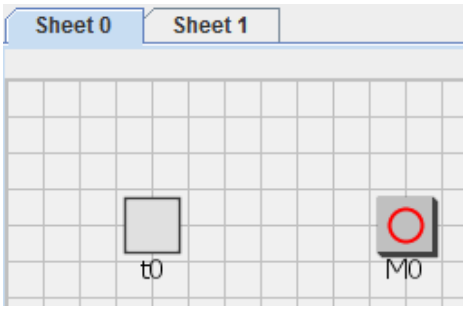
- Nets of these types have their structure divided into regions, but these regions can and should be connected with each other's.
- Their implementation has been created on the basis of Snoopy representation of such nets. There can be however some serious problems when importing, and more importantly: exporting such nets between Holmes and Snoopy.
    - Importing hierarchical (classical) net from Snoopy is possible, yet caution is advised and manual verification of the resulting export. Holmes will warn the user if any problems occurred.

    - Exporting to Snoopy is implemented **to some degree.** Due to the plans of extending this type of net in Holmes, not all features that Holmes allow can be exported at all.

    - In fact, making a hierarchical net in Holmes is still under development, yet the basic functionality is already present and can be used. It is strongly advised to store such nets as Holmes project files, because this option is the safest way of saving a model.

- There are three new elements in Holmes used to build such nets.

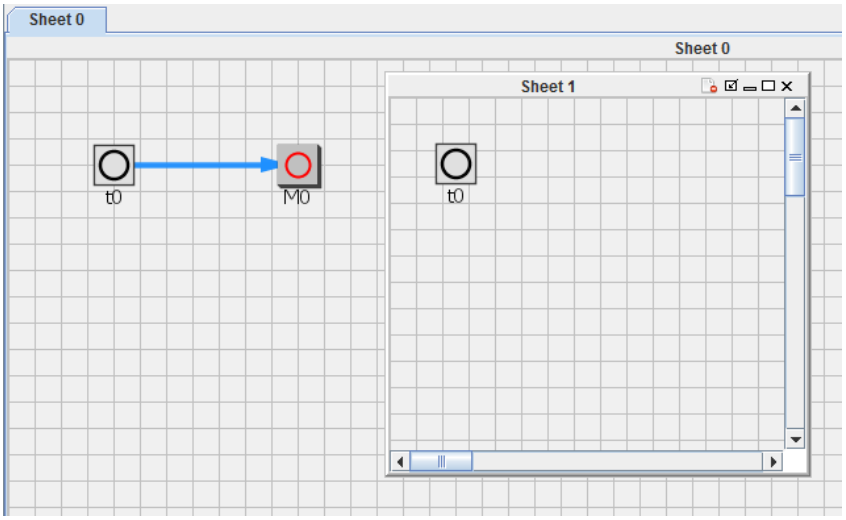
Holmes	Symbol	Snoopy	Symbol	Meaning
Subnet T-type		**Coarse Transition**		Subnet with places being the interface (connecting elements).
Subnet P-type		**Coarse Place**		Subnet with transitions being the interface (connecting elements).
General subnet		-n/a-	-n/a-	Both places and transitions can be used as the communication mechanism for such subnet. Using this option will make export to Snoopy impossible, because this feature is not available in Snoopy.



We will start our example with a net with a single t0 transition. Choosing Subnet P-type element from PN elements (Section 4) will not only draw it, but also create a new sheet for this net (Sheet 1).

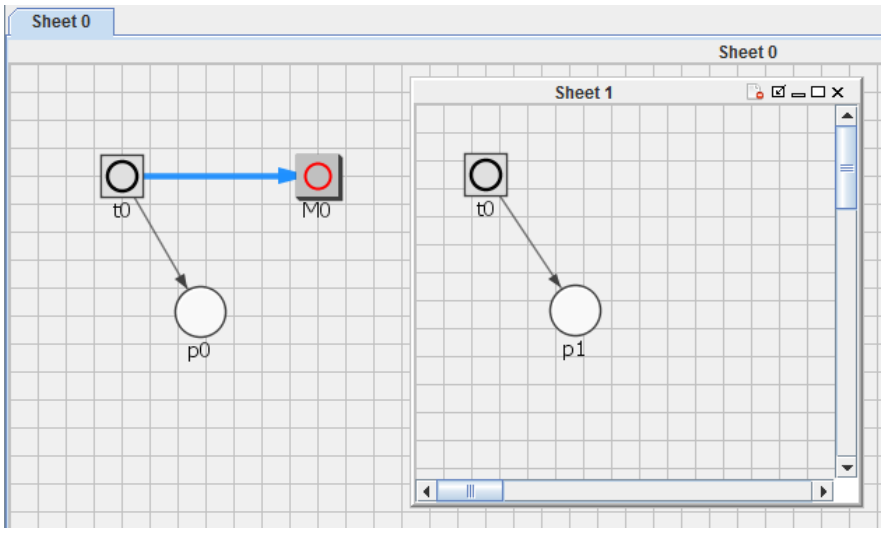
In our program graphical representations of subnets are called *meta-nodes* (their arcs being called *meta-arcs*). They are in fact only the graphical representations, to some degree similar to portals, not a distinct and separate elements like normal places or transitions.

For the next picture and Sheet 1 Externize button has been used, as already explained in 2.1



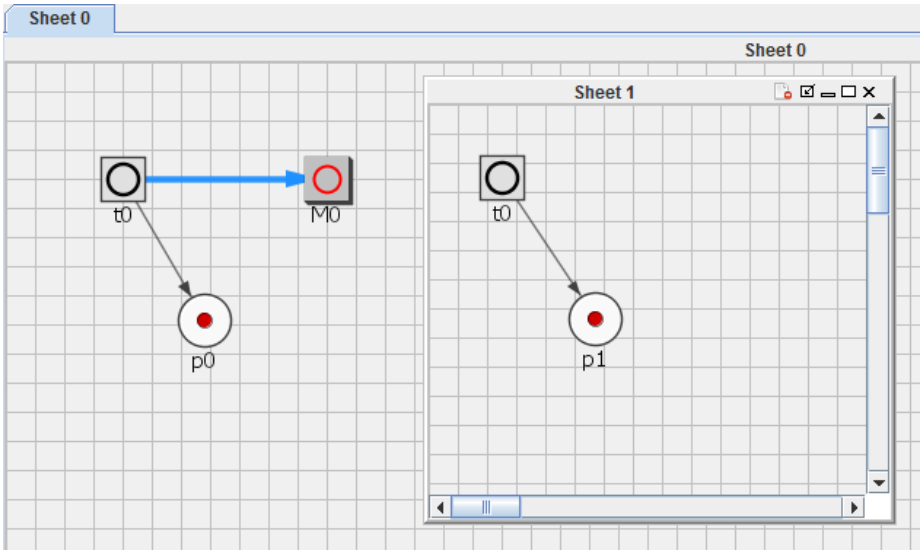
The arc in the **main** net (**always the one drawn in Sheet 0**) is drawn in blue color. The additional effect of connecting t0 with M0 is the creation of t0-portal, and its second representation in Sheet 1. In this way the connection between subnets and the main net can be drawn.

It now should be clear, why we call such arc as *meta-arc.* In fact, from the Petri net theory perspective such an arc **does not exist**. It only serves as a graphical reminder of a connection between different regions of one net/model. It the example there is only one transition: t0, so meta-arc in fact connects this transition with itself, i.e., its two portals being the graphical representation of the single transition: t0.

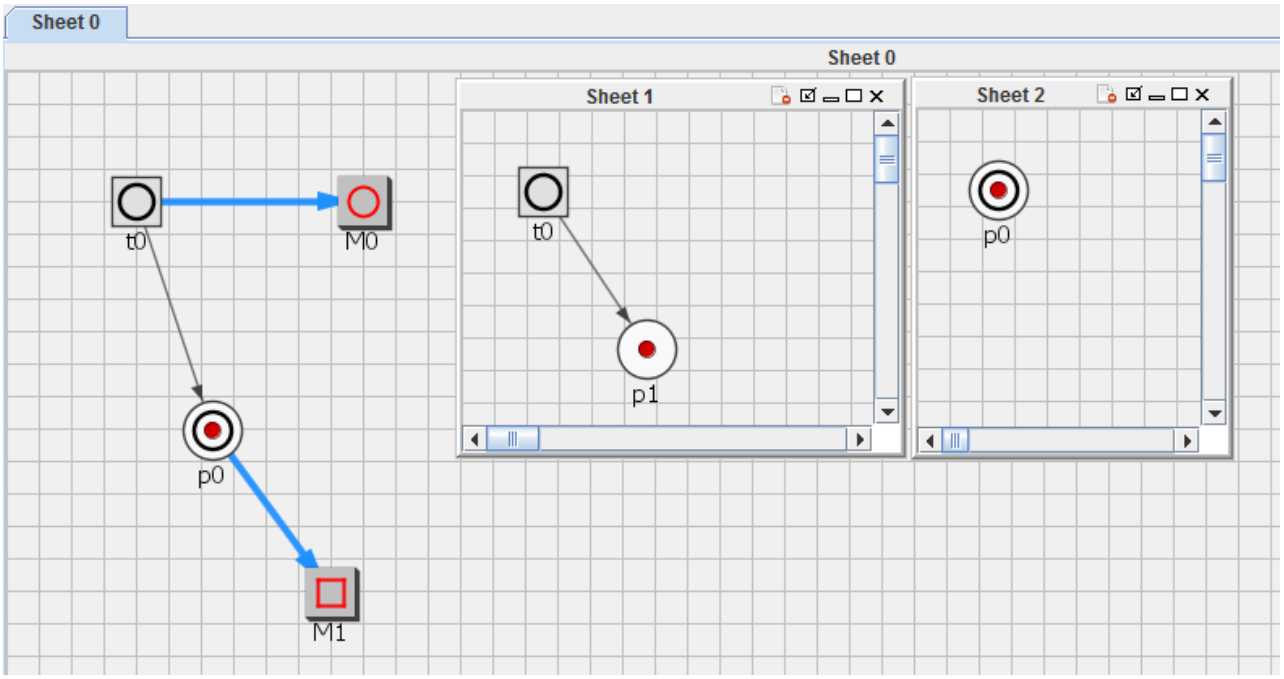


Let's add two additional places, p0 in the main net, p1 in the subnet. The result is showed in the picture.

When t0 fires (it is an input-transition, with no pre-places, therefore it is always enabled), 1 token in p0 and 1 in p1 will be produced:

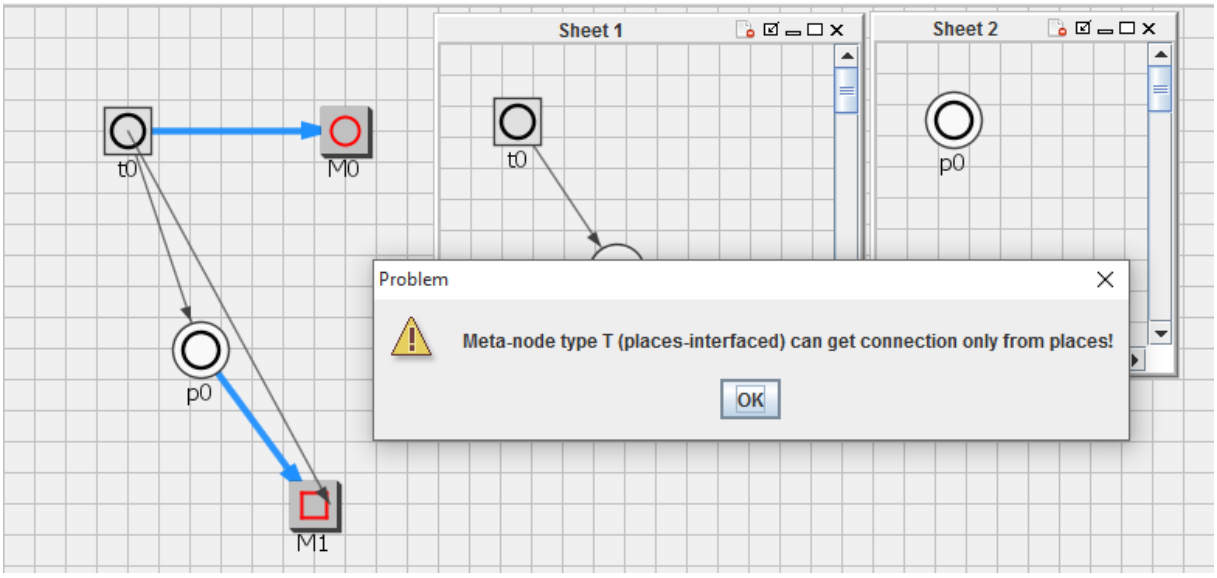


Now, let's add net Type-T Subnet, with p0 being interface places between such a subnet (Sheet 2) and the main net (Sheet 0):

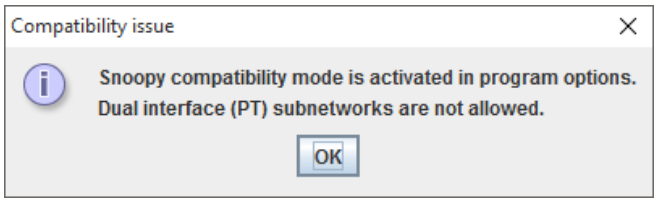


Now we have 2 subnets connected with main net by both transition (t0 for subnet M0) and place (p0 for subnet M1).

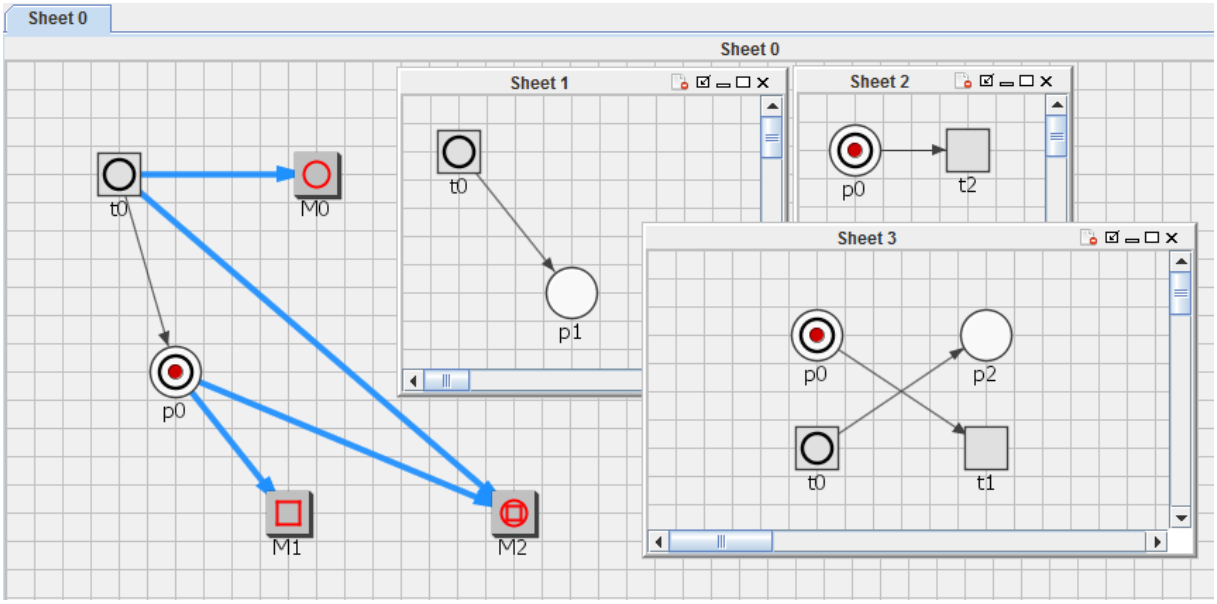
When trying to draw an arc from t0 to M1 (Subnet T-Type) a warning will show up:



Also, when trying to add a PT-type net (a *General* Subnet), another warning will show up:

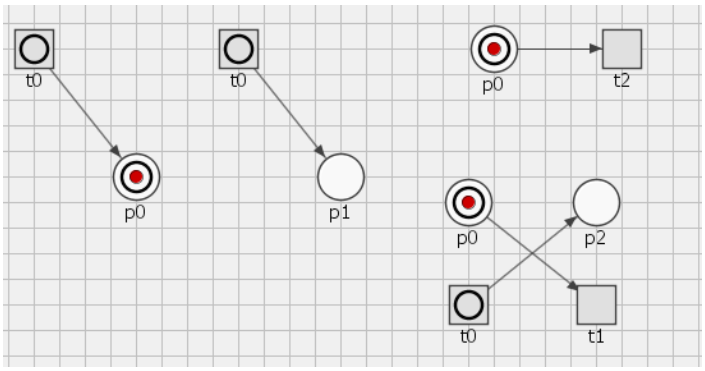


Properties window for Holmes will be described later, now let us assume, that compatibility mode (with Snoopy) has been turned off:

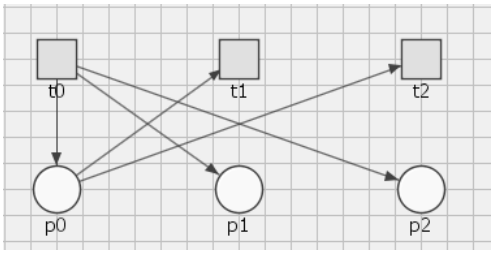


Meta-node M2 represent subnet from Sheet 3. It can have both places and transitions as interface nodes.

To close this example, let's see the very same net as in the picture above, but without the subnets:



Or without portals at all:



There are 3 useful options in Holmes, already described for the menu bar:

Option	Icon	Meaning
Compress subnets		Removes empty sheets from project.
Align to upper left		Align all the elements in subnets to upper left corner (with respect to distances) – useful for Snoopy import.
Resize panels		Adjust maximum sheet size to match the existing elements.

Double click on any meta-node will show the proper Sheet for such a subnet.

Hierarchical nets will be extended with new features and analytical tools in the future.

4.1.7 Other net types

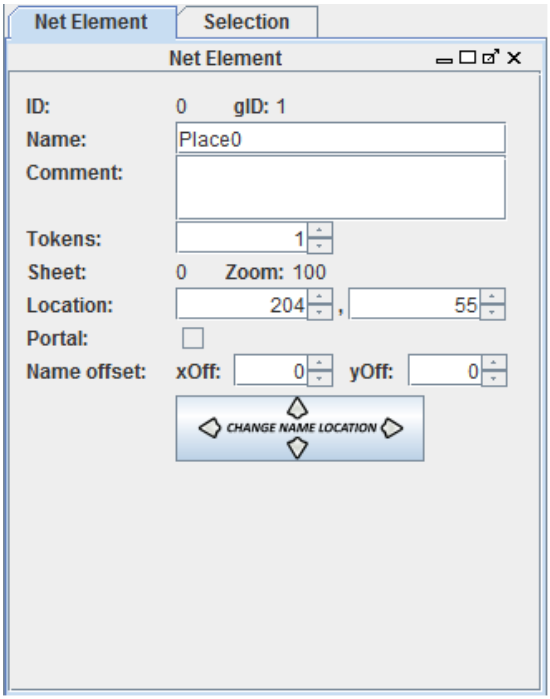
Other net types will be added to the program in the future.

4.2 PN elements data panels and subwindows

Holmes allows to see net element data in the Section 5 subwindow. Many value associated with a given selected element can be change there. In this section of the main window different views can be seen, depending on the clicked area: place, transition (transition type matters), arc and sheet (empty sheet space).

4.2.1 Place data

When a place has been clicked with LMB, its view will become available:



As one can see, there are a few elements that described a given selected place.

ID – place ID within the net. Such IDs are always counted from 0, separately for places, transitions and arcs.

gID – element global ID within the net – for all three main elements: places, transition and arcs (therefore each element will always have a different gID).

Name – place name, visible in the editor window.

Comment – field for additional comments and descriptions for a place. Not visible in the editor.

Tokens – shows the current token value in a place.

Sheet – sheet identifier in which (sheet) a place is drawn. Usually 0 (for non-hierarchical nets).

Zoom – current zoom size for a sheet. 100 is default and neutral zoom. Any other value will be given in red here.

Location – x and y coordinates of a place within a sheet. They can be changed here, but it is much easier to simply drag a place within the editor window.

Portal – shows information whether a place is a portal or not.

Name offset – x and y coordinates of a name text in relation to x,y coordinates in Location textbox. E.g., if in Location there is 300,200, and x,y of a Name offeset are e.g., 20,20, it means their global x, y in the editor are 320, 220.

They can be changed here, but it is much easier to use **CHANGE NAME LOCATION** button:

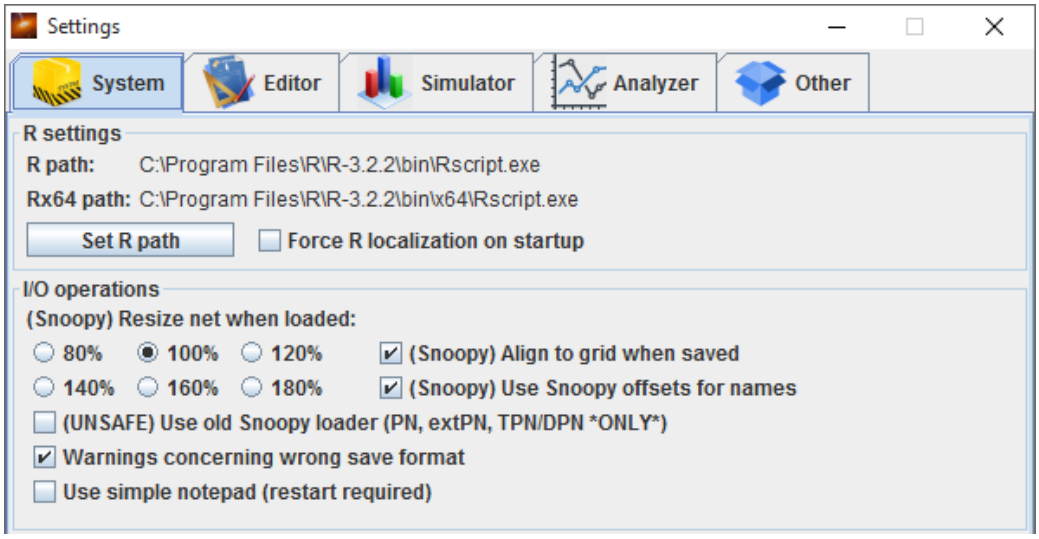
Normally the button is like:

But when clicked it will change into:

In this mode a mouse wheel with **Shift** button (or not) can be used to change the name location. E.g., when clicked and the user will roll the mouse wheel, the name will move in the editor horizontally. When the **Shift** button is pressed, the mouse wheel will change the name vertical location. To end this mode one can:

- Click **CHANGE NAME LOCATION** button again, or
- Click LMB anywhere on the sheet in the editor other than the already selected place.

When exporting/importing net to Snoopy, these values (for the name) will be saved/loaded. There are a few options in menu **Windows** -> **Properties** -> tab **System** for this feature (it will be also described later in the manual).



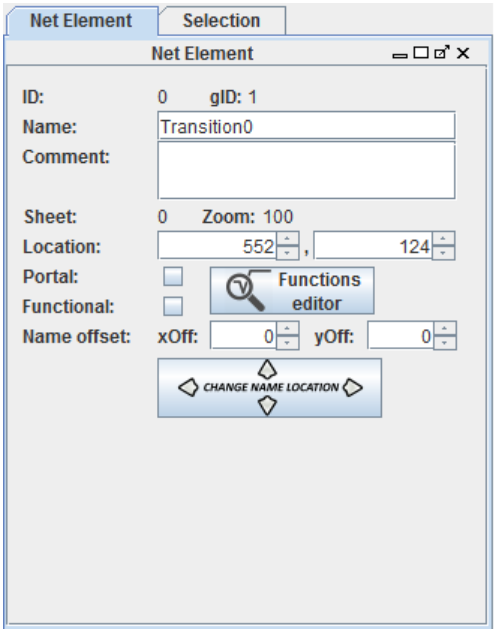
I/O operations are for **mostly** for Snoopy export/import. For example, choosing other value than 100% will automatically rescale the net when importing a net. This can be an useful option, because in general elements of a PN in Holmes are two times larger than in Snoopy (i.e., usually 40px to 20px in Snoopy). Choosing 120% will add 20% of value for every x and y coordinate for places and transition when importing a net. The same function can be obtained by the already explained buttons B1 and B2 from toolbar (2.2 Chapter of the manual).

(Snoopy) Align to grid when save will automatically act as B4 button from toolbar.

(Snoopy) Use Snoopy offsets for names will make Holmes adjusting the name coordinates to better suit it for Holmes graphical representations. Turning it off will reset the offset for every name to 0,0..

Turning this option off and exporting the net to Snoopy will save the net with all the names offsets set to 0,0.

4.2.2 Transition data



This window in Section 5 of the Holmes main window presents transition data. Some of them have identical meaning like the ones already described for places.

ID – transition main ID – same data as for places.

gID – global ID – same as for places.

Name – transition name (same as for places).

Comment – transition comments (same as for places).

Sheet – sheet ID for transition location (same as for places).

Zoom – transition's sheet zoom info (same as for places).

Location – x and y coordinates for transition (same as for places).

Portal – shows if the transition is a portal or not (same as for places).

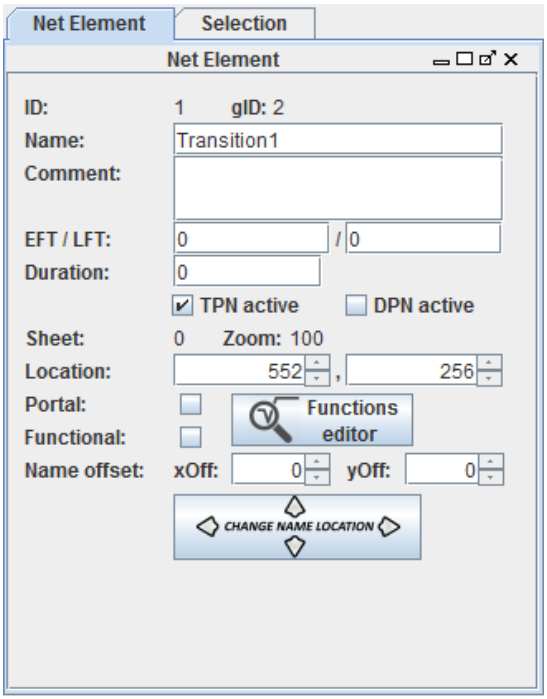
Functional – if this checkbox is pressed, the transition is considered functional **no matter if it have functions defined for its arcs or not**. In other words, when there are functions for the given transition arcs and the user wants to disregard them, all it takes is to uncheck this checkbox. **THIS WILL NOT REMOVE THE FUNCTIONS** – it will simply make them inactive until this box is checked again.

Functional editor - shows the functions editor as described already in the previous chapter.

Name offset – transition name offset (same as for places).

CHANGE NAME LOCATION – works the same as for places.

4.2.3 Time Transition data



This view has all the data boxes as described for a normal transition, with some additional ones:

EFT / LFT – such fields allow the assignment of time values for TPN net type.

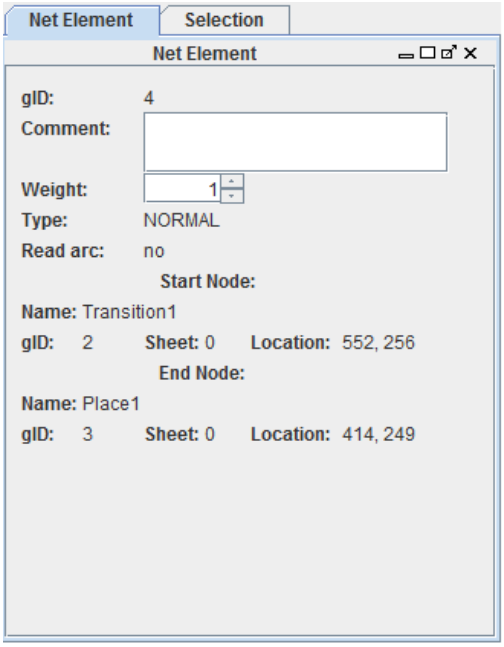
ATTENTION: since EFT ≤ LFT **must hold**, one should always start with a LFT value. Only then Holmes will allow the change of EFT from 0 to LFT at maximum.

Duration – tokens production time as described for the DPN time net.

TPN active – clicking this will enable the EFT and LFT values e.g., in the simulator. Even if they have been assigned, the simulator will use them only if this option is clicked.

DPN active – exactly as the abode, this time for the DPN type. As it has been already explained, both option can be selected at the same time.

4.2.4 Arc data



This view has less fields that can be modified directly in it.

Comment – a comment can be assigned to an arc, it will not be displayed.

Weight – for every[1] arc a weight can assigned.

Type – type of arc: **NORMAL**, **READARC**, **INHIBITOR**, **RESET**, **EQUAL**

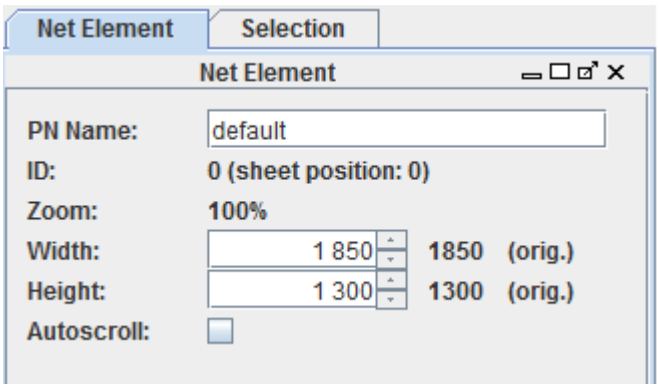
Read arc – this field will show a very useful information: whether the arc is a direct read arc, or a hidden read arc, i.e., a so called double arc („double arc (hidden readarc)" – described in the previous Chapter).

Start Node and **End Node** contains data about starting and ending nodes of an arc (either places or transitions). Additionally their gID, sheet ID and location will be displayed here.

[1] See the description of the extended arcs in the previous manual Chapter, especially the Reset arc.

4.2.5 Sheet data

This view will appear when an empty space within a sheet is clicked.



PN Name – the name of the whole net (not a single sheet!)

ID – sheet ID.

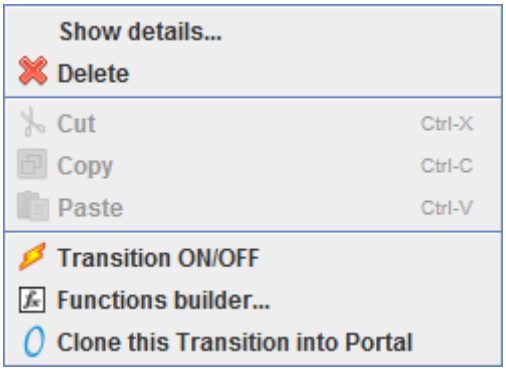
Zoom – zoom level, 100% is neutral.

Width and **Height** – allows changing the size of the sheet. They can be changed only when zoom is set to 100%. The right values shows the original size when zoom is different than 100% at the current sheet.

Autoscroll – turning it on and off will allow or disable the fast moving feature (Chapter 2.4.3).

4.3 Context menu

This menu will appear when RMB will be clicked on a sheet or a net element.



Manu for *transition*:

Show details... - this will show separate window for transition data, described later in the manual.

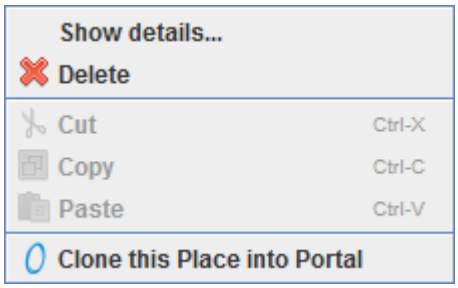
Delete – removal of the transition **from the net**.

Cut/Copy/Paste – (Holmes 1.0 : currently not implemented)

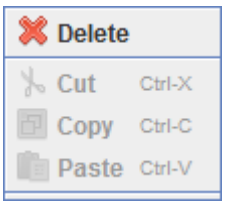
Transition ON/OFF – option for turning transition ON and OFF for the simulation. Will be explained later in great details.

Function builder… – window for functions edition.

Clone this Transition into Portal – will create a portal for the transition and it will add another graphical node for that portal.

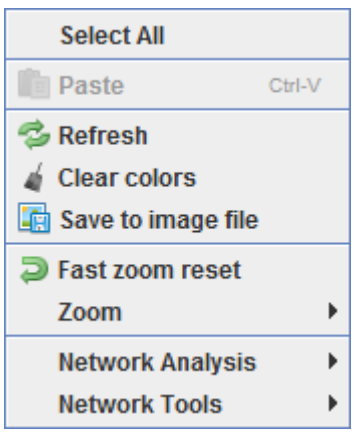


On the left there is a menu for places, on the right for arcs. There are less option here and they work analogously as the ones for transitions.



In the current version of Holmes no menu is available for meta-nodes and meta-arcs.

Context menu for a *sheet*:



It will appear when an empty space of sheet will be right-clicked.

Select All – this option will select all the nodes. It can be used to manually drag the whole net into different region of a sheet. This option will activate some hidden interface algorithms, therefore it may require some CPU time and memory!

Paste – currently not available.

Refresh – force the sheet to redraw, rarely if at all needed.

Clear color – there are many modules in Holmes which can draw

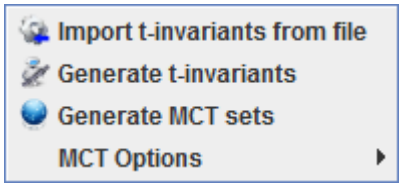
different elements of a Petri net structure in different colors. Choosing this option will restore the default colors presented in the editor.

Save to image file – this option allows saving a sheet into a picture file. **The resolution of the picture depends on the current zoom**.

Fast zoom reset – this option restores the default, 100% zoom level.

Zoom – it will open a smaller menu that allows changing the zoom into one of different values: 100%, 80%, 50% and 30%. The smaller the value, the smaller the net elements and the more of the net become visible.

Network Analysis submenu:



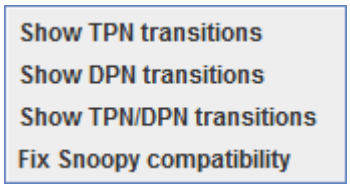
Import t-invariants – it allows reading t-invariants file created in other tools (see Invariants Generator chapter).

Generate t-invariants – this will start t-invariants calculations in the background.

Generate MCT sets – it can refresh the MCT sets. This option is not needed in the current version, because the MCT sets are always automatically computed when the t-invariants become available in the project.

MCT Options – another options concerning MCT sets will be available here, at the moment there is an option here to save MCT sets and t-invariants to file.

Network Tools submenu



Show TPN transitions – marks in light green time transitions with TPN mode enabled.

Show DPN transitions – marks in light green time transitions with DPN mode enabled.

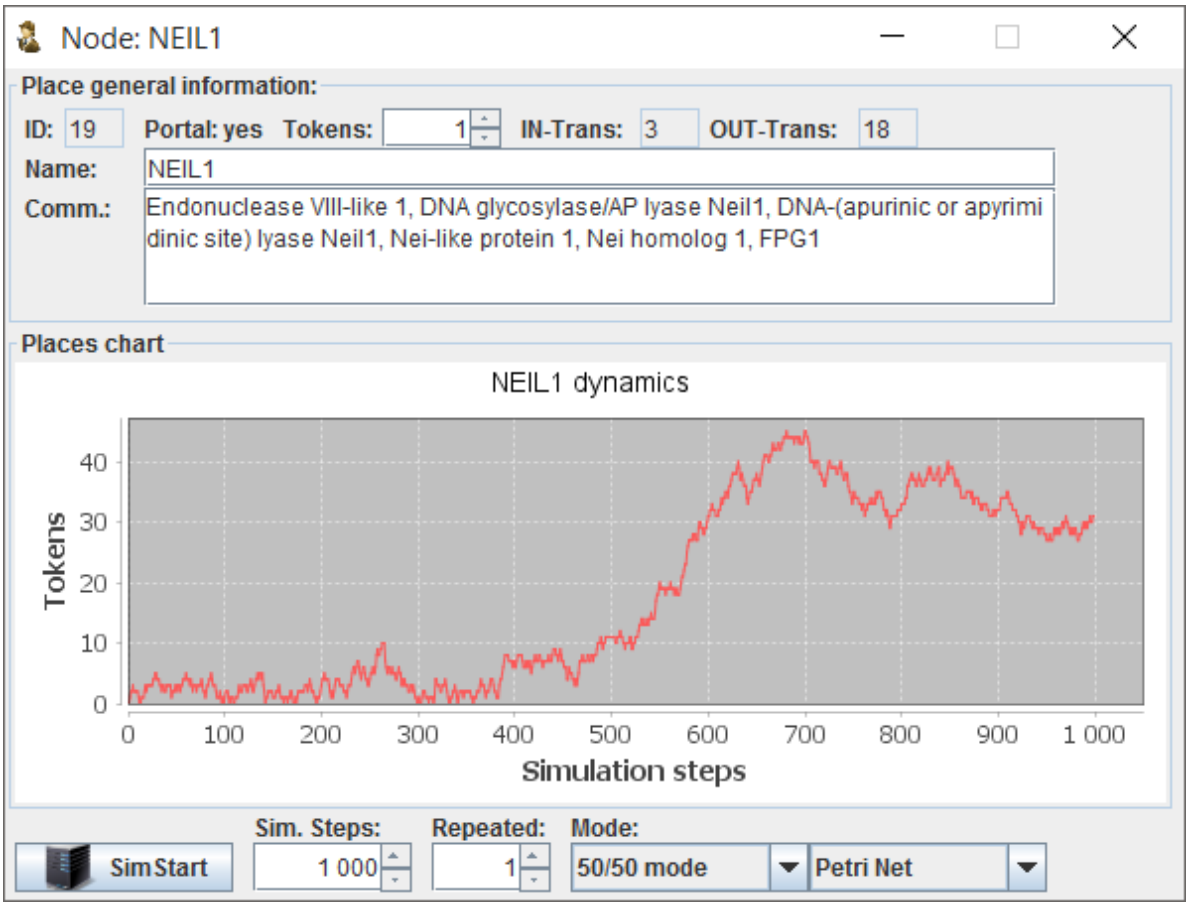
Show TPN/DPN transitions – marks both TPN and DPN transitions: pure TPN, pure DPN and mixed ones.

Fix Snoopy compatibility – this experimental option may be necessary to fix a project that consists of hierarchical Petri net imported from a Snoopy file. If some connections with the subnets seems to be lost, choosing this option may restore them. When the hierarchical net has been created in Holmes from the beginning, this option does nothing (i.e. it is not necessary).

4.4 Net elements data windows

Such windows contain detailed data for transitions and places, they can be opened by clicking Show details… menu option from the context menu.

4.4.1 Place data



Such a window has two parts, the upper one contains the following information:

ID – places ID, counted from 0 from the places set.

Portal – informs whether the place is a portal (logical place) or not.

Tokens – current number of tokens in place.

IN-Trans – the number of pre-transitions for the place (set •p)

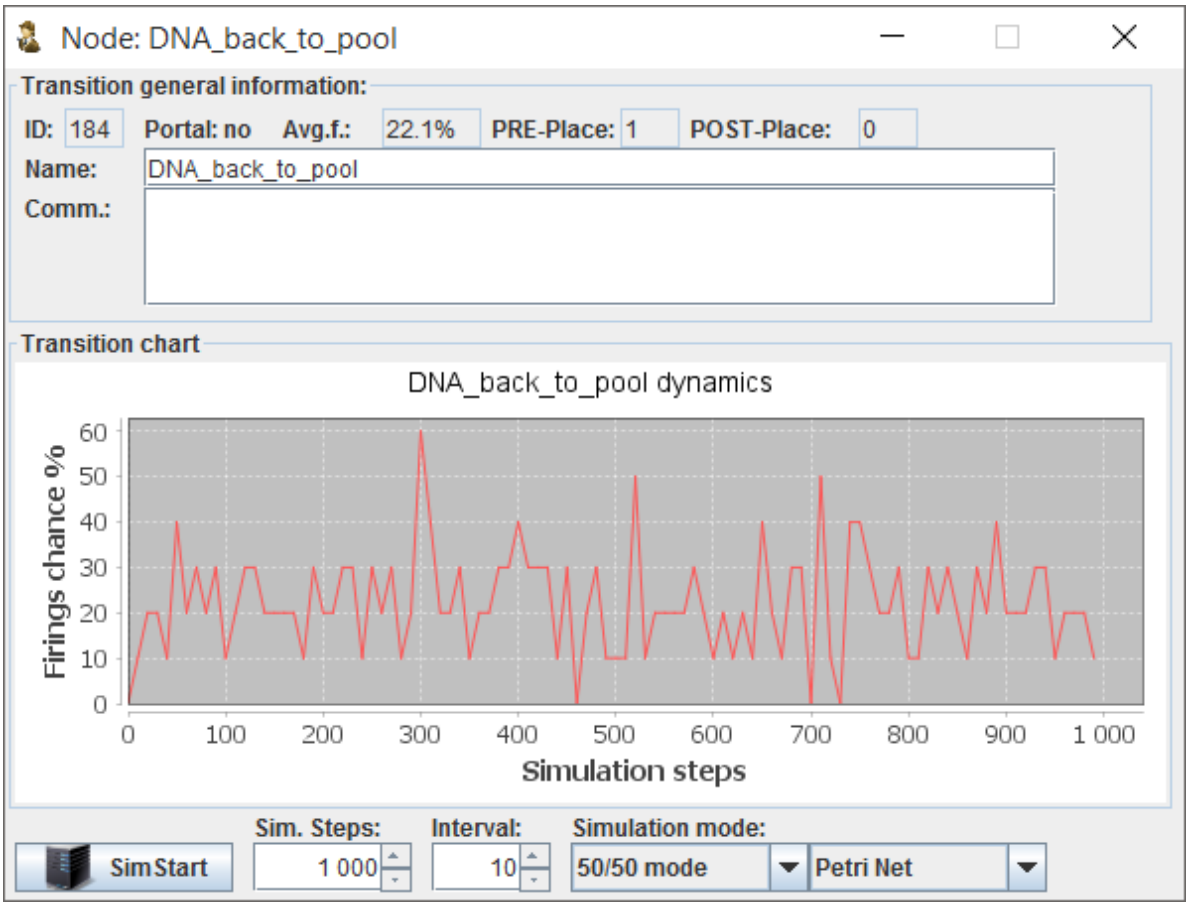
OUT-Trans – the number of post-transitions for the place (set p•)

Name – name of the place

Comm. - comment of the place

The bottom part of window contains a chart with the results of tokens distribution for a place coming from a quick simulation. SimStart button allows to make such a simulation again.

4.4.2 Transition data



The upper part of the window contains the following information:

ID – transition ID counted from 0 for the transitions set.

Portal – informs whether the transition is a portal (a logical transition) or not.

Avg.f – *average firing*, contains the average change of firing for the transition based on a fast simulation perfomed when the window has been opened.

PRE-Places – number of pre-places for the transition (set •t)

POST- Places – number of post-places for the transition (set t•)

Name – name of the transition.

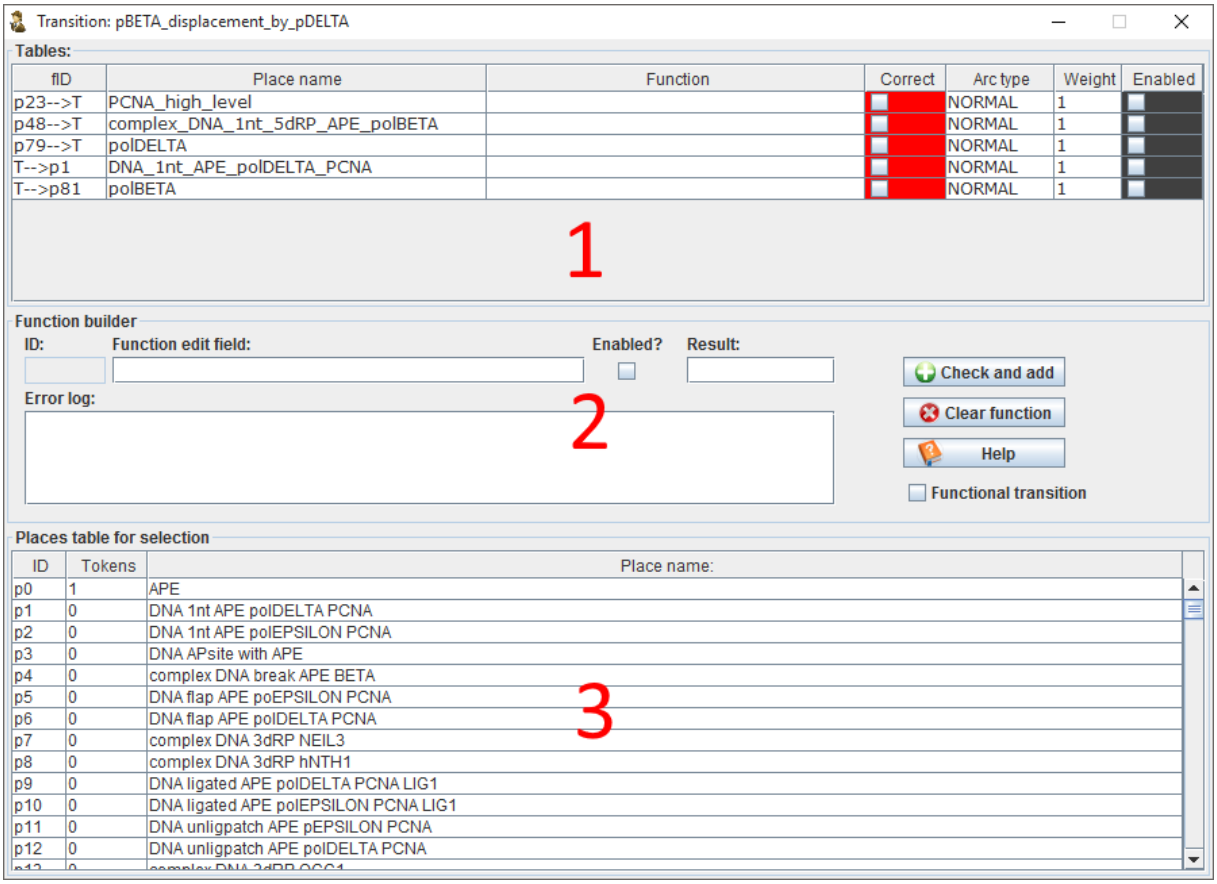
Comm. - comments for the transition.

The bottom part of window contains the average firing change for a transition in some set time interval. For example, in the picture there a 60% peak around 300th step of simulation. In means, that in 10 steps aroung 300th one the transition fired 6 times (interval is set for 10 in the picture).

4.4.3 Functions editor

This window can be open by using Functions builder… option from the context menu for a transition or by clicking a button in the transition data subwindow (Section 5).

The funtions editior look as in the following picture:



There are three main parts here:

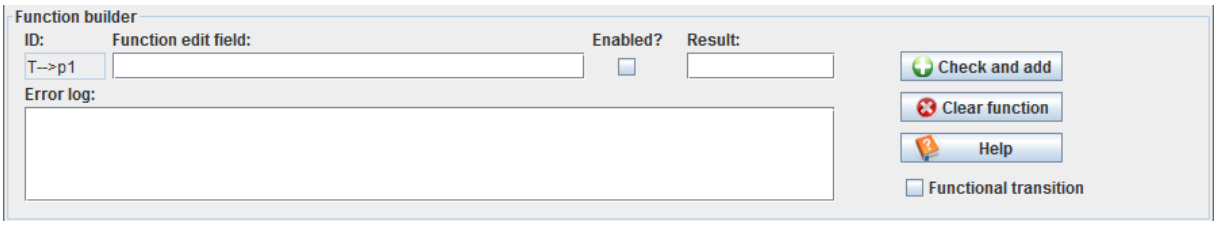
In part **1** there is a table for every incoming and outgoing arc for a transition. First column serves as the identifier, e.g., **p23-->T** describes an arc FROM place p_23 going TO transition (its name is part of the window title bar). In the example, the place name is **PCNA_high_level** (second column). Next columns are as follows: funtion/equation (if definied), function validity, type of arc (functions are possible only for the normal arc in the current version of Holmes). Last but one column denotes the weight of an arc (used if function is not correct / not enabled or not exists at all). Last column informs whether the function has been enabled or not (it should be noted, that even such an enabled, correct function will not be used IF the transition status is not set to 'functional' – explained later).

Equations/functions can be defined for both pre-places (set •t) and post-places (set t•). The difference between such functions is as follows:

- Arcs for places from set •t (with identifies like p_x-->T) define the 'activation' weight for the transition (i.e., the number of tokens necessary for a given place in order to enable the connected transition).

- Arcs for places from set t• (with identifiers like T-->p_x) denote the number of tokens that will be produced in such places if the transition fires.

By clicking a row in part **1** of the window will update the content of the part **2**, e.g. for T-->p1 from the example, part **2** of the functions editor will looks like:



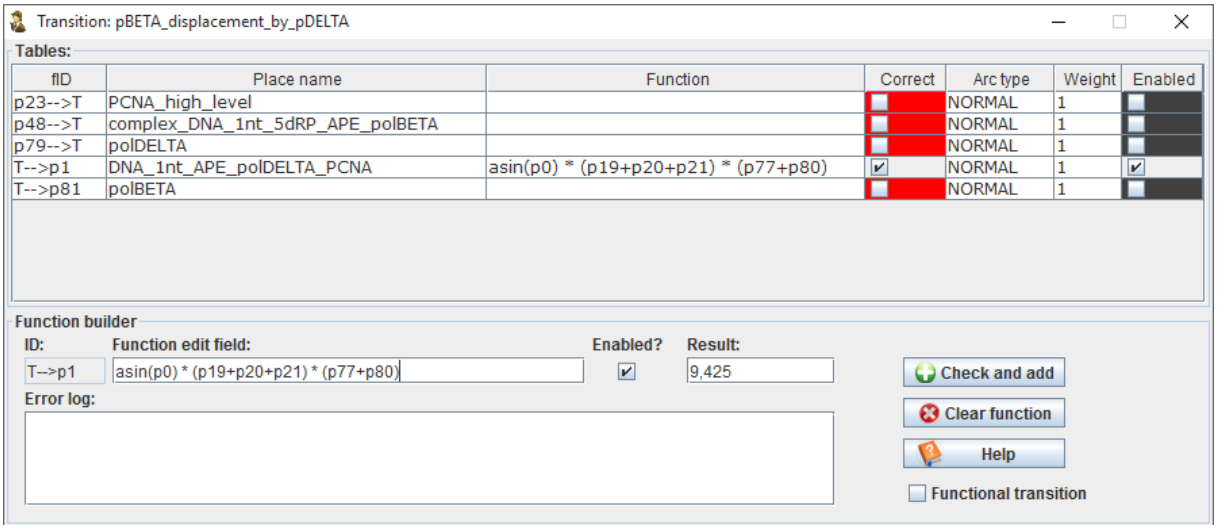
The most interesting part of this window is the field named **Function edit field**. Funtions/equation in Holmes are realised by a free java library exp4j included in our tool (*Apache Licence*, http://www.objecthunter.net/exp4j/license.html). It allows the usage of the following operators / functions:

Operator / function	Meaning
+, -, *, /	Standard arithmetical operations
+, - *(unary)*	Positive (almost unused) or negative value
x^y	x to the power of y
%	modulo – the rest from the division
( )	parenthesis
abs	Absolute values
acos	Arcus cosinus
asin	Arcus sinus
atan	Arcus tangens
cbrt	Cubic root
ceil	Upper integer value of a rational number
cos	Cosinus
cosh	Hiperbolic cosinus
exp	Euler value to the power of (a value after exp in parenthesis)
floor	Lower integer value of a rational number
log	Natural logarithm (base e)
log10	Decimal logarithm
log2	Binary logarithm (base 2)
sin	Sinus
sinh	Hiperbolic sinus
sqrt	Square root
tan	Tangens
tanh	Hiperbolic tangens

An important case of integer numbers for tokens must be clarified. In general, the value of the function is always a rational number type Double in Java. In ceil/floor function are **not used** in the equations, its value will be Double number of Holmes, **which will be conterted to Integer type to obtain tokens value**, i.e. any fraction will be cut off. It means, that a Double values 2.9 and 2.1 **will both be reduced to 2 (an Integer number)**

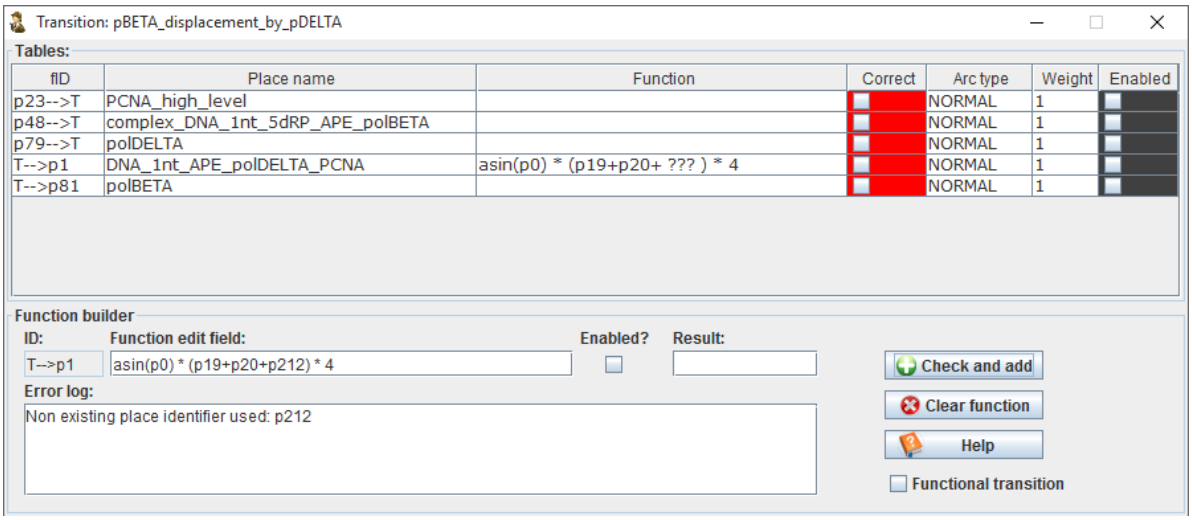
in Holmes (at least since the rational number of tokens will become available in Holmes in future versions, e.g., for continuous Petri nets). For this case, a floor/ceil functions should be used.

Another picture show more complicated function:



After providing the function and set **Enabled?** checkbox to true, button **Check and add** will start the verification/validation of the equation. If it is correct, it will be added for the proper arc. In **Enabled?** checkbox is not clicked, this button will only verify and add the function the the arc, but it will still be disabled (not used). It the function is correct, its current evaluation will be given in the **Result** field.

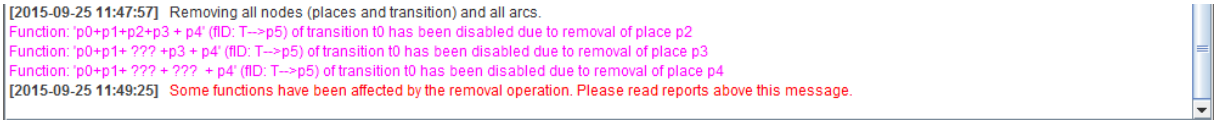
A case of an incorrect function is given below:



In error log Holmes tries to identify the problem. In this case, the used identifier for a place (p221) not exists in the net, therefore the function cannot be evaluated. In the table (Part 1 of the window) this part of the function has been replaced with "**???**". Flags **Correct** and **Enabled** are obviously set to false automatically.

IMPORTANT! When any place is removed from the net, verification of all functions is performed in the backgroud in Holmes. All functions affected by the removal will becomed disabled and set as incorrect.

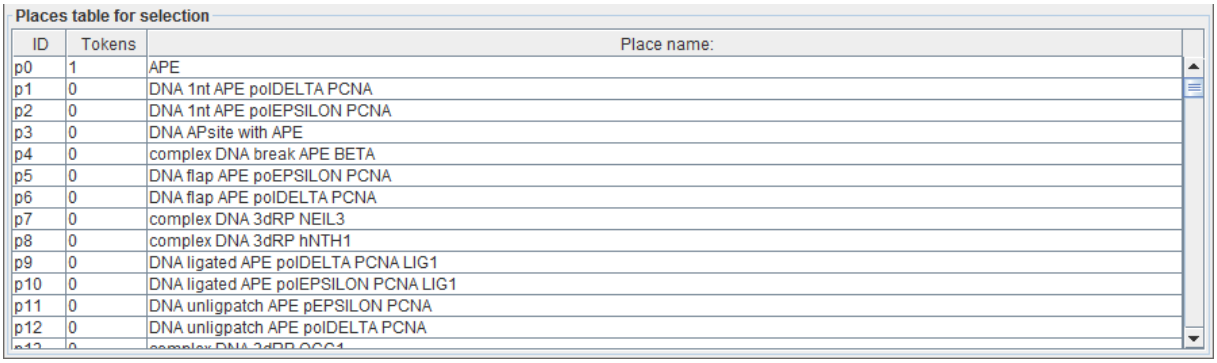
Let's assume there is an equation p0+p1+p2+p3 in some arc. If p1, p2 and p3 will be removed (let's assume, in the same moment), the Holmes log window will be shown:



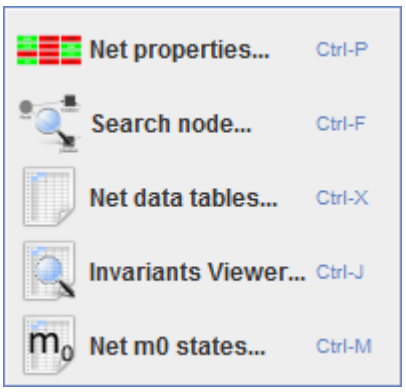
Help button shows simple notapad with informations about functions elements.

Functional transition checkbox activate or disable the 'functionality' of a transition. If transition status is "Functional" **only then any enabled functions assigned to this transition arcs will be used e.g., in the simulators.**

Part 3 of the functions editior window shows all places existing in the net with their IDs for funtions:



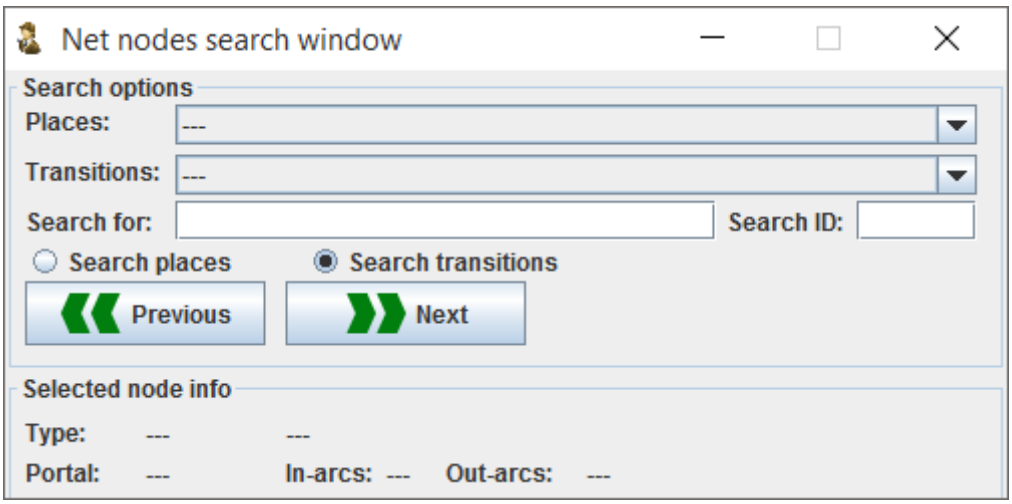
5 Net information windows



From this menu user can choose various modules containing data about the given net.

5.1 Net search

This window can also be activated by shortcut **Ctrl**+**F**. The window looks like in the picture:

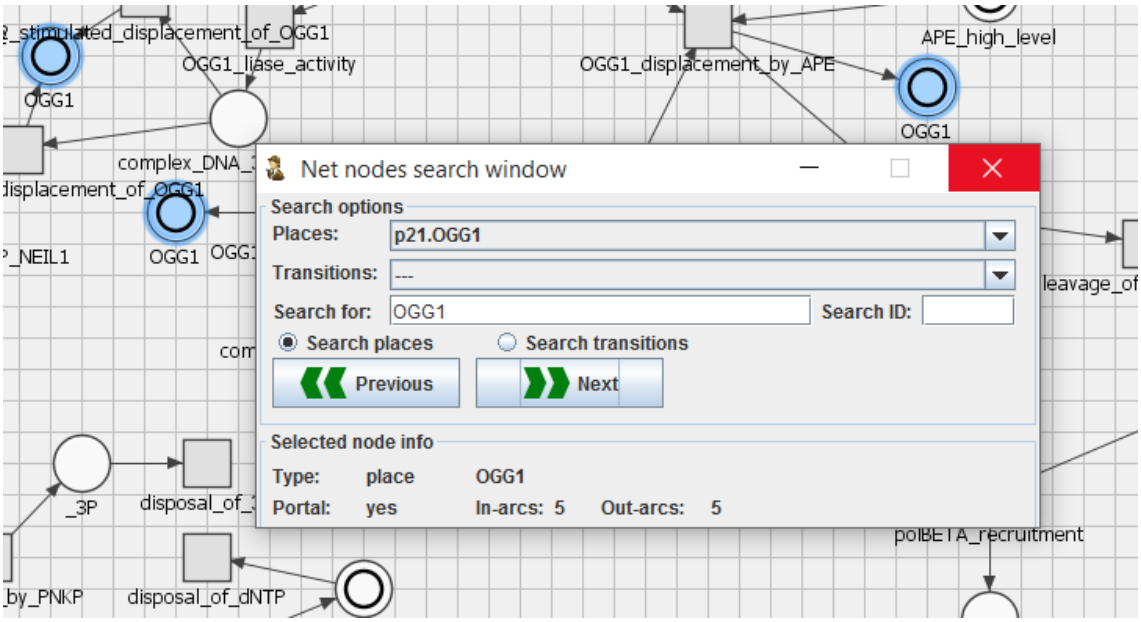


It allows the user to search for the specific place or transitions, giving the user basic information about each in the bottom part of the window.

- **Places** list – one can choose a specific place by name from the list
- **Transitions** list – similar as above for the transitions.
- **Search for** – in this field one can put a name or part of the name of a given place or transition. After pressing Enter program will search for the first matching element.
- **Search ID:** - similar as above, but if will switch the specific list into an element with the provided ID.
- **Search places** and **Search transitions** allow specifing the search targets for the two previously described fields (Search for / Search ID).

- **Previous** and **Next** buttons are connected with the **Search for** field. If more than one element matches the provided name, these button will iterate these elements.
- In the bottom part of the window, in panel **Selected node info** simple data about chosen found element will be provided.

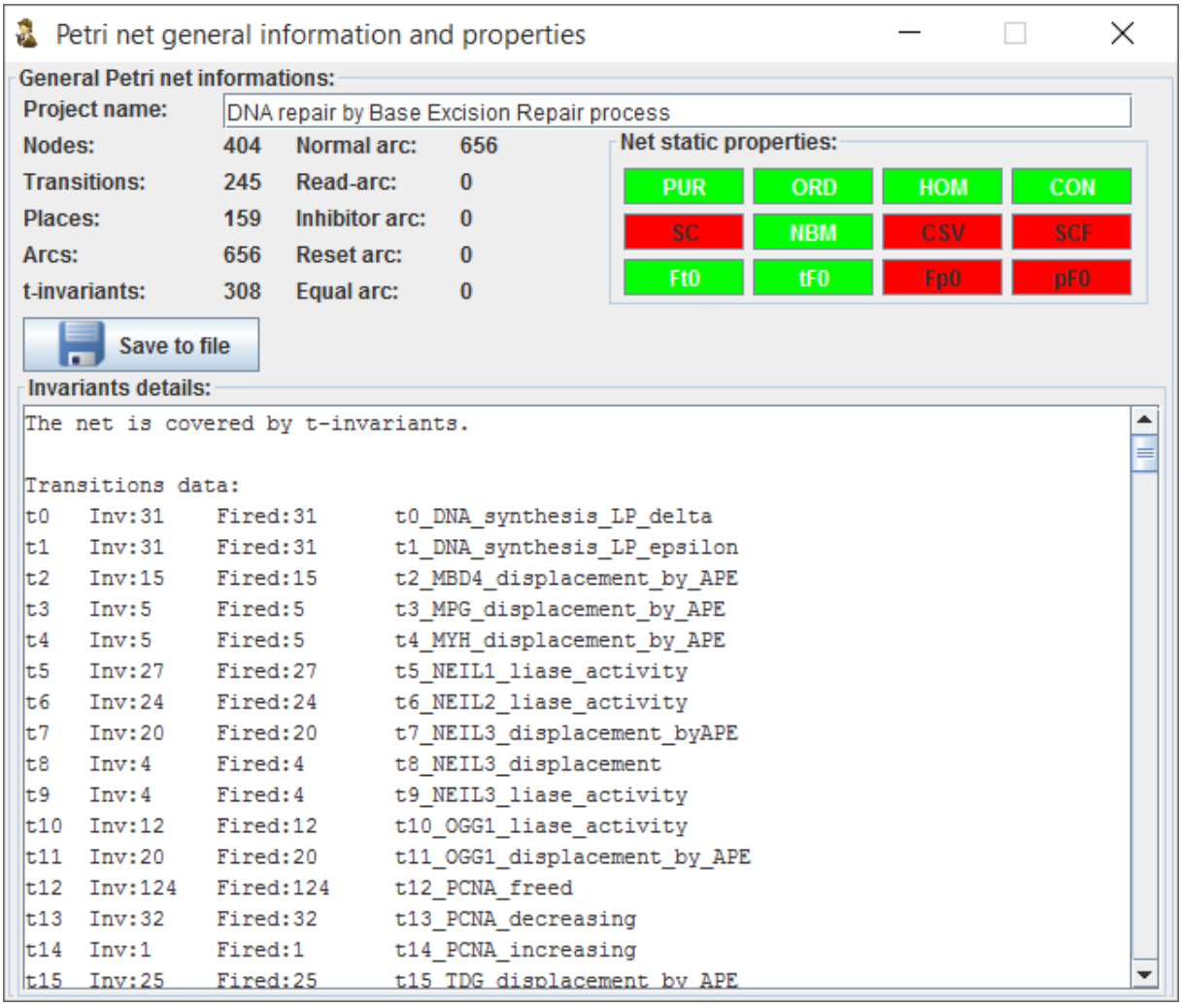
An example is given in the picture:



In the above picture an example for search by name is given. **Search places** button is active and ‘OGG1’ in **Search for** has been written. First found element is also enabled in the net, also one can notice (in real program, not in the picture) that the net will move in order to center the found element. This feature can be quite handy in the bigger nets. If the element is a portal, all its graphical nodes will also be marked in color. If **Next** button will be clicked, next place (if exists) with a phrase ‘OGG1’ in its name will be found and selected (analogously for the **Previous** button).

5.2 Net properties

This module can be also activated with a shortcut **Ctrl+P**. Picture below shows the example of the window, for some, already created net.



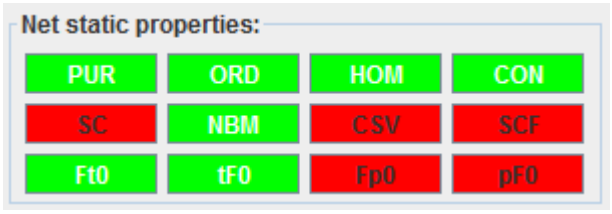
The window will be now explained on the specific net example, obviously it will look different and show different data for different nets.

The main data window shows are:

- **Project name** - it can be changed here
- **Nodes** - sum of places and transitions
- **Transitions** - sum of all transitions (all types)
- **Places** - sum of all places
- **Arcs** - number of arcs (all types)
- **t-invariants** - number of t-invariants (if already generated / loaded)
- **Normal arc** - number of normal arcs
- **Read-arc** - number of read arcs

- **Inhibitor arc** - number of inhibitor arcs
- **Reset arc** - number of reset arcs
- **Equal arc** - number of equal arcs

In the top right part there is a panel with net structural properties

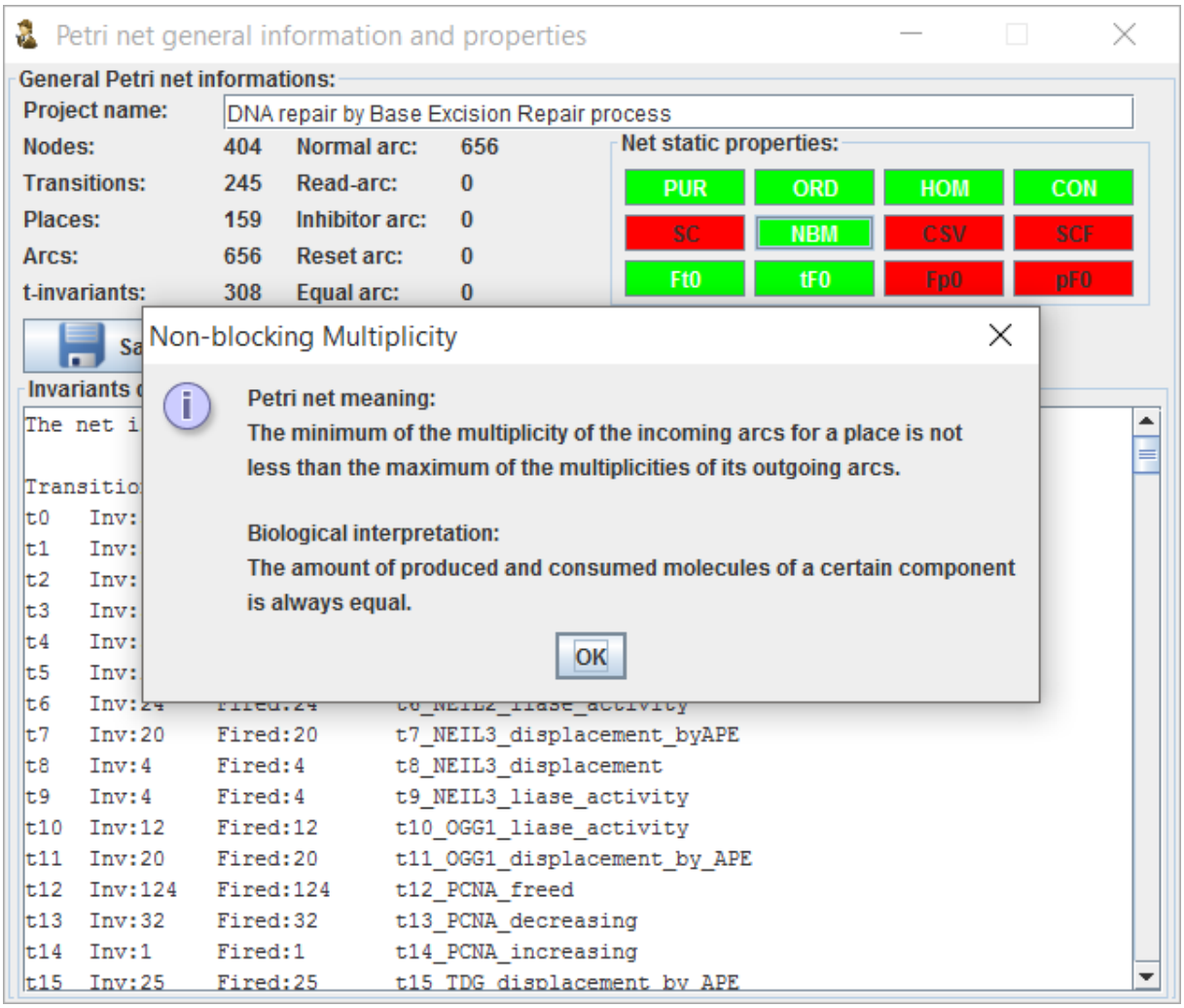


Green color mean the net has the property, red – that it hasn't.

For the moment Holmes checks the following net properties:

- **PUR** - **Pure** – the net does not have read arcs.
- **ORD** - **Ordinary** – all arcs have weight equal to 1.
- **HOM** - **Homogeneous** – all arcs going from a place have the same weight.
- **CON** - **Connected** – there is no pair of vertices which does not have an undirected path in between.
- **SC** - **Strongly Connected** – for every pair of vertices a directed path can be found.
- **NBM** - **Non Blocking Multiplicity** – minimal weight from in arcs going into some place is no less than minimal weight of arcs going out of that place.
- **CSV** - **Conservative** – every transition produces exactly as many tokens in set t● as in consumes from the set ●t
- **SCF** - **Static Conflict Free** – there are no places shared as pre-places for two or more transitions.
- **Ft0** - **Ft = {}, Input Transitions** – transitions without set ●t
- **tF0** - **tF = {}, Output Transitions** – transitions without set t●
- **Fp0** - **Fp = {}, Input Places** – places without set ●p
- **pF0** - **pF = {}, Output Places** – places without set p●

Clicking any property button in panel will shot a brief explanation, as presented in the example below.



With a **Save to file** button one can write basic net properties into a text file.

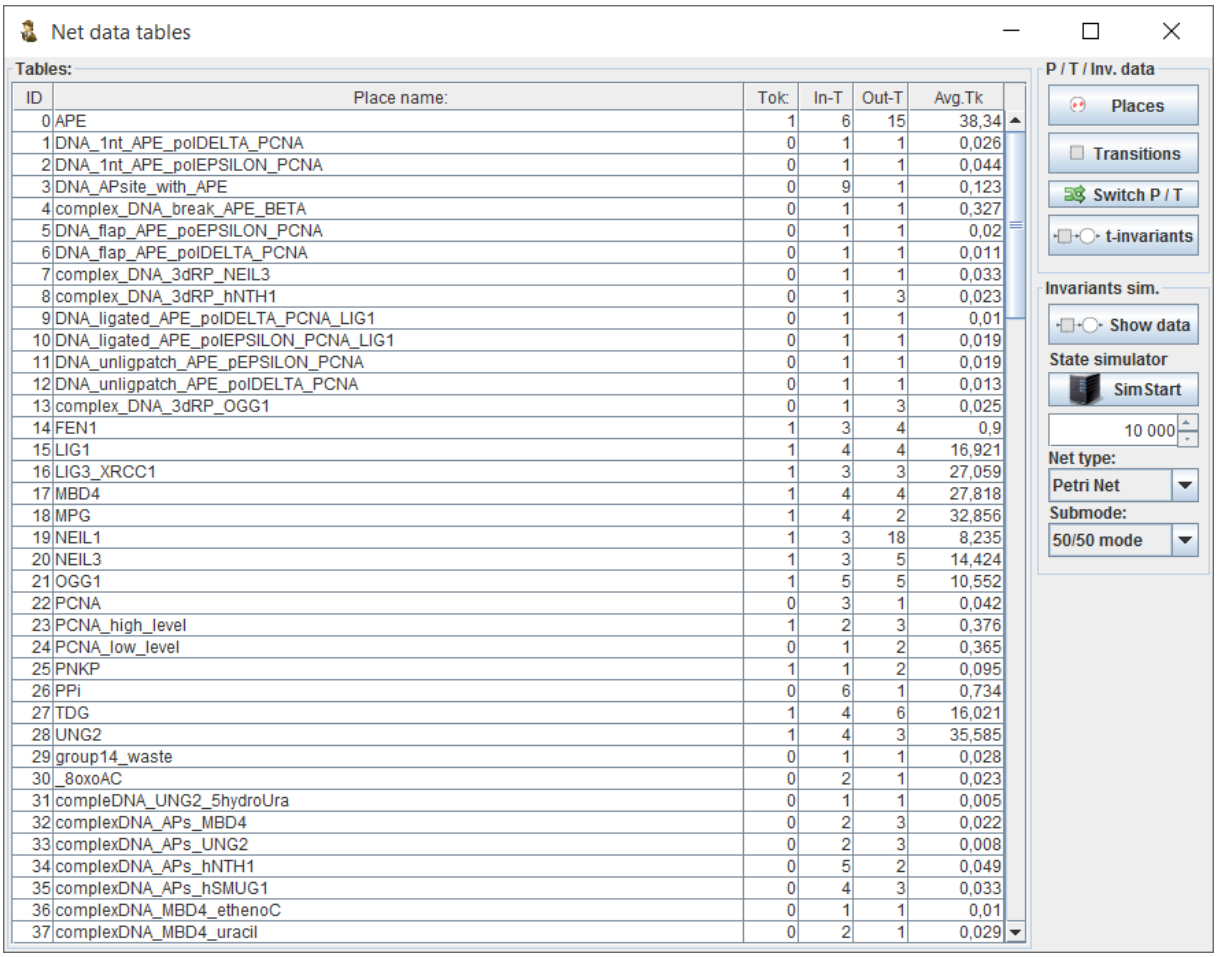
Invariants details will only show information if the invariants have been generated. If it is so, it will start from the information whether the net is covered by them or not. If the net is not covered by the t-invariants, all not covered transition will be written here.

Next comes the list of all covered transitions with some t-invariants related informations.

- **t_x** - where x counted from 0 is the ID of the transition.
- **Inv: xx** - where xx is the number of t-invariants which have t_x in their supports.
- **Fired: xx** - information how many a given transition (in sum) will be fired in all the t-invariants in which supports it is present.
- Last column is the full name of the transition.

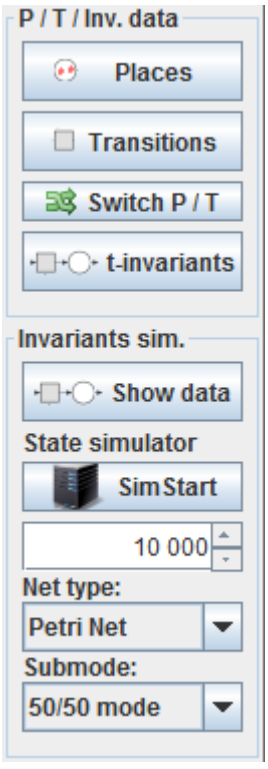
5.3 Net tables

Shortcut for this window is **Ctrl+X**. The window looks like in the example picture:



ID	Place name:	Tok:	In-T	Out-T	Avg.Tk
0	APE	1	6	15	38,34
1	DNA_1nt_APE_polDELTA_PCNA	0	1	1	0,026
2	DNA_1nt_APE_polEPSILON_PCNA	0	1	1	0,044
3	DNA_APsite_with_APE	0	9	1	0,123
4	complex_DNA_break_APE_BETA	0	1	1	0,327
5	DNA_flap_APE_poEPSILON_PCNA	0	1	1	0,02
6	DNA_flap_APE_polDELTA_PCNA	0	1	1	0,011
7	complex_DNA_3dRP_NEIL3	0	1	1	0,033
8	complex_DNA_3dRP_hNTH1	0	1	3	0,023
9	DNA_ligated_APE_polDELTA_PCNA_LIG1	0	1	1	0,01
10	DNA_ligated_APE_polEPSILON_PCNA_LIG1	0	1	1	0,019
11	DNA_unligpatch_APE_pEPSILON_PCNA	0	1	1	0,019
12	DNA_unligpatch_APE_polDELTA_PCNA	0	1	1	0,013
13	complex_DNA_3dRP_OGG1	0	1	3	0,025
14	FEN1	1	3	4	0,9
15	LIG1	1	4	4	16,921
16	LIG3_XRCC1	1	3	3	27,059
17	MBD4	1	4	4	27,818
18	MPG	1	4	2	32,856
19	NEIL1	1	3	18	8,235
20	NEIL3	1	3	5	14,424
21	OGG1	1	5	5	10,552
22	PCNA	0	3	1	0,042
23	PCNA_high_level	1	2	3	0,376
24	PCNA_low_level	0	1	2	0,365
25	PNKP	1	1	2	0,095
26	PPi	0	6	1	0,734
27	TDG	1	4	6	16,021
28	UNG2	1	4	3	35,585
29	group14_waste	0	1	1	0,028
30	_8oxoAC	0	2	1	0,023
31	compleDNA_UNG2_5hydroUra	0	1	1	0,005
32	complexDNA_APs_MBD4	0	2	3	0,022
33	complexDNA_APs_UNG2	0	2	3	0,008
34	complexDNA_APs_hNTH1	0	5	2	0,049
35	complexDNA_APs_hSMUG1	0	4	3	0,033
36	complexDNA_MBD4_ethenoC	0	1	1	0,01
37	complexDNA_MBD4_uracil	0	2	1	0,029

Available button which govern the tables here are as follows:



Places – shows the table for places

Transitions – table for transitions.

Switch P or T – it will switch the IDs in a pair of selected places / transitions (e.g., if place $\mathbf{P_{abc}}$ has ID=11, and place $\mathbf{P_{xyz}}$ has ID =77, then after selecting both of them (with Ctrl button pressed), this button will make $\mathbf{P_{abc}}$ ID=77 and $\mathbf{P_{xyz}}$ ID = 11.

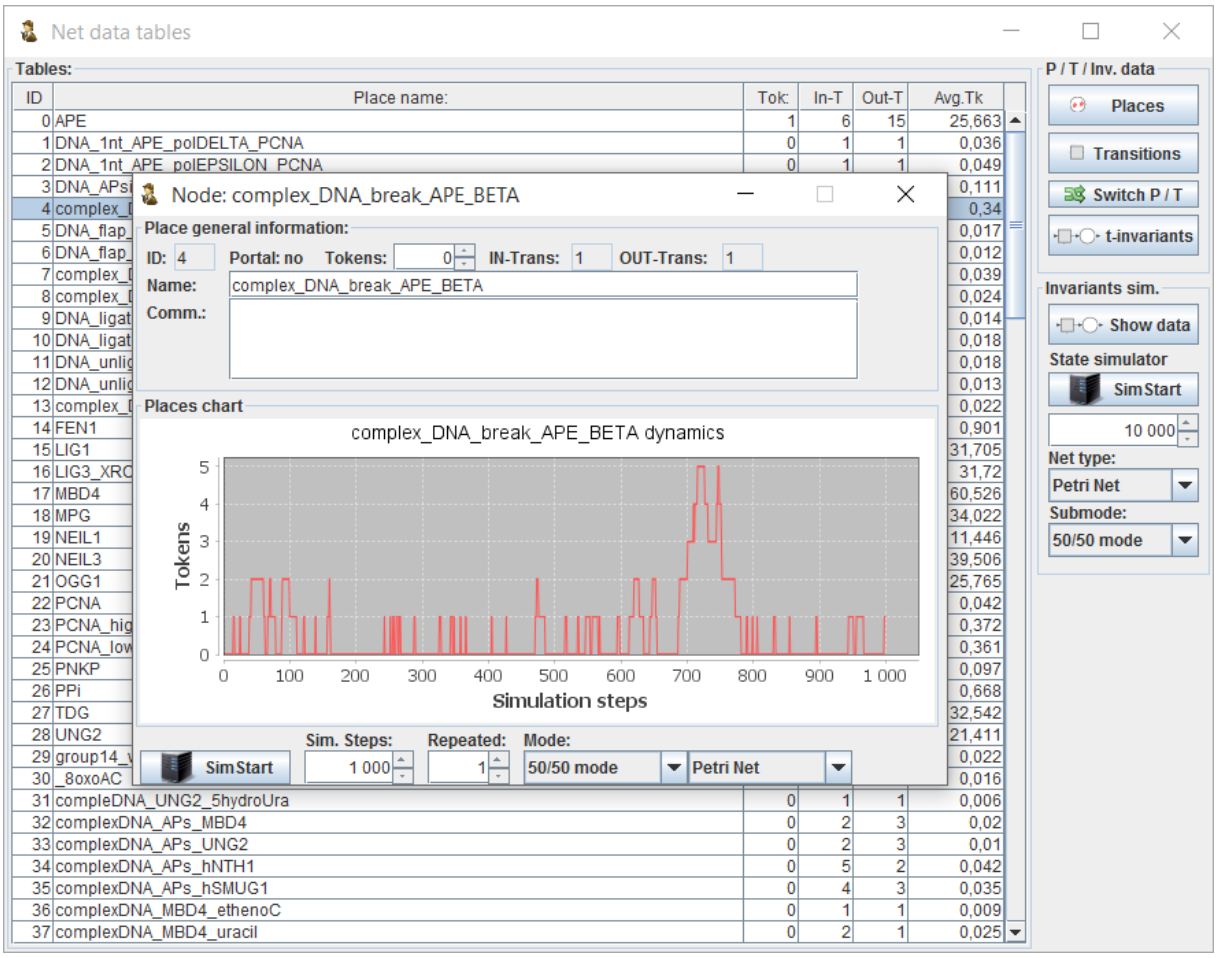
ATTENTION: ID is used to identify transition in various stored data packages in Holmes, therefore using this option can make these data obsolete

t-invariants – shows the table with t-invariants basic data

Subpanel **Invariants sim.** is connected with the second, extended t-invariants table. It will be explained later in the manual, after explanations about simulation algorithms.

5.3.1 Places table

Places button will show table with places, as in the example below:

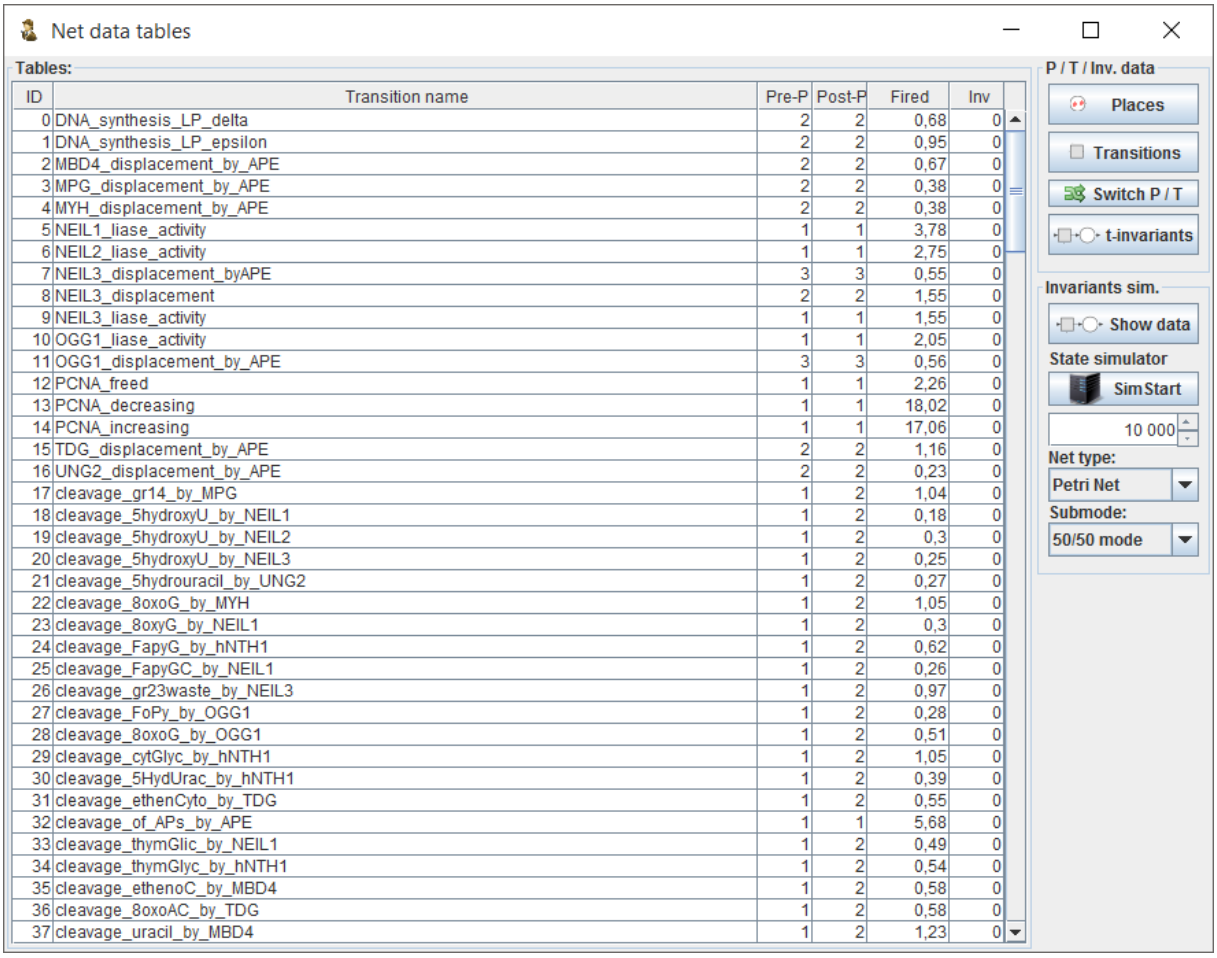


All columns except **Avg.Tk** show data already explained in the manual (places sub-window section). The last mentioned column shows data from fast background simulation from the given net state for 1000 steps. The lower the value, the less tokens the place accumulated in such a simulation. For example value 0.026 means, that in 1000 steps **sum of tokens in all steps** has been equal to 27. 1000.0 would mean, that for the whole 1000-step simulation the place had (in average) 1 token in it.

Clicking any row will show place details, as seen in the above picture.

5.3.2 Transition table

Pressing **Transitions** button will show a table with data concerning the net transitions. Example window is given below:



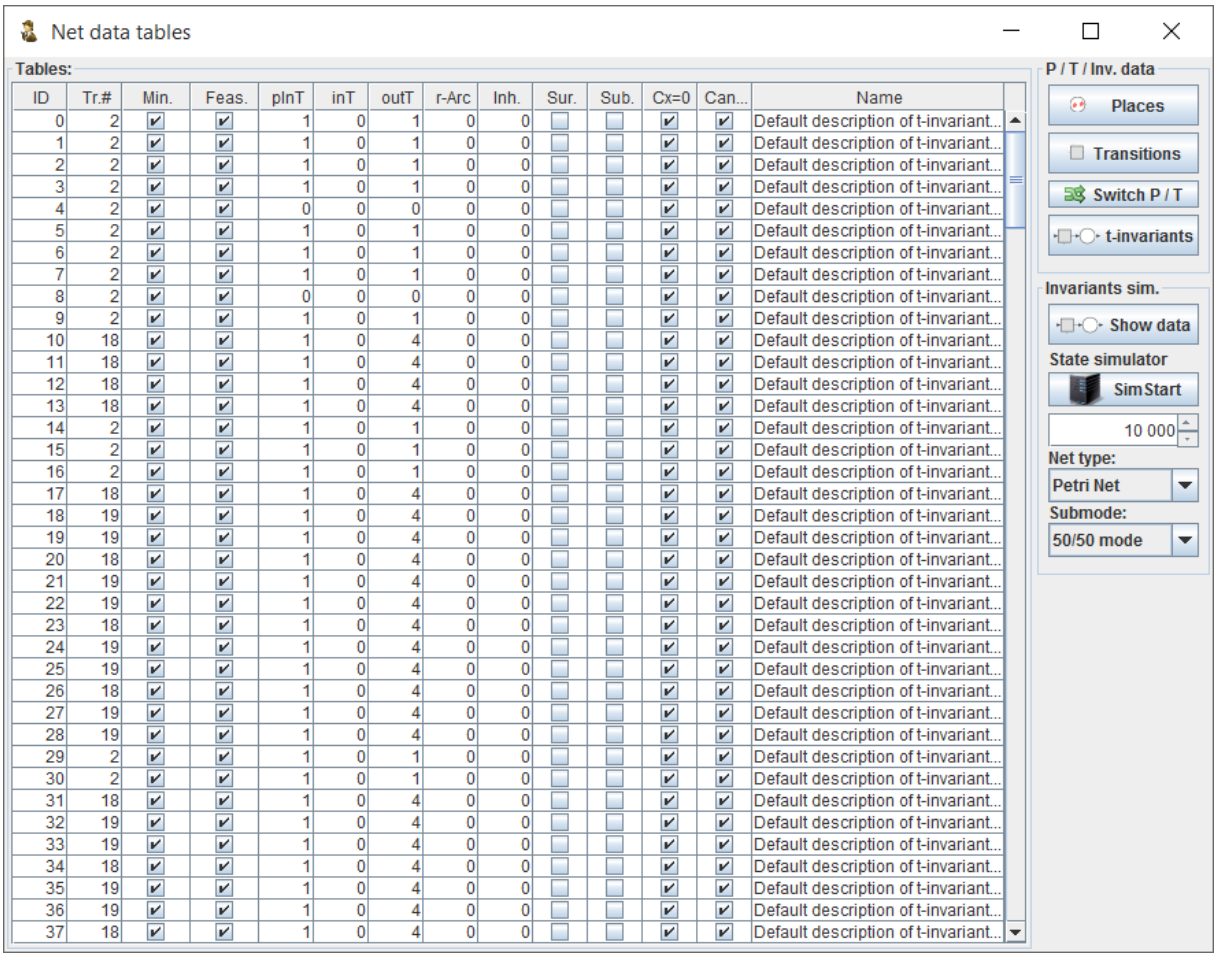
ID	Transition name	Pre-P	Post-P	Fired	Inv
0	DNA_synthesis_LP_delta	2	2	0,68	0
1	DNA_synthesis_LP_epsilon	2	2	0,95	0
2	MBD4_displacement_by_APE	2	2	0,67	0
3	MPG_displacement_by_APE	2	2	0,38	0
4	MYH_displacement_by_APE	2	2	0,38	0
5	NEIL1_liase_activity	1	1	3,78	0
6	NEIL2_liase_activity	1	1	2,75	0
7	NEIL3_displacement_byAPE	3	3	0,55	0
8	NEIL3_displacement	2	2	1,55	0
9	NEIL3_liase_activity	1	1	1,55	0
10	OGG1_liase_activity	1	1	2,05	0
11	OGG1_displacement_by_APE	3	3	0,56	0
12	PCNA_freed	1	1	2,26	0
13	PCNA_decreasing	1	1	18,02	0
14	PCNA_increasing	1	1	17,06	0
15	TDG_displacement_by_APE	2	2	1,16	0
16	UNG2_displacement_by_APE	2	2	0,23	0
17	cleavage_gr14_by_MPG	1	2	1,04	0
18	cleavage_5hydroxyU_by_NEIL1	1	2	0,18	0
19	cleavage_5hydroxyU_by_NEIL2	1	2	0,3	0
20	cleavage_5hydroxyU_by_NEIL3	1	2	0,25	0
21	cleavage_5hydrouracil_by_UNG2	1	2	0,27	0
22	cleavage_8oxoG_by_MYH	1	2	1,05	0
23	cleavage_8oxyG_by_NEIL1	1	2	0,3	0
24	cleavage_FapyG_by_hNTH1	1	2	0,62	0
25	cleavage_FapyGC_by_NEIL1	1	2	0,26	0
26	cleavage_gr23waste_by_NEIL3	1	2	0,97	0
27	cleavage_FoPy_by_OGG1	1	2	0,28	0
28	cleavage_8oxoG_by_OGG1	1	2	0,51	0
29	cleavage_cytGlyc_by_hNTH1	1	2	1,05	0
30	cleavage_5HydUrac_by_hNTH1	1	2	0,39	0
31	cleavage_ethenCyto_by_TDG	1	2	0,55	0
32	cleavage_of_APs_by_APE	1	1	5,68	0
33	cleavage_thymGlic_by_NEIL1	1	2	0,49	0
34	cleavage_thymGlyc_by_hNTH1	1	2	0,54	0
35	cleavage_ethenoC_by_MBD4	1	2	0,58	0
36	cleavage_8oxoAC_by_TDG	1	2	0,58	0
37	cleavage_uracil_by_MBD4	1	2	1,23	0

Pre-P and Post-P columns gives information about each transition pre- and post-places. Fired column tells about average transition firing in the fast, background simulation (1000 steps), e.g., 0.18 means that in the simulation transition had 18% chance of firing. Last column – Inv tells in how many t-invariants support a transition is, if they are computed.

As before, clicking any row will activate transition data window already explained in the previous chapter of the manual.

5.3.3 Simple t-invariants table

Pressing **t-invariants** will show the table about t-invariants. They will be briefly explained here, but in order to fully understand some of them, it is advised to read the theory about the t-invariants generator later in the manual. The example window of such a table is given below:



ID	Tr.#	Min.	Feas.	pInT	inT	outT	r-Arc	Inh.	Sur.	Sub.	Cx=0	Can...	Name
0	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
1	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
2	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
3	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
4	2	✔	✔	0	0	0	0	0			✔	✔	Default description of t-invariant...
5	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
6	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
7	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
8	2	✔	✔	0	0	0	0	0			✔	✔	Default description of t-invariant...
9	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
10	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
11	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
12	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
13	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
14	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
15	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
16	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
17	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
18	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
19	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
20	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
21	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
22	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
23	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
24	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
25	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
26	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
27	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
28	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
29	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
30	2	✔	✔	1	0	1	0	0			✔	✔	Default description of t-invariant...
31	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
32	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
33	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
34	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
35	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
36	19	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...
37	18	✔	✔	1	0	4	0	0			✔	✔	Default description of t-invariant...

Columns from left to right are as follows:

- **ID** - t-invariant number counted from 0
- **Tr.#** - number of transition in the t-inv. support
- **Min.** - is the t-inv. minimal
- **Feas.** - is it *feasible*?
- **pInT** - *pure Input Transition* – how many transition in the t-inv. support does not have ANY ingoing arcs / pre-places
- **inT** - *Input Transitions* – similar as above, but the read arcs and inhibitor arcs are not counted as 'arcs'. The motivation here is, that such arc will not 'provide' any tokens, they can only influence the transition firing chances.
- **outT** - *Output Transitions* – analogously as above
- **r-Arc** - how many read arcs, also double arcs are counted
- **Inh.** - Inhibitor arcs connected with transition in the t-inv. support
- **Sur.** - is it sur-invariant?
- **Sub.** - is it sub-invariant?
- **Cx=0** - is it real t-invariant (nor sur or sub)
- **Can.** - is the t-inv. canonical, i.e., maximal common divider of t-inv. non-zero entries is = 1
- **Name** - name/function of the t-inv

5.3.4 t-invariants extended table

This window is strongly connected with the simulation algorithm. Pressing **Show Data** will show the table. Example picture is given below.

Tables:

ID	Trans. #	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20	t21	t22	t23	t24	t25	t26	t27
135	18						1(2.15%)																						
136	20				1(0.52%)									1(1.69%)					1(0.83%)										
137	20				1(0.52%)									1(1.69%)					1(0.83%)										
138	18				1(0.52%)														1(0.83%)										
139	20	1(0.48%)			1(0.52%)									1(1.69%)					1(0.83%)										
140	21		1(0.7%)		1(0.52%)									1(1.69%)	1(21%)				1(0.83%)										
141	20					1(0.56%)								1(1.69%)										1(0.95%)					
142	20					1(0.56%)								1(1.69%)										1(0.95%)					
143	18					1(0.56%)																		1(0.95%)					
144	20	1(0.48%)				1(0.56%)								1(1.69%)										1(0.95%)					
145	21		1(0.7%)			1(0.56%)								1(1.69%)	1(21%)									1(0.95%)					
146	20													1(1.69%)				1(0.65%)					1(0.55%)						
147	20													1(1.69%)				1(0.65%)											
148	21								1(0%)					1(1.69%)														1(0.07%)	
149	21								1(0%)					1(1.69%)								1(0.18%)							
150	21								1(0%)					1(1.69%)															
151	21								1(0%)					1(1.69%)															
152	20			1(0.64%)										1(1.69%)															
153	20			1(0.64%)										1(1.69%)															
154	20			1(0.64%)										1(1.69%)															
155	20													1(1.69%)			1(0.53%)												
156	20													1(1.69%)			1(0.53%)												
157	20													1(1.69%)			1(0.53%)												
158	20													1(1.69%)			1(0.53%)												
159	20													1(1.69%)			1(0.53%)												
160	21												1(0%)	1(1.69%)															
161	21												1(0%)	1(1.69%)															1(0.19%
162	21												1(0%)	1(1.69%)															
163	21												1(0%)	1(1.69%)															
164	20													1(1.69%)															
165	20													1(1.69%)															
166	20													1(1.69%)															
167	20													1(1.69%)															
168	20													1(1.69%)															
169	20													1(1.69%)															
170	21													1(1.69%)															
171	21													1(1.69%)															
172	21													1(1.69%)															
173	21													1(1.69%)												1(0.17%)			
174	21													1(1.69%)															
175	20													1(1.69%)				1(0.65%)					1(0.55%)						
176	20													1(1.69%)				1(0.65%)											
177	21								1(0%)					1(1.69%)														1(0.07%)	
178	21								1(0%)					1(1.69%)								1(0.18%)							
179	21								1(0%)					1(1.69%)															
180	21								1(0%)					1(1.69%)															
181	20			1(0.64%)										1(1.69%)															
182	20			1(0.64%)										1(1.69%)															
183	20			1(0.64%)										1(1.69%)															
184	20													1(1.69%)			1(0.53%)												
185	20													1(1.69%)			1(0.53%)												
186	20													1(1.69%)			1(0.53%)												
187	20													1(1.69%)			1(0.53%)												

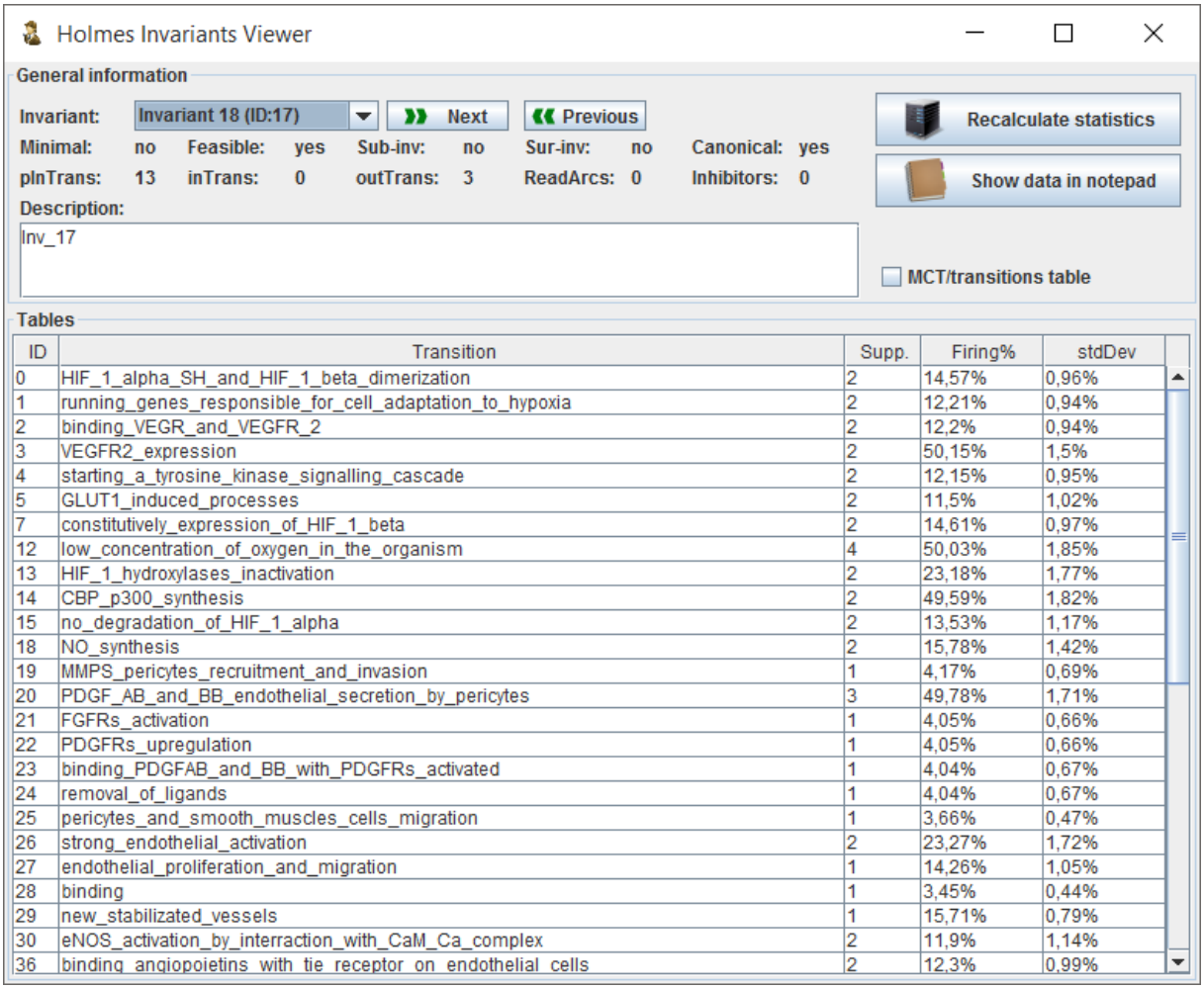
First column is the t-invariants ID (number), second gives the number of transitions in its support. All the next columns are connected with the net transition, and the percentage value given in them is the average chance for every transition to fire (from the simulation results). For example, if in a cell is **1(16.36%)** it means, that such a transition is used once in the t-invariants (its non-zero entry is equal to 1, this of course can be any other positive integer value, cf. invariants generator theory) and its average chance for firing is 16.36%. This can give the information about the chances of firing for all the t-invariants transitions, i.e., the average chance for the whole inv. to perform its function.

Columns painted in dark grey informs about the transition that did not fire even once (background simulation steps = 10 000 for this table). Button Start Sim (with icon) will make the simulation again and the values can change. Other simulation modes will be explained later.

T-invariant row painted light grey means, that at least one its transition did not fire in the simulation (its column is painted dark grey).

5.4 t-invariants window

Such a window gives some detailed information about every invariant.



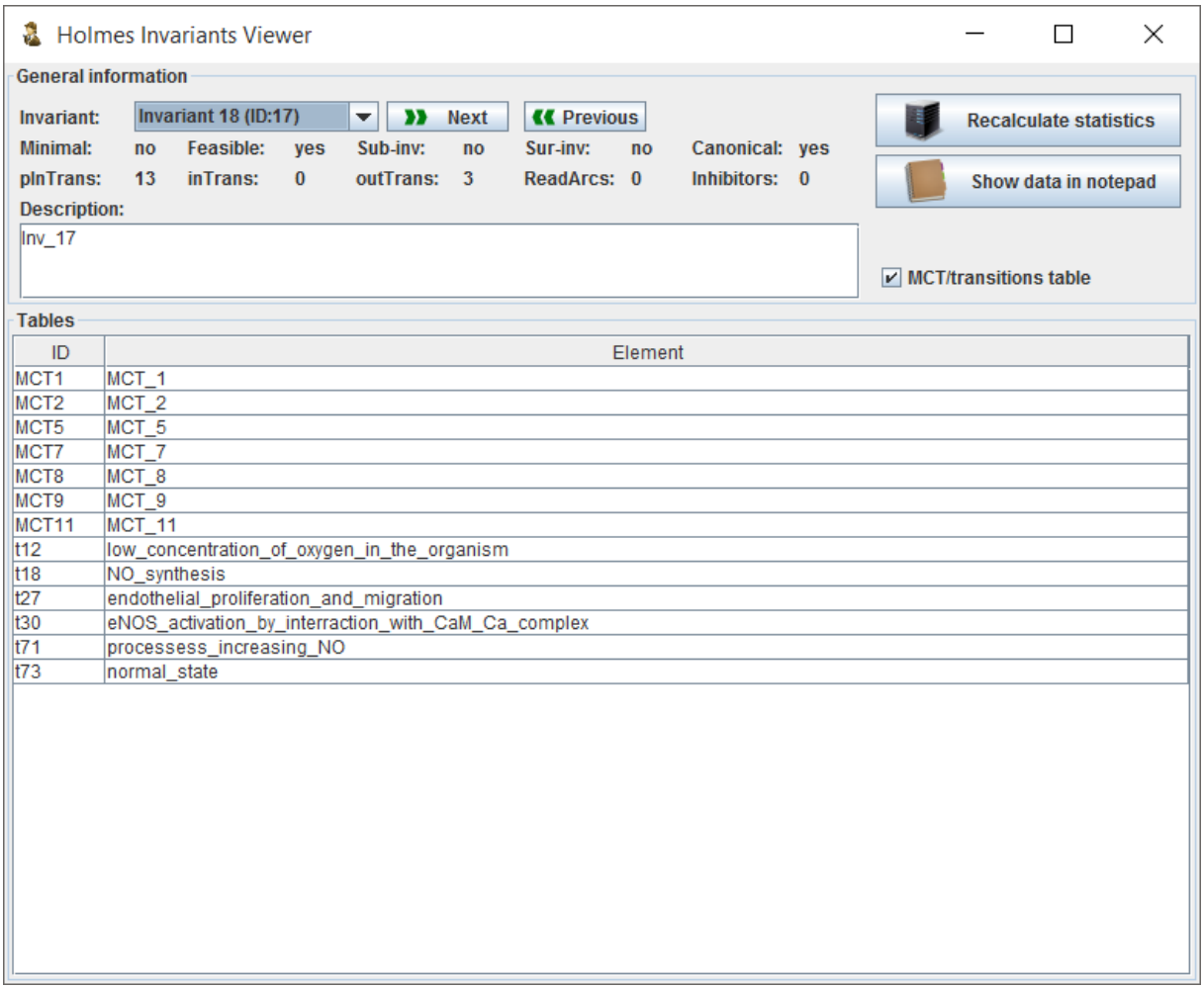
ID	Transition	Supp.	Firing%	stdDev
0	HIF_1_alpha_SH_and_HIF_1_beta_dimerization	2	14,57%	0,96%
1	running_genes_responsible_for_cell_adaptation_to_hypoxia	2	12,21%	0,94%
2	binding_VEGR_and_VEGFR_2	2	12,2%	0,94%
3	VEGFR2_expression	2	50,15%	1,5%
4	starting_a_tyrosine_kinase_signalling_cascade	2	12,15%	0,95%
5	GLUT1_induced_processes	2	11,5%	1,02%
7	constitutively_expression_of_HIF_1_beta	2	14,61%	0,97%
12	low_concentration_of_oxygen_in_the_organism	4	50,03%	1,85%
13	HIF_1_hydroxylases_inactivation	2	23,18%	1,77%
14	CBP_p300_synthesis	2	49,59%	1,82%
15	no_degradation_of_HIF_1_alpha	2	13,53%	1,17%
18	NO_synthesis	2	15,78%	1,42%
19	MMPS_pericytes_recruitment_and_invasion	1	4,17%	0,69%
20	PDGF_AB_and_BB_endothelial_secretion_by_pericytes	3	49,78%	1,71%
21	FGFRs_activation	1	4,05%	0,66%
22	PDGFRs_upregulation	1	4,05%	0,66%
23	binding_PDGFAB_and_BB_with_PDGFRs_activated	1	4,04%	0,67%
24	removal_of_ligands	1	4,04%	0,67%
25	pericytes_and_smooth_muscles_cells_migration	1	3,66%	0,47%
26	strong_endothelial_activation	2	23,27%	1,72%
27	endothelial_proliferation_and_migration	1	14,26%	1,05%
28	binding	1	3,45%	0,44%
29	new_stabilizated_vessels	1	15,71%	0,79%
30	eNOS_activation_by_interraction_with_CaM_Ca_complex	2	11,9%	1,14%
36	binding_angiopoietins_with_tie_receptor_on_endothelial_cells	2	12,3%	0,99%

t-invariant can be chosen from the list, and also by using button **Next** and **Previous**. Below the button are detailed information about invariant transitions, as explained in 5.3.3 chapter.

In **Description** field one can assign a comment for every t-invariant.

Table show the average chances of firing for the t-invariant transitions, as explained before. This time each simulation has 1000 steps, and 20 simulation are computed. The average value of firing chances is given, with the standard deviation as well.

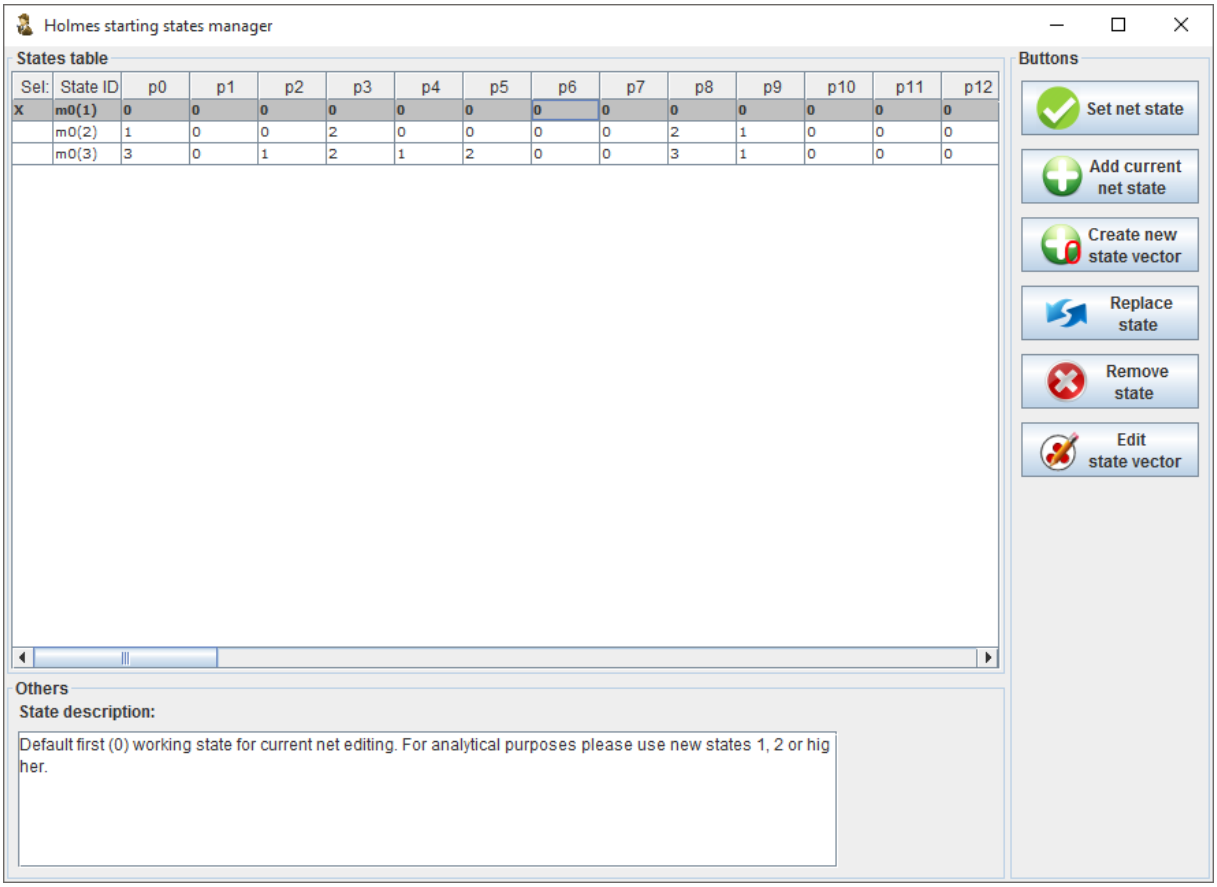
Button **MCT/transitions table** will switch the table into MCT sets view, where non-trivial sets (having more than 1 transition) and single transition outside them are given.



Show data in Notepad button will export the information about t-invariant into a text file.

5.5 Initial states management window

Window is available in the **Net** menu, by shortcut **CTRL+M** and in some other relevant windows of Holmes. Example window is given below.



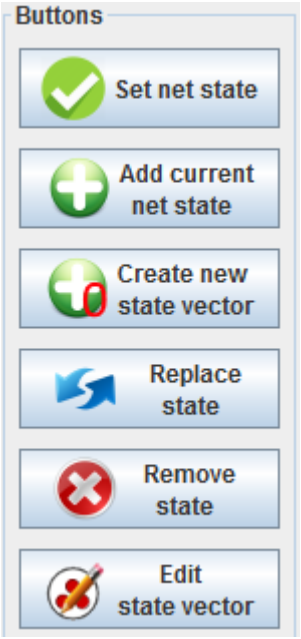
Its goal it to help manage net initial/starting states, e.g. for simulations.

In the main table stored states are given (they are saved in the Holmes project files). It should be noted, that states depend on the set of places. Adding or removing a place from the net will alter stored states (automatically, by Holmes, i.e. columns can be removes/added with 0 value by default).

Currently chosen stated is marked by **X** in the first column. It does not mean it is the actual current state, but only that **this state (vector) will be used by the simulation algorithms**.

ATTENTION! First state in the table is a specific one . If it is active (marked X), every change in tokens in places (in net editor!) will be automatically written into such a state (only to the first one/row in the table). If, for example, second state is chosen and the user will change the tokens in the net editor (**not** in the state manager directly), such a change will not be marked (automatically) in the states manager, but it can be activated by the user (explained later).

Simply put, state m0(1) can be changed directly in the net editor (and only this default first row state in the table). Every state, also that one, can be changed in the states manager (outside of net editor). It can be done by simply changing any cell except the first two columns. Assigning any negative value will be automatically corrected by changing it to 0.



Available buttons are as follows:

Set net state – set the chosen/clicked state (in the table) as the net current one. **All places in the net will acquire tokens in numbers defined by the selected m0 state.**

Add current net state – makes the current tokens distribution in the net as the next state (will add the state to the table).

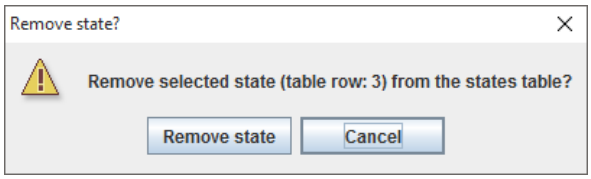
Create new state vector - creates new state with all entries set to 0.

Replace state – when a state row is clicked in table, this button will replace its values by the values of tokens currently residing in the net places in the editor.

Remove state – removes clicked state from the table.

Edit state vector – another window for changing tokens values in the state.

It should be noted, that chosen/clicked state is different from the selected state. The latter is marked by X, the first ones refers to clicking a row in the table (row will be painted light grey). Buttons described here works for clicked states. For example when user click the third (m0(3)) row/state and click **Remove state** button, a window will appear:



It should be noted, that in this example still the first state (m0(1)) is selected (marked X), but the third row has be clicked, and for this state/row the button refers to.

States are stored in the Holmes project file.

Button  will open a window similar to the example below.

Holmes state editor

State vector data

State vector ID: 0

Default first (0) working state for current net editing. For analytical purposes please use new states 1, 2 or higher.

Set tokens in all places

New tokens number: 0

State vector table

ID	Place name	Tokens
0	VCAM1	0.0
1	early_leukocytes_in_inflammed_endothelium	0.0
2	monocytes_in_intima_media	0.0
3	CCR2	0.0
4	intimal_macrophages	0.0
5	macrophages_with_scavanger_receptors_SRA_or_CD36	0.0
6	lipoprotein_particles_modified_by_oxidation	0.0
7	scavanger_receptors_modyfied_lipoprotein_particles_complex	0.0
8	modyfied_by_glycation_lipoprotein_particles	0.0
9	M_CSF	0.0
10	macrophage_foam_cell	0.0
11	MCP1	0.0
12	apoptotic_bodies	0.0
13	ICAM1	0.0
14	selectin_E	0.0
15	highly_toxic_hydroxyl_radical	0.0
16	iron_ion_Fe3	0.0
17	iron_ions_Fe2	1.0
18	superoxide_radical_anion	0.0
19	hydrogen_peroxide	0.0
20	hydroxide_ion	0.0
21	newly_formed_NADPH_oxidase_enzyme_complex	0.0
22	superoxide_dismutase_SOD	0.0
23	peroxynitrite_ONOO_anion	0.0
24	oxygen	0.0
25	L_arginine	0.0
26	citruline	1.0
27	iNOS	0.0
28	nitric_oxide	0.0

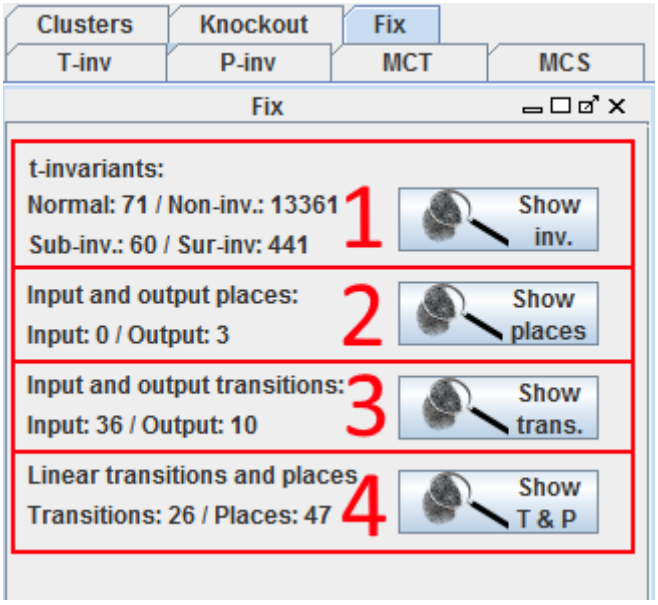
One can change tokens for every places for a given state, assign same value to all places (Set tokens in places button with the text box in top right corner). Also a state can be assigned with a comment.

5.6. Holmes interface section 6 sub-windows

In Holmes main window, its bottom right part has been denoted as section number 6. There are a few tabs there connected with various modules: **Clusters**, **Knockout**, **Fix**, **T-inv**, **P-inv**, **MCT**, **MCS** oraz **qSim**. In this chapter (5) four of them: **Fix**, **T-inv**, **P-inv** and **MCT** will be explained, as they are connected with topics of this chapter.

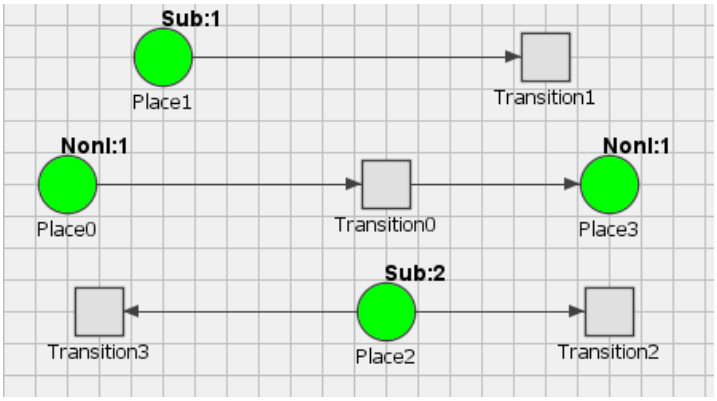
5.6.1 Net fix/checking tab

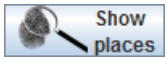
At the moment, there are four different options in this tab, offering various option for net structure checking. One of them (first) is dependent on the t-invariants calculations. The tab is given in picture below:



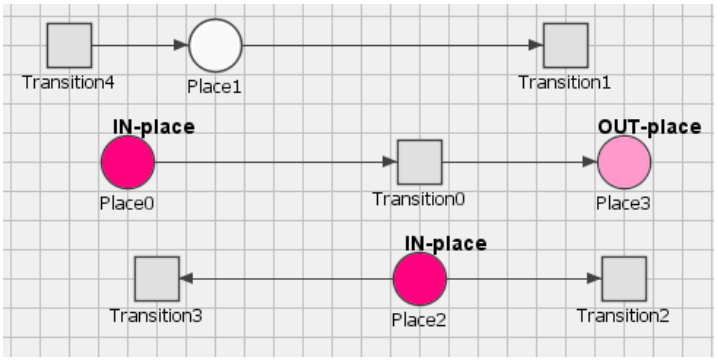
Button 

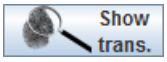
will recalculate the presented values in the tab on the left, based on the t-inv. set. Also, every place connected with the non-proper t-invariants (if exist, sub-, sur- or non-inv vectors, cf. invariants generator theory) will mark places where the problem exists. For the moment, without the theory, let us just say that such non-invariants (if their transition will fire) will produce excessive tokens or consume more of them than they produce (in other words for such vectors equation Cx=0 does not hold). All places where such a problem exists will be marked. For example (picture below), Place1 is marked Sub:1, because in the net there is one sub-invariant, which firing will take token from Place1, but within such sub-inv. there are no transitions that could produce the consumed token.



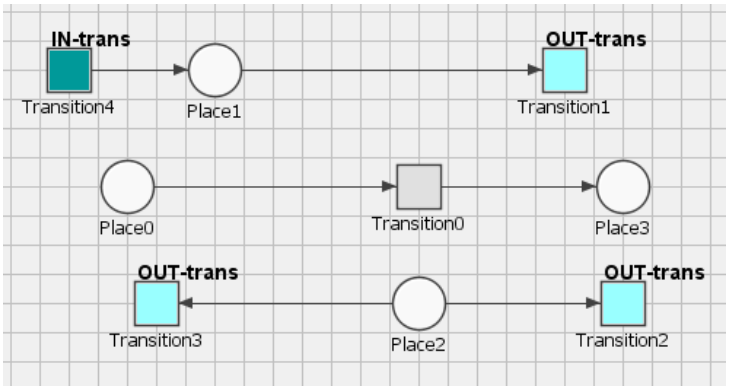
Button: 

mark all input and output places of the net (respectively, without sets •p and p•). Example is given in the picture:



Button 

marks all input and output transitions (respectively without sets •t and t•). Example is given in the picture below:



Before last button: 

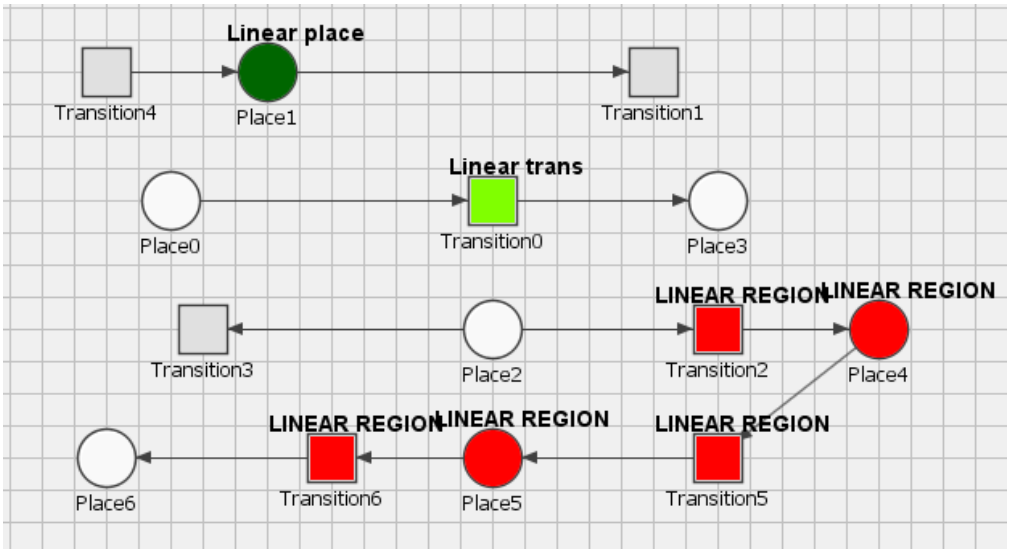
will be explained, some definitions are necessary.

Linear Transition we call a transition having exactly one ingoing and one outgoing arc.

Linear Place we call a place having exactly one ingoing and one outgoing arc.

Linear region we call an area of the net where at least 2-elements chain of linear places/transitions exists.

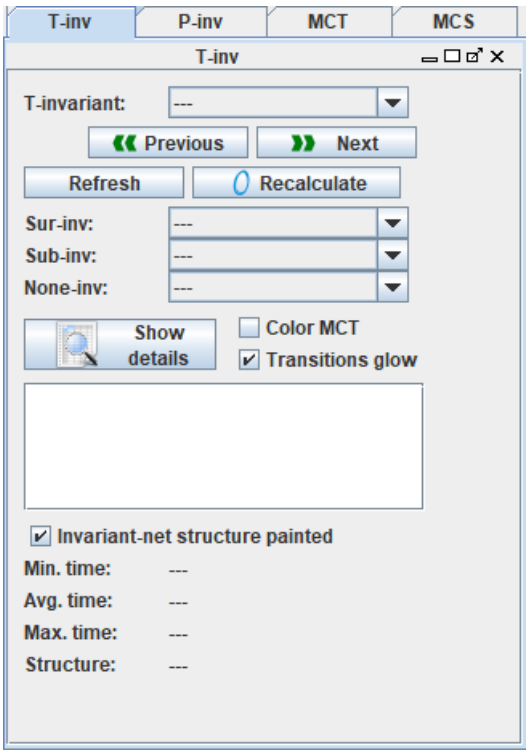
Example effect of such button is given in the picture below:



Single linear places and transition (green) are usually not a problem. However, linear region can indicate an area which could be easily reduced to one transition and/or place having the same function for the e.g. t-invariants perspective. In other words, the flow of tokens in such region is and always be the same, no matter how long such a region will be. Therefore it can possibly be simplified and replaced by a simpler structure.

5.6.2 t-invariants tab

Besides window explained in chapter 5.4, this tab allows among other things to draw invariant structure on the net. It looks like in the picture below:



T-invariant – a specific t-invariant can be chosen from this box (keyboard arrows up/down work when this box is selected).

Previous / **Next** – other way of changing active t-invariant.

Refresh – if invariants are already calculated, it will refresh comboboxes with sur/sub/none vectors.

Recalculate – can take longer, forces t-invariants calculations, then activate Refresh function as well.

Sur/Sub/None – these comboboxes allow choosing non canonical vectors (if exist).

Show details – open window explained in 5.4.

After choosing any vector, its structure will be painted on the net. Each painted transition will have value

assigned from the t-invariant non-zero entry it corresponds to.

Color MCT – if checked, every transition belonging to the non-trivial MCT set will be marked in MCT set color on the t-invariant structure (see 5.6.4).

Transitions glow – another graphical feature, transitions slightly glow.

In the first combobox (T-invariants) there are few interesting option on the end of the list:

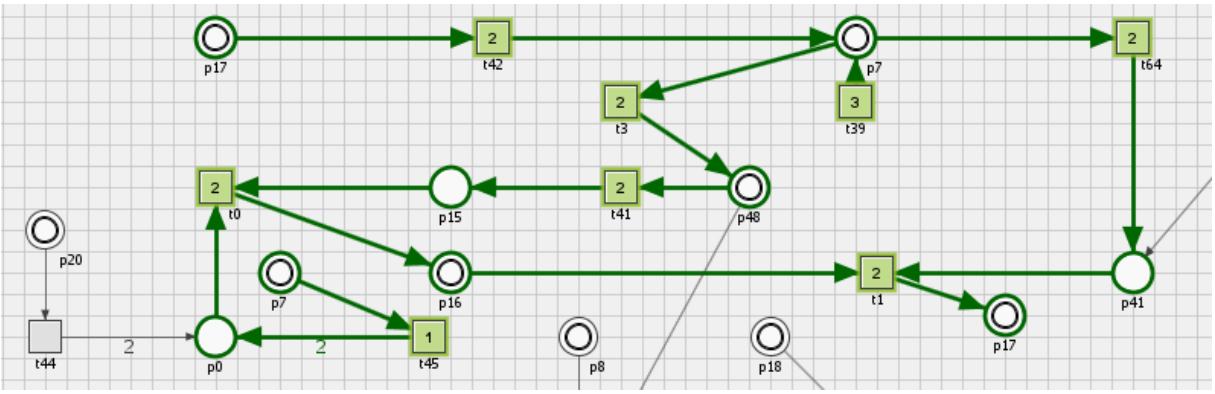
null transitions - marks all transitions not covered by the (real, classical) t-invariants.

inv/trans frequency – for every transition it shows the number of t-invariants, the transition belong to.

For cleaning colors these button make, one can:

- Choose first „---” combobox entry.
- Choose icon  from toolbar (section 3 button C5, chapter 2.2)
- Chose „**Clear Colors**” from context menu in the editor

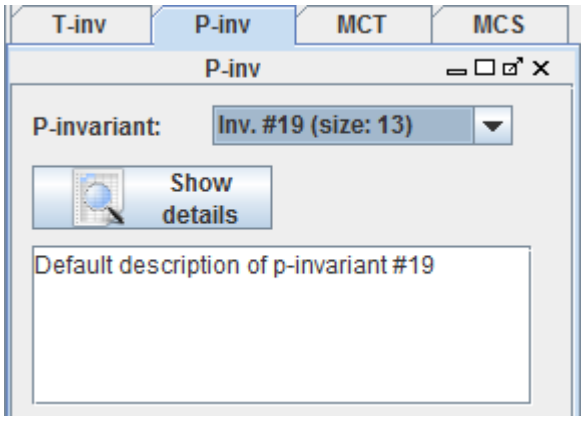
If last option is checked (default) „**Invariant-net structure painted**” it will paint the t-invariant structure. The example is given below:



Obviously, transition painted green belong to the t-invariant support. All places marked green have at least one input and one output transitions belonging to the support.

5.6.3 p-invariants tab

Similar to the t-invariants explained already.

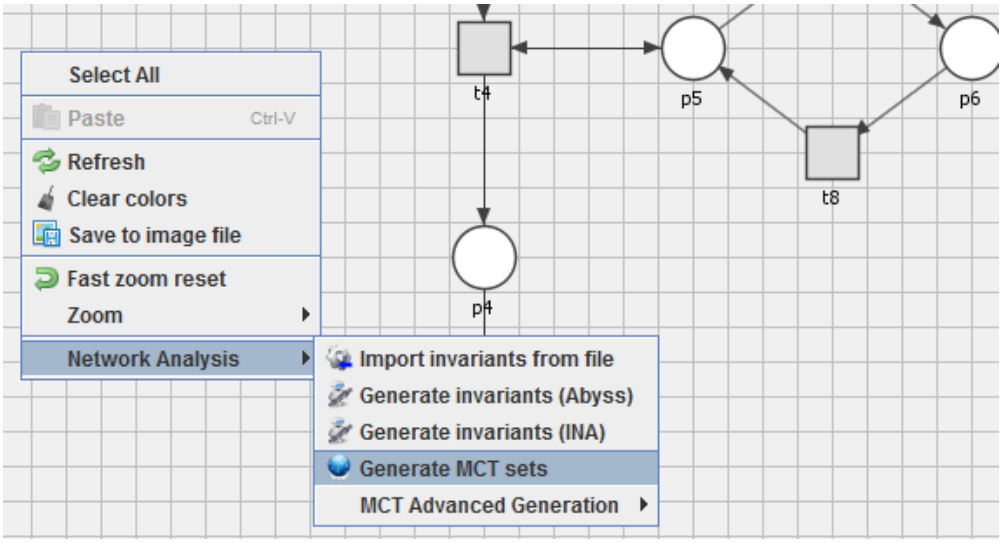


At the moment possible option is to check the invariant structure and obtain its details in the separate window.

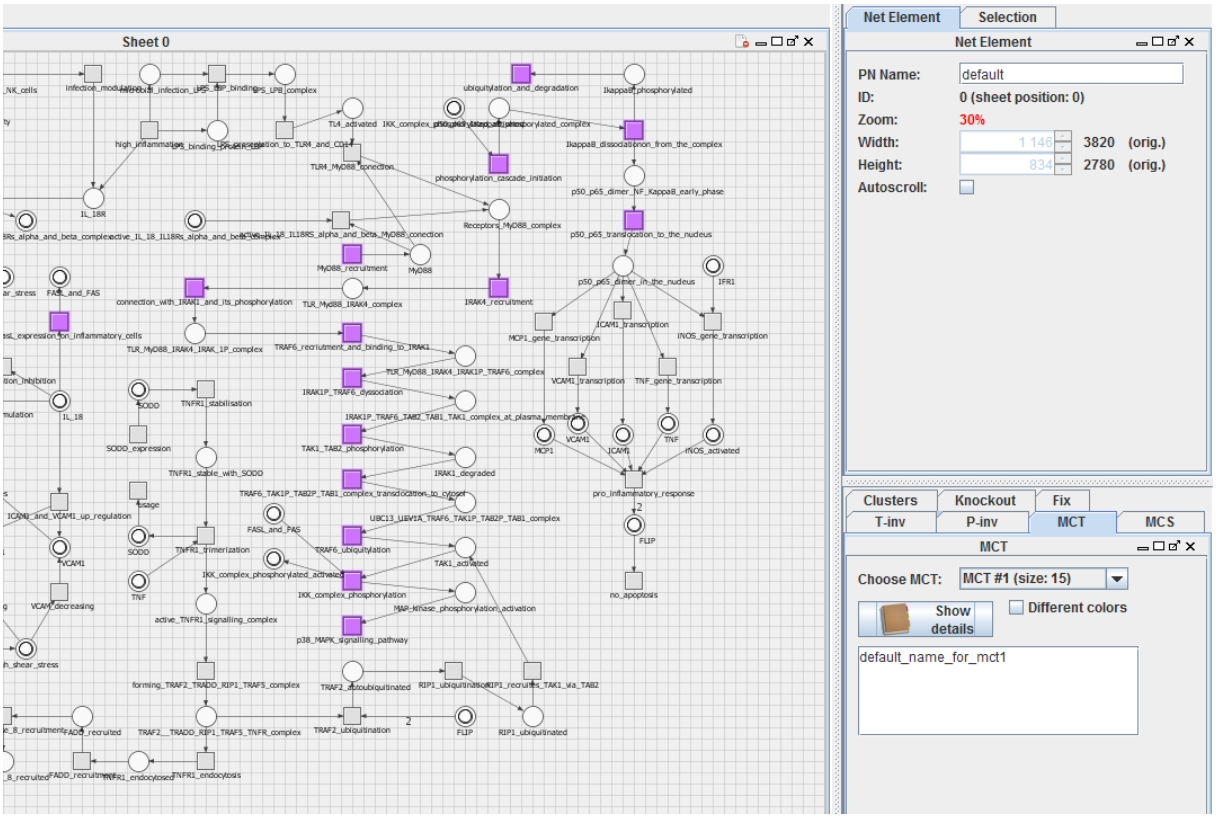
5.6.4 MCT sets tab

MCT (*Maximum Common Transitions sets*) or mADT (maximal *Abstract Dependend Transitions sets*) are automatically generated when t-invariants set become available within the project.

They are computed from the *feasible* t-invariants (cf. chapter 7.1). They group transition always working together in all available t-invariants. Formally, single transition is also an MCT set on its own. We call such sets trivial ones and in practice only consider non-trivial, 2 or more elements ones. As for their theory and biological application, the reader can see e.g. (Sackmann, Heiner i Koch, 2006) and (Formanowicz, Sackmann, Kozak, Błażewicz i Formanowicz, 2011).



MCT can be seen in the net structure, separately or all at once (because they are disjoint subsets of a set of transitions). Example is given below:

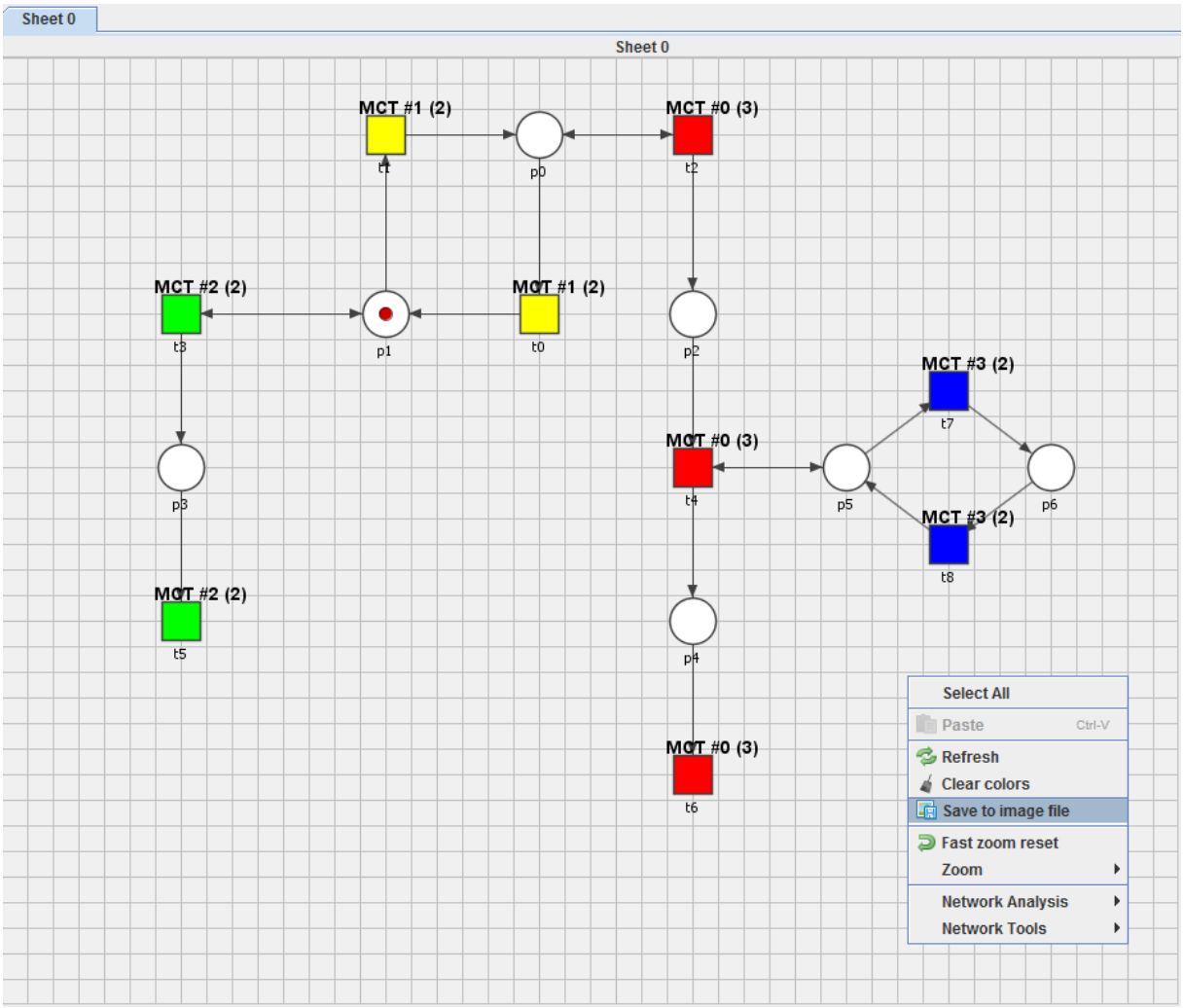


Chosing option **No-MCT transitions** from the list will also show all trivial MCT sets.

Show details will show separate window with MCT set data.

Show all options shows all non-trivial MCT sets painted in different colors. Every transition has (apart from color) a number of MCT set and (in parenthesis) the number indicating to size of the given MCT set.

Chosing option **Different colors** will show every single set in an unique color.



Cleaning colors is the same as for the t-invariants structure view:

- by choosing first option „---” from the list.
- by icon in the  toolbar.
- by choosing **Clear Colors** from the context menu.

6 Simulation algorithms

From the very beginning it should be noted that there is more than one simulation algorithm implemented with different modes that can be switched. Current Holmes simulation modes are present in the table.

Name	Brief description
Graphical simulation	Explained below the table.
State simulator (*fast simulation*, *Standard Token Simulator*)	In fact this is the same algorithm as the so called 'graphical simulation'. The different lies in the presentation of the results. Graphical simulation concerns the net structure and the results are given as the flow / accumulation of tokens in places. State simulator (as it will be referred to later in this chapter) gives the results in a form of data vectors that are used by different modules in different ways in Holmes. Example of such simulator can be the firing chances of transitions within the t-invariants, as it has been explained in the previous chapter.
Stochastic simulator	Used mainly in the stochastic Petri nets (SPN), requires transitions firing rates provided by the user (e.g. as firing rates vector similar to the already explained initial states vectors)
Stochastic Simulation Algorithms (**SSA**), Gillespie SSA	 **MORE SOMPHISTICATED SIMULATOR FOR SPN NETS.** This algorithm is not yet fully implemented in Holmes 1.0, although its components are already present. However, in the current version it is still inactive (estimated time for full version: end od 2017).

As it has been already stated in the table, graphical and state simulators refer to different visualization of the results, but are in fact the same algorithm. Therefore, it is sometimes possible to see rather strange message, that 'in order to run some simulator, a simulator must be stopped'. If in any form similar message appears, it always should be interpreted as a warning that a graphical simulation is in progress and should be stopped, because the state (background) simulation is needed and cannot perform its function due to graphical simulation enabled.

As a general rule it should be established that whenever the user wants to perform some simulation except the graphical one, the latter should be stopped and the initial state restored (via e.g., initial states manager).

6.1 Graphical simulator

Its options are available in the second tab of the section 4 of the main window (left one). Default tab is **Tools** (with Petri net components to be selected in order to draw / modify a net). Second tab is **Simulator**.

Graphical simulator is simpler than a full state simulator, i.e. it is the same algorithm, but with less functions / modes available. Its main goal it to visualize flow of tokens in the net structure. Obviously, only simple observations of the net behavior can be seen that way.

Mode - available modes are **Petri Net, Timed Petri Net** nad **Hybrid Mode.** Will be explained later.

Time/step – current simulation step

Available button:

1 - retract simulation by one step (one firing)
2 - makes the one step forward in the simulation
3 - retract the simulation by whole state (multiple firing)
4 - makes multiple firing forward
5 - **starts continuous simulation forward until paused / stopped**
6 - as above, but single firing per step
7 - pause, active if buttons 1-6 have been activated
8 - stop the simulation when all firings in the current step are finished (may take short time to occur)
9 - resets state to the selected m_0 state **selected in the states manager (chapter 5.5)**

10 - saves to current tokens distribution as the net state m_0 (cf. chapter 5.5)

Maximum mode – if active, all enabled transition will fire simultaneously (if enough tokens are available).

Single mode – similar to button  (**6**), with one important exception. In default **50/50 mode** (opposite to **maximum mode**) some transitions may not fire even when enabled. In this mode (i.e. single mode) only one transition per step can fire. Therefore combining **50/50 mode** (i.e. when Maximum mode **IS NOT** enabled) and **Single mode** (enabled) **would** result in some 'empty-steps' when none transition fires (even when some are enabled). Therefore by default checking Single mode **WILL ENABLE** maximum mode as well, what in turn guarantee that if at least one transition is enabled, it will fire. In order to disable automatic **maximum mode** in **single mode** the user should check the option "**Single-maximum mode (single-50/50 when unchecked"** in **Simulator Engine Options** in the **Simulator** panel in Properties window (menu **Windows**-> **Properties**).

The option on the top of the simulation panel (**Mode:** , see: previous picture) can take three modes. **Petri Net** mode makes all transition behave as a classical ones (even if they are not classical in the current model, e.g. timed.) **Timed Petri Net** mode is available only for pure time Petri nets. If there are some other transition than time ones, **Hybrid mode** will be automatically used (cf. chapter 4.1).

To retract the simulation one can use button  (1) or  (3), of course if the simulation went any further from the step: 0.

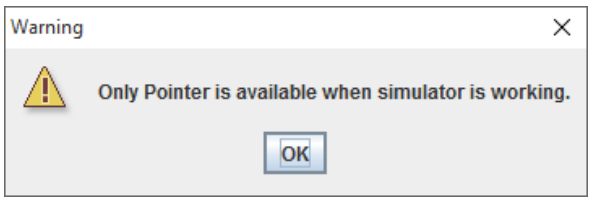
Button  (5) is the most commonly used one. It utilize the selected mode and show the results on the net structure automatically.

Button  (7) i  (8) are active when simulation is started by using button  (5).  (7) – **pause**, freezes the simulation, it can be used e.g. for making pictures of ongoing simulation.

IMPORTANT. Stop button:  (8) . First of all, its activation may take some time, because in order to stop a simulation, all the firing (i.e. consumption and production of tokens must be completed).

This button will not reset the state of the net to the selected initial one. It will only stop the simulation that is already in progress. In order to reset the state, other button is necessary:  (9).

It should be remembered, that it is not possible to add or remove net elements when any simulation is in progress. If one tries to do this, the following message will appear:



In order to change the net, all simulations must be stopped.

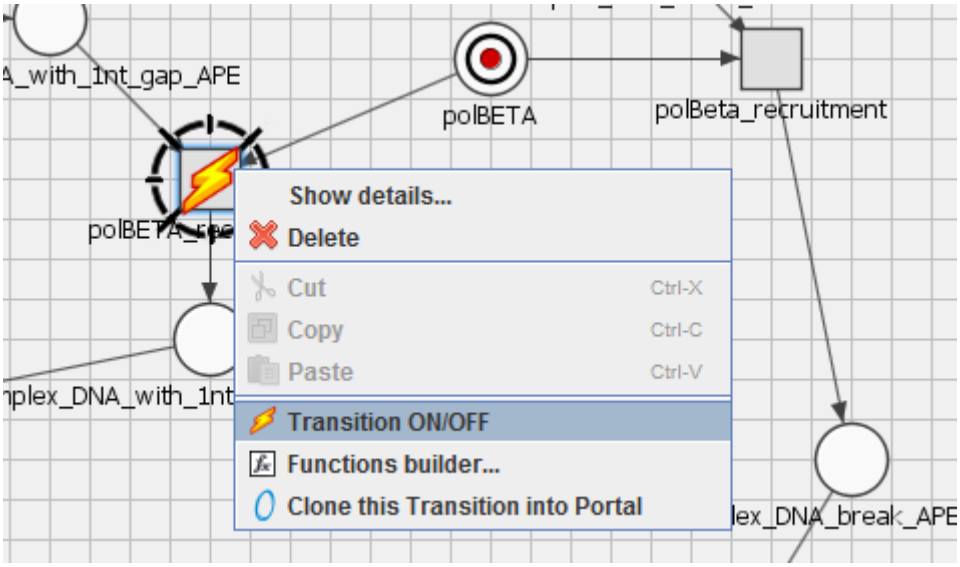
If reset button has not been pressed  **and some changes in the net structure will be performed by the user, first Holmes will restore the selected initial step automatically (as if the reset has been pressed), and then the change in the net structure will be executed.**

The last button is  (10) – when the simulation is stopped it will add a new initial state for the stored states – based on the current tokens distribution from the current simulation step.

It should be noticed that the reset arc influence the graphical simulator capability of retracting steps (going backward). In the current version of Holmes, if the reset arc is present in the net structure, the graphical simulator will not be able to restore tokens taken by this kind of arc. This is a minor inconvenience however, because it concerns only graphical simulation. All other simulators work forward only, therefore all arcs can be used without the fear of losing any information.

6.1.1 Transition deactivation (simulation knockout)

The option is present in the following picture showing a context menu utilized on a transition.

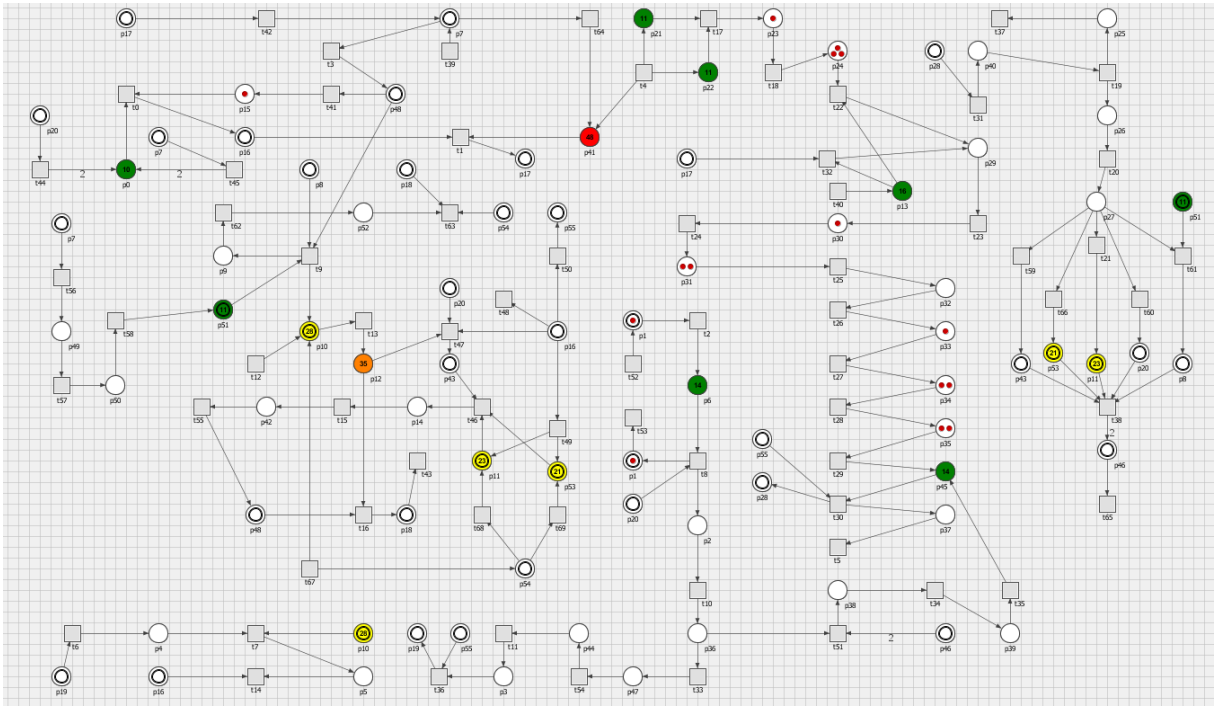


The selected transition has a thunder icon. Such a transition will be considered **permanently disabled no matter how many tokens are in its pre-places (set ●t).** It is true for any simulator in Holmes, not only graphical one. In other words, when a user sets some transition of the net as disabled, all simulators will treat such transition as disabled. This can be used for analyzing knockout behavior of some reactions/functions in the simulation analysis.

This feature does not concern t-invariants based analysis (this type of analysis has its own knockout modules and algorithms, which will be explained later in the manual).

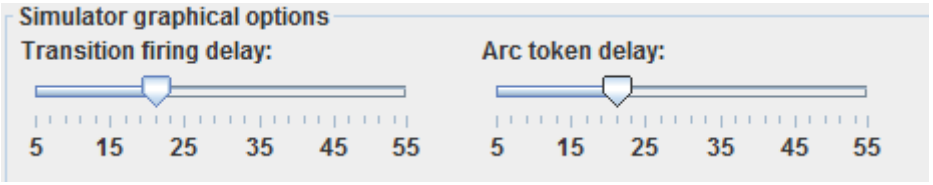
6.1.2 Marking multiple places with tokens in a simulation

In **Properties** window (**CTRL+W**) in a tab **Simulator** there is an option „**Places change color during simulation**". If it is selected, after 10 steps of the graphical simulation, Holmes starts painting places with more than 10 tokens in different colors. This allows the observation of net regions that accumulates more tokens than the other ones. Example is given in the picture below:



6.1.3 Graphical simulation speed

In the Properties window, Simulator tab there are two interesting bars:

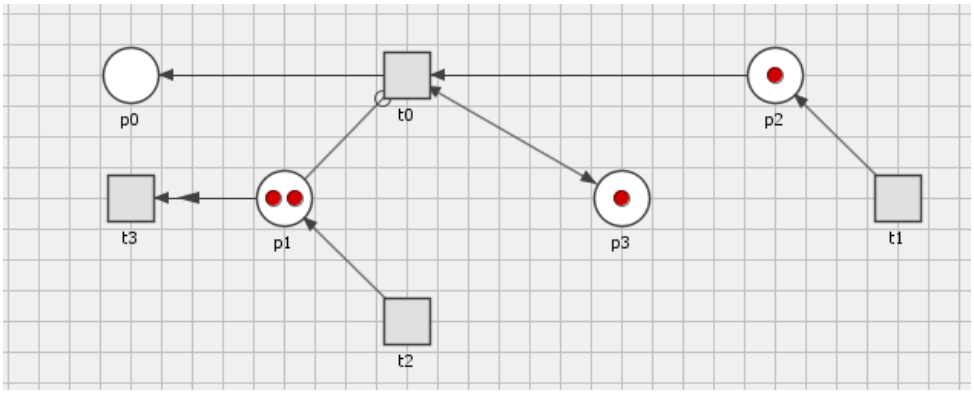


In general, the lower the selected value, the faster the graphical simulation will be performed in the Holmes editor window. First bar cannot be lower than the second one. The fastest way for fastening the simulation is to draw the left bar on the left – the right one will follow. Speed **can be** changed during the active simulation.

6.2 State simulator

This chapter extends the basic theory given in chapters 4.1.1 – 4.1.4. Three main simulation modes will be explained here: **Petri Net, Timed Petri Net** oraz **Hybrid Net** with other possible sub-modes for them (50-50 / maximum / single mode.

6.2.1 Petri Net mode



The example uses extended net: read arc (p3-t0), inhibitor arc (p1-t0) and reset arc (p1-t3). For such a net (or obviously for the one without extended arcs) default mode is **Petri Net**.

Enabled transitions in the net state (m_0 = {0,2,1,1}) are:

- t1 and t2 as input transitions
- t3 active by reset arc from p1 (p1 has more than 0 tokens)

Transition t0 **is not enabled**. p3 has 1 tokens (enough for read arc) and p2 also has 1, however there is inhibitor arc going from p1 (with 2 tokens, inhibitor weight=1).

Simulator will activate transition depending on the sub-mode: **50/50** or **maximum**. For **50/50** sub-mode there are 5 possible scenarios:

- t1 fires
- t2 fires
- t3 fires
- random combination of the above, however there is one not so obvious scenario here: if t3 fire before t2, it will take 2 tokens from p1, then t2 produces there 1 token. **However, if** first t2 fires (before t3), then it will produce third token in p1, then t3 will take all of them by its reset arcs. **Sequence of transitions firing is random in 50/50 and it is determined by the simulator before the actual firing begins**.
- no transition fires. It should be notet, that this possible scenario is not very usefull in the

simulations, however it is possible. If it happens, simulator 'reset' such a simulation step and tries to fire transitions again (in another randomly selected sequence) in the same step, until at least one fires.

For the **Maximum** sub-mode there are firing scenarios::

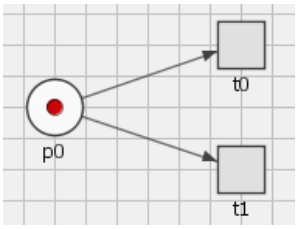
- t1 always fire
- ***alternative***:
  - first t2 fires, then t3, tokens in p1 = 0, in the next step t0 will not be blocked and can fire (if t2 won't fire before it)
  - first fires t3 (taken 2 tokens from p1), then t2 which produces 1 token in p1. In the next step t0 is blocked.

In the maximum mode all enabled transitions must fire, the only thing that can prevent it is that some other transition which fire first in the same step, consume tokens necessary for other ones to be enabled. If in maximum mode no transition can fire, it means that there is a deadlock and it is the very last state of the net – simulation cannot continue.

Sub-mode **single** does not change rules of transitions choosing, the only modification is that **only one transitions can fire in this mode in one simulation step**.

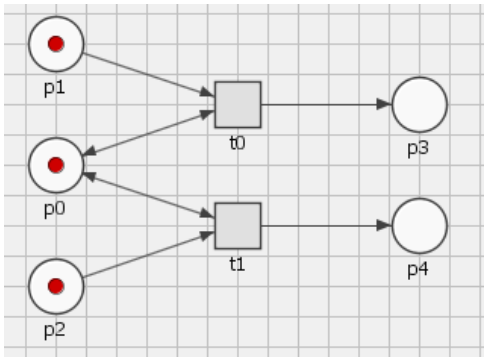
6.2.1.1 Tokens reservation by read arc

It is a rather technical issue connected with the simulator decisions. When active transitions are being added to the *activation list*, they *reserver* tokens from their pre-places (one by one). In this way it is not possible to have in the *activation list* more transition than actually can be fired (taking tokens in pre-places into consideration).



An example is given by the following picture. Let us assume that the (random) order in which transition will be added to the *activation list* is as follows {t1, t0}. Transition t1 is enabled and it should be added to the list. It is indeed added (not fired yet!). Then, without the reservation mechanism, adding enabled t0 into activation list (while 1 token is in p0) could results in firing both transition, when there is a conflict in their pre-place (not enough tokens for both transition to fire). Therefore, every added transition (to the activation list) reserves tokens before firing.

Such a mechanism for the above example does not have any practical implications – it just works within the simulation algorithm. However, when read arcs are considered, such mechanism do have more serious implications.

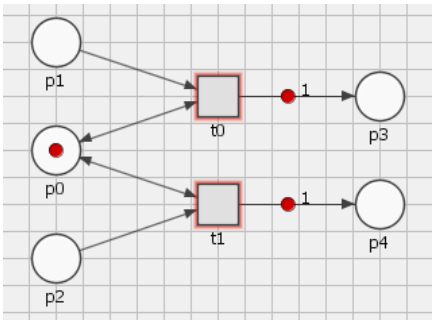


There are read arcs between p0-t0 and p0-t1. It means that token in 0 is necessary to activate t0 and t1, but it will never be taken during these transitions firing phase.

The question arise is it possible to fire both t0 and t1 in one step when there is only one token in p0?

The question can have some serious impact on the behavior of e.g. a biological system. Catalytic compound in p0 does not take place in reactions t0 nor t1, but it makes them possible. It is only 1 token, so can it be used for both reactions?

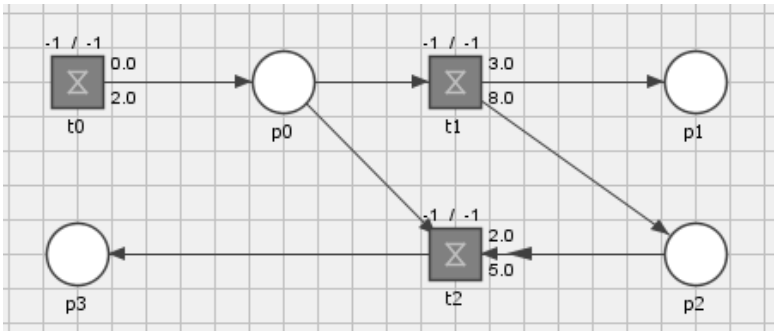
The default behavior (window **Properties**, tab **Simulator**, option „**Transitions reserve tokens in places via read-arc**") is **on**. It means, that in this example, only one transition will be able to fire in one simulation step. Turning this option off will enable (e.g. in maximum mode) faring of both reactions in the same step.



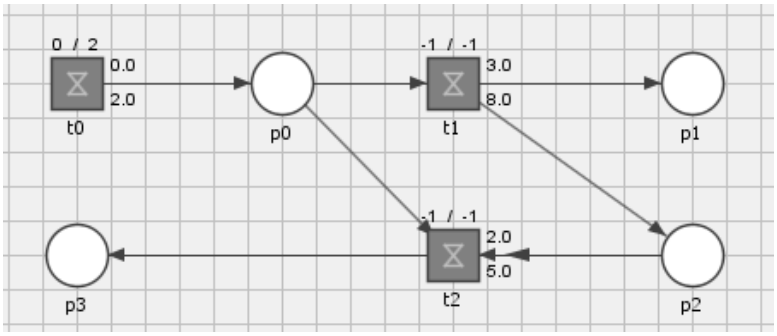
Both reaction are activated by p0 and fire because:

a they cannot take it anyway (*read-arc*)
b option mentioned before is turned off.

6.2.2. Time Petri Net mode

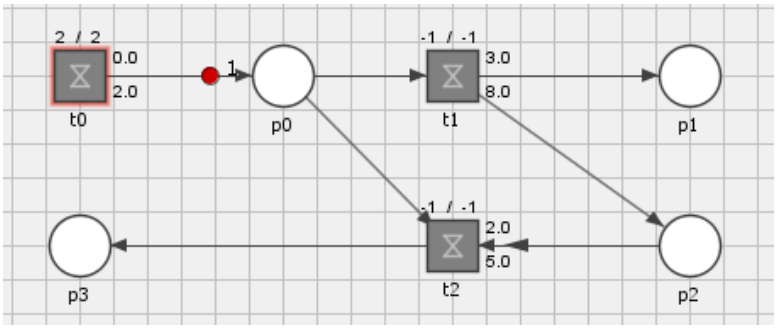


This mode is available only for TPN/DPN/TDPN time nets. The example is given in the picture. The number following transition has been already explained later, to example for t0: EFT=0, LFT=2, -1/-1 means the simulation has not yet started.

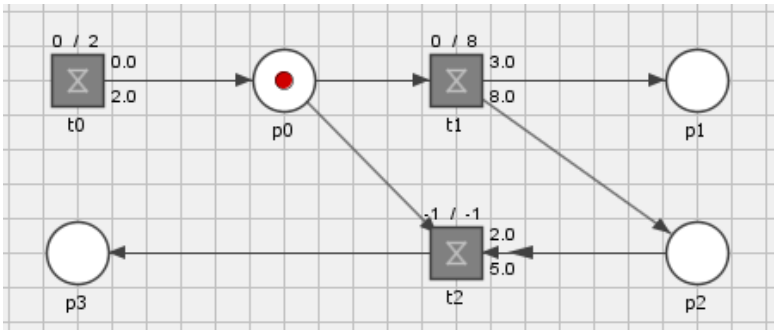


In this step, t0 real firing time (random between EFT and LFT) is 2. For the next two steps the simulator will count from 0 to 2, then t0 will fire.

Effects are given in the following picture:



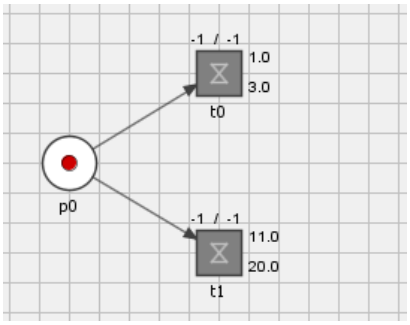
Simulation at the beginning of the third step:



Transition t0 again drawn firing time equal to 2 (between EFT / LFT). Transition t1 has firing time 8 – maximal LFT time (available integer times are 3,4,5,6,7 and 8).

6.2.2.1 Time conflicts

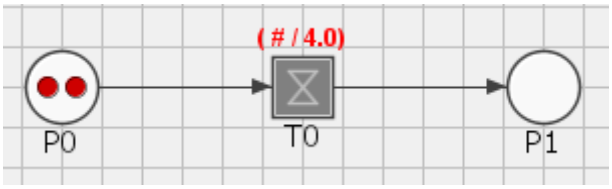
One important issue concerning time should be explained. When two transitions are in conflict (they share the same pre-place), it is important to remember, that their time constraint my cause an inhibition effect on each other.



Transition t0 and t1 are in conflict. The problem here is that t1 has no chance to fire due to its high EFT, much higher than the other transition LFT. Of course there may be some source of tokens which will make enough tokens in p0 for t1 to remain active until it fires, but it is still advisable to remember about such time conflicts when making a net.

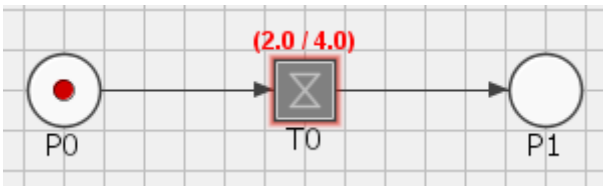
6.2.2.2 DPN nets

Their functioning has been already briefly explained in chapter 4.1.3.2.



Transition T0 may be enabled, but since the simulation has not started yet, there is a # symbol. Duration for this transition is 4

Let us assume 2 steps of the simulation passed.



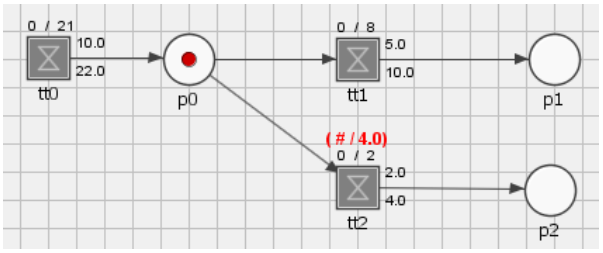
Transition T0 consumed tokens immediately when it was enabled. Then is started counting towards its duration time.

Since 2 steps have passed already, two more are

necessary for T0 to produce token in P1. **Deactivation of T0 is not possible when it starts counting (and consumes tokens).**

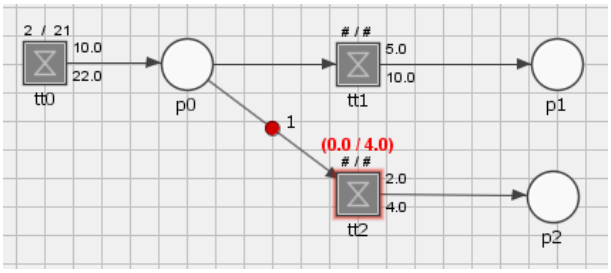
6.2.2.3 TDPN nets

Let us assume the following net:



When the simulation started, transitions tt1 and tt2 became enabled. First drawn T_x = 8 (in range <5, 10>), second T_x = 2 (range <2,4>). Transition tt0 time T_x = 21 (<10, 22>).

There is a question what will happen in simulation step 2.



We see in the picture how tt2 transition consumes tokens, and its clock (DPN type) starts counting towards 4 (6th step, then tt2 will produce token in p2).

What is important here is that tt1 stops being enabled. In a TPN model deactivation of a transition is possible up to the point when its counter is almost at T_x . On the other hand, DPN transition or TDPN when its TPN counter reaches T_x **must fire immediately**. Tokens will be produced after counting to d_x value.

The problem with the above net is that tt1 will never fire. Its EFT is longer than LFT of tt2 (5>4) what means this scenario will repeat. tt0 works to slow to produce more than 1 tokens in p0 (tt2 will very soon takes one).

Charts from transition data window (chapter 4.4.2) in a time simulation for **tt0**:

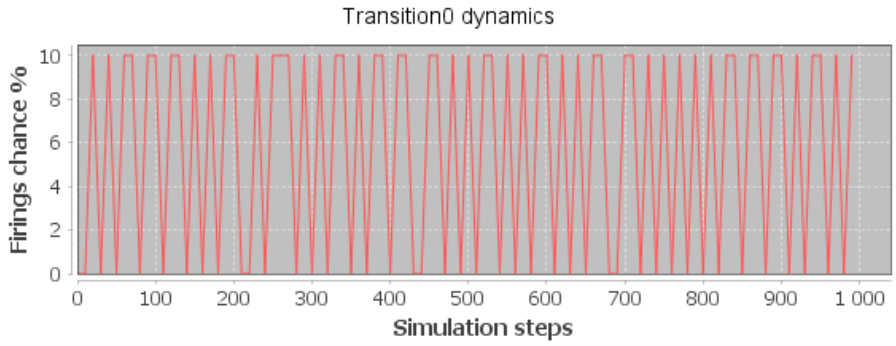


Chart for **tt1** is given below – the transition cannot fire.

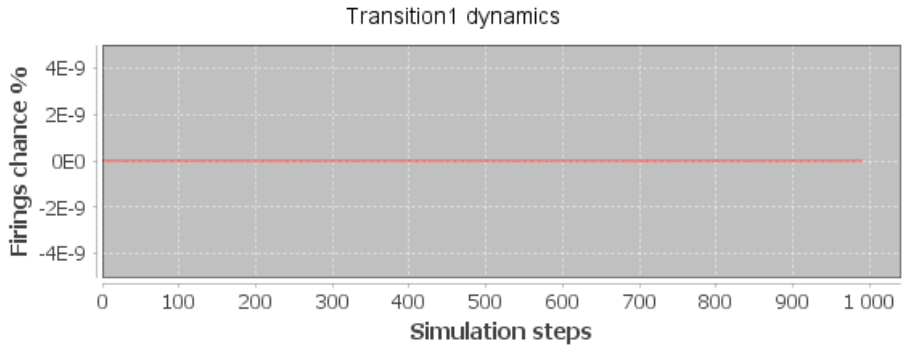
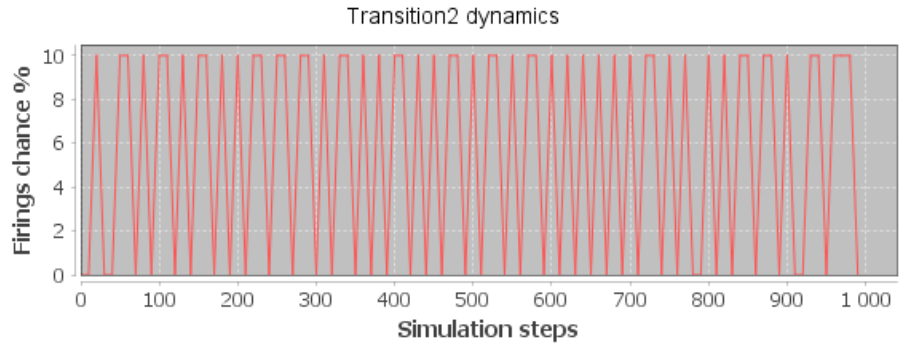
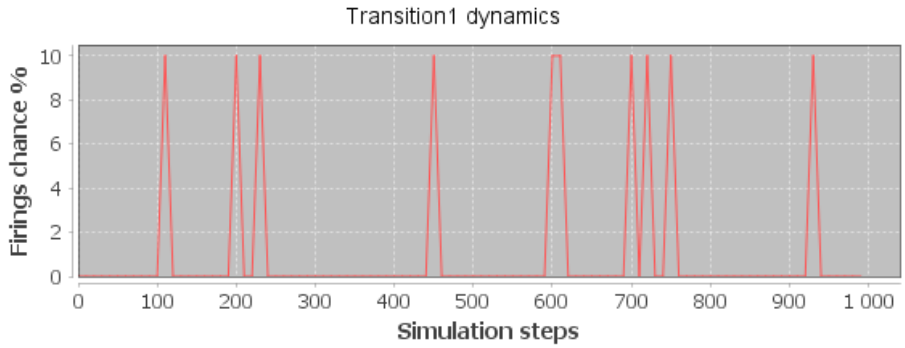


Chart for **tt2**:

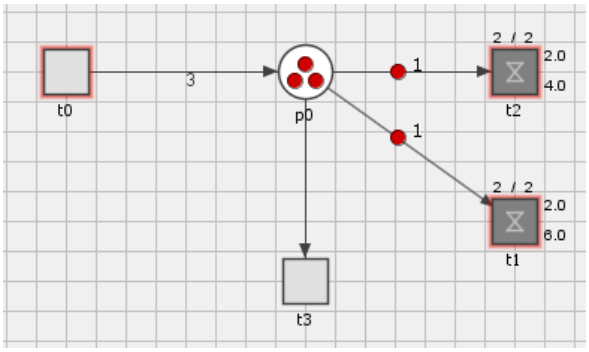


When tt1 EFT will be change to 0, it will give it a rather small, but non-zero chance of firing. The new chart to tt1 is given below:



6.2.3 Hybrid Mode

This mode will automatically be activated, when among timed transitions (TPN, DPN or TDPN) there will be any other transition type, e.g. a classical one.



The picture shows a state from a hybrid mode simulation.

6.3 Stochastic simulator

Two main differences between such an algorithm and the ones described so far are:

- Probability of firing any transition depends (among other things) on the transition *firing rate*. The default value is 1.0, the lower the value (from 1.0), the less chances the transition has, the higher than 1.0 – the higher the chances.
- The probability can also depend on the marking of the net, specifically – of the number of tokens in pre-places of every transition.

By default, the algorithm chooses one transition in each simulation step for firing.

6.3.1 Stochastic algorithm for SPN

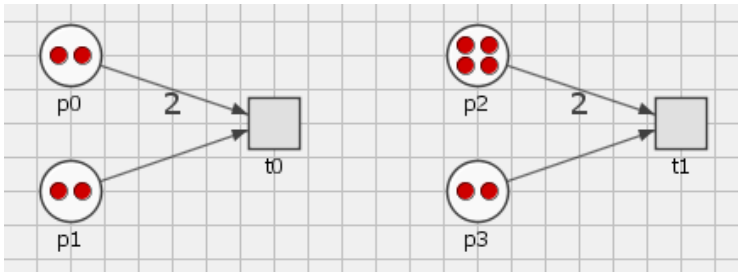
The goal of the algorithm is to calculate the chances of firing for every enabled transition by obtaining a probabilistic firing time. Transition with the lowest value will be fired next. In a *mass action kinetics* mode, the number of tokens in every transition pre-places are also considered in the equation.

Firing (or waiting time) *dt* are obtained from the following equation (Balazki i Einloft, 2014):

$$f(dt) = r(T_i) \times e^{-r(T_i) \times dt}$$

where $r(T_i)$ is the transition *firing rate*.

When *mass action kinetics* enabled, different scenario is considered. The example is given below:



Transitions t0 and t1 are enabled. It should be noted however, that t1 (if allowed) could fire two times, because there are enough tokens in its pre-places (t0 can only fire once).

Value $q(T_i)$ – is a number of potential firings of a transition in a time unit dependent on the number of tokens in the pre-places. A new equation for firing probability is given for such a scenario:

$$f(dt) = q(T_i) \times r(T_i) \times e^{-q(T_i) \times r(T_i) \times dt}$$

The algorithm simulates a stochastic choice by calculating for every transition a *stochastic firing time*:

$$time_i = \frac{-ln(1-x)}{q(T_i) \times r(T_i)}$$

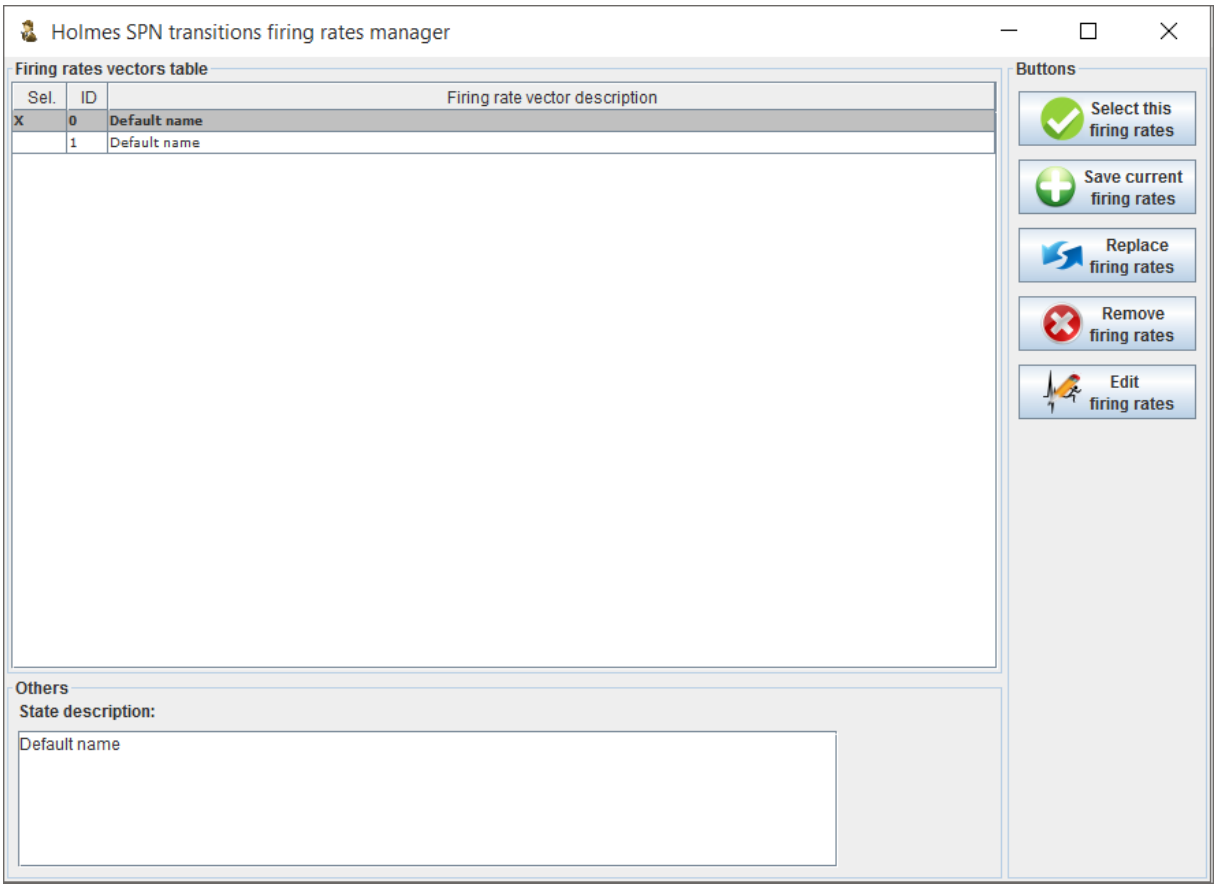
Where *i* is a number of enabled transition, *x* is a real random value from (0, 1), $r(T_i)$ is a transition *firing rate*.

When *mass action kinetics* is turned off $q(T_i) = 1$. When it is active $q(T_i)$ for transition T_i denotes the maximal possible firing times for a given transition (in the above example $q(t_0) = 1$, $q(t_1) = 2$)

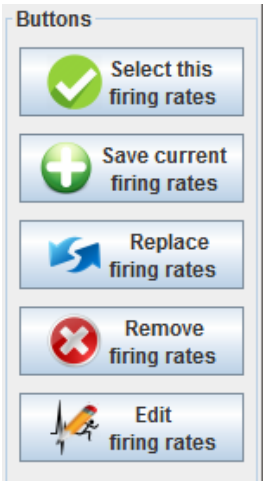
6.3.2 Firing rates manager

This is a similar window to initial states manager already described. It can be activated from the simulation options by a button 

.



Selected *firing rates* vector is marked by X.



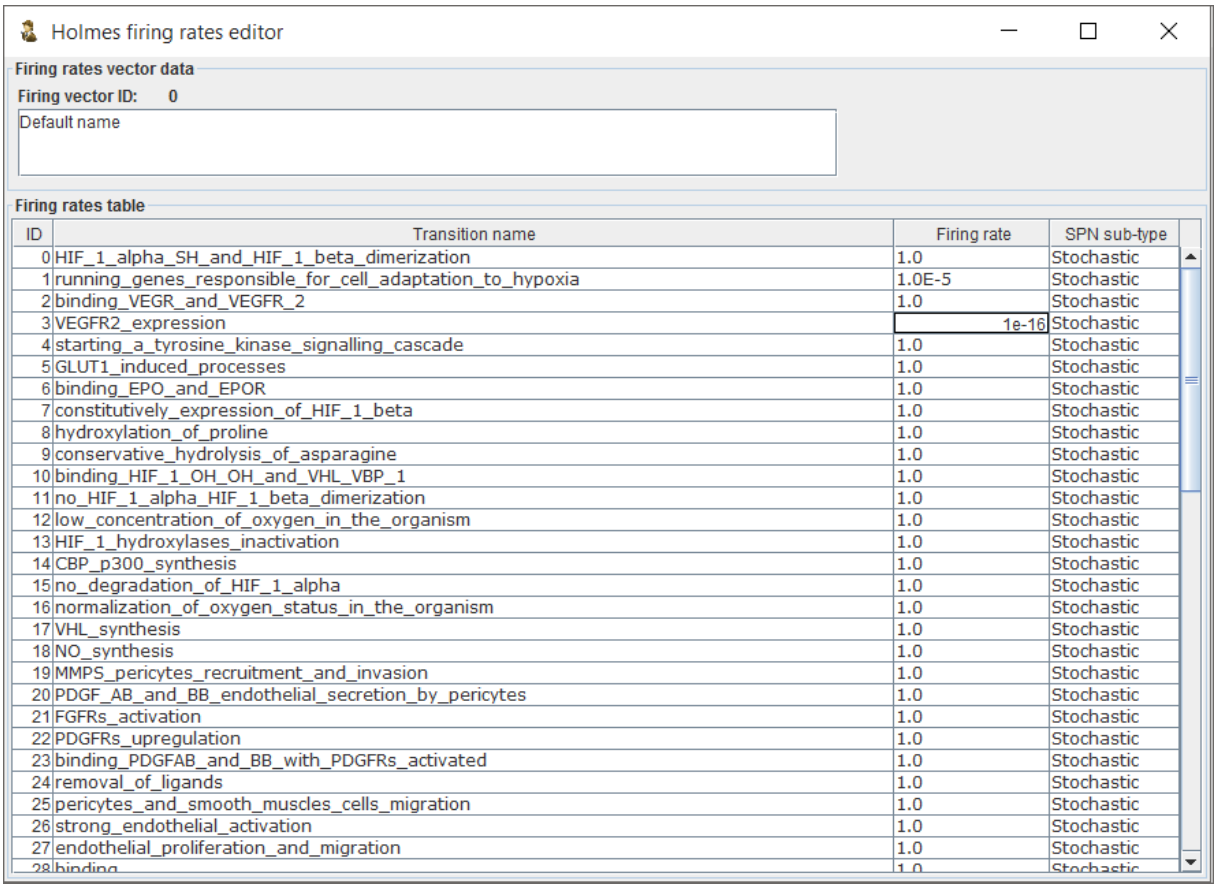
Select this firing rates – this button select the firing rates vector to be used in the stochastic simulation.

Save current firing rates – creates a new *firing rates* vector from the values currently assigned to transitions (via e.g., the editor).

Replace firing rates – similar as above, but the selected vector is replaced.

Remove firing rates – removes the vector from the list / project.

Edit firing rates – activates editor window.



ID	Transition name	Firing rate	SPN sub-type
0	HIF_1_alpha_SH_and_HIF_1_beta_dimerization	1.0	Stochastic
1	running_genes_responsible_for_cell_adaptation_to_hypoxia	1.0E-5	Stochastic
2	binding_VEGR_and_VEGFR_2	1.0	Stochastic
3	VEGFR2_expression	1e-16	Stochastic
4	starting_a_tyrosine_kinase_signalling_cascade	1.0	Stochastic
5	GLUT1_induced_processes	1.0	Stochastic
6	binding_EPO_and_EPOR	1.0	Stochastic
7	constitutively_expression_of_HIF_1_beta	1.0	Stochastic
8	hydroxylation_of_proline	1.0	Stochastic
9	conservative_hydrolysis_of_asparagine	1.0	Stochastic
10	binding_HIF_1_OH_OH_and_VHL_VBP_1	1.0	Stochastic
11	no_HIF_1_alpha_HIF_1_beta_dimerization	1.0	Stochastic
12	low_concentration_of_oxygen_in_the_organism	1.0	Stochastic
13	HIF_1_hydroxylases_inactivation	1.0	Stochastic
14	CBP_p300_synthesis	1.0	Stochastic
15	no_degradation_of_HIF_1_alpha	1.0	Stochastic
16	normalization_of_oxygen_status_in_the_organism	1.0	Stochastic
17	VHL_synthesis	1.0	Stochastic
18	NO_synthesis	1.0	Stochastic
19	MMPS_pericytes_recruitment_and_invasion	1.0	Stochastic
20	PDGF_AB_and_BB_endothelial_secretion_by_pericytes	1.0	Stochastic
21	FGFRs_activation	1.0	Stochastic
22	PDGFRs_upregulation	1.0	Stochastic
23	binding_PDGFAB_and_BB_with_PDGFRs_activated	1.0	Stochastic
24	removal_of_ligands	1.0	Stochastic
25	pericytes_and_smooth_muscles_cells_migration	1.0	Stochastic
26	strong_endothelial_activation	1.0	Stochastic
27	endothelial_proliferation_and_migration	1.0	Stochastic
28	binding	1.0	Stochastic

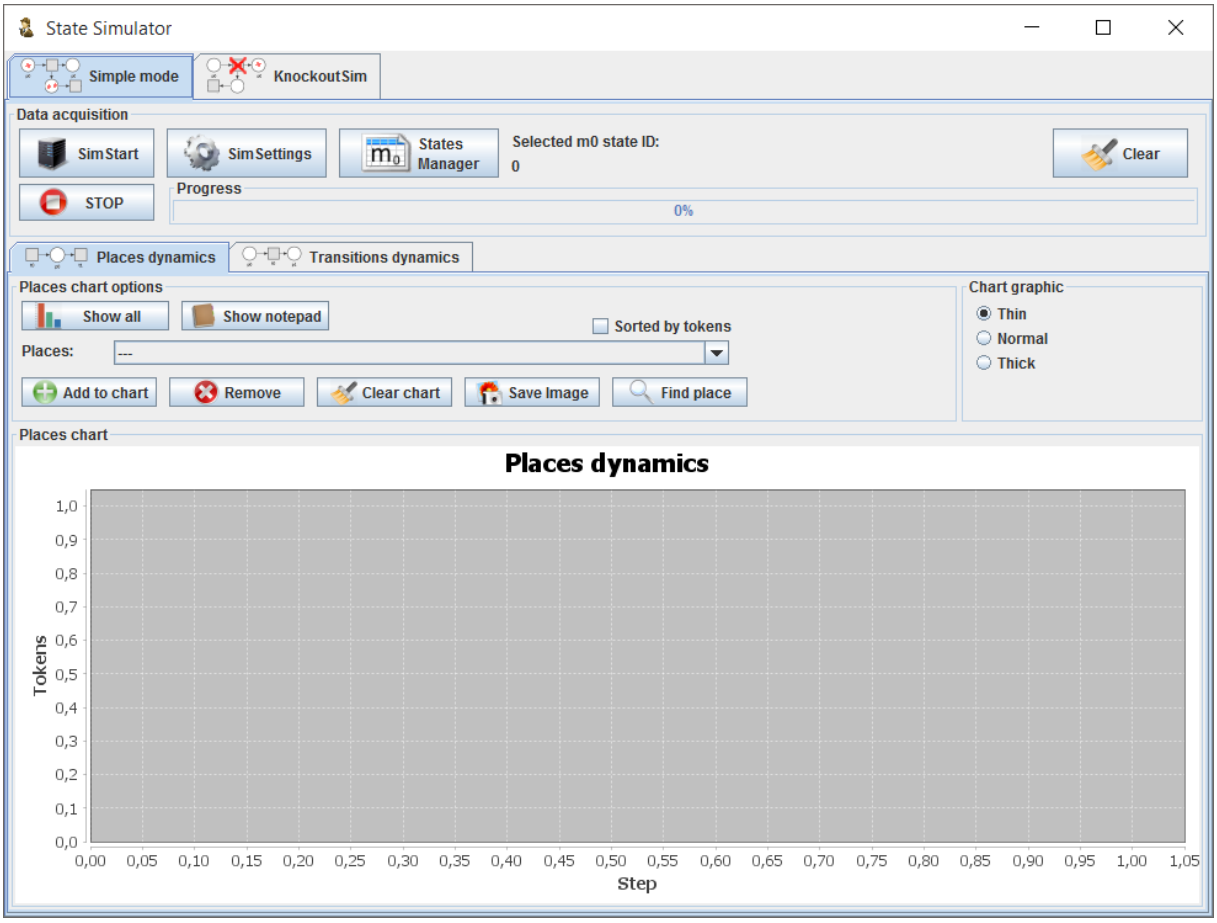
6.4 Stochastic Simulation Algorithm (SSA)

Not yet implemented, estimated time: end of 2017.

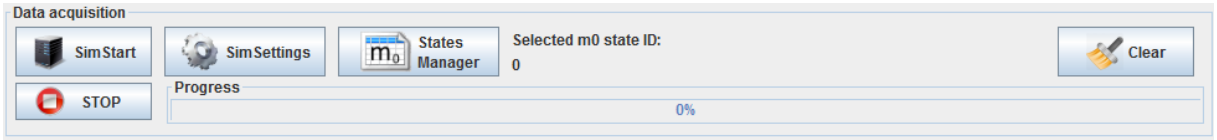
(Gillespie, 2001).

6.5 Simulator – main window

Simulators in Holmes have their own window. It is available from the menu **Analysis** -> **State Simulator…** or by shortcut (**Ctrl+Q**):



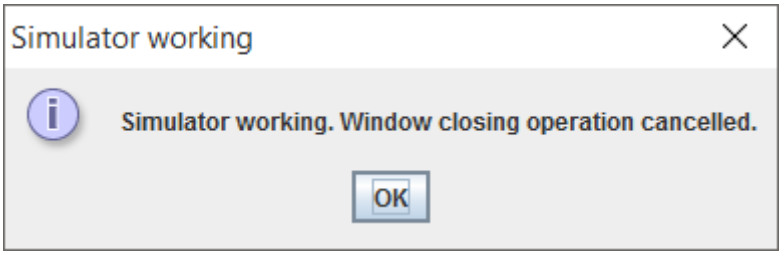
In this chapter the Simple mode tab will be described (KnockoutSim in the next one). They both can use same simulators, the difference lies in the data gathering procedures.



Button **SimStart** enables simulator (working in background). Simulation can be stopped using **STOP** button. Types of simulation depends on the configuration of the simulator in the **SimSetting** options.

There are safety measures implemented, because the simulator uses the actual net structure, which during this process cannot be changed. For these reason the main window (Holmes) is blocked when the simulation is being performed.

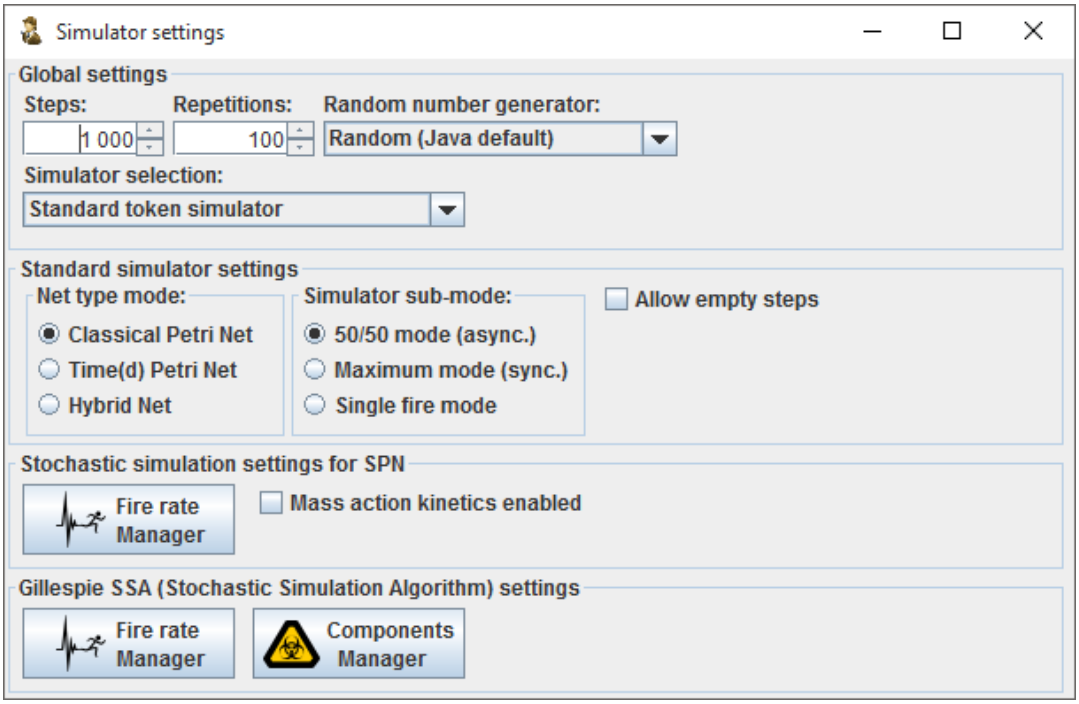
When trying to close the simulator window and return to main Holmes window during the active simulation, the warning will be given:



Button **States Manager** opens the window already described in chapter 5.5. Clear button should clear the memory after the simulation, but this feature depends on the JRE, so the clearing may not be instantaneous.

SimSettings 

allows to configure simulators.



- **Global settings**:
  - **Steps** - set the maximal[2] number of simulation steps.
  - **Repetitions** – in some modules multiple simulation are used and the results averaged. This option concerns the KnockoutSim module to be described in chapter 6.6.
  - **Random number generator** – selection of random generator, Java default one works fastest in the current version of Holmes.
  - **Simulator selection** :
    - **Standard token simulator** – described in chapters 6.1 – 6.2

[2] If the state space is small, the simulation may end sooner due to net deadlock.

        - **Stochastic simulation for SPN** – described in 6.3
        - **Gillespie SSA (exact version)** – not yet available
        - **Gillespie SSA (fast version)** – not yet available
- **Standard simulation settings:**
    - **Net type mode** – see chapter 6.2
        - **Classical Petri Net** – chapter 6.2.1
        - **Time Petri Net** – chapter 6.2.2
        - **Hybrid Mode** – chapter 6.2.3
    - **Simulator sub-mode** - explained in chapters 6.1 and 6.2
        - **50/50 mode (async**.)
        - **Maximum mode (sync**.)
        - **Single fire mode**
    - **Allow empty steps** – in 50/50 mode this will allows steps without firing any (active) transition.
- **Stochastic simulation settings for SPN** – see chapter 6.3.
    - **Fire rate manager** – chapter 6.3.2
    - **Mass action kinetics enabled** – when enabled, tokens in pre-pleces have influence of transition firing chances
- **Gillespie SSA (Stochastic Simulation Algorithm) settings** –:
    - 

feature not yet functional.
    - 

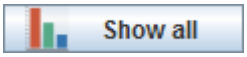
feature not yet functional.

The main (active) simulator is ALWAYS set in the **Simulator selection**.

6.5.1 Places analysis

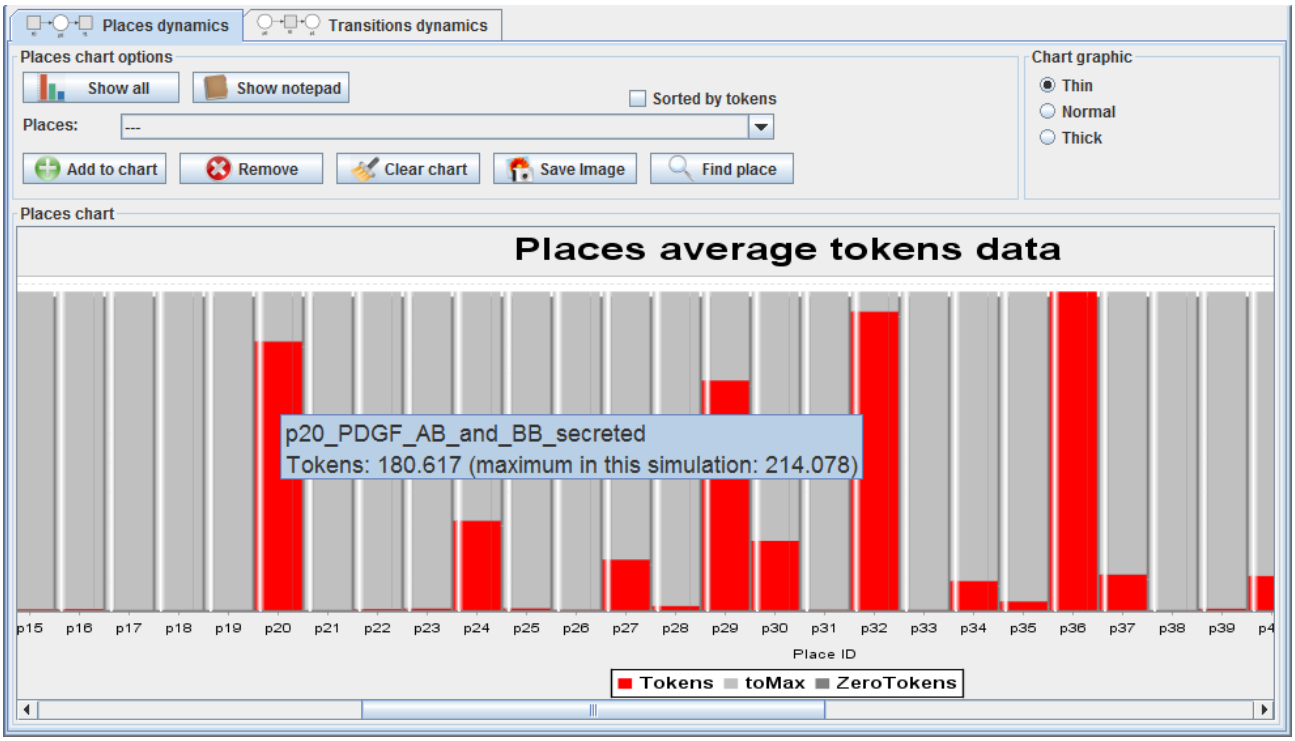
The bottom part of simulator window has two tabs – separately for places and transitions.

For the example let us assume, that there is some net in Holmes, **Standard tokens simulator** for **Petri Net / 50-50 mode** is enabled, and 1000 steps of simulation will be performed.

After the simulation has finished, the button 

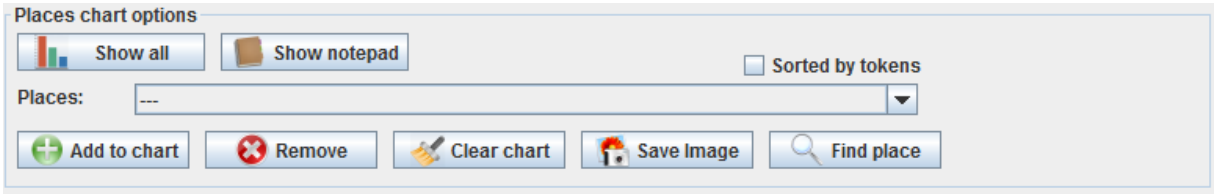
has been pressed.

The bottom part of the simulator window can look like in the example:



In the chart there is information about tokens accumulation (sum of all tokens in all steps) in places during the simulation.

- Bar type:
  - **Tokens** – red bar – what was the sum of tokens during the simulation steps
  - **toMax** – light-grey bar – difference between max number of tokens accumulated in one of the places and the number of tokens in the currently selected place
  - **ZeroTokens** – dark-grey bar – means that during the simulation the place did not acquired any token (not present on the picture above).

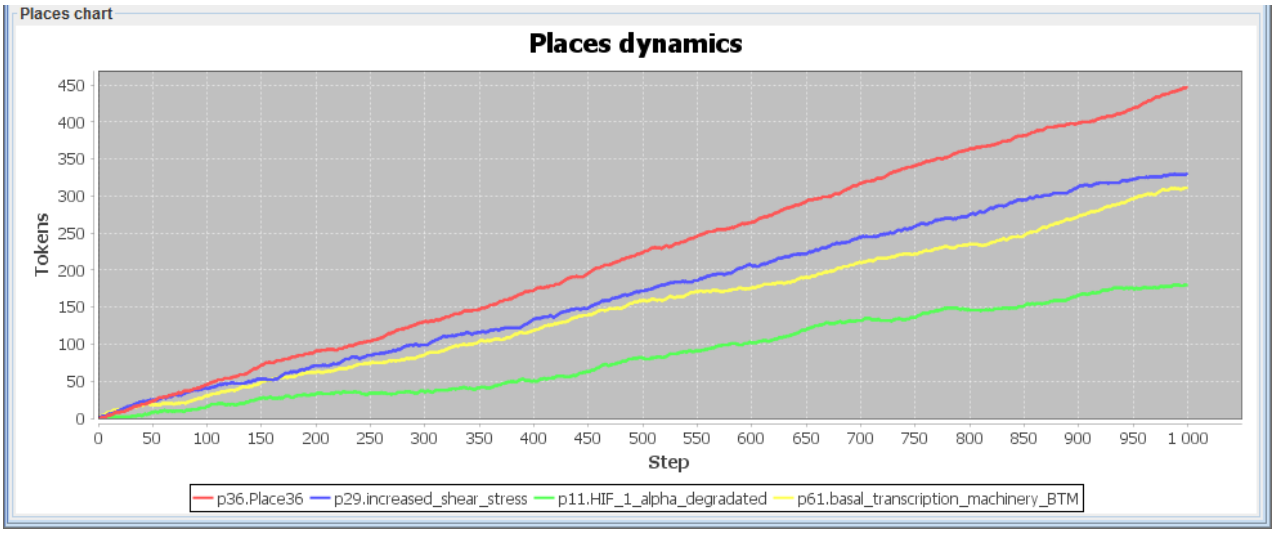


Available options:

- **Show notepad** – shows data in notepad.
- **Places** – list of places.
- **Sorted by tokens** – changes the sequence of places on the above list.
- **Add to chart** – will add place history during the simulation to the chart (will change the chart type, see below)
- **Remove** – remove the place from the chart (see below)
- **Clear chart** – clears the chart
- **Save Image** – saves the chart as a picture

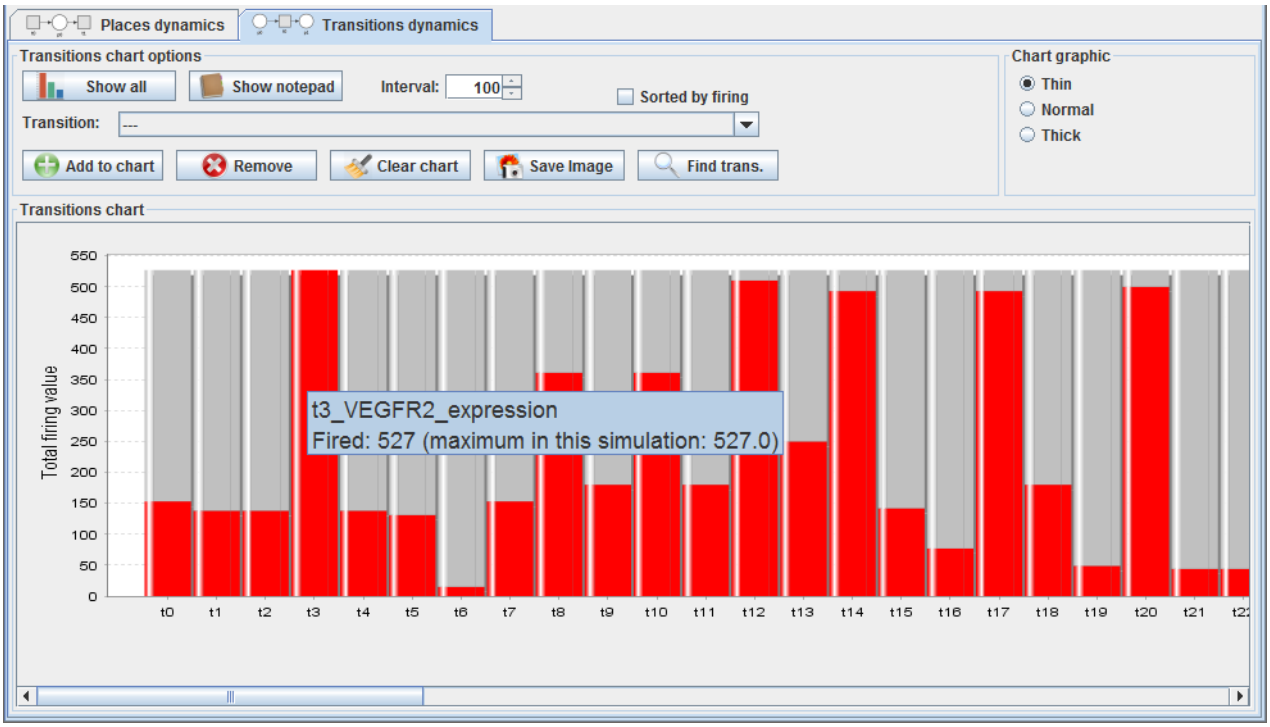
- **Find places** – will mark and center the net editor on the selected place.

When some place will be added to the chart, it can look like in the example below:

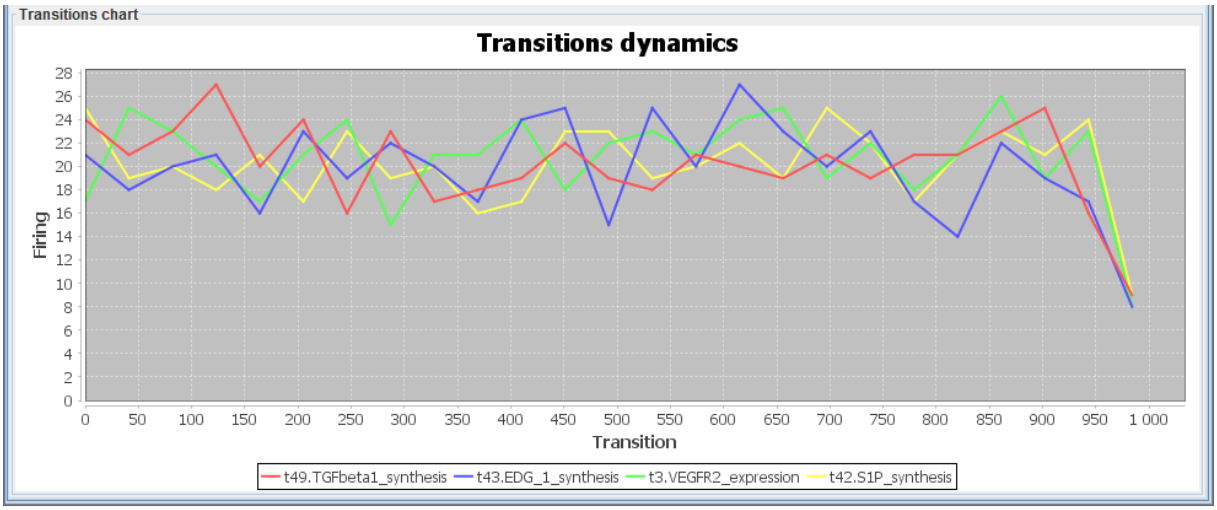


6.5.2 Transitions analysis

This panel has the very similar view as the places analysis panel, the only difference here is that *average transition firing* during the simulation is considered here.



Transition comparison chart:



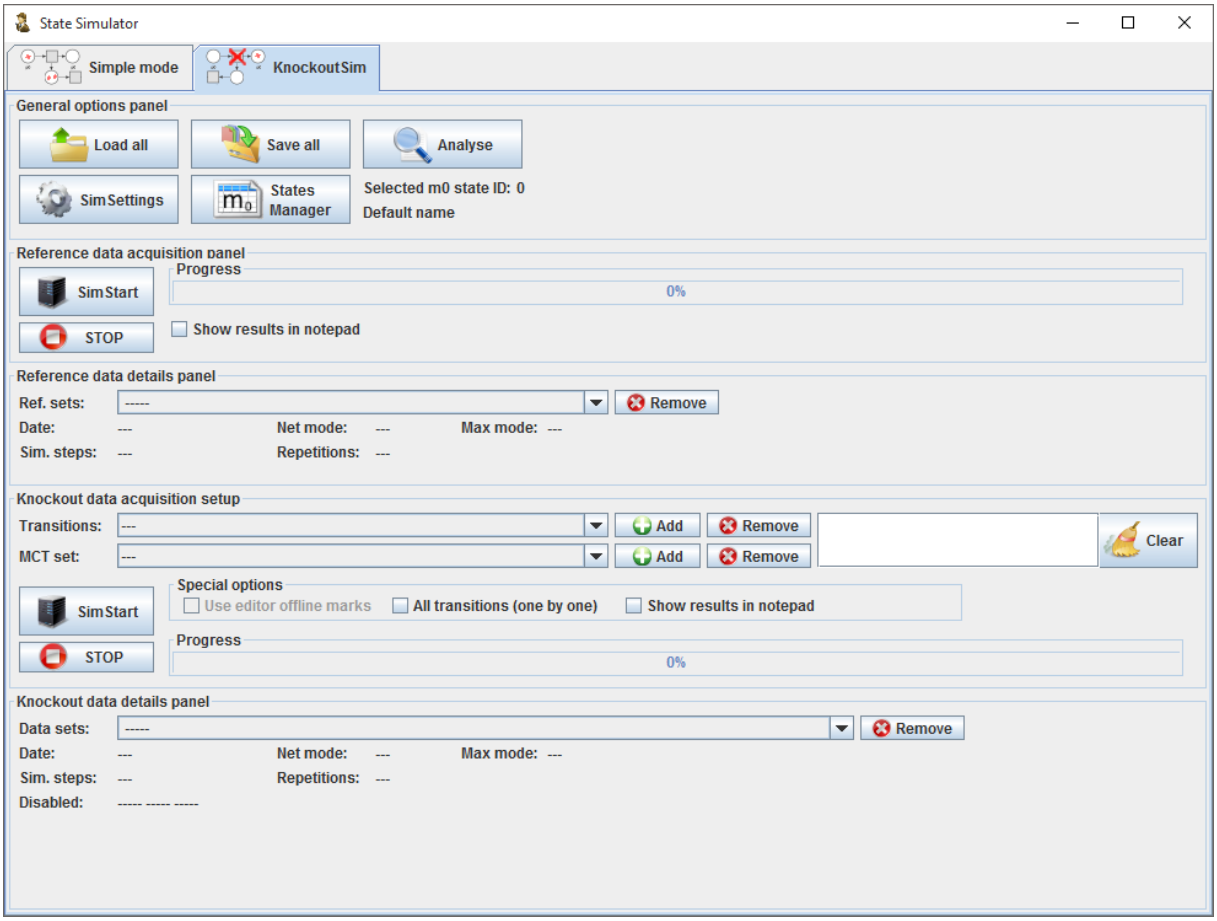
One more feature should be explained here, the **Interval** field. On the example above, it has been set to 40. It means that in the simulation having 1000 steps, every 40 are averaged. For example, setting this value to 1 will draw every single step of the simulation for the selected transitions (no average in this case).

NOTE 1: modification of the Interval value clears the chart, however the data remains in memory (i.e., simulation does not need to be repeated).

NOTE 2: Interval maximal value is: number of sim. steps / 5. For 1000 steps, this is 200.

6.6 Simulation *knockout* analysis

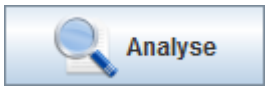
For more detailed data gathering during the simulation, second tab utilizing knockout feature can be used.

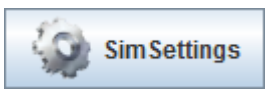


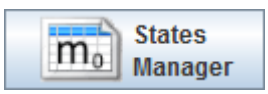
The button are:

- 

- load simulation knockout results from a separate file (.sim extension, **outside of project file**).
- 

- saves the simulation results to .sim file.
- 

- activate analytical window (see below).
- 

- chapter 6.5, simulators settings.
- 

- states manager window.

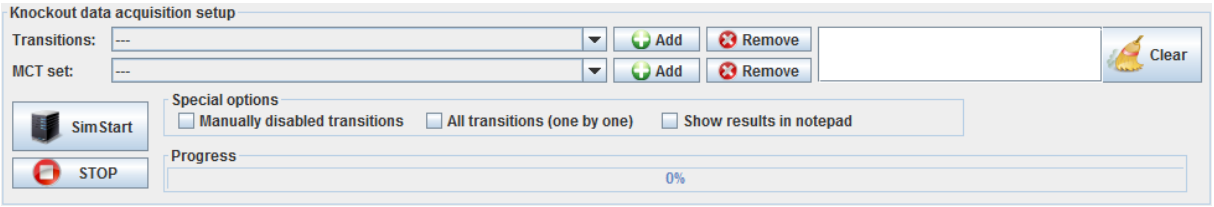
Panel for gathering reference data (i.e., **when NO KNOCKOUT is set**):

It should be remembered, that whole **KnockoutSim** window uses repetition of simulation. Depending on the number of steps and repeats, this may take some time to complete, but the more accurate the results will be.

When **Show results in notepad** is set, notepad with brief summary will show when the simulation finish. Simulator window cannot be close until the simulation is finished or stopped by **STOP** button.

Second panel shows the reference data set signature on the list.

Third panel is more complicated, its goal it to gather knockout data.



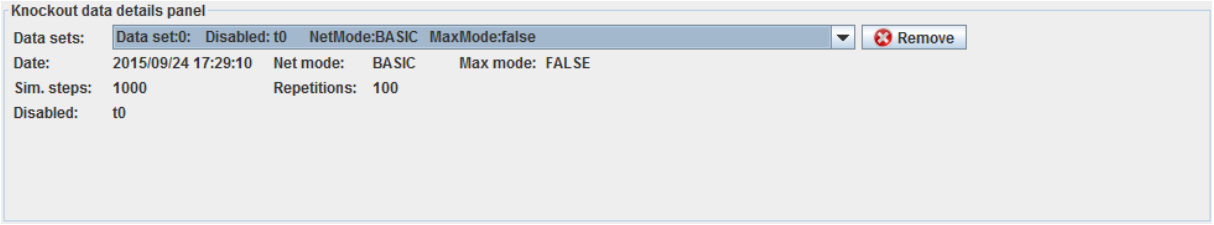
In this panel the user can set the transitions or whole MCT set to be knocked out during the simulation. **Add**, **Remove** and **Clear** buttons are used for this goal. When **SimStart** is clicked, every selected transition and MCT set will be disabled and the simulation will gather data about the net behavior.

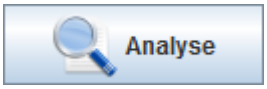
Special options:

- **Manually disabled transitions** – when this option is set, the SimStart button will work differently – it will test the net in its current 'knockout' state, i.e. if some transitions are set to be disabled (see chapter 4.3), such a simulation data will be gathered. Transitions and MCT sets in the right top text field are ignored in this mode.
- **All transitions (one by one)** – this will also ignore the upper part of the panel. It will make a separate simulation for every transition of the net, when it is disabled (separately). This make take some time, depending on the number of simulation steps and repeats. **It is strongly advised to save the data after such a long simulation** (Save all button).

Choosing both above options and clicking **SimStart** is not possible:

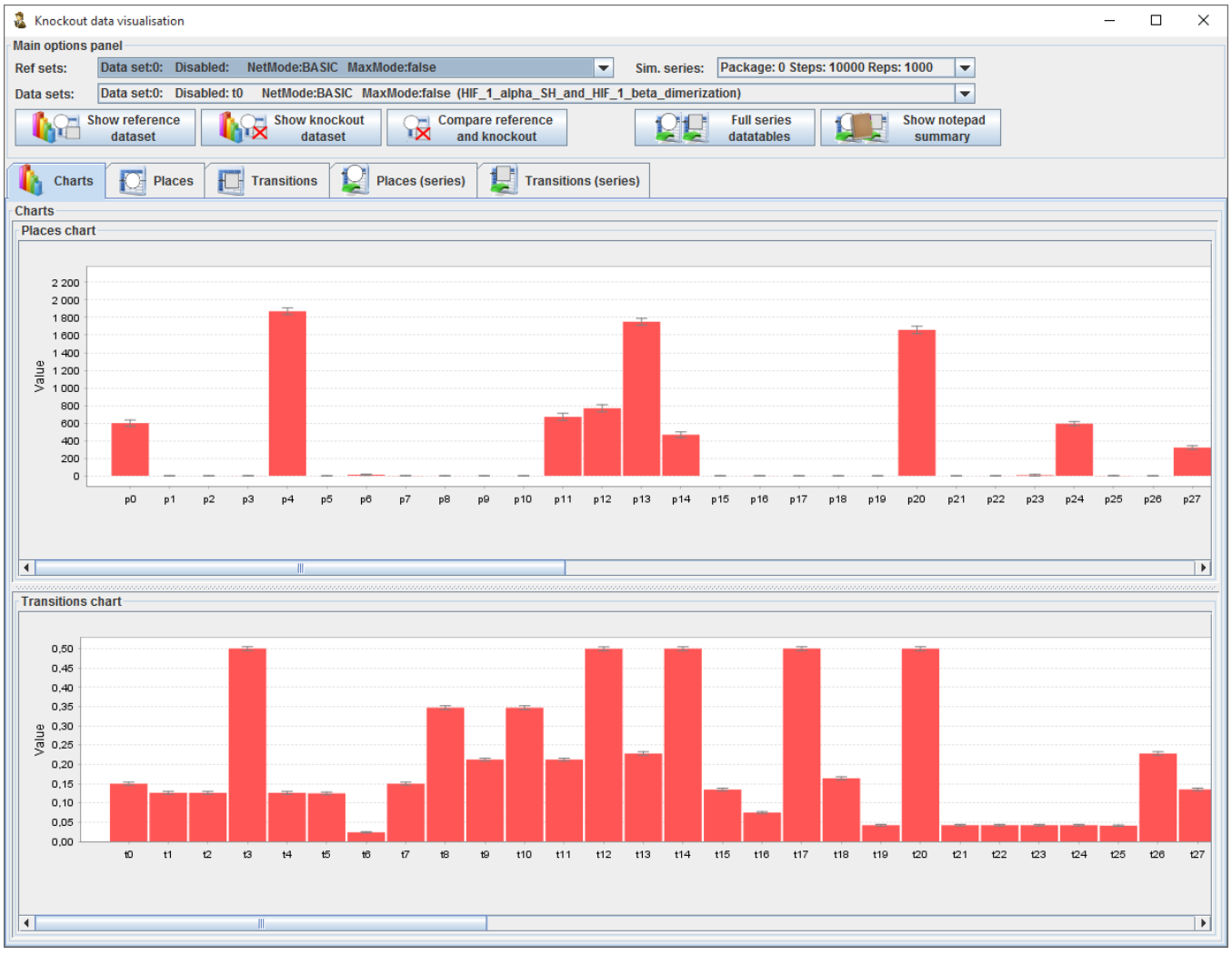
Last, fourth panel of the **KnockoutSim** tab shows the signatures for every knockout simulation performed using the third panel:



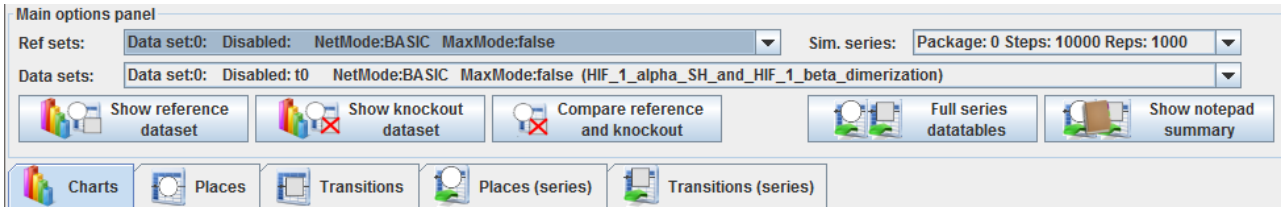
The real analysis is hidden behind the 

button and the window it activates.

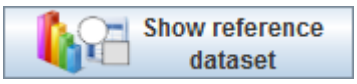
For the example below it is assumed that the reference set has been calculated, and all sets for all disabled transitions one by one as well are available (**All transitions (one by one)** mode used).

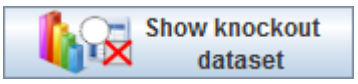


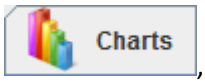
In this window the reference set (if more than one is available) and the knockout set can be compared.

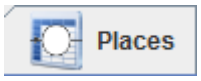


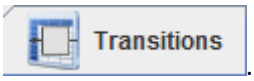
Fields **Ref sets**, **Data sets** and **Sim. series** mean respectively: the reference set, the (one, selected) knockout set or the data package for knockout, available when the **All transitions (one by one)** option in the third panel of the last described window has been used in the knockout simulation.

Button 

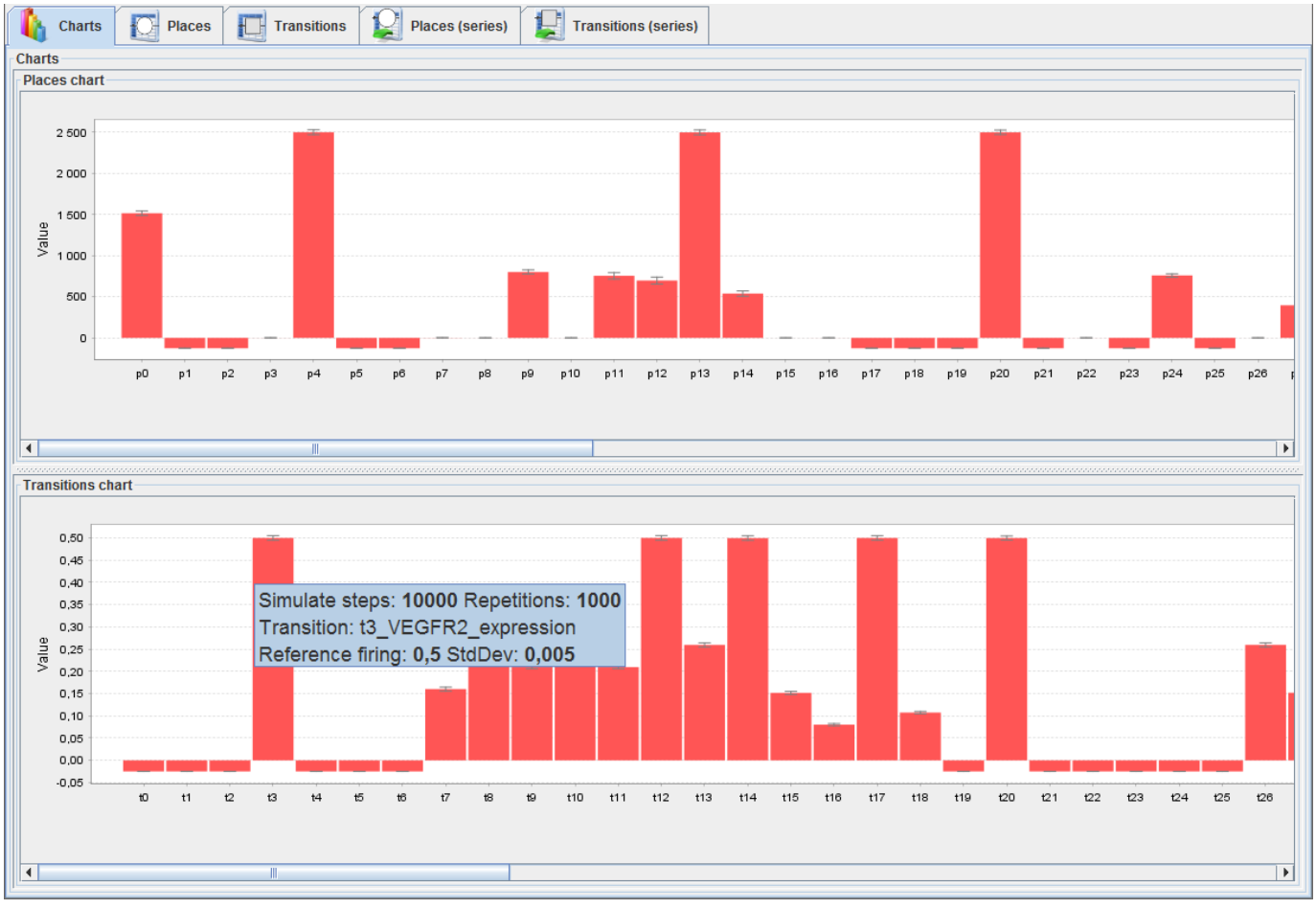
and 

shows data tables for the reference and the knockout sets. The example below concerns the second button. Its pressing will fill the table in three tabs: 

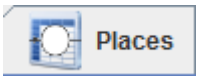
, 

and 

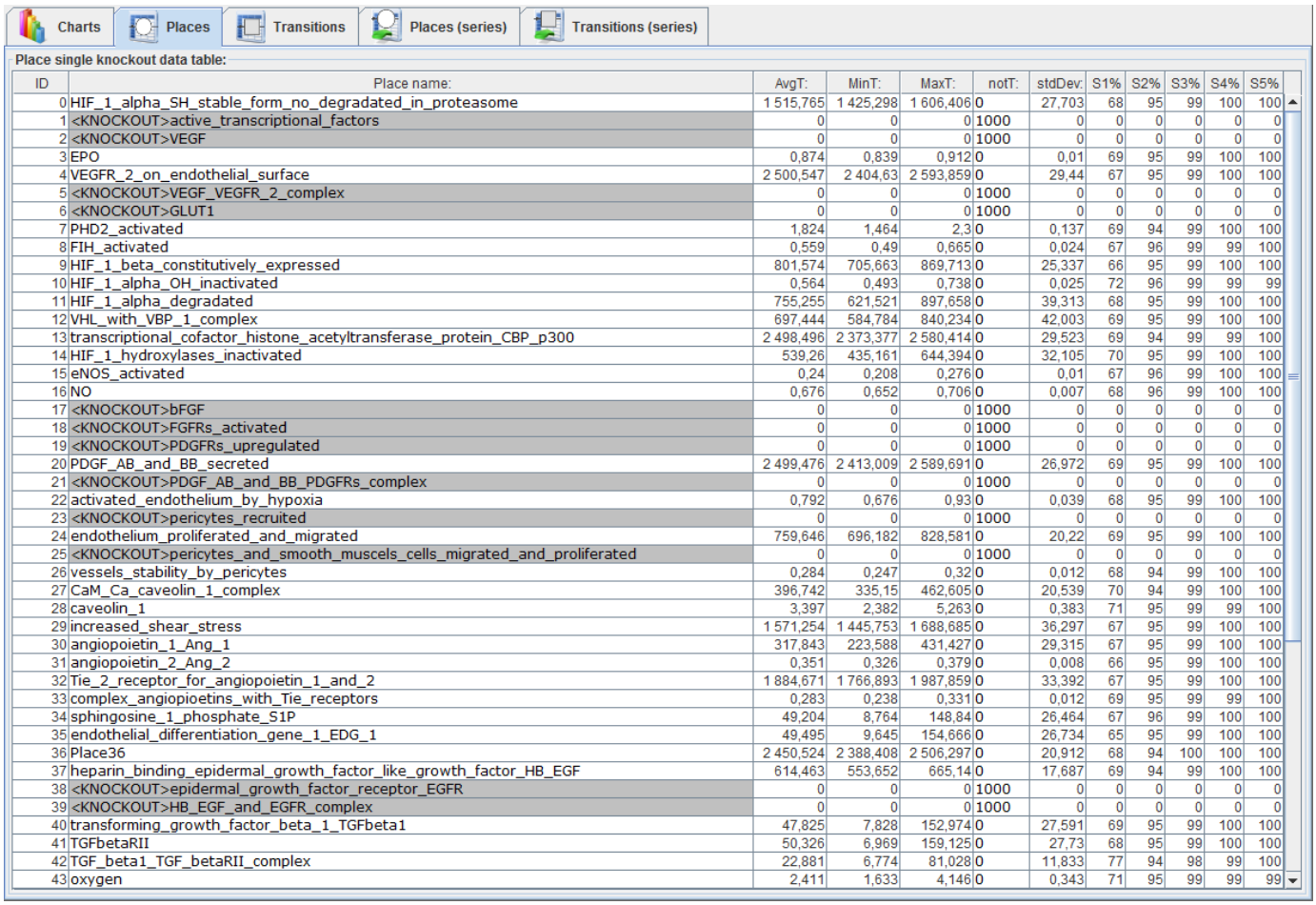
. First one shows the data in charts:



The upper chart is for places, the lower for transitions. First one shows the sum of accumulated tokens, the second – average transition firing. The 'negative' bars are only used to mark transitions/places which did not fired/did not accumulated tokens – in order to clearly distinguish them from the ones that e.g. fired rarely, but more than 0 during the simulation. Such bars often indicate places and transitions which were affected the most by a knockout of some other transition.

In a tab 

more detailed data are available.

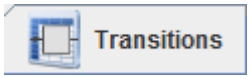


ID	Place name:	AvgT:	MinT:	MaxT:	notT:	stdDev:	S1%	S2%	S3%	S4%	S5%
0	HIF_1_alpha_SH_stable_form_no_degradated_in_proteasome	1 515,765	1 425,298	1 606,406	0	27,703	68	95	99	100	100
1	<KNOCKOUT>active_transcriptional_factors	0	0	0	1000	0	0	0	0	0	0
2	<KNOCKOUT>VEGF	0	0	0	1000	0	0	0	0	0	0
3	EPO	0,874	0,839	0,912	0	0,01	69	95	99	100	100
4	VEGFR_2_on_endothelial_surface	2 500,547	2 404,63	2 593,859	0	29,44	67	95	99	100	100
5	<KNOCKOUT>VEGF_VEGFR_2_complex	0	0	0	1000	0	0	0	0	0	0
6	<KNOCKOUT>GLUT1	0	0	0	1000	0	0	0	0	0	0
7	PHD2_activated	1,824	1,464	2,3	0	0,137	69	94	99	100	100
8	FIH_activated	0,559	0,49	0,665	0	0,024	67	96	99	99	100
9	HIF_1_beta_constitutively_expressed	801,574	705,663	869,713	0	25,337	66	95	99	100	100
10	HIF_1_alpha_OH_inactivated	0,564	0,493	0,738	0	0,025	72	96	99	99	99
11	HIF_1_alpha_degradated	755,255	621,521	897,658	0	39,313	68	95	99	100	100
12	VHL_with_VBP_1_complex	697,444	584,784	840,234	0	42,003	69	95	99	100	100
13	transcriptional_cofactor_histone_acetyltransferase_protein_CBP_p300	2 498,496	2 373,377	2 580,414	0	29,523	69	94	99	99	100
14	HIF_1_hydroxylases_inactivated	539,26	435,161	644,394	0	32,105	70	95	99	100	100
15	eNOS_activated	0,24	0,208	0,276	0	0,01	67	96	99	100	100
16	NO	0,676	0,652	0,706	0	0,007	68	96	99	100	100
17	<KNOCKOUT>bFGF	0	0	0	1000	0	0	0	0	0	0
18	<KNOCKOUT>FGFRs_activated	0	0	0	1000	0	0	0	0	0	0
19	<KNOCKOUT>PDGFRs_upregulated	0	0	0	1000	0	0	0	0	0	0
20	PDGF_AB_and_BB_secreted	2 499,476	2 413,009	2 589,691	0	26,972	69	95	99	100	100
21	<KNOCKOUT>PDGF_AB_and_BB_PDGFRs_complex	0	0	0	1000	0	0	0	0	0	0
22	activated_endothelium_by_hypoxia	0,792	0,676	0,93	0	0,039	68	95	99	100	100
23	<KNOCKOUT>pericytes_recruited	0	0	0	1000	0	0	0	0	0	0
24	endothelium_proliferated_and_migrated	759,646	696,182	828,581	0	20,22	69	95	99	100	100
25	<KNOCKOUT>pericytes_and_smooth_muscels_cells_migrated_and_proliferated	0	0	0	1000	0	0	0	0	0	0
26	vessels_stability_by_pericytes	0,284	0,247	0,32	0	0,012	68	94	99	100	100
27	CaM_Ca_caveolin_1_complex	396,742	335,15	462,605	0	20,539	70	94	99	100	100
28	caveolin_1	3,397	2,382	5,263	0	0,383	71	95	99	99	100
29	increased_shear_stress	1 571,254	1 445,753	1 688,685	0	36,297	67	95	99	100	100
30	angiopoietin_1_Ang_1	317,843	223,588	431,427	0	29,315	67	95	99	100	100
31	angiopoietin_2_Ang_2	0,351	0,326	0,379	0	0,008	66	95	99	100	100
32	Tie_2_receptor_for_angiopoietin_1_and_2	1 884,671	1 766,893	1 987,859	0	33,392	67	95	99	100	100
33	complex_angiopioetins_with_Tie_receptors	0,283	0,238	0,331	0	0,012	69	95	99	99	100
34	sphingosine_1_phosphate_S1P	49,204	8,764	148,84	0	26,464	67	96	99	100	100
35	endothelial_differentiation_gene_1_EDG_1	49,495	9,645	154,666	0	26,734	65	95	99	100	100
36	Place36	2 450,524	2 388,408	2 506,297	0	20,912	68	94	100	100	100
37	heparin_binding_epidermal_growth_factor_like_growth_factor_HB_EGF	614,463	553,652	665,14	0	17,687	69	94	99	100	100
38	<KNOCKOUT>epidermal_growth_factor_receptor_EGFR	0	0	0	1000	0	0	0	0	0	0
39	<KNOCKOUT>HB_EGF_and_EGFR_complex	0	0	0	1000	0	0	0	0	0	0
40	transforming_growth_factor_beta_1_TGFbeta1	47,825	7,828	152,974	0	27,591	69	95	99	100	100
41	TGFbetaRII	50,326	6,969	159,125	0	27,73	68	95	99	100	100
42	TGF_beta1_TGF_betaRII_complex	22,881	6,774	81,028	0	11,833	77	94	98	99	100
43	oxygen	2,411	1,633	4,146	0	0,343	71	95	99	99	99

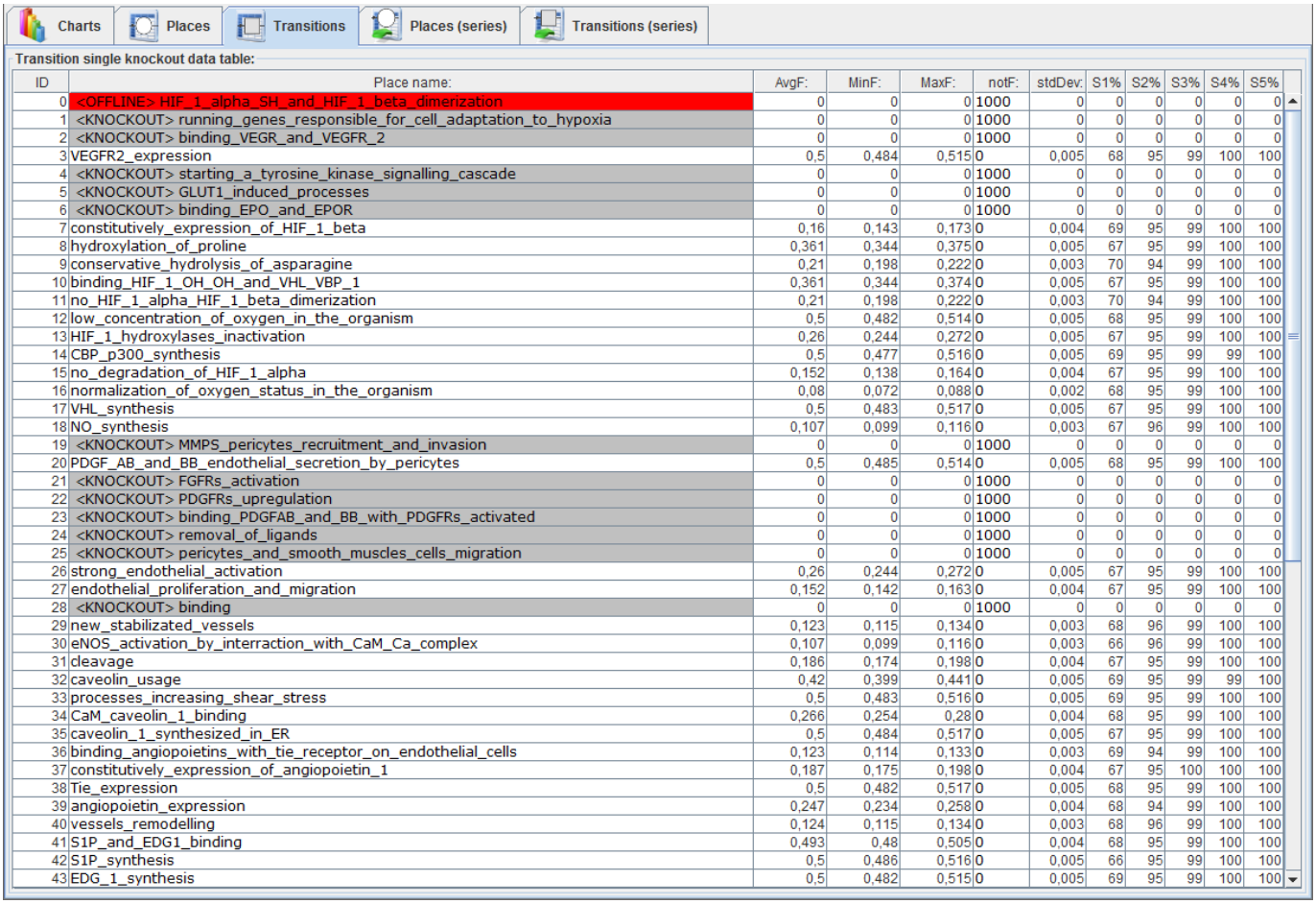
In the example rows are places. Names grayed indicated places which did not receive any tokens, possibly because of the knockout effect. The columns are as follows:

- **AvgT** – average number of sum of tokens in all steps of the simulation averaged by the number of repeated simulations.
- **MinT** – average minimal number of sum of tokens during all repeated simulations.
- **MaxT** – average maximal number of sum of tokens during all repeated simulations.
- **notT** – informs in how many simulation the place did not receive any tokens.

- **stdDev** – standard deviation of the results.
- **S1% - S5%** . Column S1% shows how many percent of results is the range <**AvgT** - **stdDev**, **AvgT** + **stdDev**>, S2% : <**AvgT** - 2*__stdDev__, **AvgT** + 2*__stdDev__>. Can be useful for checking Chebyshev's inequality.

Tab 

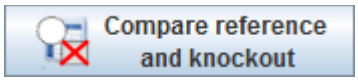
:

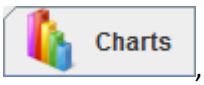


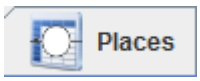
Transition single knockout data table:

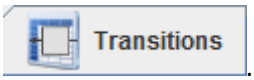
ID	Place name:	AvgF:	MinF:	MaxF:	notF:	stdDev:	S1%	S2%	S3%	S4%	S5%
0	<OFFLINE> HIF_1_alpha_SH_and_HIF_1_beta_dimerization	0	0	0	1000	0	0	0	0	0	0
1	<KNOCKOUT> running_genes_responsible_for_cell_adaptation_to_hypoxia	0	0	0	1000	0	0	0	0	0	0
2	<KNOCKOUT> binding_VEGR_and_VEGFR_2	0	0	0	1000	0	0	0	0	0	0
3	VEGFR2_expression	0,5	0,484	0,515	0	0,005	68	95	99	100	100
4	<KNOCKOUT> starting_a_tyrosine_kinase_signalling_cascade	0	0	0	1000	0	0	0	0	0	0
5	<KNOCKOUT> GLUT1_induced_processes	0	0	0	1000	0	0	0	0	0	0
6	<KNOCKOUT> binding_EPO_and_EPOR	0	0	0	1000	0	0	0	0	0	0
7	constitutively_expression_of_HIF_1_beta	0,16	0,143	0,173	0	0,004	69	95	99	100	100
8	hydroxylation_of_proline	0,361	0,344	0,375	0	0,005	67	95	99	100	100
9	conservative_hydrolysis_of_asparagine	0,21	0,198	0,222	0	0,003	70	94	99	100	100
10	binding_HIF_1_OH_OH_and_VHL_VBP_1	0,361	0,344	0,374	0	0,005	67	95	99	100	100
11	no_HIF_1_alpha_HIF_1_beta_dimerization	0,21	0,198	0,222	0	0,003	70	94	99	100	100
12	low_concentration_of_oxygen_in_the_organism	0,5	0,482	0,514	0	0,005	68	95	99	100	100
13	HIF_1_hydroxylases_inactivation	0,26	0,244	0,272	0	0,005	67	95	99	100	100
14	CBP_p300_synthesis	0,5	0,477	0,516	0	0,005	69	95	99	99	100
15	no_degradation_of_HIF_1_alpha	0,152	0,138	0,164	0	0,004	67	95	99	100	100
16	normalization_of_oxygen_status_in_the_organism	0,08	0,072	0,088	0	0,002	68	95	99	100	100
17	VHL_synthesis	0,5	0,483	0,517	0	0,005	67	95	99	100	100
18	NO_synthesis	0,107	0,099	0,116	0	0,003	67	96	99	100	100
19	<KNOCKOUT> MMPS_pericytes_recruitment_and_invasion	0	0	0	1000	0	0	0	0	0	0
20	PDGF_AB_and_BB_endothelial_secretion_by_pericytes	0,5	0,485	0,514	0	0,005	68	95	99	100	100
21	<KNOCKOUT> FGFRs_activation	0	0	0	1000	0	0	0	0	0	0
22	<KNOCKOUT> PDGFRs_upregulation	0	0	0	1000	0	0	0	0	0	0
23	<KNOCKOUT> binding_PDGFAB_and_BB_with_PDGFRs_activated	0	0	0	1000	0	0	0	0	0	0
24	<KNOCKOUT> removal_of_ligands	0	0	0	1000	0	0	0	0	0	0
25	<KNOCKOUT> pericytes_and_smooth_muscles_cells_migration	0	0	0	1000	0	0	0	0	0	0
26	strong_endothelial_activation	0,26	0,244	0,272	0	0,005	67	95	99	100	100
27	endothelial_proliferation_and_migration	0,152	0,142	0,163	0	0,004	67	95	99	100	100
28	<KNOCKOUT> binding	0	0	0	1000	0	0	0	0	0	0
29	new_stabilizated_vessels	0,123	0,115	0,134	0	0,003	68	96	99	100	100
30	eNOS_activation_by_interraction_with_CaM_Ca_complex	0,107	0,099	0,116	0	0,003	66	96	99	100	100
31	cleavage	0,186	0,174	0,198	0	0,004	67	95	99	100	100
32	caveolin_usage	0,42	0,399	0,441	0	0,005	69	95	99	99	100
33	processes_increasing_shear_stress	0,5	0,483	0,516	0	0,005	69	95	99	100	100
34	CaM_caveolin_1_binding	0,266	0,254	0,28	0	0,004	68	95	99	100	100
35	caveolin_1_synthesized_in_ER	0,5	0,484	0,517	0	0,005	67	95	99	100	100
36	binding_angiopoietins_with_tie_receptor_on_endothelial_cells	0,123	0,114	0,133	0	0,003	69	94	99	100	100
37	constitutively_expression_of_angiopoietin_1	0,187	0,175	0,198	0	0,004	67	95	100	100	100
38	Tie_expression	0,5	0,482	0,517	0	0,005	68	95	99	100	100
39	angiopoietin_expression	0,247	0,234	0,258	0	0,004	68	94	99	100	100
40	vessels_remodelling	0,124	0,115	0,134	0	0,003	68	96	99	100	100
41	S1P_and_EDG1_binding	0,493	0,48	0,505	0	0,004	68	95	99	100	100
42	S1P_synthesis	0,5	0,486	0,516	0	0,005	66	95	99	100	100
43	EDG_1_synthesis	0,5	0,482	0,515	0	0,005	69	95	99	100	100

Except that the average firing of transitions is considered here, the columns has the same properties as in the places tab. Row marked red is the transition manually set as disabled..

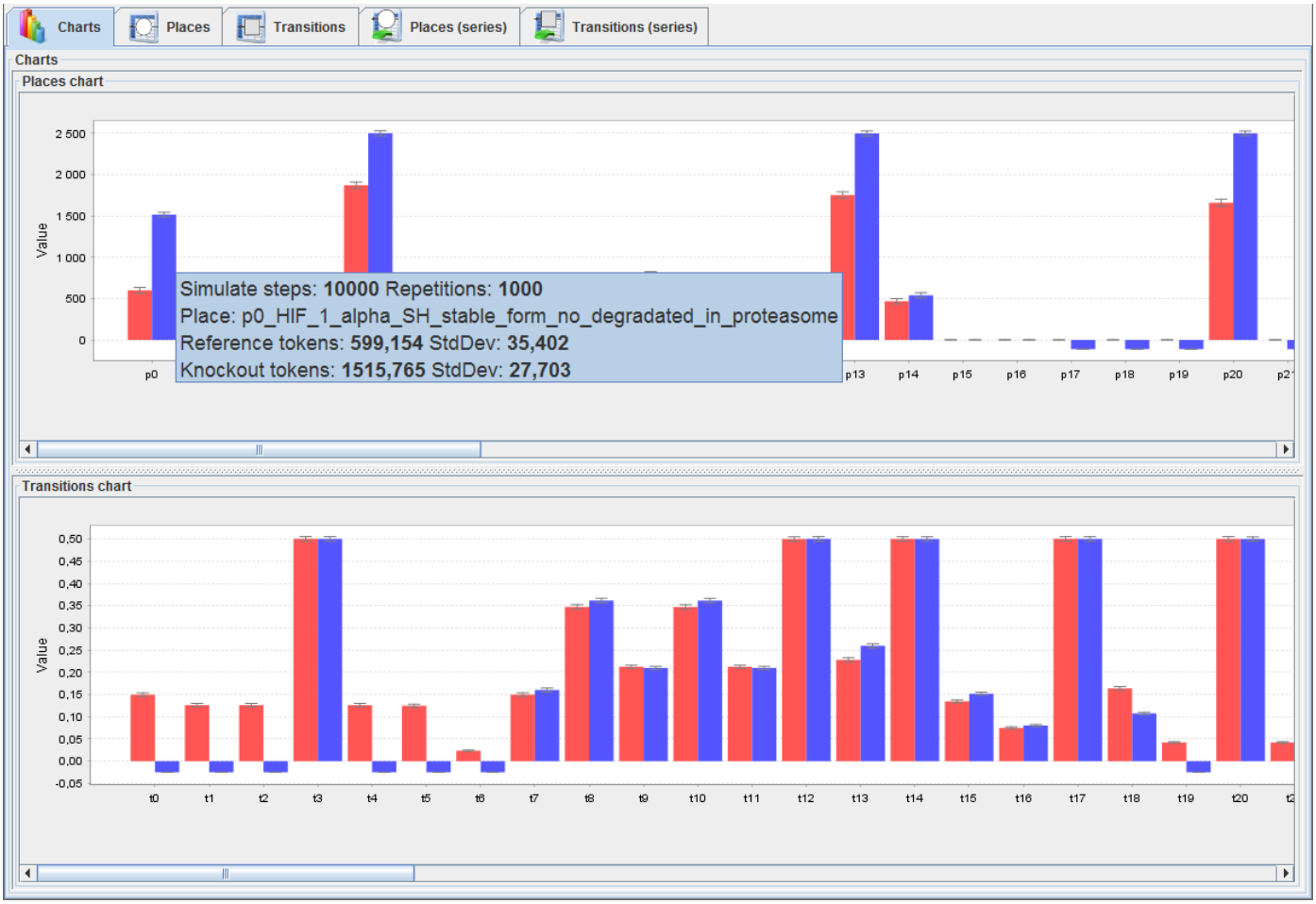
Button 

will change the data in: 

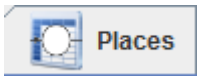
, 

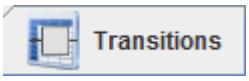
and 

. They will look like in the example picture:



Now the charts show the difference between reference set and knockout sets for places and transitions.

Also in 

and 

the tables changed.

In the example, transition t73 has been set to be disabled. There are no completely knocked out transitions, yet more subtle changes can be observed.

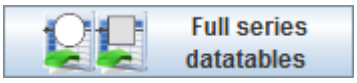
Charts | Places | Transitions | Places (series) | Transitions (series)

Transition single knockout data table:

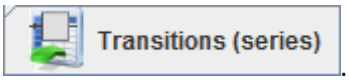
ID	Place name	AvgFRef	stdDevRef	AvgFKnock	stdDev...	diff: ▲	noFire	Sign1	Sign2
73	<OFFLINE> normal_state	0,5	0,005	0	0	-inf	1000	OK	OK
25	pericytes_and_smooth_muscles_cells_migration	0,041	0,002	0,007	0,001	-83,22%	0	OK	OK
28	binding	0,041	0,002	0,007	0,001	-83,17%	0	OK	OK
24	removal_of_ligands	0,042	0,002	0,007	0,001	-82,6%	0	OK	OK
23	binding_PDGFAB_and_BB_with_PDGFRs_activated	0,042	0,002	0,007	0,001	-82,6%	0	OK	OK
22	PDGFRs_upregulation	0,042	0,002	0,007	0,001	-82,6%	0	OK	OK
52	synthesis_factors_that_variously_stimulate_proliferation_survival_bFGF	0,042	0,002	0,007	0,001	-82,6%	0	OK	OK
21	FGFRs_activation	0,042	0,002	0,007	0,001	-82,6%	0	OK	OK
2	binding_VEGR_and_VEGFR_2	0,126	0,004	0,022	0,001	-82,57%	0	OK	OK
51	synthesis_factors_that_variously_stimulate_vessels_permability	0,042	0,002	0,007	0,001	-82,57%	0	OK	OK
1	running_genes_responsible_for_cell_adaptation_to_hypoxia	0,126	0,004	0,022	0,001	-82,57%	0	OK	OK
4	starting_a_tyrosine_kinase_signalling_cascade	0,126	0,004	0,022	0,001	-82,57%	0	OK	OK
19	MMPS_pericytes_recruitment_and_invasion	0,042	0,002	0,007	0,001	-82,54%	0	OK	OK
45	binding_HB_EGF_with_EGFR	0,042	0,002	0,007	0,001	-82,54%	0	OK	OK
53	synthesis_factors_that_variously_stimulate_migration_MMPs	0,042	0,002	0,007	0,001	-82,54%	0	OK	OK
0	HIF_1_alpha_SH_and_HIF_1_beta_dimerization	0,15	0,004	0,026	0,001	-82,4%	0	OK	OK
7	constitutively_expression_of_HIF_1_beta	0,15	0,004	0,026	0,001	-82,4%	0	OK	OK
50	complex_binding_to_HREs_hypoxia_responsive_elements	0,15	0,004	0,026	0,001	-82,4%	0	OK	OK
5	GLUT1_induced_processes	0,125	0,003	0,022	0,001	-82,38%	0	OK	OK
6	binding_EPO_and_EPOR	0,024	0,001	0,004	0,001	-81,48%	0	OK	OK
72	EPOR_synthesis_induced_by_hypoxia	0,024	0,001	0,004	0,001	-81,48%	0	OK	OK
55	CaM_EPOR_binding	0,024	0,001	0,004	0,001	-81,48%	0	OK	OK
56	upregulation_of_EPOR_signalling_pathways	0,024	0,001	0,004	0,001	-81,48%	0	OK	OK
64	HIF_1_alpha_proline_hydroxylases_ativation	0,212	0,004	0,053	0,002	-74,92%	0	OK	OK
63	HIF_1_alpha_asparagine_hydroxylases_activation	0,212	0,004	0,053	0,002	-74,92%	0	OK	OK
9	conservative_hydrolysis_of_asparagine	0,212	0,004	0,053	0,002	-74,92%	0	OK	OK
11	no_HIF_1_alpha_HIF_1_beta_dimerization	0,212	0,004	0,053	0,002	-74,91%	0	OK	OK
8	hydroxylation_of_proline	0,347	0,005	0,199	0,004	-42,78%	0	OK	OK
10	binding_HIF_1_OH_OH_and_VHL_VBP_1	0,347	0,005	0,198	0,004	-42,77%	0	OK	OK
66	activation_of_proteasome_degradation	0,347	0,005	0,198	0,004	-42,76%	0	OK	OK
18	NO_synthesis	0,164	0,004	0,104	0,003	-36,71%	0	OK	OK
30	eNOS_activation_by_interraction_with_CaM_Ca_complex	0,122	0,003	0,096	0,003	-20,91%	0	OK	OK
29	new_stabilizated_vessels	0,164	0,004	0,13	0,003	-20,59%	0	OK	OK
54	increase_in_intracellular_Ca	0,21	0,004	0,177	0,003	-15,61%	0	OK	OK
57	CaM_binds_Ca	0,21	0,004	0,177	0,003	-15,61%	0	OK	OK
37	constitutively_expression_of_angiopoietin_1	0,21	0,004	0,177	0,004	-15,61%	0	OK	OK
31	cleavage	0,209	0,004	0,177	0,004	-15,6%	0	OK	OK
34	CaM_caveolin_1_binding	0,274	0,004	0,253	0,004	-7,59%	0	n/a	OK
32	caveolin_usage	0,435	0,005	0,423	0,005	-2,8%	0	n/a	OK
46	TGFbeta1_TGFbetaRII_binding	0,494	0,004	0,494	0,004	-0,14%	0	n/a	n/a
47	activin_like_kinase_5_ALK_5_activation	0,49	0,004	0,49	0,004	-0,12%	0	n/a	n/a
3	VEGFR2_expression	0,5	0,005	0,5	0,005	-0,11%	0	n/a	n/a
17	VHL_synthesis	0,5	0,005	0,5	0,005	-0,1%	0	n/a	n/a
35	caveolin_1_synthesized_in_ER	0,5	0,005	0,5	0,005	-0,07%	0	n/a	n/a

Columns:

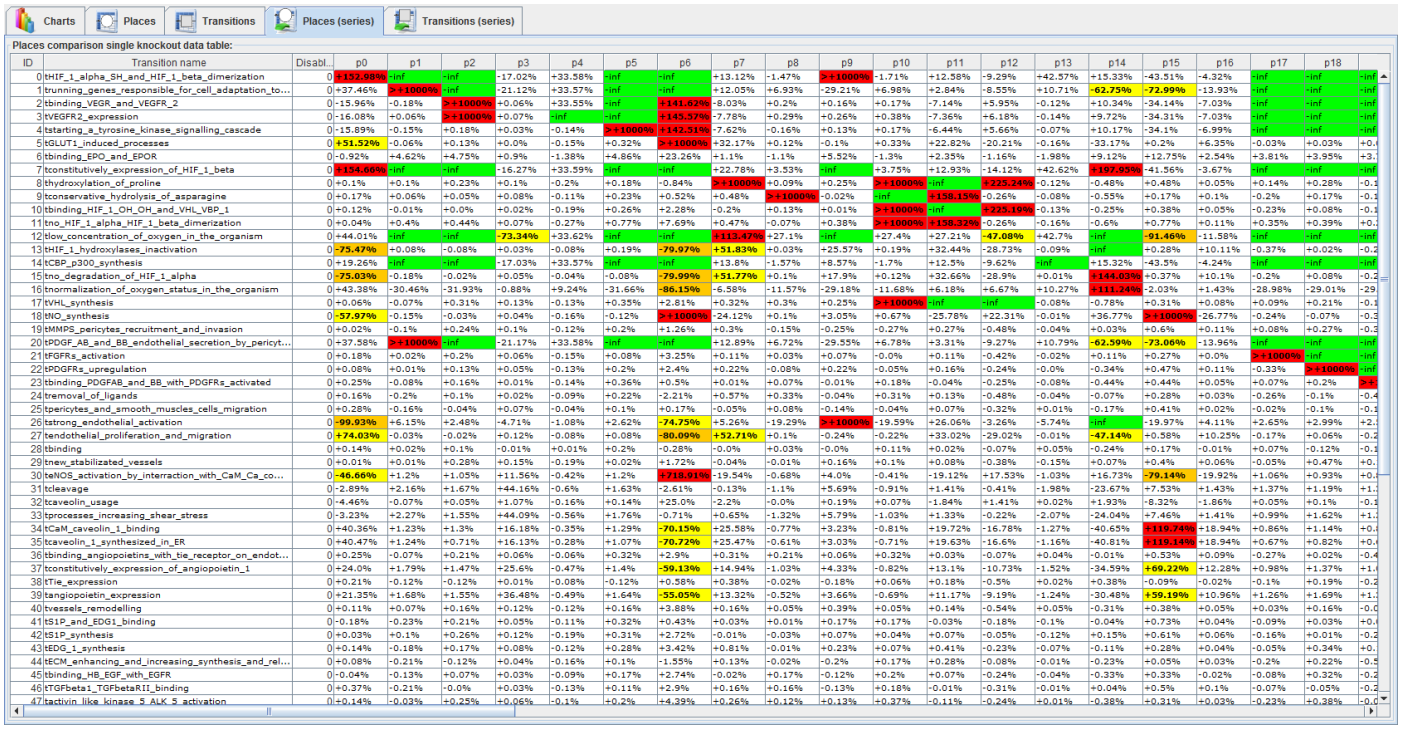
- **AvgFRef** – average number of firing in all steps in all simulations for the reference set.
- **stdDevRef** – standard deviation (1000 repeated simulation has been performed in the example).
- **AvgFKnock** i **stdDevKnock** – as the two above, for the knockout set.
- **Diff** – difference in average firing of a transition
- **noFire** – in how many simulation a given transition did not fire a single time.

Last button 

compares all places and all transitions together (from the reference and knockout data sets). Results will fill tab 

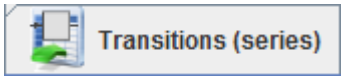
and 

. Example for the first one:

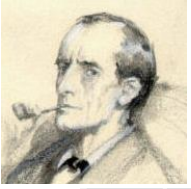


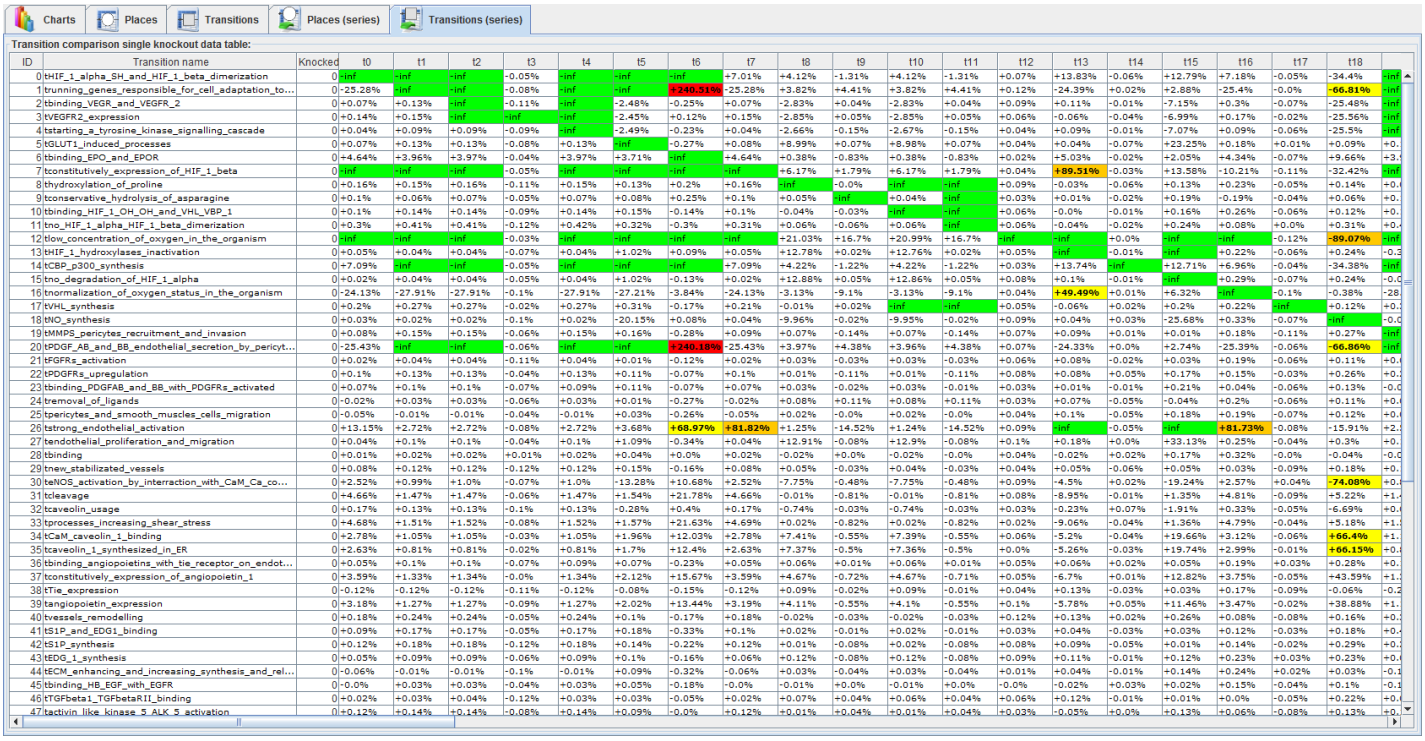
ID	Transition name	Disabl.	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18
0	tHIF_1_alpha_SH_and_HIF_1_beta_dimerization	0	+152.98%	-inf	-inf	-17.02%	+33.58%	-inf	-inf	+13.12%	-1.47%	>+1000%	-1.71%	+12.58%	-9.29%	+42.57%	+15.33%	-43.51%	-4.32%	-inf	-inf	-inf
1	trunning_genes_responsible_for_cell_adaptation_to...	0	+37.46%	>+1000%	-inf	-21.12%	+33.57%	-inf	-inf	+12.05%	+6.93%	-29.21%	+6.98%	+2.84%	-8.55%	+10.71%	-62.75%	-72.99%	-13.93%	-inf	-inf	-inf
2	tbinding_VEGR_and_VEGFR_2	0	-15.96%	-0.18%	>+1000%	+0.06%	+33.55%	-inf	+141.62%	-8.03%	+0.2%	+0.16%	+0.17%	-7.14%	+5.95%	-0.12%	+10.34%	-34.14%	-7.03%	-inf	-inf	-inf
3	tVEGFR2_expression	0	-16.08%	+0.06%	>+1000%	+0.07%	-inf	-inf	+145.57%	-7.78%	+0.29%	+0.26%	+0.38%	-7.36%	+6.18%	-0.14%	+9.72%	-34.31%	-7.03%	-inf	-inf	-inf
4	tstarting_a_tyrosine_kinase_signalling_cascade	0	-15.89%	-0.15%	+0.18%	+0.03%	-0.14%	>+1000%	+142.51%	-7.62%	-0.16%	+0.13%	+0.17%	-6.44%	+5.66%	-0.07%	+10.17%	-34.1%	-6.99%	-inf	-inf	-inf
5	tGLUT1_induced_processes	0	+51.52%	-0.06%	+0.13%	+0.0%	-0.15%	+0.32%	>+1000%	+32.17%	+0.12%	-0.1%	+0.33%	+22.82%	-20.21%	-0.16%	-33.17%	+0.2%	+6.35%	-0.03%	+0.03%	+0.0
6	tbinding_EPO_and_EPOR	0	-0.92%	+4.62%	+4.75%	+0.9%	-1.38%	+4.86%	+23.26%	+1.1%	-1.1%	+5.52%	-1.3%	+2.35%	-1.16%	-1.98%	+9.12%	+12.75%	+2.54%	+3.81%	+3.95%	+3.
7	tconstitutively_expression_of_HIF_1_beta	0	+154.66%	-inf	-inf	-16.27%	+33.59%	-inf	-inf	+22.78%	+3.53%	-inf	+3.75%	+12.93%	-14.12%	+42.62%	+197.93%	-41.56%	-3.67%	-inf	-inf	-inf
8	thydroxylation_of_proline	0	+0.1%	+0.1%	+0.23%	+0.1%	-0.2%	+0.18%	-0.84%	>+1000%	+0.09%	+0.25%	>+1000%	-inf	+225.24%	-0.12%	-0.48%	+0.48%	+0.05%	+0.14%	+0.28%	-0.1
9	tconservative_hydrolysis_of_asparagine	0	+0.17%	+0.06%	+0.05%	+0.08%	-0.11%	+0.23%	+0.52%	+0.48%	>+1000%	-0.02%	-inf	+158.15%	-0.26%	-0.08%	-0.55%	+0.17%	+0.1%	-0.2%	+0.17%	-0.1
10	tbinding_HIF_1_OH_OH_and_VHL_VBP_1	0	+0.12%	-0.01%	+0.0%	+0.02%	-0.19%	+0.26%	+2.28%	-0.2%	+0.13%	+0.01%	>+1000%	-inf	+225.19%	-0.13%	-0.25%	+0.38%	+0.05%	-0.23%	+0.08%	-0.1
11	tno_HIF_1_alpha_HIF_1_beta_dimerization	0	+0.04%	+0.4%	+0.44%	+0.07%	-0.27%	+0.77%	+7.69%	+0.47%	-0.07%	+0.38%	>+1000%	+158.32%	-0.26%	-0.16%	-0.6%	+0.77%	+0.11%	+0.35%	+0.39%	+0.
12	tlow_concentration_of_oxygen_in_the_organism	0	+44.01%	-inf	-inf	-73.34%	+33.62%	-inf	-inf	+113.47%	+27.1%	-inf	+27.4%	+27.21%	-47.08%	+42.7%	-inf	-91.46%	-11.58%	-inf	-inf	-inf
13	tHIF_1_hydroxylases_inactivation	0	-75.47%	+0.08%	-0.08%	+0.03%	-0.08%	+0.19%	-79.97%	+51.83%	+0.03%	+25.57%	+0.19%	+32.44%	-28.73%	-0.09%	-inf	+0.28%	+10.11%	-0.37%	+0.02%	-0.2
14	tCBP_p300_synthesis	0	+19.26%	-inf	-inf	-17.03%	+33.57%	-inf	-inf	+13.8%	-1.57%	+8.57%	-1.7%	+12.5%	-9.62%	-inf	+15.32%	-43.5%	-4.24%	-inf	-inf	-inf
15	tno_degradation_of_HIF_1_alpha	0	-75.03%	-0.18%	-0.02%	+0.05%	-0.04%	-0.08%	-79.99%	+51.77%	+0.1%	+17.9%	+0.12%	+32.66%	-28.9%	+0.01%	+144.03%	+0.37%	+10.1%	-0.2%	+0.08%	-0.2
16	tnormalization_of_oxygen_status_in_the_organism	0	+43.38%	-30.46%	-31.93%	-0.88%	+9.24%	-31.66%	-86.15%	-6.58%	-11.57%	-29.18%	-11.68%	+6.18%	+6.67%	+10.27%	+111.24%	-2.03%	+1.43%	-28.98%	-29.01%	-29
17	tVHL_synthesis	0	+0.06%	-0.07%	+0.31%	+0.13%	-0.13%	+0.35%	+2.81%	+0.32%	+0.3%	+0.25%	>+1000%	-inf	-inf	-0.08%	-0.78%	+0.31%	+0.08%	+0.09%	+0.21%	-0.1
18	tNO_synthesis	0	-57.97%	-0.15%	-0.03%	+0.04%	-0.16%	-0.12%	>+1000%	-24.12%	+0.1%	+3.05%	+0.67%	-25.78%	+22.31%	-0.01%	+36.77%	>+1000%	-26.77%	-0.24%	-0.07%	-0.3
19	tMMPS_pericytes_recruitment_and_invasion	0	+0.02%	-0.1%	+0.24%	+0.1%	-0.12%	+0.2%	+1.26%	+0.3%	-0.15%	-0.25%	-0.27%	+0.27%	-0.48%	-0.04%	+0.03%	+0.6%	+0.11%	+0.08%	+0.27%	-0.3
20	tPDGF_AB_and_BB_endothelial_secretion_by_pericyt...	0	+37.58%	>+1000%	-inf	-21.17%	+33.58%	-inf	-inf	+12.89%	+6.72%	-29.55%	+6.78%	+3.31%	-9.27%	+10.79%	-62.59%	-73.06%	-13.96%	-inf	-inf	-inf
21	tFGFRs_activation	0	+0.18%	+0.02%	+0.2%	+0.06%	-0.15%	+0.08%	+3.25%	+0.11%	+0.03%	+0.07%	-0.0%	+0.11%	-0.42%	-0.02%	+0.11%	+0.27%	+0.0%	>+1000%	-inf	-inf
22	tPDGFRs_upregulation	0	+0.08%	+0.01%	+0.13%	+0.05%	-0.13%	+0.2%	+2.4%	+0.22%	-0.08%	+0.22%	-0.05%	+0.16%	-0.24%	-0.0%	-0.34%	+0.47%	+0.11%	-0.33%	>+1000%	-inf
23	tbinding_PDGFAB_and_BB_with_PDGFRs_activated	0	+0.25%	-0.08%	+0.16%	+0.01%	-0.14%	+0.26%	+0.5%	+0.01%	+0.07%	-0.01%	+0.18%	-0.04%	-0.25%	-0.08%	-0.44%	+0.44%	+0.05%	+0.07%	+0.2%	>-1
24	tremoval_of_ligands	0	+0.16%	-0.2%	+0.1%	+0.02%	-0.09%	+0.22%	-2.21%	+0.57%	+0.33%	-0.04%	+0.31%	+0.13%	-0.48%	-0.04%	-0.07%	+0.28%	+0.03%	-0.26%	-0.1%	-0.4
25	tpericytes_and_smooth_muscles_cells_migration	0	+0.28%	-0.16%	-0.04%	+0.07%	-0.04%	+0.1%	+0.17%	-0.05%	+0.08%	-0.14%	-0.04%	+0.07%	-0.32%	+0.01%	-0.17%	+0.41%	+0.02%	-0.02%	-0.1%	-0.1
26	tstrong_endothelial_activation	0	-99.93%	+6.15%	+2.48%	-4.71%	-1.08%	+2.62%	-74.75%	+5.26%	-19.29%	>+1000%	-19.59%	+26.06%	-3.26%	-5.74%	-inf	-19.97%	+4.11%	+2.65%	+2.99%	+2.
27	tendothelial_proliferation_and_migration	0	+74.03%	-0.03%	-0.02%	+0.12%	-0.08%	+0.08%	-80.09%	+52.71%	+0.1%	-0.24%	-0.22%	+33.02%	-29.02%	-0.01%	-47.14%	+0.58%	+10.25%	-0.17%	+0.06%	-0.2
28	tbinding	0	+0.14%	+0.02%	+0.1%	-0.01%	+0.01%	+0.2%	-0.28%	-0.0%	+0.03%	-0.0%	+0.11%	+0.02%	-0.07%	+0.05%	-0.24%	+0.17%	-0.01%	+0.07%	-0.12%	-0.1
29	tnew_stabilizated_vessels	0	+0.01%	+0.01%	+0.28%	+0.15%	-0.19%	+0.02%	+1.72%	-0.04%	-0.01%	+0.16%	+0.1%	+0.08%	-0.38%	-0.15%	+0.07%	+0.4%	+0.06%	-0.05%	+0.47%	+0.
30	teNOS_activation_by_interraction_with_CaM_Ca_co...	0	-46.66%	+1.2%	+1.05%	+11.56%	-0.42%	+1.2%	+718.91%	-19.54%	-0.68%	+4.0%	-0.41%	-19.12%	+17.53%	-1.03%	+16.73%	-79.14%	-19.92%	+1.06%	+0.93%	+0.
31	tcleavage	0	-2.89%	+2.16%	+1.67%	+44.16%	-0.6%	+1.63%	-2.61%	-0.13%	-1.1%	+5.69%	-0.91%	+1.41%	-0.41%	-1.98%	-23.67%	+7.53%	+1.43%	+1.37%	+1.19%	+1.
32	tcaveolin_usage	0	-4.46%	-0.07%	+0.05%	+1.07%	-0.16%	+0.14%	+25.0%	-2.2%	-0.0%	+0.19%	+0.07%	-1.84%	+1.41%	+0.02%	+1.93%	-8.32%	-1.86%	+0.05%	+0.1%	-0.1
33	tprocesses_increasing_shear_stress	0	-3.23%	+2.27%	+1.55%	+44.09%	-0.56%	+1.76%	-0.71%	+0.65%	-1.32%	+5.79%	-1.03%	+1.33%	-0.22%	-2.07%	-24.04%	+7.46%	+1.41%	+0.99%	+1.62%	+1.
34	tCaM_caveolin_1_binding	0	+40.36%	+1.23%	+1.3%	+16.18%	-0.35%	+1.29%	-70.15%	+25.58%	-0.77%	+3.23%	-0.81%	+19.72%	-16.78%	-1.27%	-40.65%	+119.74%	+18.94%	+0.86%	+1.14%	+0.
35	tcaveolin_1_synthesized_in_ER	0	+40.47%	+1.24%	+0.71%	+16.13%	-0.28%	+1.07%	-70.72%	+25.47%	-0.61%	+3.03%	-0.71%	+19.63%	-16.6%	-1.16%	-40.81%	+119.14%	+18.94%	+0.67%	+0.82%	+0.
36	tbinding_angiopoietins_with_tie_receptor_on_endot...	0	+0.25%	-0.07%	+0.21%	+0.06%	-0.06%	+0.32%	+2.9%	+0.31%	+0.21%	+0.06%	+0.32%	+0.03%	-0.07%	+0.04%	-0.01%	+0.53%	+0.09%	-0.27%	+0.02%	-0.4
37	tconstitutively_expression_of_angiopoietin_1	0	+24.0%	+1.79%	+1.47%	+25.6%	-0.47%	+1.4%	-59.13%	+14.94%	-1.03%	+4.33%	-0.82%	+13.1%	-10.73%	-1.52%	-34.59%	+69.22%	+12.28%	+0.98%	+1.37%	+1.
38	tTie_expression	0	+0.21%	-0.12%	-0.12%	+0.01%	-0.08%	-0.12%	+0.58%	+0.38%	-0.02%	-0.18%	+0.06%	+0.18%	-0.5%	+0.02%	+0.38%	-0.09%	-0.02%	-0.1%	+0.19%	-0.2
39	tangiopoietin_expression	0	+21.35%	+1.68%	+1.55%	+36.48%	-0.49%	+1.64%	-55.05%	+13.32%	-0.52%	+3.66%	-0.69%	+11.17%	-9.19%	-1.24%	-30.48%	+59.19%	+10.96%	+1.26%	+1.69%	+1.
40	tvessels_remodelling	0	+0.11%	+0.07%	+0.16%	+0.12%	-0.12%	+0.16%	+3.88%	+0.16%	+0.05%	+0.39%	+0.05%	+0.14%	-0.54%	+0.05%	-0.31%	+0.38%	+0.05%	+0.03%	+0.16%	-0.0
41	tS1P_and_EDG1_binding	0	-0.18%	-0.23%	+0.21%	+0.05%	-0.11%	+0.32%	+0.43%	+0.03%	+0.01%	+0.17%	+0.17%	-0.03%	-0.18%	-0.1%	-0.04%	+0.73%	+0.04%	-0.09%	+0.03%	+0.
42	tS1P_synthesis	0	+0.03%	+0.1%	+0.26%	+0.12%	-0.19%	+0.31%	+2.72%	-0.01%	-0.03%	+0.07%	+0.04%	+0.07%	-0.05%	-0.12%	+0.15%	+0.61%	+0.06%	-0.16%	+0.01%	-0.2
43	tEDG_1_synthesis	0	+0.14%	-0.18%	+0.17%	+0.08%	-0.12%	+0.28%	+3.42%	+0.81%	-0.01%	+0.23%	+0.07%	+0.41%	-0.23%	-0.07%	-0.11%	+0.28%	+0.04%	-0.05%	+0.34%	+0.
44	tECM_enhancing_and_increasing_synthesis_and_rel...	0	+0.08%	-0.21%	-0.12%	+0.04%	-0.16%	+0.1%	-1.55%	+0.13%	-0.02%	-0.2%	+0.17%	+0.28%	-0.08%	-0.01%	-0.23%	+0.05%	+0.03%	-0.2%	+0.22%	-0.5
45	tbinding_HB_EGF_with_EGFR	0	-0.04%	-0.13%	+0.07%	+0.03%	-0.09%	+0.17%	+2.74%	-0.02%	+0.17%	-0.12%	+0.2%	+0.07%	-0.24%	-0.04%	-0.33%	+0.33%	-0.02%	-0.08%	+0.32%	-0.2
46	tTGFbeta1_TGFbetaRII_binding	0	+0.37%	-0.21%	-0.0%	+0.03%	-0.13%	+0.11%	+2.9%	+0.16%	+0.16%	-0.13%	+0.18%	-0.01%	-0.31%	-0.01%	+0.04%	+0.5%	+0.1%	-0.07%	-0.05%	-0.2
47	tactivin_like_kinase_5_ALK_5_activation	0	+0.14%	-0.03%	+0.25%	+0.06%	-0.1%	+0.2%	+4.39%	+0.26%	+0.12%	+0.13%	+0.37%	-0.11%	-0.24%	+0.01%	-0.38%	+0.31%	+0.03%	-0.23%	+0.38%	-0.0

Simply speaking, average number of (sum of) tokens is compared here (reference and knockout sets) and the difference given. In the first row (ID=0) indicates that transition t0 has been disabled. The column p0 (for first place) has value +152.98%. It means that much more tokens were accumulated in the knockout scenario compared to the reference scenario in p0. –inf in p1 means, that disabling transition t0 (we are still in the first row) crippled the production of tokens in p1 (it did not receive a single token in the knockout scenario simulations).

Table in tab 

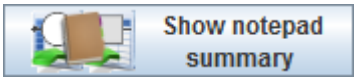
looks similar, here transitions are compared.





Transition comparison single knockout data table:

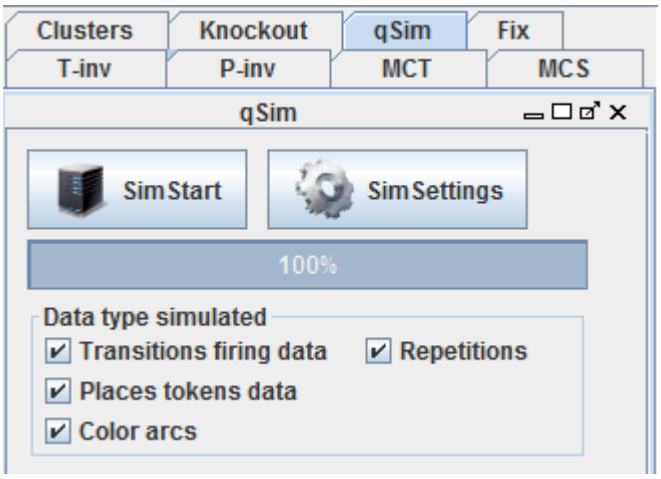
ID	Transition name	Knocked	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18
0	tHIF_1_alpha_SH_and_HIF_1_beta_dimerization	0	-inf	-inf	-inf	-0.05%	-inf	-inf	-inf	+7.01%	+4.12%	-1.31%	+4.12%	-1.31%	+0.07%	+13.83%	-0.06%	+12.79%	+7.18%	-0.05%	-34.4%
1	trunning_genes_responsible_for_cell_adaptation_to...	0	-25.28%	-inf	-inf	-0.08%	-inf	-inf	+240.51%	-25.28%	+3.82%	+4.41%	+3.82%	+4.41%	+0.12%	-24.39%	+0.02%	+2.88%	-25.4%	-0.0%	-66.81%
2	tbinding_VEGR_and_VEGFR_2	0	+0.07%	+0.13%	-inf	-0.11%	-inf	-2.48%	-0.25%	+0.07%	-2.83%	+0.04%	-2.83%	+0.04%	+0.09%	+0.11%	-0.01%	-7.15%	+0.3%	-0.07%	-25.48%
3	tVEGFR2_expression	0	+0.14%	+0.15%	-inf	-inf	-inf	-2.45%	+0.12%	+0.15%	-2.85%	+0.05%	-2.85%	+0.05%	+0.06%	-0.06%	-0.04%	-6.99%	+0.17%	-0.02%	-25.56%
4	tstarting_a_tyrosine_kinase_signalling_cascade	0	+0.04%	+0.09%	+0.09%	-0.09%	-inf	-2.49%	-0.23%	+0.04%	-2.66%	-0.15%	-2.67%	-0.15%	+0.04%	+0.09%	-0.01%	-7.07%	+0.09%	-0.06%	-25.5%
5	tGLUT1_induced_processes	0	+0.07%	+0.13%	+0.13%	-0.08%	+0.13%	-inf	-0.27%	+0.08%	+8.99%	+0.07%	+8.98%	+0.07%	+0.04%	+0.04%	-0.07%	+23.25%	+0.18%	+0.01%	+0.09%
6	tbinding_EPO_and_EPOR	0	+4.64%	+3.96%	+3.97%	-0.04%	+3.97%	+3.71%	-inf	+4.64%	+0.38%	-0.83%	+0.38%	-0.83%	+0.02%	+5.03%	-0.02%	+2.05%	+4.34%	-0.07%	+9.66%
7	tconstitutively_expression_of_HIF_1_beta	0	-inf	-inf	-inf	-0.05%	-inf	-inf	-inf	-inf	+6.17%	+1.79%	+6.17%	+1.79%	+0.04%	+89.51%	-0.03%	+13.58%	-10.21%	-0.11%	-32.42%
8	thydroxylation_of_proline	0	+0.16%	+0.15%	+0.16%	-0.11%	+0.15%	+0.13%	+0.2%	+0.16%	-inf	-0.0%	-inf	-inf	+0.09%	-0.03%	-0.06%	+0.13%	+0.23%	-0.05%	+0.14%
9	tconservative_hydrolysis_of_asparagine	0	+0.1%	+0.06%	+0.07%	-0.05%	+0.07%	+0.08%	+0.25%	+0.1%	+0.05%	-inf	+0.04%	-inf	+0.03%	+0.01%	-0.02%	+0.19%	-0.19%	-0.04%	+0.06%
10	tbinding_HIF_1_OH_OH_and_VHL_VBP_1	0	+0.1%	+0.14%	+0.14%	-0.09%	+0.14%	+0.15%	-0.14%	+0.1%	-0.04%	-0.03%	-inf	-inf	+0.06%	-0.0%	-0.01%	+0.16%	+0.26%	-0.06%	+0.12%
11	tno_HIF_1_alpha_HIF_1_beta_dimerization	0	+0.3%	+0.41%	+0.41%	-0.12%	+0.42%	+0.32%	-0.3%	+0.31%	+0.06%	-0.06%	+0.06%	-inf	+0.06%	-0.04%	-0.02%	+0.24%	+0.08%	+0.0%	+0.31%
12	tlow_concentration_of_oxygen_in_the_organism	0	-inf	-inf	-inf	-0.03%	-inf	-inf	-inf	-inf	+21.03%	+16.7%	+20.99%	+16.7%	+inf	-inf	+0.0%	-inf	-inf	-0.12%	-89.07%
13	tHIF_1_hydroxylases_inactivation	0	+0.05%	+0.04%	+0.04%	-0.07%	+0.04%	+1.02%	+0.09%	+0.05%	+12.78%	+0.02%	+12.76%	+0.02%	+0.05%	-inf	-0.01%	-inf	+0.22%	-0.06%	+0.24%
14	tCBP_p300_synthesis	0	+7.09%	-inf	-inf	-0.05%	-inf	-inf	-inf	+7.09%	+4.22%	-1.22%	+4.22%	-1.22%	+0.03%	+13.74%	-inf	+12.71%	+6.96%	-0.04%	-34.38%
15	tno_degradation_of_HIF_1_alpha	0	+0.02%	+0.04%	+0.04%	-0.05%	+0.04%	+1.02%	-0.13%	+0.02%	+12.88%	+0.05%	+12.86%	+0.05%	+0.08%	+0.1%	-0.01%	-inf	+0.29%	-0.07%	+0.24%
16	tnormalization_of_oxygen_status_in_the_organism	0	-24.13%	-27.91%	-27.91%	-0.1%	-27.91%	-27.21%	-3.84%	-24.13%	-3.13%	-9.1%	-3.13%	-9.1%	+0.04%	+49.49%	+0.01%	+6.32%	-inf	-0.1%	-0.38%
17	tVHL_synthesis	0	+0.2%	+0.27%	+0.27%	-0.02%	+0.27%	+0.31%	-0.17%	+0.21%	-0.01%	+0.02%	-inf	-inf	+0.05%	-0.06%	+0.02%	+0.2%	+0.22%	-inf	+0.12%
18	tNO_synthesis	0	+0.03%	+0.02%	+0.02%	-0.1%	+0.02%	-20.15%	+0.08%	+0.04%	-9.96%	-0.02%	-9.95%	-0.02%	+0.09%	+0.04%	+0.03%	-25.68%	+0.33%	-0.07%	-inf
19	tMMP9_pericytes_recruitment_and_invasion	0	+0.08%	+0.15%	+0.15%	-0.06%	+0.15%	+0.16%	-0.28%	+0.09%	+0.07%	-0.14%	+0.07%	-0.14%	+0.07%	+0.09%	+0.01%	+0.01%	+0.18%	-0.11%	+0.27%
20	tPDGF_AB_and_BB_endothelial_secretion_by_pericyt...	0	-25.43%	-inf	-inf	-0.06%	-inf	-inf	+240.18%	-25.43%	+3.97%	+4.38%	+3.96%	+4.38%	+0.07%	-24.33%	+0.0%	+2.74%	-25.39%	-0.06%	-66.86%
21	tFGFRs_activation	0	+0.02%	+0.04%	+0.04%	-0.11%	+0.04%	+0.01%	-0.12%	+0.02%	+0.03%	-0.03%	+0.03%	-0.03%	+0.06%	+0.08%	-0.02%	+0.03%	+0.19%	-0.06%	+0.11%
22	tPDGFRs_upregulation	0	+0.1%	+0.13%	+0.13%	-0.04%	+0.13%	+0.11%	-0.07%	+0.1%	+0.01%	-0.11%	+0.01%	-0.11%	+0.08%	+0.08%	+0.05%	+0.17%	+0.15%	-0.03%	+0.26%
23	tbinding_PDGFAB_and_BB_with_PDGFRs_activated	0	+0.07%	+0.1%	+0.1%	-0.07%	+0.09%	+0.11%	-0.07%	+0.07%	+0.03%	-0.02%	+0.03%	-0.01%	+0.03%	+0.01%	-0.01%	+0.21%	+0.04%	-0.06%	+0.13%
24	tremoval_of_ligands	0	-0.02%	+0.03%	+0.03%	-0.06%	+0.03%	+0.01%	-0.27%	-0.02%	+0.08%	+0.11%	+0.08%	+0.11%	+0.03%	+0.07%	-0.05%	-0.04%	+0.2%	-0.06%	+0.11%
25	tpericytes_and_smooth_muscles_cells_migration	0	-0.05%	-0.01%	-0.01%	-0.04%	-0.01%	+0.03%	-0.26%	-0.05%	+0.02%	-0.0%	+0.02%	-0.0%	+0.04%	+0.1%	-0.05%	+0.18%	+0.19%	-0.07%	+0.12%
26	tstrong_endothelial_activation	0	+13.15%	+2.72%	+2.72%	-0.08%	+2.72%	+3.68%	+68.97%	+81.82%	+1.25%	-14.52%	+1.24%	-14.52%	+0.09%	-inf	-0.05%	-inf	+81.73%	-0.08%	-15.91%
27	tendothelial_proliferation_and_migration	0	+0.04%	+0.1%	+0.1%	-0.04%	+0.1%	+1.09%	-0.34%	+0.04%	+12.91%	-0.08%	+12.9%	-0.08%	+0.1%	+0.18%	+0.0%	+33.13%	+0.25%	-0.04%	+0.3%
28	tbinding	0	+0.01%	+0.02%	+0.02%	+0.01%	+0.02%	+0.04%	+0.0%	+0.02%	-0.02%	+0.0%	-0.02%	-0.0%	+0.04%	-0.02%	+0.02%	+0.17%	+0.32%	-0.0%	-0.04%
29	tnew_stabilizated_vessels	0	+0.08%	+0.12%	+0.12%	-0.12%	+0.12%	+0.15%	-0.16%	+0.08%	+0.05%	-0.03%	+0.04%	-0.03%	+0.04%	+0.05%	-0.06%	+0.05%	+0.03%	-0.09%	+0.18%
30	teNOS_activation_by_interaction_with_CaM_Ca_co...	0	+2.52%	+0.99%	+1.0%	-0.07%	+1.0%	-13.28%	+10.68%	+2.52%	-7.75%	-0.48%	-7.75%	-0.48%	+0.09%	-4.5%	+0.02%	-19.24%	+2.57%	+0.04%	-74.08%
31	tcleavage	0	+4.66%	+1.47%	+1.47%	-0.06%	+1.47%	+1.54%	+21.78%	+4.66%	-0.01%	-0.81%	-0.01%	-0.81%	+0.08%	-8.95%	-0.01%	+1.35%	+4.81%	-0.09%	+5.22%
32	tcaveolin_usage	0	+0.17%	+0.13%	+0.13%	-0.1%	+0.13%	-0.28%	+0.4%	+0.17%	-0.74%	-0.03%	-0.74%	-0.03%	+0.03%	-0.23%	+0.07%	-1.91%	+0.33%	-0.05%	-6.69%
33	tprocesses_increasing_shear_stress	0	+4.68%	+1.51%	+1.52%	-0.08%	+1.52%	+1.57%	+21.63%	+4.69%	+0.02%	-0.82%	+0.02%	-0.82%	+0.02%	-9.06%	-0.04%	+1.36%	+4.79%	-0.04%	+5.18%
34	tCaM_caveolin_1_binding	0	+2.78%	+1.05%	+1.05%	-0.03%	+1.05%	+1.96%	+12.03%	+2.78%	+7.41%	-0.55%	+7.39%	-0.55%	+0.06%	-5.2%	-0.04%	+19.66%	+3.12%	-0.06%	+66.4%
35	tcaveolin_1_synthesized_in_ER	0	+2.63%	+0.81%	+0.81%	-0.02%	+0.81%	+1.7%	+12.4%	+2.63%	+7.37%	-0.5%	+7.36%	-0.5%	+0.0%	-5.26%	-0.03%	+19.74%	+2.99%	-0.01%	+66.15%
36	tbinding_angiopoietins_with_tie_receptor_on_endot...	0	+0.05%	+0.1%	+0.1%	-0.07%	+0.09%	+0.07%	-0.23%	+0.05%	+0.06%	+0.01%	+0.06%	+0.01%	+0.05%	+0.06%	+0.02%	+0.05%	+0.19%	+0.03%	+0.28%
37	tconstitutively_expression_of_angiopoietin_1	0	+3.59%	+1.33%	+1.34%	-0.0%	+1.34%	+2.12%	+15.67%	+3.59%	+4.67%	-0.72%	+4.67%	-0.71%	+0.05%	-6.7%	+0.01%	+12.82%	+3.75%	-0.05%	+43.59%
38	tTie_expression	0	-0.12%	-0.12%	-0.12%	-0.11%	-0.12%	-0.08%	-0.15%	-0.12%	+0.09%	-0.02%	+0.09%	-0.01%	+0.04%	+0.13%	-0.03%	+0.03%	+0.17%	-0.09%	-0.06%
39	tangiopoietin_expression	0	+3.18%	+1.27%	+1.27%	-0.09%	+1.27%	+2.02%	+13.44%	+3.19%	+4.11%	-0.55%	+4.1%	-0.55%	+0.1%	-5.78%	+0.05%	+11.46%	+3.47%	-0.02%	+38.88%
40	tvessels_remodelling	0	+0.18%	+0.24%	+0.24%	-0.05%	+0.24%	+0.1%	-0.17%	+0.18%	-0.02%	-0.03%	-0.02%	-0.03%	+0.12%	+0.13%	+0.02%	+0.26%	+0.08%	-0.08%	+0.16%
41	tS1P_and_EDG1_binding	0	+0.09%	+0.17%	+0.17%	-0.05%	+0.17%	+0.18%	-0.33%	+0.1%	+0.02%	-0.01%	+0.02%	-0.01%	+0.03%	+0.04%	-0.03%	+0.03%	+0.12%	-0.03%	+0.18%
42	tS1P_synthesis	0	+0.12%	+0.18%	+0.18%	-0.12%	+0.18%	+0.14%	-0.22%	+0.12%	+0.01%	-0.08%	+0.02%	-0.08%	+0.08%	+0.09%	-0.05%	+0.01%	+0.14%	-0.02%	+0.29%
43	tEDG_1_synthesis	0	+0.05%	+0.09%	+0.09%	-0.06%	+0.09%	+0.1%	-0.16%	+0.06%	+0.12%	-0.08%	+0.12%	-0.08%	+0.09%	+0.11%	-0.01%	+0.12%	+0.23%	+0.03%	+0.23%
44	tECM_enhancing_and_increasing_synthesis_and_rel...	0	-0.06%	-0.01%	-0.01%	-0.1%	-0.01%	+0.09%	-0.32%	-0.06%	+0.03%	-0.04%	+0.03%	-0.04%	+0.01%	+0.04%	-0.01%	+0.14%	+0.24%	+0.02%	+0.03%
45	tbinding_HB_EGF_with_EGFR	0	-0.0%	+0.03%	+0.03%	-0.04%	+0.03%	+0.05%	-0.18%	-0.0%	-0.01%	+0.0%	-0.01%	+0.0%	-0.0%	-0.02%	+0.03%	+0.02%	+0.15%	-0.04%	+0.1%
46	tTGFbeta1_TGFbetaRII_binding	0	+0.02%	+0.03%	+0.04%	-0.12%	+0.03%	+0.03%	-0.05%	+0.02%	+0.07%	+0.04%	+0.06%	+0.04%	+0.06%	+0.12%	-0.01%	+0.01%	+0.0%	-0.05%	+0.22%
47	tactivin_like_kinase_5_ALK_5_activation	0	+0.12%	+0.14%	+0.14%	-0.08%	+0.14%	+0.09%	-0.0%	+0.12%	+0.01%	+0.04%	+0.01%	+0.04%	+0.03%	-0.05%	+0.0%	+0.13%	+0.06%	-0.08%	+0.13%

Last button 

open the notepad with results in a text form.

6.7 quickSim module (qSim)

In Section 6 of the Holmes main window there is a qSim tab:



Button **SimStart** and **SimSettings** has the same function as already explained in this (6) chapter.

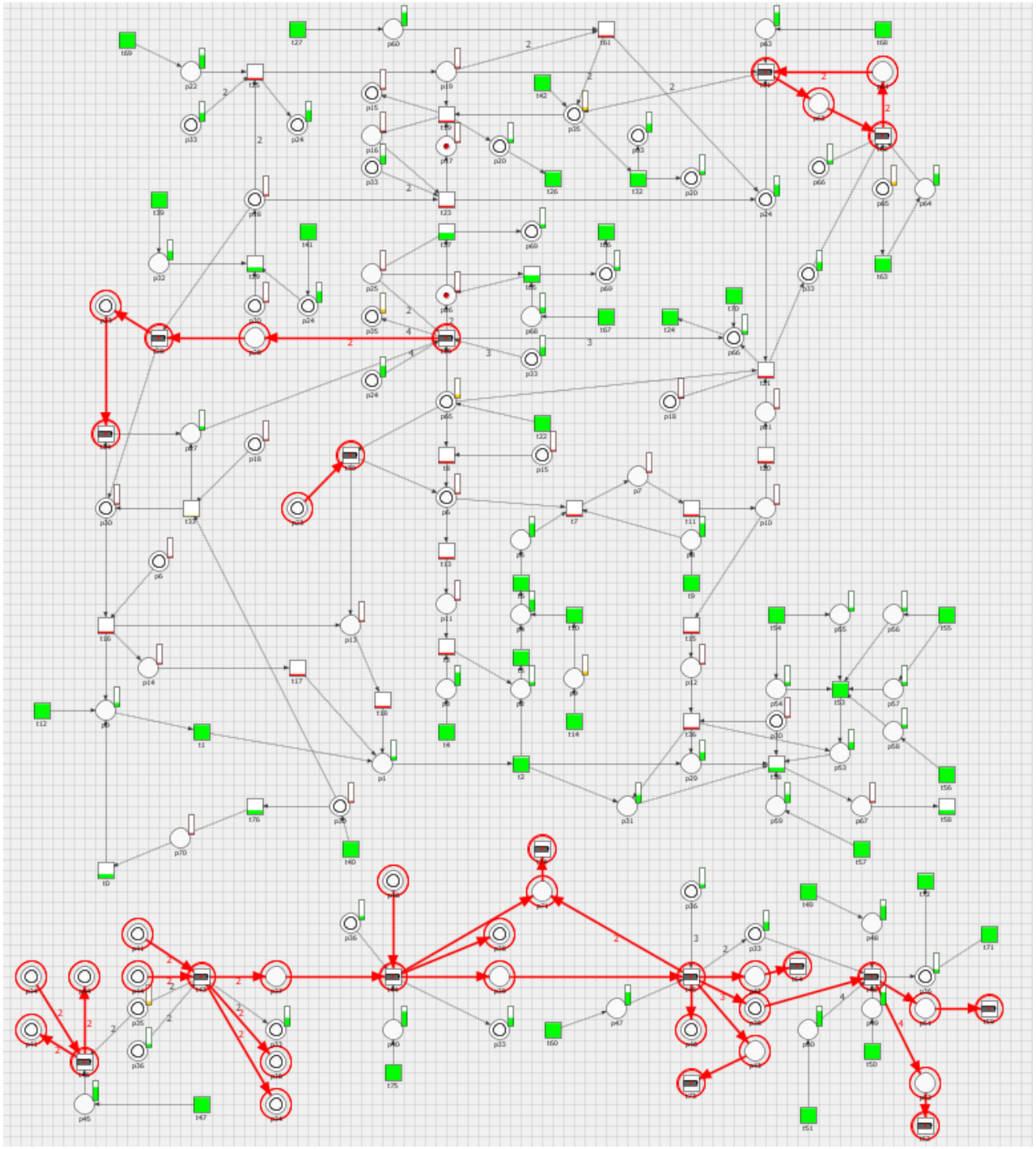
The main difference is that this tab shows the simplified knockout simulation results on the net structure.

Let us use an example. Let us assume we have a net, where nothing is disabled. However, some places do not receive tokens, nor some transition fire. It means the net is not fully live from the given initial state m0.

Data type simulated panel.

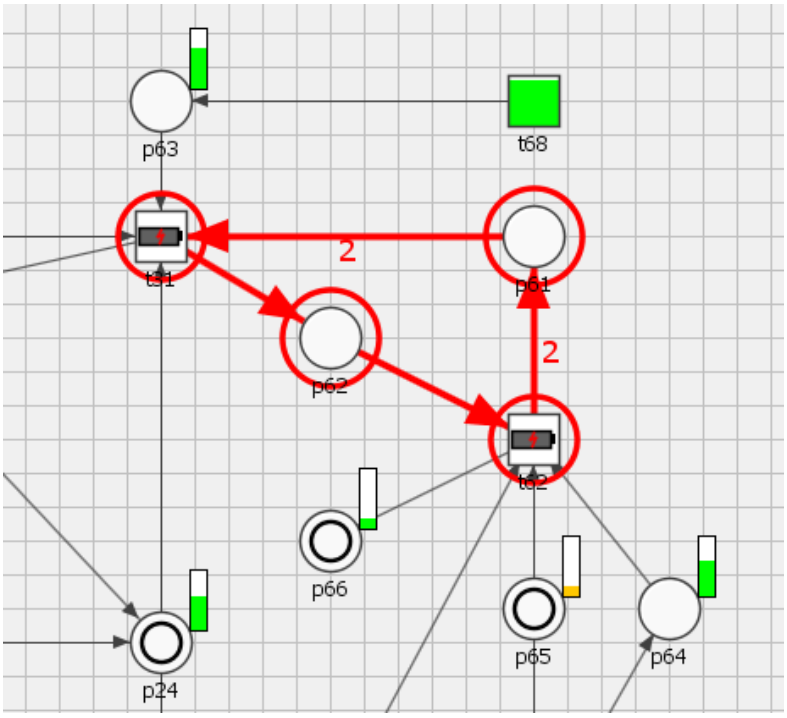
- **Transitions firing data** – transition data (should they be showed?)
- **Places tokens data** – places tokens data
- **Color arc** – should the knockout structure be drawn (see example below)
- **Repetitions** – for this tab maximal number is 20. Turning this option off will ignore the repetitions value in the **SimSettings** and only one simulation will be performed.

After choosing **SimStart**, the simulation results will be show on the net:



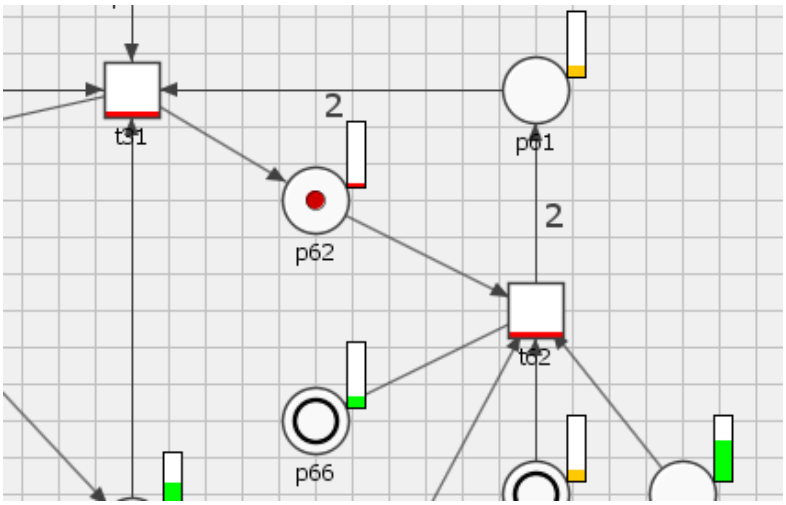
The more green the transition square is, the more times it fired. The same is true for every bar connected with every net place.

Regions marked red are dead – their transition did not fire and their places did not receive any tokens during the simulation.

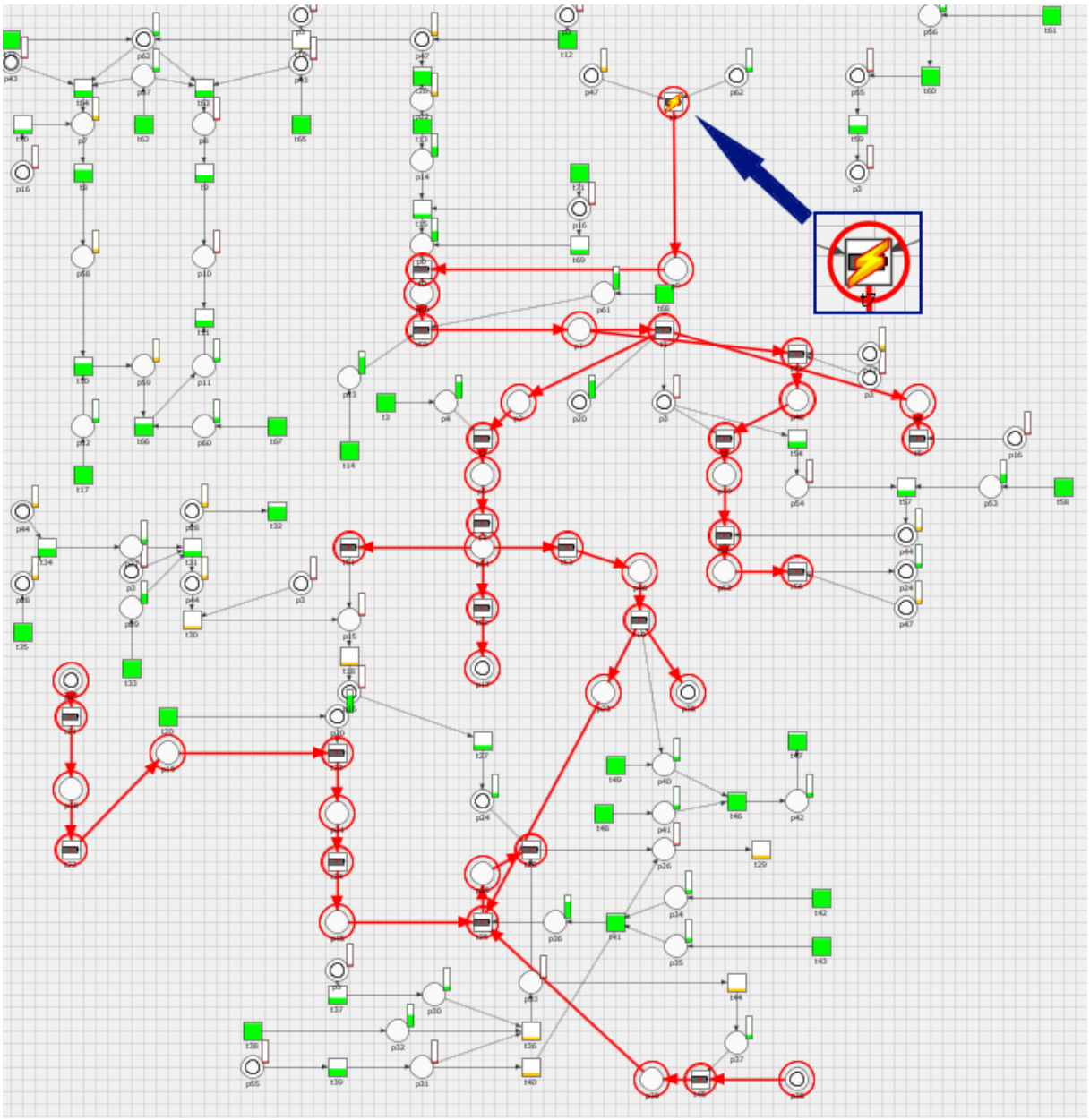


Places p66, p65 and p64 have some tokens accumulated, however t62 requires token from p62, where there is none. t31 requires 2 from p61, which should be provided by… t62. In the above example, one token in p62 OR 2 in p61 are necessary to initiate this cyclic reaction.

Results after adding a token in p62 are given below:



This module (**qSim**) can be very useful for observing the knockout behavior:



Transition t7 has been manually disabled (option from the context menu, already described in the manual), the impact on the net is given in the above picture.

7 Other analytical modules

Modules explained in this chapter requires t-invariants or are directly involved in their generation.

7.1 Net invariants

7.1.1 Theory

The following papers deal with the necessary theory:

(Murata, 1989)

(Law, Gwee i Chang, 2006)

Algorithm foundations:

Fourier-Motzkin Gauss elimination (Fourier, 1826) (Motzkin, 1936).

Algorithm implementation:

(Colom i Silva, 1991).

Other algorithms:

(Marinescu, Beaven i Stansifer, 1991)

(Law, Gwee i Chang, 2006)

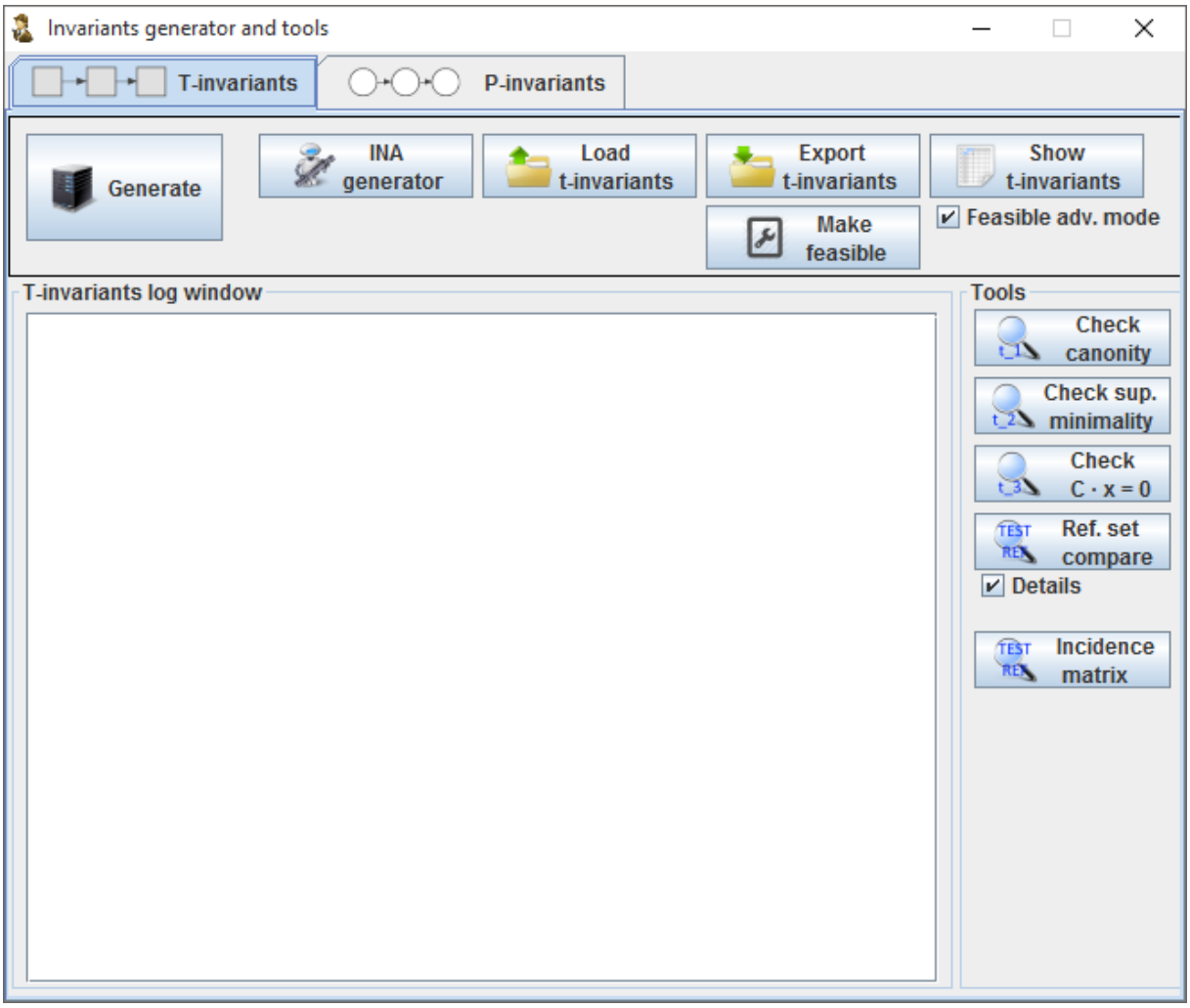
(Law, Gwee i Chang, 2007)

Feasible t-invariants:

(Sackmann, Heiner i Koch, 2006).

7.1.2 t-invariants generator

Main window is available from the **Net**->**Invariants generator…** or by shorcut **Ctrl+I**. Window looks like in the example below:



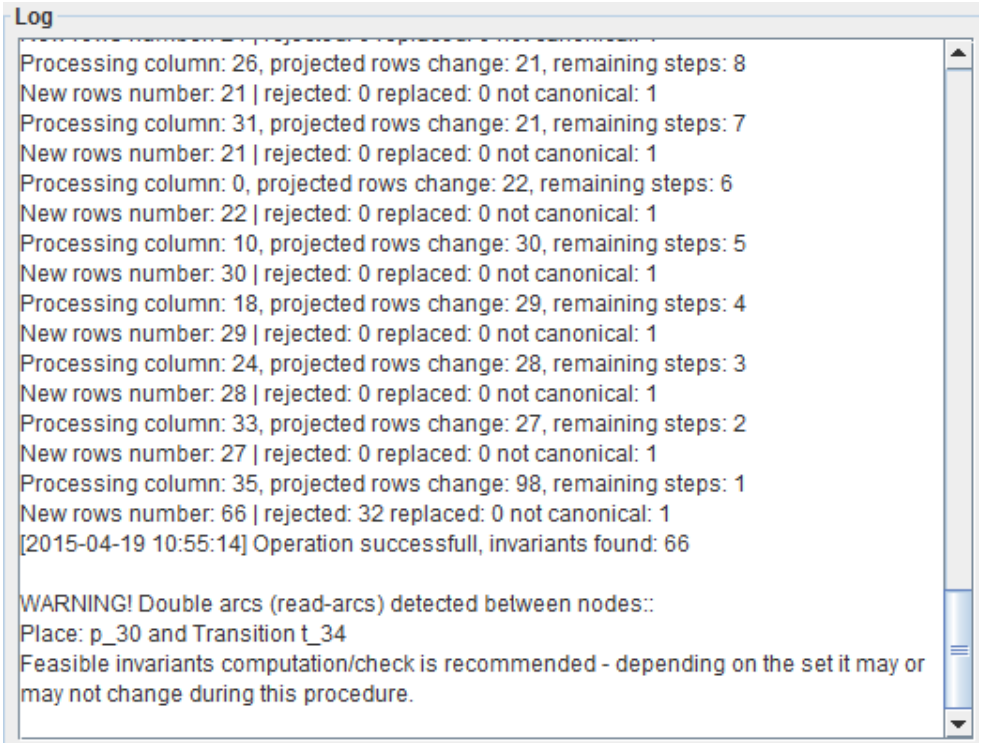
Two separate tabs are available for transitions- and places-invariants. For the t-invariants it looks like in the picture above.

Buttons:

- **Generate**- starts the invariant calculations, depending on the net structure it may take some time.
- **Ina Generator** - When INAwin32.exe program is available in the Holmes /Tools subdirectory, it can be used as the auxiliary generator (Integrated Net Analyzer (Starke, 1992)). When the INAwin32.exe console appears, user should press 'Y' key to accept pre-defined generation sequence. Window should close automatically after calculation, if it not happens and the cursor is available in the console window, 'N' key should be pressed at the very end.
- **Load Invariants** – Holmes can read the following invariants file formats::
  - **CSV** - *Comma Separated Values*, defaults file, .csv extension
  - **MonaLisa** - extension: .inv, from MonaLisa (Einloft, Ackermann, Nothen i Koch, 2013)
  - **Charlie** - extension: .inv, from Charlie (Heiner, Schwarick i Wegener, 2015)
  - **INA** - extension: .inv, from Integrated Net Analyzer already mentioned
- **Export Invariants** – as CSV or .inv (INA, Charlie)
- **Show t-invariant** – shows t-invariants in the notepad (CSV).

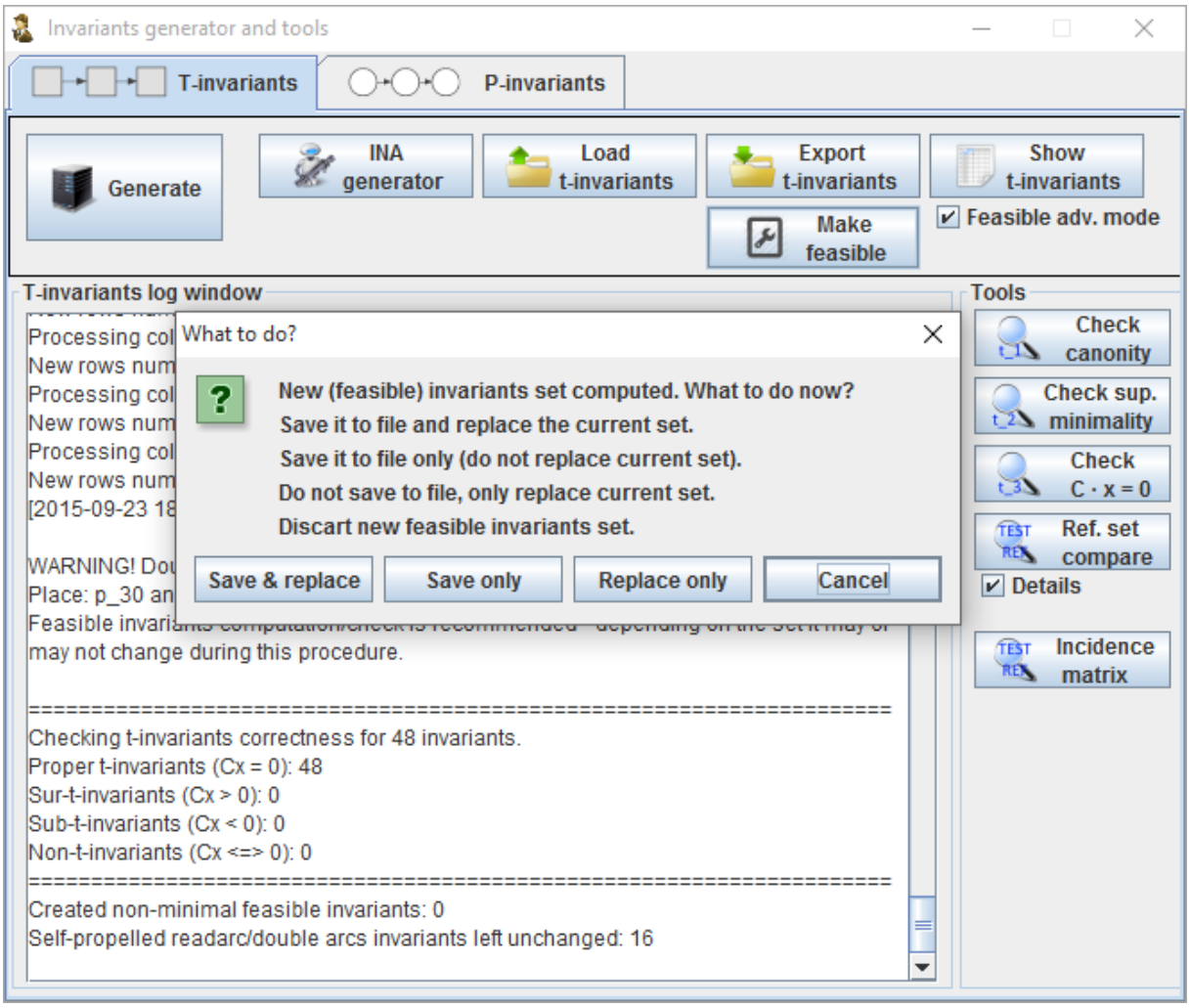
When generated by INA, Holmes will immediately ask if the invariants should be save to file. When native generator is used, invariants can be exported using **Export Invariants**, but they still be the part of the project file.

When read arcs / double arcs are detected, additional warning will be given:



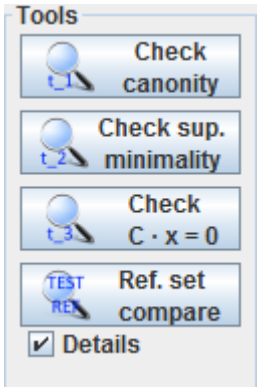
Button **Make Feasible** will make the t-invariants feasible according to the literature given in 7.1.1. It is advised to leave option **Feasible adv. mode** on (however it may need more time to calculate feasible set). The effects will be given in the log window, for example as report:

```
Created non-minimal feasible invariants: 0
Self-propelled readarc/double arcs invariants left unchanged: 22
```



- **Save & replace** – new set can be stored as a file, it will immediately replace the current t-invariants set
- **Save only** - save to file only.
- **Replace only** - replace the current set with the feasible one.
- **Cancel** - disregard computed feasible check (standard t-invariants set will remain)

Right panel buttons:



Check canonity – checks invariants for greatest common divisor = 1 (part of the native algorithm, may be useful if some unknown set of invariants has been just read from the file, not generated by Holmes).

Check sup. Minimality – test of support minimality (comment same as above)

Check Cx=0 – checks the equation C x = 0

Effect in log:

```
========================================================================
Checking canonicality for 66 invariants.
Non canonical invariants: 0
========================================================================

========================================================================
Checking support minimality for 66 invariants.
Non support-minimal invariants: 0
========================================================================

========================================================================
Checking invariants correctness for 66 invariants.
Proper invariants (Cx = 0): 66
Sur-invariants (Cx > 0): 0
Sub-invariants (Cx < 0): 0
Non-invariants (Cx <=> 0): 0
========================================================================
```

In (Starke, 1992) a few other types of vectors have been given: sur-invariant and sub-invariant, for which does not hold the $C \cdot x = 0$ equation, but the respective inequalities hold: $C^T \cdot x \geq 0$ (sur) and $C^T \cdot x \leq 0$ (sub).

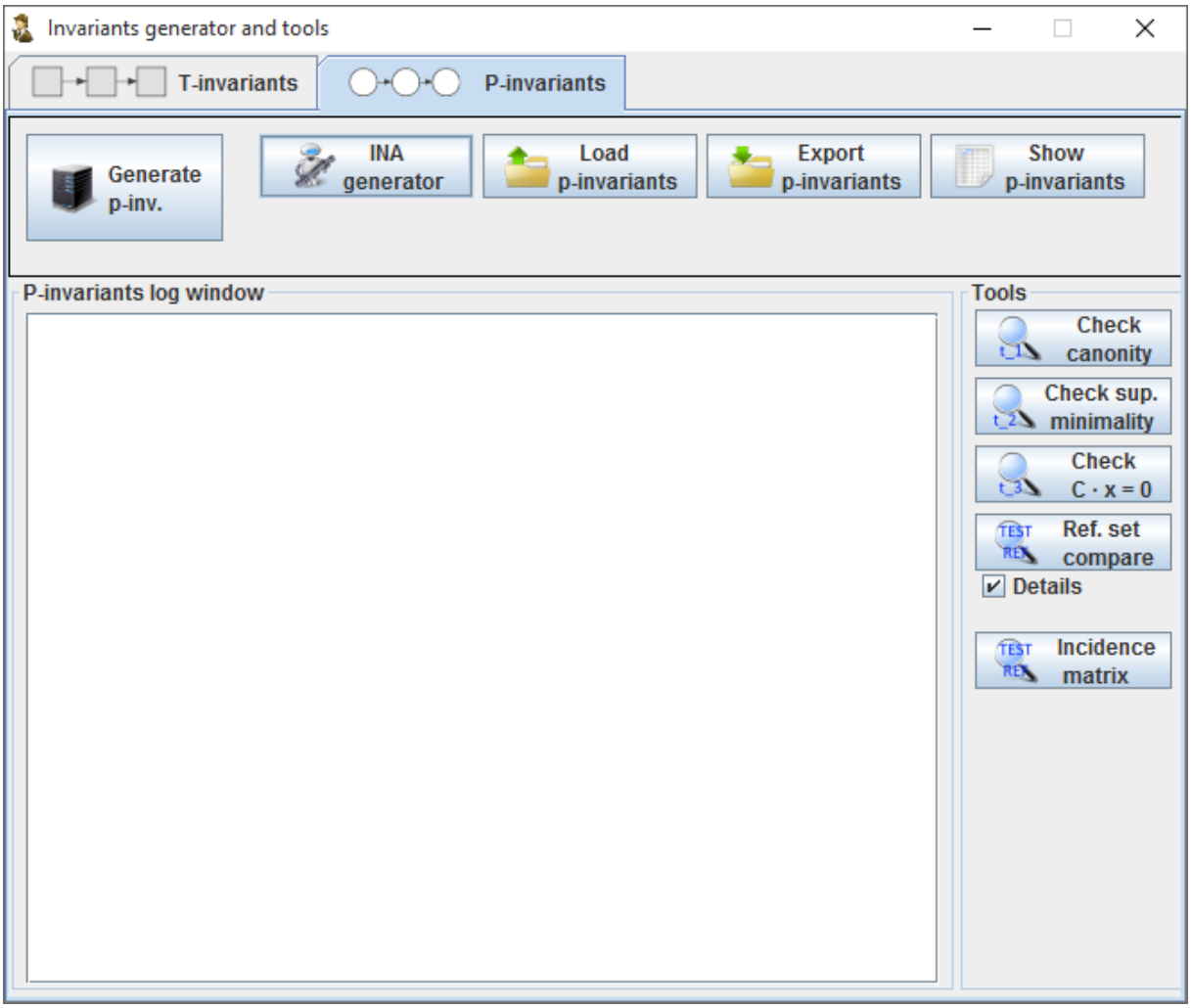
Last button **Ref. Set compare** can compare the t-invariants from a file with the already computed set stored currently in Holmes project memory. Effects will be given in a log:

```
========================================================================
Prev. computed set size:   66
Loaded (now) set size:    66
Common set size (load & ref): 44
Loaded invariants not in a computed set:  22
Computed invariants not in a loaded set:  22
Repetitions in common set: 0
Total repetitions in loaded:0

Inititating further tests for the loaded set of 66 invariants.
-> Non canonical invariants found : 0
-> Non support-minimal inv. found: 0
Proper invariants (Cx = 0): 44
Sur-invariants (Cx > 0): 0
Sun-invariants (Cx < 0): 22
Non-invariants (Cx <=> 0): 0
========================================================================
```

7.1.3 p-invariants tab

This tab has similar buttons like the previously described one, the only major difference is lack of feasible invariants sub-generator.



7.2 Cluster analysis

7.2.1 Theory

Clusters algorithms and distance metrics:

(Górecki, 2011).

Evaluation measures:

Calinski-Harabasz index:

(Caliński i Harabasz, 1974)

Mean Split Silhouette:

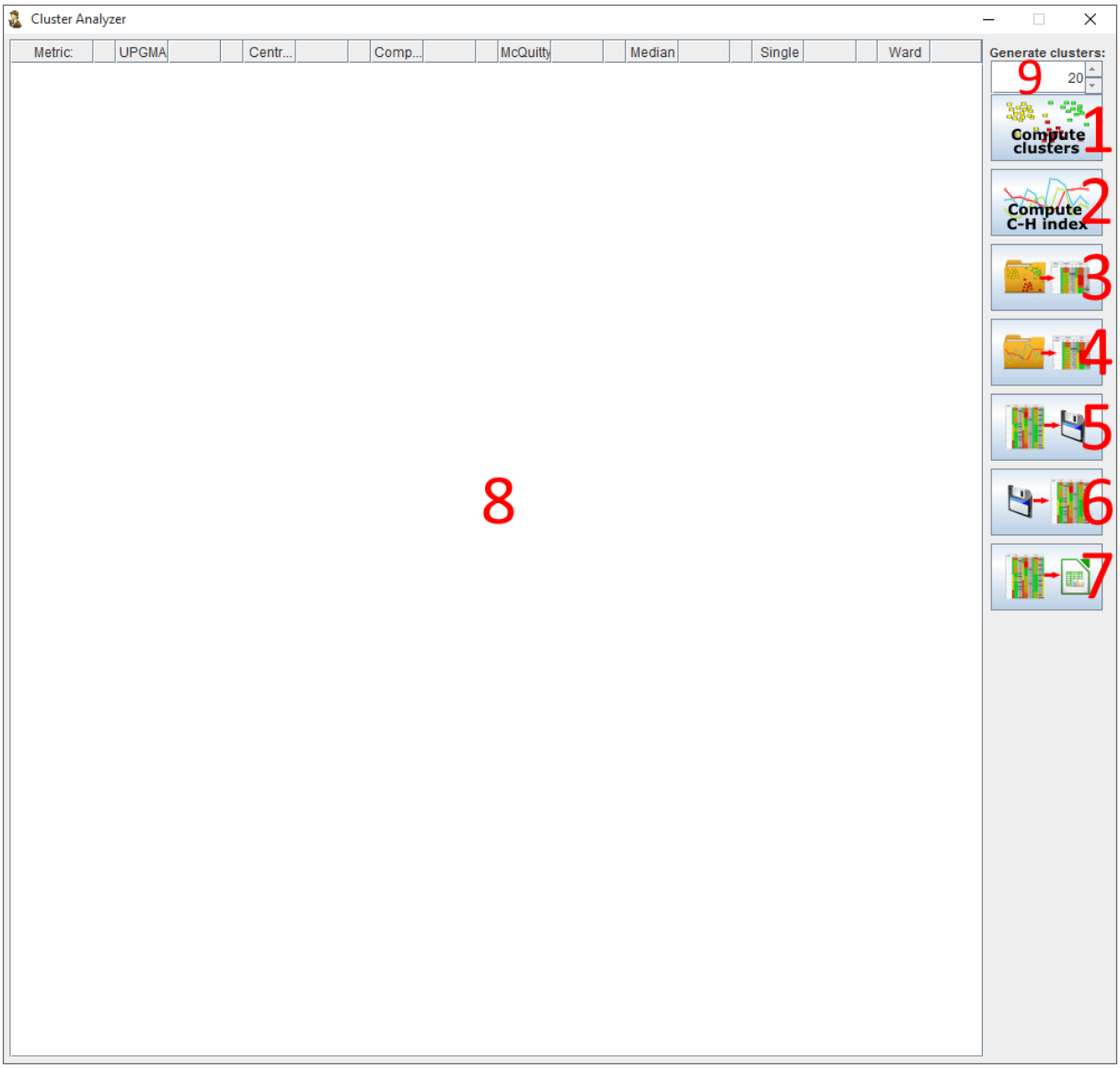
(Rousseeuw, 1987).

7.2.2 Holmes cluster module

All clustering procedures are performed using R Language environment, which should be installed. The communication between Holmes and R is automatic, assuming the R files path is given correctly (see 8.1 chapter, R path button). As for the R language, the required steps are:

- install R, at least version 3.1 (http://www.r-project.org/)
- in R the following libraries must be installed:
  - amap
  - clusters
  - fpc
- optionally a BioConductor tools can be installed in R libraries: http://www.bioconductor.org/

Cluster analysis window is available from menu **Analysis** -> **Cluster analysis** (**Ctrl+C**) or by C1 button from Holmes toolbar (Sekcion 2). The window looks similar to:



In part **8** there will be a table with the results, when available. The number of clusters to be computed (from 2 to this value) must be provided in provided field **9** before the computations start.

In the current version of Holmes one more button is avaible below the 7th one on the picture. Using the window this button activates, the user can select different combinations of clustering algorithms and distance metrics to be calculated. Default scenario assumes calculating all 56 combinations of 7 algorithms and 8 distance metrics. This will change the time button 1 and 2 require to finish the calculations using R environment in the background.

Buttons:

1 **Compute clusters** - the user should create a new folder for the calculations (option will be given). By default all 56 combinations of clustering algorithm and distance metric will be computed, each one for cluster number from 2 to the given in field **9**.

2 **Compute C-H index** - similar as above, but it computes the auxiliary evaluation metric based on Caliński-Harabasz. The default evaluation metric is MSS, calculated using button **1**.

klastrów (**ATTENTION:** when t-invariants number exceed thousands, this may take a VERY long time to finish, assuming the user has at least 8GB of RAM memory.)

3 **Load Clusters** - when button 1 calculations are finished (the log will inform about this), the user must click this button and select the folder assigned when using button 1.

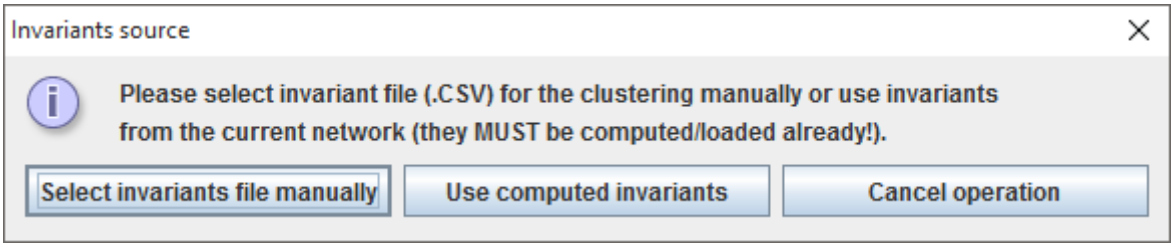
4 **Load C-H Results** - as above for the C-H metrics and folder assigned by using button 2.

5 **Save table to file** - when button 3 (and 4) has been used to draw the table, this button can save it to a single file for future faster usage.

6 **Load table from file** - loads table save by button 5.

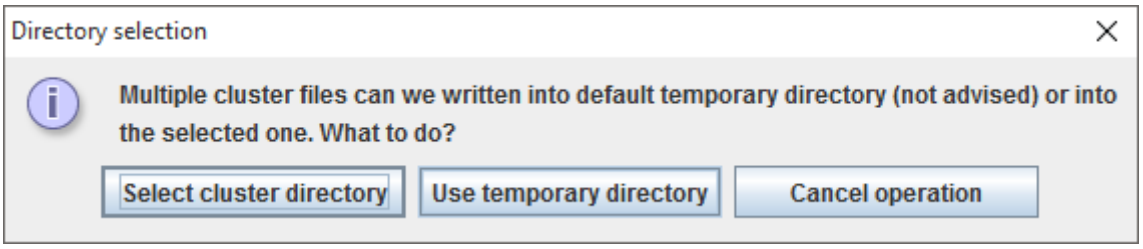
7 **Export to Excel** - export table to Excel file (2003).

Let us assume we have a net with computed t-invariants. Clicking button 1 and choosing folder for calculations will initiate the computations. Before it happens, the following message will be shown:



Advised option is the second one: **Use computed invariants**.

Second question concerns the folder for calculations.

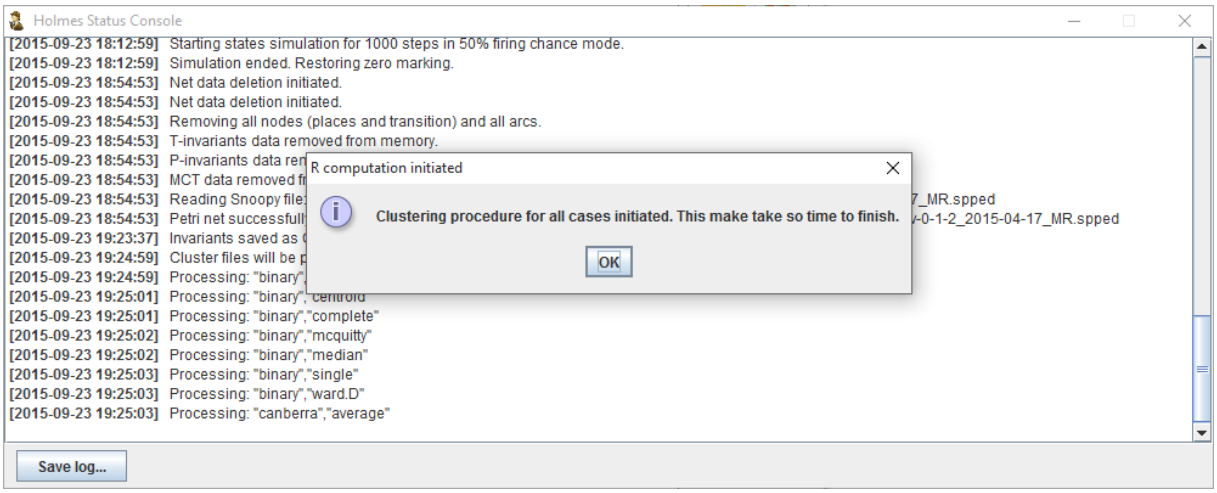


ATTENTION! IT IS STRONGLY ADIVESED TO USE 'SELECT CLUSTER DIRECTORY' OPTION.

When the computations started, the user can use Button 2 and initiate C-H calculations in the second thread, assuming having the multi-core CPU in a computer.

ATTENTION: folder for C-H calculations (button 2) should be different than the one selected for MSS cluster computations (button 1).

When the calculations are in progress, informations in log will be given:



If not selected otherwise, the last to process will be always correlation (Pearson) metrics with ward clustering algorithm:

```
[2015-04-19 13:49:03]   Processing: "correlation","ward"
[2015-04-19 13:49:04]   All clusterings has been computed.
```

Then button 3 can be used to read the table from the already selected folder (button 1 procedure), and button 4 to read C-H metrics (button 2 procedure). The final table can look like in the example picture below:

Cluster Analyzer

Metric:		UPGMA			Centr...			Comp...			McQui...			Median			Single			Ward	
Correlation	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H
2	1	0.5696	5.9	1	0.5696	5.9	0	0.6701	45.4	0	0.5709	8.2	1	0.5696	5.9	1	0.5696	5.9	0	0.6701	45.4
3	2	0.5549	6.1	2	0.5549	6.1	0	0.7124	42.4	2	0.5549	6.1	2	0.5549	6.1	2	0.5549	6.1	0	0.6054	41.5
4	2	0.7005	31.5	2	0.7005	31.5	1	0.7108	36.7	2	0.7005	31.5	2	0.7005	31.5	3	0.5398	6.2	0	0.6528	60.5
5	3	0.6996	36.8	3	0.6996	36.8	3	0.6996	36.8	3	0.6996	36.8	3	0.6996	36.8	3	0.6996	36.8	0	0.5342	53.7
6	3	0.7023	178.6	3	0.7023	178.6	3	0.7023	178.6	3	0.7023	178.6	3	0.7023	178.6	3	0.6914	152.8	0	0.6441	52
7	3	0.6436	181.3	4	0.5809	171.2	3	0.6436	181.3	3	0.6436	181.3	3	0.6436	181.3	4	0.6172	134.6	1	0.6452	80.8
8	5	0.5674	159.5	5	0.5674	159.5	3	0.6036	328.6	5	0.5674	159.5	5	0.5674	159.5	5	0.5674	159.5	3	0.6491	305.2
9	5	0.5622	312.4	5	0.5622	312.4	3	0.6336	462.4	5	0.5574	265.9	5	0.5574	265.9	5	0.5574	265.9	3	0.6336	462.4
10	5	0.6153	480.8	5	0.6153	480.8	3	0.5777	531.9	5	0.5265	272.2	6	0.5011	254.9	6	0.5011	254.9	3	0.6298	470.7
Pearson	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H
2	1	0.6232	7.5	1	0.6232	7.5	0	0.6182	11.7	1	0.6232	7.5	1	0.6232	7.5	1	0.6232	7.5	0	0.6832	40.2
3	2	0.6066	7.8	2	0.6066	7.8	1	0.6126	8.9	2	0.6066	7.8	2	0.6066	7.8	2	0.6066	7.8	0	0.596	32
4	3	0.6043	8.4	3	0.6043	8.4	3	0.6043	8.4	3	0.6043	8.4	3	0.6043	8.4	3	0.6043	8.4	0	0.6502	50
5	3	0.7285	47.1	3	0.7285	47.1	3	0.7285	47.1	3	0.7285	47.1	3	0.7285	47.1	3	0.7285	47.1	0	0.5082	41.8
6	3	0.7057	221	3	0.7057	221	3	0.7057	221	3	0.7057	221	3	0.7057	221	3	0.6944	186.6	1	0.5114	59.5
7	3	0.6661	237.9	4	0.6057	218.3	3	0.6661	237.9	3	0.6661	237.9	3	0.6661	237.9	4	0.638	169.5	3	0.5164	152.5
8	5	0.599	213.8	5	0.599	213.8	3	0.6015	385.3	5	0.599	213.8	5	0.599	213.8	5	0.599	213.8	3	0.6353	308.6
9	5	0.5699	378.3	5	0.5699	378.3	3	0.538	386.7	5	0.561	325.8	5	0.561	325.8	5	0.561	325.8	3	0.6347	496.3
10	5	0.6223	534.5	5	0.6223	534.5	3	0.5723	553.8	5	0.5279	343.5	6	0.4967	314.4	6	0.4967	314.4	3	0.5872	588.3
Binary	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H
2	1	0.5356	6.2	1	0.5356	6.2	0	0.5492	11.9	1	0.5356	6.2	1	0.5356	6.2	1	0.5356	6.2	0	0.2966	9.9
3	2	0.5372	6.8	2	0.5372	6.8	1	0.5432	8.6	2	0.5332	6.7	2	0.5332	6.7	2	0.5332	6.7	0	0.3403	14.4
4	3	0.5423	7.7	3	0.5423	7.7	3	0.5423	7.7	3	0.5423	7.7	3	0.5423	7.7	3	0.5423	7.7	0	0.3992	17.2
5	4	0.5202	8.8	4	0.5202	8.8	3	0.4288	11.8	3	0.4288	11.8	4	0.5202	8.8	4	0.5202	8.8	0	0.3482	16.8
6	4	0.4673	11.2	4	0.4673	11.2	3	0.4574	18.1	3	0.4541	15	4	0.4673	11.2	4	0.4673	11.2	0	0.326	15.3
7	4	0.3823	14.4	6	0.439	9.7	3	0.4422	19.7	4	0.4042	14.8	6	0.439	9.7	6	0.439	9.7	0	0.3553	16.3
8	4	0.4269	20.8	6	0.3874	13.3	4	0.4269	20.8	4	0.4269	20.8	6	0.3874	13.3	6	0.3874	13.3	1	0.3564	19.6
9	6	0.4158	19.5	6	0.4158	19.5	4	0.3954	27.6	6	0.4158	19.5	6	0.4158	19.5	6	0.4158	19.5	3	0.3571	26.5
10	6	0.3938	18.8	6	0.3938	18.8	6	0.3844	27.1	6	0.3844	27.1	6	0.3938	18.8	6	0.3938	18.8	4	0.375	33
Canberra	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H
2	1	0.2897	2.7	1	0.2897	2.7	0	0.2965	4.4	1	0.2897	2.7	1	0.2897	2.7	1	0.2897	2.7	0	0.2166	13.6
3	2	0.2902	2.8	2	0.2902	2.8	1	0.2906	3.2	2	0.2861	2.7	2	0.2861	2.7	2	0.2861	2.7	0	0.2369	11.3
4	3	0.2889	2.9	3	0.2889	2.9	3	0.2889	2.9	3	0.2889	2.9	3	0.2889	2.9	3	0.2889	2.9	0	0.2434	11.4
5	4	0.2627	2.9	4	0.2627	2.9	3	0.1905	4	3	0.1905	4	4	0.2627	2.9	4	0.2627	2.9	0	0.2542	10.9
6	4	0.2082	3.3	5	0.1848	2.9	3	0.2662	8.1	3	0.1934	4.3	5	0.1848	2.9	5	0.1849	2.9	0	0.2944	12
7	4	0.2302	6.8	6	0.1823	2.8	3	0.2804	10.4	3	0.2662	8.3	6	0.1823	2.8	6	0.1823	2.8	0	0.2816	11.4
8	4	0.2719	9.3	7	0.1002	2.7	4	0.2765	9.7	3	0.2943	10.6	6	0.2132	6.1	6	0.1977	4.6	0	0.2924	11.9
9	6	0.2633	8.4	8	0.05	2.7	4	0.2926	10.1	4	0.2926	10.1	6	0.1597	5.9	6	0.2208	5.5	0	0.2807	11.3
10	6	0.2842	9.4	8	0.0256	3.6	4	0.3065	12.1	6	0.2842	9.4	6	0.2471	8.5	6	0.1694	5.4	0	0.2794	11
Euclidean	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H
2	0	0.7271	65.6	1	0.5653	4.5	0	0.7197	74.2	0	0.6617	63	1	0.5653	4.5	1	0.7019	17.2	0	0.7271	65.6
3	1	0.6563	46.5	1	0.667	53.3	1	0.667	53.3	1	0.6169	44.6	2	0.5574	12.6	2	0.5574	12.6	0	0.4618	56.6
4	2	0.6311	39	2	0.6395	46.4	2	0.6395	46.4	2	0.6076	44	3	0.5719	16.9	3	0.5719	16.9	0	0.5263	59.2
5	3	0.6066	32.1	3	0.6154	40.7	3	0.6154	40.7	3	0.5796	42.9	4	0.5823	21.9	3	0.6066	32.1	1	0.5149	63.1
6	3	0.5038	42.8	4	0.5458	35	3	0.4612	65.4	3	0.5038	42.8	5	0.5856	26.9	5	0.5856	26.9	2	0.5211	66.7
7	5	0.4971	38.4	6	0.5442	31	4	0.45	62.6	5	0.4971	38.4	6	0.5442	31	6	0.5442	31	2	0.5123	74.1
8	6	0.4279	35.5	6	0.4279	35.5	6	0.4483	61.2	6	0.4279	35.5	6	0.4279	35.5	6	0.3784	35.3	2	0.3588	68.4
9	6	0.3801	44.3	8	0.3725	31.1	6	0.4217	61.9	6	0.3801	44.3	6	0.3801	44.3	6	0.3801	44.3	2	0.3838	67
10	6	0.502	110.7	8	0.3432	39.8	6	0.502	110.7	6	0.502	110.7	8	0.3432	39.8	8	0.3432	39.8	3	0.3797	88.6
Manhattan	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H	0:	MSS	C-H
2	0	0.7461	100.8	0	0.7461	100.8	0	0.7461	100.8	0	0.6139	63.5	0	0.7461	100.8	1	0.7098	18.5	0	0.7461	100.8
3	1	0.6967	62.3	1	0.6967	62.3	0	0.6727	65	0	0.5748	68.6	1	0.6967	62.3	2	0.6919	22.3	0	0.5262	90.6
4	2	0.6632	49.1	2	0.6632	49.1	0	0.521	88.2	2	0.5528	51.9	2	0.6632	49.1	2	0.6646	37.6	0	0.5269	75.9
5	3	0.5254	44.3	3	0.5254	44.3	2	0.5121	82.1	2	0.5026	59	3	0.5764	38.7	4	0.6262	28.8	0	0.5222	91.1
6	4	0.4772	27.7	4	0.4772	27.7	2	0.5161	89.7	3	0.4881	60.3	5	0.5716	31.0	5	0.479	27.3	0	0.3573	78.7

Generate clusters: 10

Compute clusters

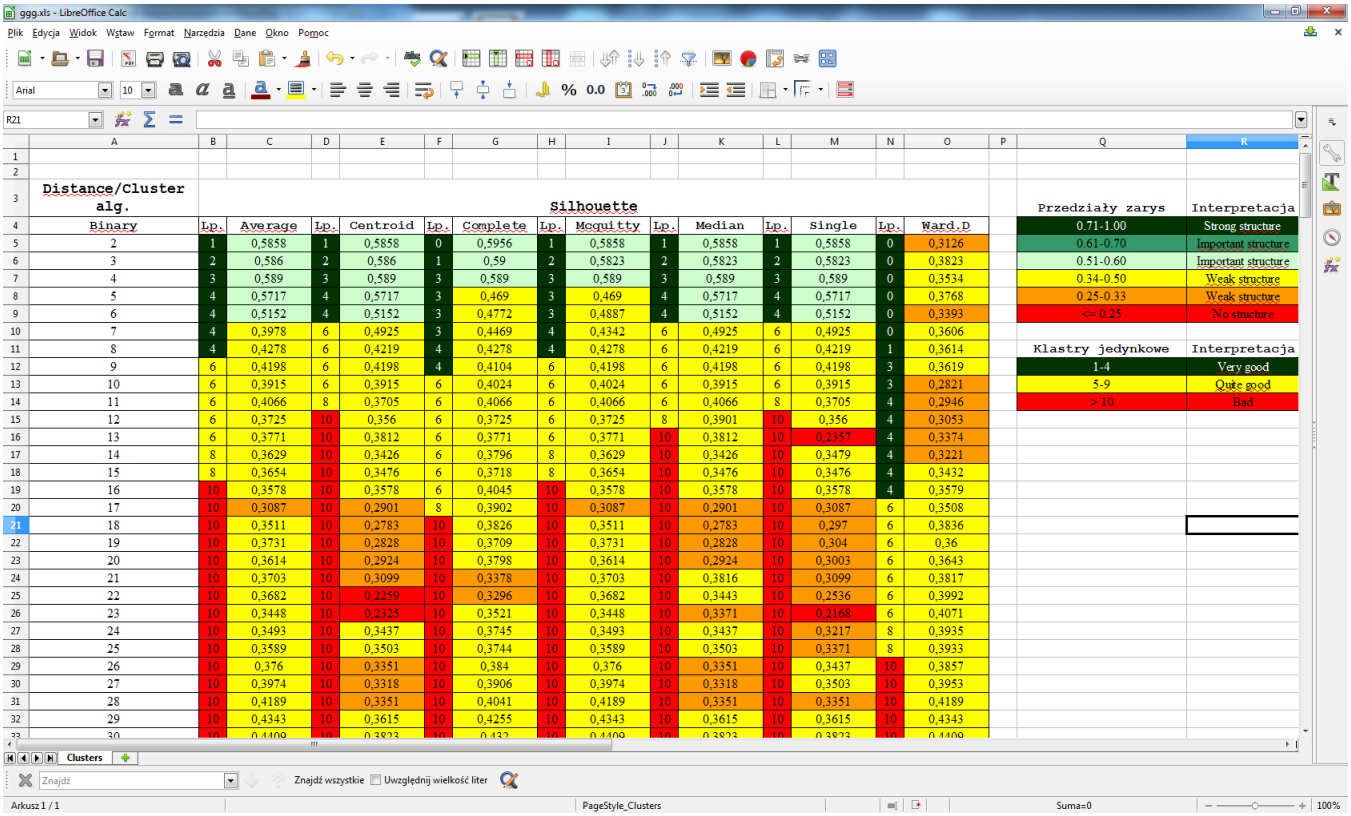
Compute C-H index

Not all combinations of algorithm and metrics are visible in the picture (they can be scrolled in Holmes). The table is divided by blocks, each one (rows) consisting of chosen distance metric (e.g. from the top: Correlated Person, (uncorrelated) Person, Binary, Canberra, etc. used for the selected number of clusters (from 2 to value given in top right part of the window). Columns in general represent clustering algorithms, each one having always 3 columns, named: '0:', 'MSS' and 'C-H'. First three-columns block is for UPGMA (***U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic means* (Sokal i Michener, 1958)) clustering algorithm, second block for Centroid, then Complete, McQuitty, Median, Single and Ward algorithm.

As for the colors, they are arbitrarily assigned, in general: the more green the better the results are. '0:' column shows the number of single-invariant clusters (in most cases: the lowest, the better). Second column 'MSS:' shows the Mean Split Silhuette value for a given clustering (total, will be described later). Finally 'C-H' column show the Calinski-Harabasz value if computed. For this column 'peaks' of values are marked with darker gray color.

As mentioned before, when using 1-4 button to generate the table, it is advisable to store in in separate file with button 5 (and read from such a file with button 6).

Button 7 export the table to Excel file (.xls):



ggg.xls - LibreOffice Calc

Plik Edycja Widok Wstaw Format Narzędzia Dane Okno Pomoc

Distance/Cluster alg.	Silhouette														Przedziały zarys	Interpretacja
Binary	Lp.	Average	Lp.	Centroid	Lp.	Complete	Lp.	Mcquitty	Lp.	Median	Lp.	Single	Lp.	Ward.D	0.71-1.00	Strong structure
2	1	0,5858	1	0,5858	0	0,5956	1	0,5858	1	0,5858	1	0,5858	0	0,3126	0.61-0.70	Important structure
3	2	0,586	2	0,586	1	0,59	2	0,5823	2	0,5823	2	0,5823	0	0,3823	0.51-0.60	Important structure
4	3	0,589	3	0,589	3	0,589	3	0,589	3	0,589	3	0,589	0	0,3534	0.34-0.50	Weak structure
5	4	0,5717	4	0,5717	3	0,469	3	0,469	4	0,5717	4	0,5717	0	0,3768	0.25-0.33	Weak structure
6	4	0,5152	4	0,5152	3	0,4772	3	0,4887	4	0,5152	4	0,5152	0	0,3393	<= 0.25	No structure
7	4	0,3978	6	0,4925	3	0,4469	4	0,4342	6	0,4925	6	0,4925	0	0,3606		
8	4	0,4278	6	0,4219	4	0,4278	4	0,4278	6	0,4219	6	0,4219	1	0,3614	Klastry jedynkowe	Interpretacja
9	6	0,4198	6	0,4198	4	0,4104	6	0,4198	6	0,4198	6	0,4198	3	0,3619	1-4	Very good
10	6	0,3915	6	0,3915	6	0,4024	6	0,4024	6	0,3915	6	0,3915	3	0,2821	5-9	Quite good
11	6	0,4066	8	0,3705	6	0,4066	6	0,4066	6	0,4066	8	0,3705	4	0,2946	> 10	Bad
12	6	0,3725	10	0,356	6	0,3725	6	0,3725	8	0,3901	10	0,356	4	0,3053		
13	6	0,3771	10	0,3812	6	0,3771	6	0,3771	10	0,3812	10	0,2357	4	0,3374		
14	8	0,3629	10	0,3426	6	0,3796	8	0,3629	10	0,3426	10	0,3479	4	0,3221		
15	8	0,3654	10	0,3476	6	0,3718	8	0,3654	10	0,3476	10	0,3476	4	0,3432		
16	10	0,3578	10	0,3578	6	0,4045	10	0,3578	10	0,3578	10	0,3578	4	0,3579		
17	10	0,3087	10	0,2901	8	0,3902	10	0,3087	10	0,2901	10	0,3087	6	0,3508		
18	10	0,3511	10	0,2783	10	0,3826	10	0,3511	10	0,2783	10	0,297	6	0,3836		
19	10	0,3731	10	0,2828	10	0,3709	10	0,3731	10	0,2828	10	0,304	6	0,36		
20	10	0,3614	10	0,2924	10	0,3798	10	0,3614	10	0,2924	10	0,3003	6	0,3643		
21	10	0,3703	10	0,3099	10	0,3378	10	0,3703	10	0,3816	10	0,3099	6	0,3817		
22	10	0,3682	10	0,2259	10	0,3296	10	0,3682	10	0,3443	10	0,2536	6	0,3992		
23	10	0,3448	10	0,2325	10	0,3521	10	0,3448	10	0,3371	10	0,2168	6	0,4071		
24	10	0,3493	10	0,3437	10	0,3745	10	0,3493	10	0,3437	10	0,3217	8	0,3935		
25	10	0,3589	10	0,3503	10	0,3744	10	0,3589	10	0,3503	10	0,3371	8	0,3933		
26	10	0,376	10	0,3351	10	0,384	10	0,376	10	0,3351	10	0,3437	10	0,3857		
27	10	0,3974	10	0,3318	10	0,3906	10	0,3974	10	0,3318	10	0,3503	10	0,3953		
28	10	0,4189	10	0,3351	10	0,4041	10	0,4189	10	0,3351	10	0,3351	10	0,4189		
29	10	0,4343	10	0,3615	10	0,4255	10	0,4343	10	0,3615	10	0,3615	10	0,4343		

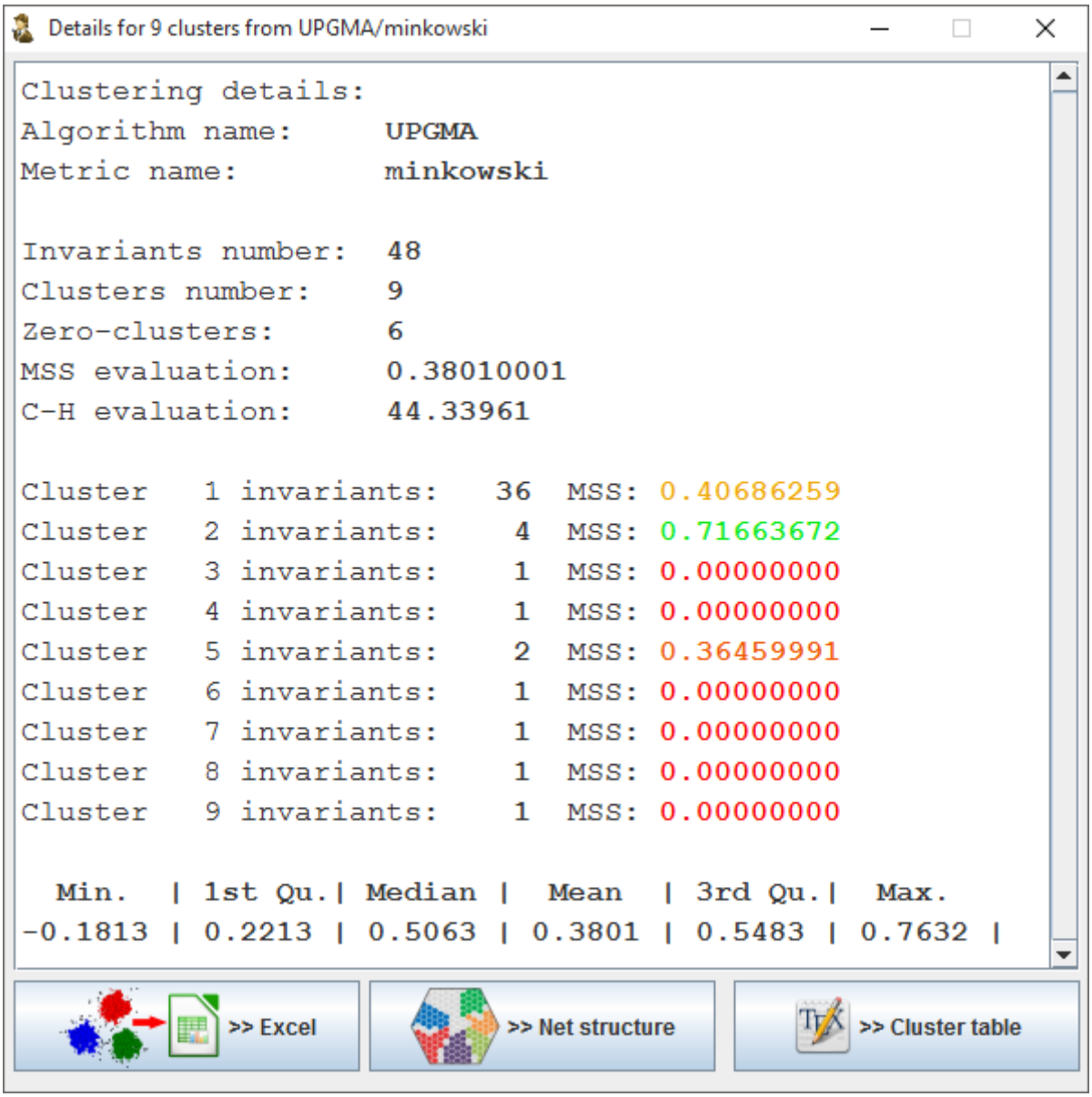
Clusters

Znajdź wszystkie Uwzględnij wielkość liter

Arkusz 1 / 1 PageStyle_Clusters Suma=0 100%

ATTENTION: in order to create a file, a folder with MSS values must be selected – the one assigned when running calculation by button 1 (the same folder is chosen with button 3 as already explained).

7.2.3 Details about clustering



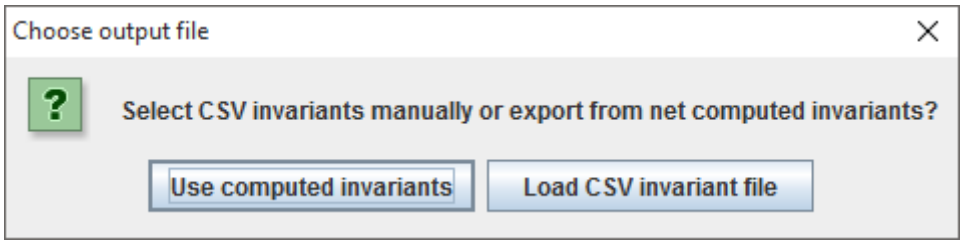
Clicking any cell corresponding to a given combination of metric (row) and clustering algorithm (column) will show a window with detailing information. MSS values are presented evaluating each cluster within the clustering.

Three possible button have the following functions (from left to right):

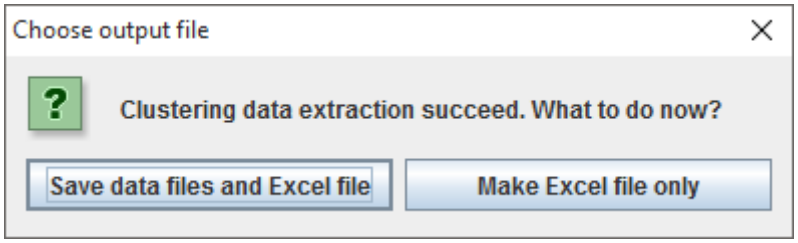
- Export this specific clustering into Excel file.
- Show clustering data on the net structure.
- Export cluster table into Tex file

Exporting data to Excel will require using R script one more time, this may take a few second or more, depending on the number of t-invariants.

After clicking that button, a question will appear:



After choosing one option (computed invariants in Holmes memory is a recommended one), one more question will show up:



First option will write additional PDF files with the Excel document about the clustering:

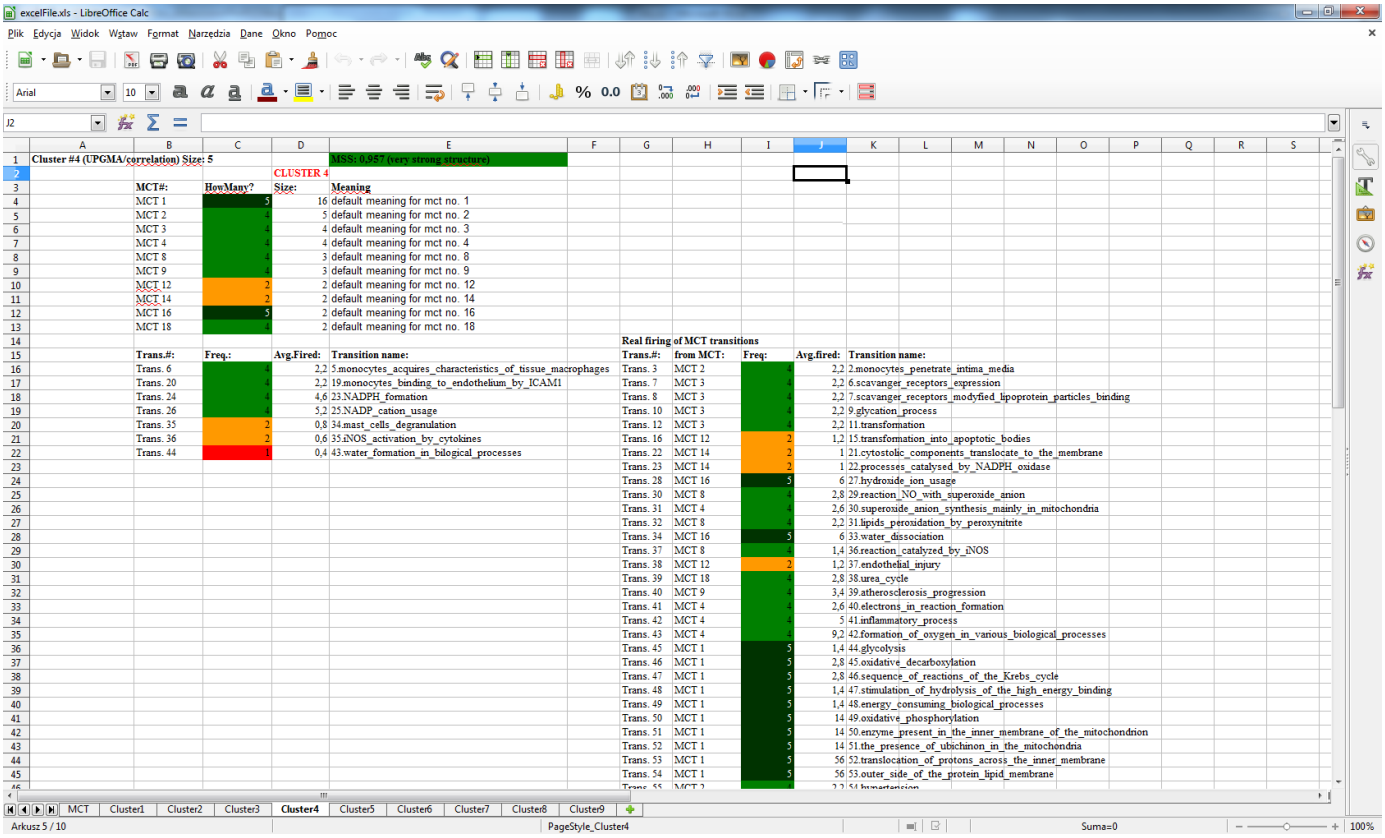
- average_correlation_clusters_ext_9.pdf
- average_correlation_dendrogram_ext_9.pdf
- cluster.csv.analysed.txt
- excelFile.xls (*file name assigned by the user in a separate window*)

First file contains graphical MSS representation for every cluster and every (its) invariant. Second file contains the clusters dendrogram. Fourth file is divided into 4 blocks:

- full names of transition in every t-invariant
- t-invariant CSV block
- t-invariants with MCT sets (if exists within)
- MCT sets

Fourth (Excel) file contains detailed data about each cluster. First tab contains MCT sets data, next tabs – detailed data about every cluster composition.

An example of a cluster tab within the file:



excelFile.xls - LibreOffice Calc

Plik Edycja Widok Wstaw Format Narzędzia Dane Okno Pomoc

	A	B	C	D	E
1	Cluster #4 (UPGMA/correlation) Size: 5				MSS: 0.957 (very strong structure)
2				CLUSTER 4	
3		MCT#:	HowMany?	Size:	Meaning
4		MCT 1	5	16	default meaning for mct no. 1
5		MCT 2	4	5	default meaning for mct no. 2
6		MCT 3	4	4	default meaning for mct no. 3
7		MCT 4	4	4	default meaning for mct no. 4
8		MCT 8	4	3	default meaning for mct no. 8
9		MCT 9	4	3	default meaning for mct no. 9
10		MCT 12	2	2	default meaning for mct no. 12
11		MCT 14	2	2	default meaning for mct no. 14
12		MCT 16	5	2	default meaning for mct no. 16
13		MCT 18	4	2	default meaning for mct no. 18
14					
15		Trans.#:	Freq.:	Avg.Fired:	Transition name:
16		Trans. 6	4	2,2	5.monocytes_acquires_characteristics_of_tissue_macrophages
17		Trans. 20	4	2,2	19.monocytes_binding_to_endothelium_by_ICAM1
18		Trans. 24	4	4,6	23.NADPH_formation
19		Trans. 26	4	5,2	25.NADP_cation_usage
20		Trans. 35	2	0,8	34.mast_cells_degranulation
21		Trans. 36	2	0,6	35.iNOS_activation_by_cytokines
22		Trans. 44	1	0,4	43.water_fomation_in_bilogical_processes

Real firing of MCT transitions

	G	H	I	J	K
14					
15	Trans.#:	from MCT:	Freq:	Avg.fired:	Transition name:
16	Trans. 3	MCT 2	4	2,2	2.monocytes_penetrate_intima_media
17	Trans. 7	MCT 3	4	2,2	6.scavanger_receptors_expression
18	Trans. 8	MCT 3	4	2,2	7.scavanger_receptors_modyfied_lipoprotein_particles_binding
19	Trans. 10	MCT 3	4	2,2	9.glycation_process
20	Trans. 12	MCT 3	4	2,2	11.transformation
21	Trans. 16	MCT 12	2	1,2	15.transformation_into_apoptotic_bodies
22	Trans. 22	MCT 14	2	1	21.cytostolic_components_translocate_to_the_membrane
23	Trans. 23	MCT 14	2	1	22.processes_catalysed_by_NADPH_oxidase
24	Trans. 28	MCT 16	5	6	27.hydroxide_ion_usage
25	Trans. 30	MCT 8	4	2,8	29.reaction_NO_with_superoxide_anion
26	Trans. 31	MCT 4	4	2,6	30.superoxide_anion_synthesis_mainly_in_mitochondria
27	Trans. 32	MCT 8	4	2,2	31.lipids_peroxidation_by_peroxynitrite
28	Trans. 34	MCT 16	5	6	33.water_dissociation
29	Trans. 37	MCT 8	4	1,4	36.reaction_catalyzed_by_iNOS
30	Trans. 38	MCT 12	2	1,2	37.endothelial_injury
31	Trans. 39	MCT 18	4	2,8	38.urea_cycle
32	Trans. 40	MCT 9	4	3,4	39.atherosclerosis_progression
33	Trans. 41	MCT 4	4	2,6	40.electrons_in_reaction_formation
34	Trans. 42	MCT 4	4	5	41.inflammatory_process
35	Trans. 43	MCT 4	4	9,2	42.formation_of_oxygen_in_various_biological_processes
36	Trans. 45	MCT 1	5	1,4	44.glycolysis
37	Trans. 46	MCT 1	5	2,8	45.oxidative_decarboxylation
38	Trans. 47	MCT 1	5	2,8	46.sequence_of_reactions_of_the_Krebs_cycle
39	Trans. 48	MCT 1	5	1,4	47.stimulation_of_hydrolysis_of_the_high_energy_binding
40	Trans. 49	MCT 1	5	1,4	48.energy_consuming_biological_processes
41	Trans. 50	MCT 1	5	14	49.oxidative_phosphorylation
42	Trans. 51	MCT 1	5	14	50.enzyme_present_in_the_inner_membrane_of_the_mitochondrion
43	Trans. 52	MCT 1	5	14	51.the_presence_of_ubichinon_in_the_mitochondria
44	Trans. 53	MCT 1	5	56	52.translocation_of_protons_across_the_inner_membrane
45	Trans. 54	MCT 1	5	56	53.outer_side_of_the_protein_lipid_membrane

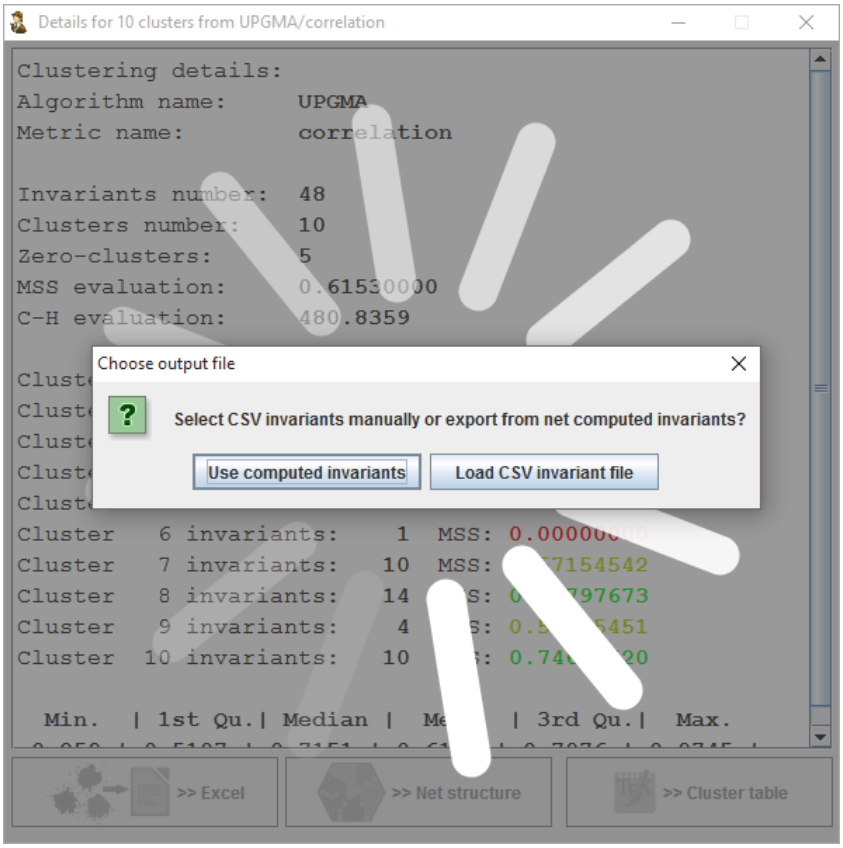
MCT | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 | Cluster6 | Cluster7 | Cluster8 | Cluster9

Arkusz 5 / 10 | PageStyle_Cluster4 | Suma=0 | 100%

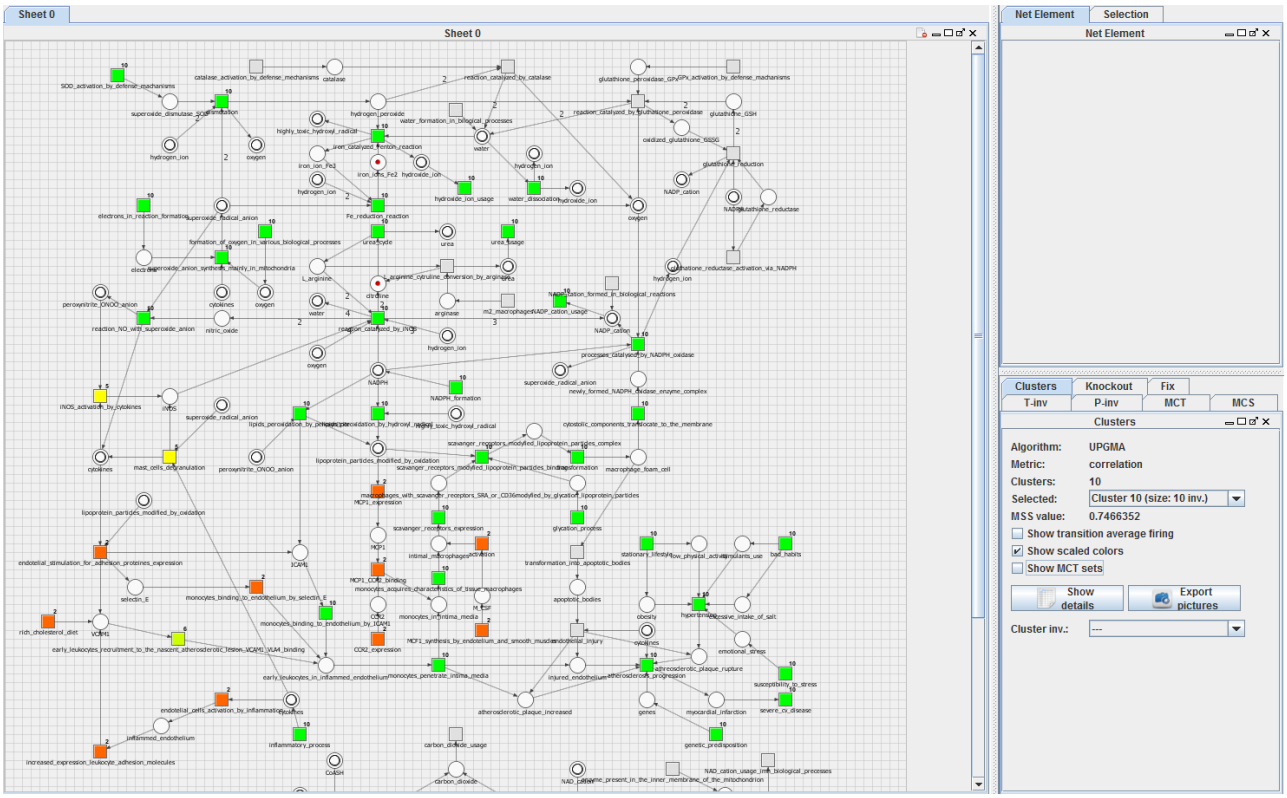
.

7.2.4 Clusters on a net structure

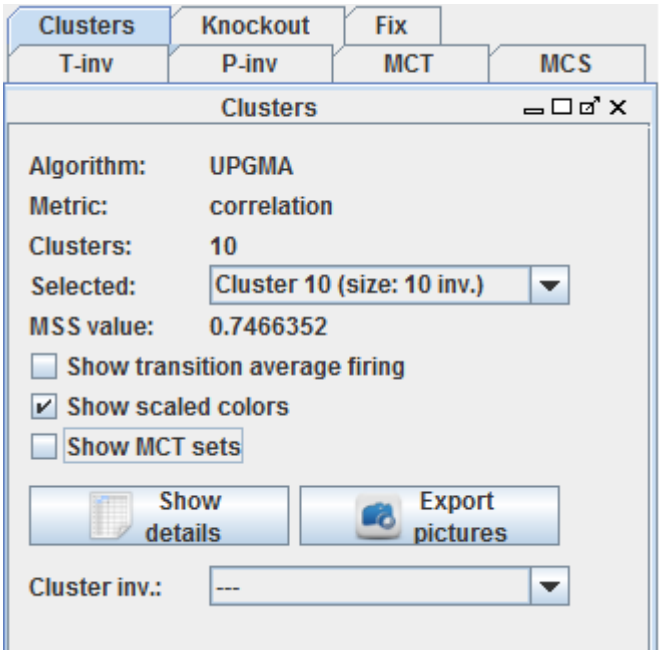
This button will again require R computations for a selected clustering. After short time, a message will appear and the clustering information window described in the last chapter (7.2.3) can be closed. Cluster information in such a form requires net editor and the option from one of the tab of the 6th section of the main interface window.



The clusters view example:



As one can see, Clusters tab is selected from the Section 6 of the Holmes window (bottom right part of the window). From there, a specific cluster can be selected, and its structure will be drawn (using colors) on a net structure.



Possible options:

- **Show transition average firing**
- **Show scaled colors**
- **Show MCT sets**

First one will compute each transition average firing within the cluster (based on the values on non-zero entries of the t-invariants from the cluster). When this option is not selected (by default), the number of times each transition exists in a cluster is presented (i.e. the number of this cluster t-invariants in which the transition exists).

Second button (turned on by default) shows the colors on a net structure in a scaled way – from red (lower values) to green (greatest values), going through degrees of orange and yellow colors.

Third option will mark MCT sets within the cluster in different colors.

Show details will show data about cluster in a notepad.

Export pictures will require selecting a directory. When selected, each cluster will be saved there as picture.

Cluster inv.: allows selecting each t-invariant from a previously selected cluster.

7.3 Minimal Cutting Seys (MCS)

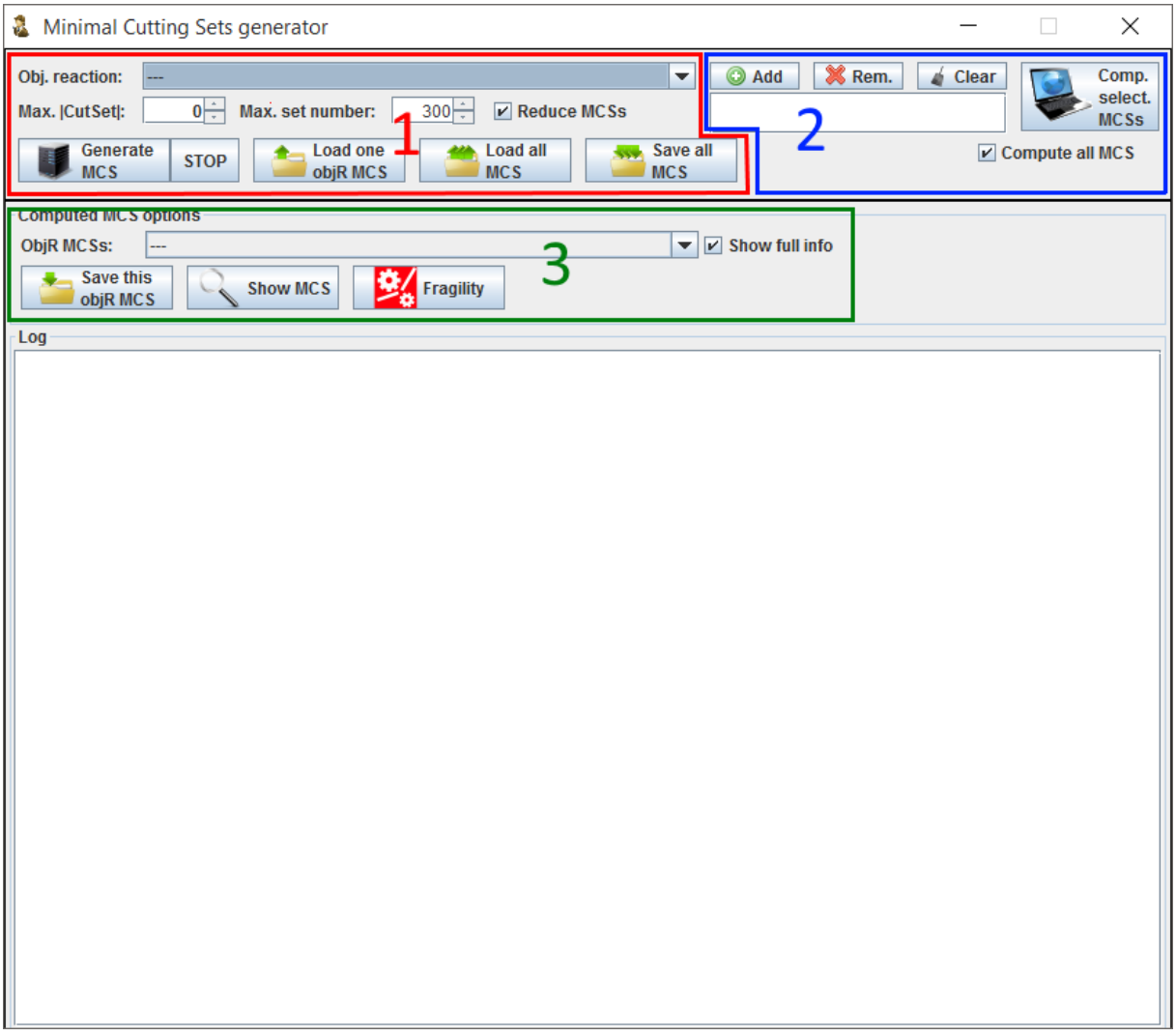
This module can be selected from menu **Analysis** -> **Minimal Cut Sets…** (**Ctrl+G**)..

In theory, such MC Sets provide an answer to the following question: what transition(s) should be disabled, in order to completely disable all t-invariants containing a pre-selected single transition.

Algorithm generating such sets implemented in Holmes is taken from (Klamt i Gilles, 2004). Other algorithms for generating such sets are also known in the literature, e.g., (Ballerstein, von Kamp, Klamt i Haus, 2012).

One very important thing should be noted here: this is a straight-forward bruteforce algorithm trying to search the whole search space. In simple words in means, that selecting sets greater than (usually) 4 elements, can results in a very long computations – hours or more. Therefore it is advised to compute sets from a range 1 to 3(4) at maximum.

The window looks like in the picture:

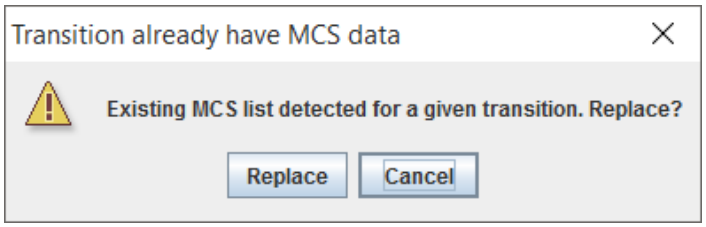


It has been divided into three parts. Panel **1** is devoted for creating MCS separately for each transition. Panel **2** can generate MCS for the whole net (for every transition but automatically). Panel **3** has the tools for viewing the sets.

Elements of panel **1** are as follows:

- **Obj. reaction** - here and *objective reaction* must be selected, i.e., a transition for which MCS will be computed.
- **Max. |CutSet|** - maximal sizes for generated sets, **IT IS ADVISABLE NO THE EXCEED 3-4**.
- **Max. set number** – maximal number of sets generated, computation will stop when the results will exceed this number
- **Reduce MCS** – will be explained later
- **Generate MCS** – starts the computations.
- **Stop** – it will stop the computations, but it can take some time to 'trigger'.
- **Load one objR MCS** – load a file with MCS for a single transition (extension: .objR)
- **Load all MCS** – load a file with all (computed) sets for all or a subset of transitions (extension: .mcs)
- **Save all MCS** – saves the .mcs file

If loaded file will contain data for a transition which already has some recently computed MCS, the question will appear when loading:

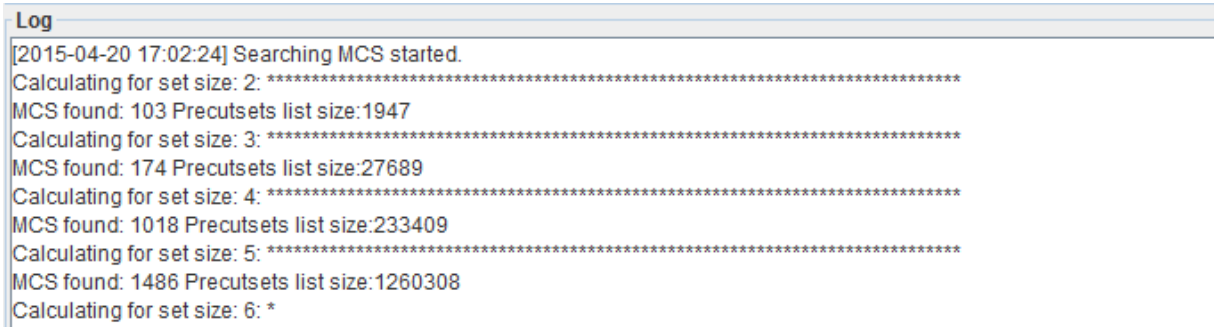


ATTENTION: One more time it should be stated, that computing MCS with greater sizes (more than 3-4) can take a long time.

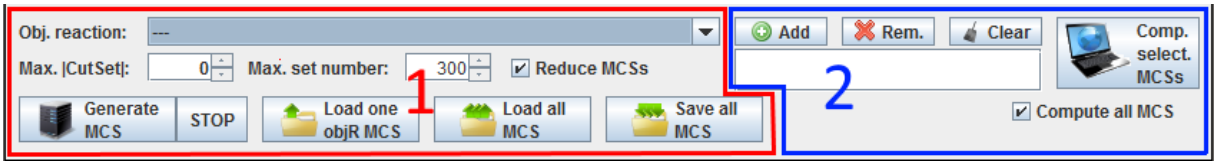
```
[2015-04-20 16:54:05] Searching MCS started.
Calculating for set size: 2: **********************************************************************************
MCS found: 41 Precutsets list size:490
Calculating for set size: 3: **********************************************************************************
MCS found: 42 Precutsets list size:2067
Calculating for set size: 4: **********************************************************************************
MCS found: 42 Precutsets list size:3565
Calculating for set size: 5: **********************************************************************************
MCS found: 42 Precutsets list size:2050
Calculating for set size: 6: **********************************************************************************
MCS found: 42 Precutsets list size:0
[2015-04-20 16:54:06] MCS list created.
```

In the above picture there is a very optimistic scenario, when not only bigger MCS have been computed, but it seems all of them (all existing one) – because the Precutsets list containing new candidates for MCS is 0.

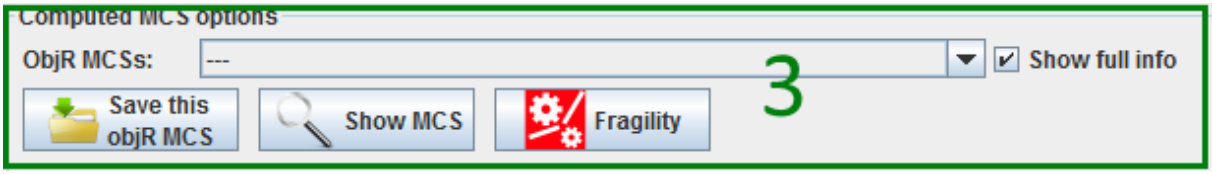
Sadly, more realistic scenario can look like this:



As one can see, after a minute of computations, we have 1 260 308 new candidates list (and rising).



Panel **2** allows to add transitions to the list (below) the buttons. The top right button will initiate the calculations. If "Compute all MCS" option is enabled, the list is disregarded and sets for all transitions will be computed (with respect to their size and their total maximum number provided in the panel **1**.

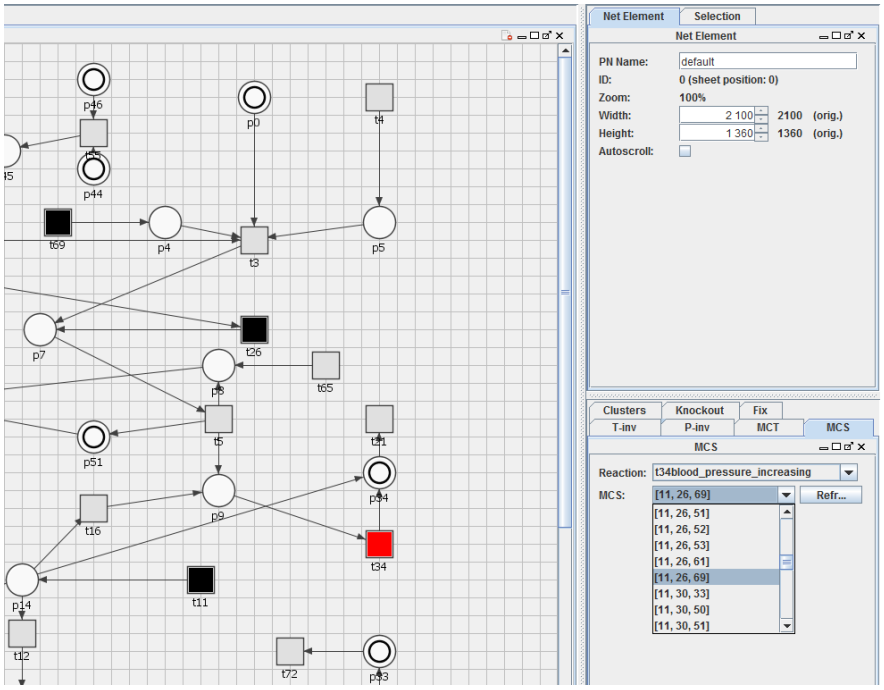


Panel **3** provides options for analyzing the MC Sets:

- **ObjR MCSs** - first the transition must be selected for which the lower button will refer.
- **Save this objR MCS** – saves the sets for the selected transition into a file.
- **Show MCS** - show MCS in a textbox below the button.
- **Fragility** - for each transition a *fragility function* is computed (Klamt i Gilles, 2004)
- **Show full info** – if enabled, full names of transition will be used when showing MCS data.

7.3.1 Showing MCS on a net structure

Similarly as for the other modules, MCS can be seen on a net structure. The view looks like in the picture below:



Objective reaction is a transition t34. One set is selected from a list, consisting of transition t11, t26 and t69. Disabling these transition will knockout every t-invariant containing in its support the transition t34.

This is a different mechanism than the simulation knockout, but it can be verified that sometimes both methods provide similar results.

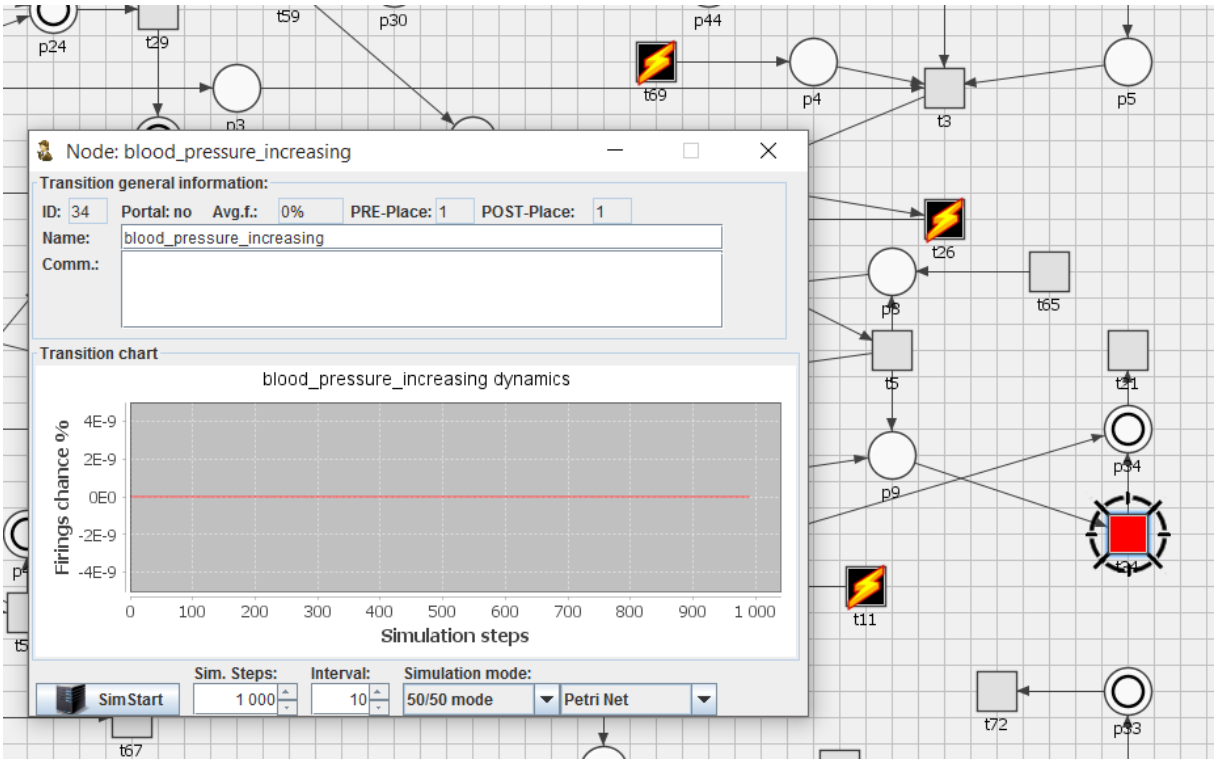
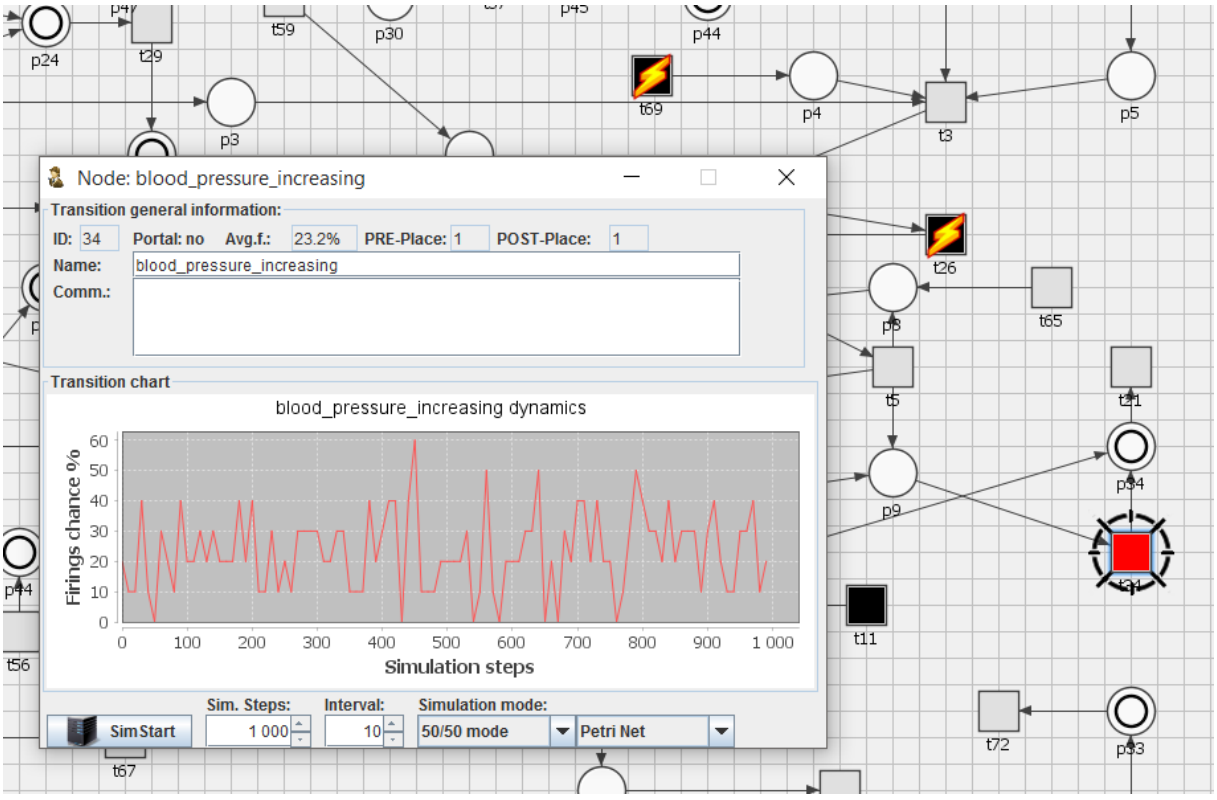


Chart for t34 for disabled t69 and t26:

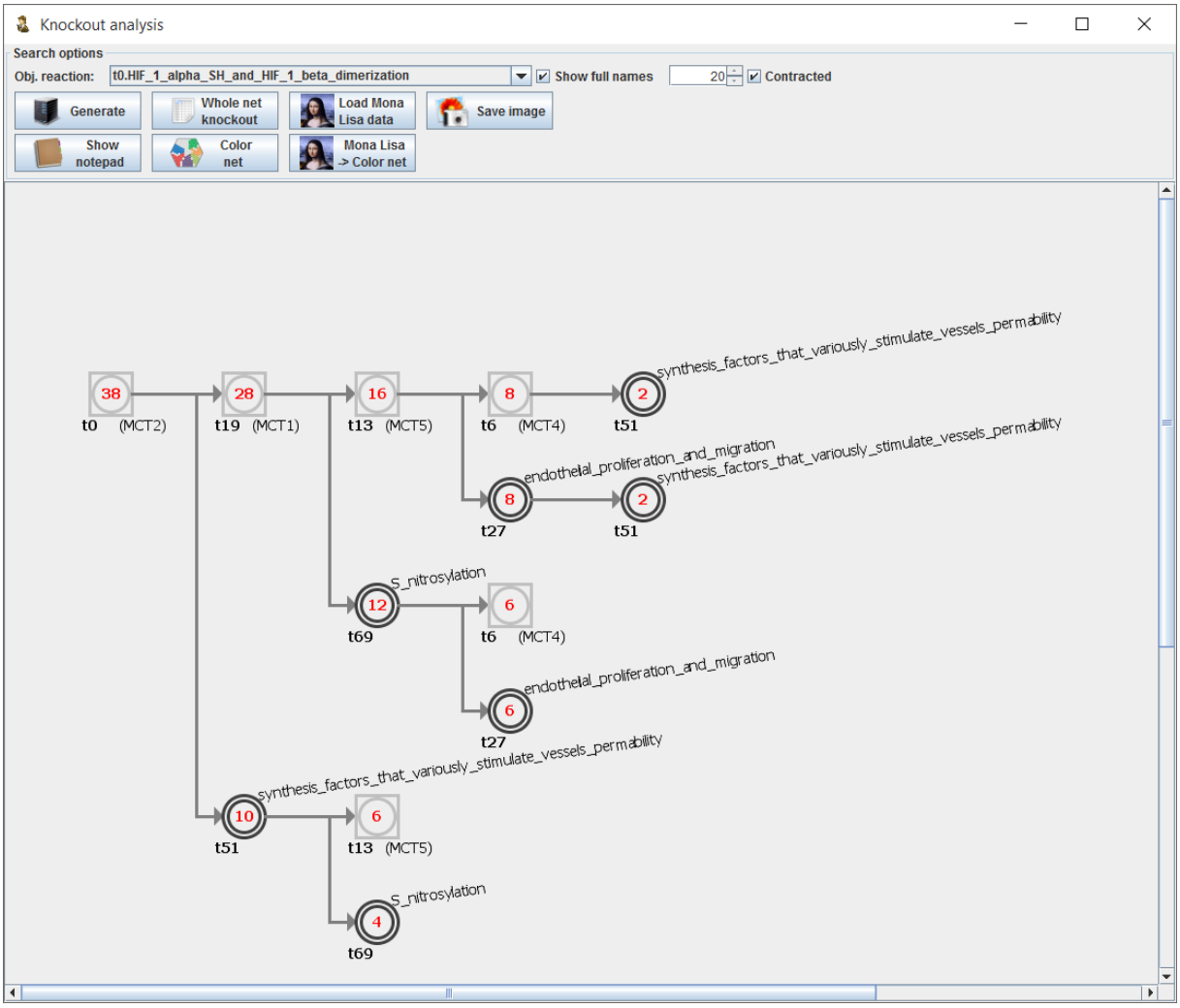

Node: blood_pressure_increasing
Transition general information:
ID: 34
Portal: no
Avg.f.: 23.2%
PRE-Place: 1
POST-Place: 1
Name: blood_pressure_increasing
Comm.:
Transition chart
blood_pressure_increasing dynamics
Firings chance %
Simulation steps
Sim. Steps:
1 000
Interval:
10
Simulation mode:
50/50 mode
Petri Net
SimStart
p24
t29
t59
p30
p44
t69
p4
t3
p5
p3
t26
t65
p8
t5
t21
p34
p9
t11
t72
p33
t67
t56
p44

7.4 t-invariants *knockout* analysis

Implemented algorithm uses the algorithm and idea presented in (Grunwald, Speer, Ackermann i Koch, 2008).

It should be noted, that this knockout analysis is completely different from a simulation knockout already explained.

Module can be activated from menu **Analysis** -> **Knockout analysis** (**Ctrl+K**). The window can look like in the following example (for already existing net):



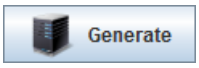
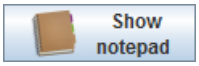
Double circles indicate transitions, squares with circles – MCT sets and one of its transition (e.g. for MCT2, t0 is shown). The idea behind the connections (drawn from the idea of *Mauritius maps*) is fully explained in in (Grunwald, Speer, Ackermann i Koch, 2008).

There is a simple zoom mechanism for the view panel, activated with mouse wheel and CTRL button pressed.

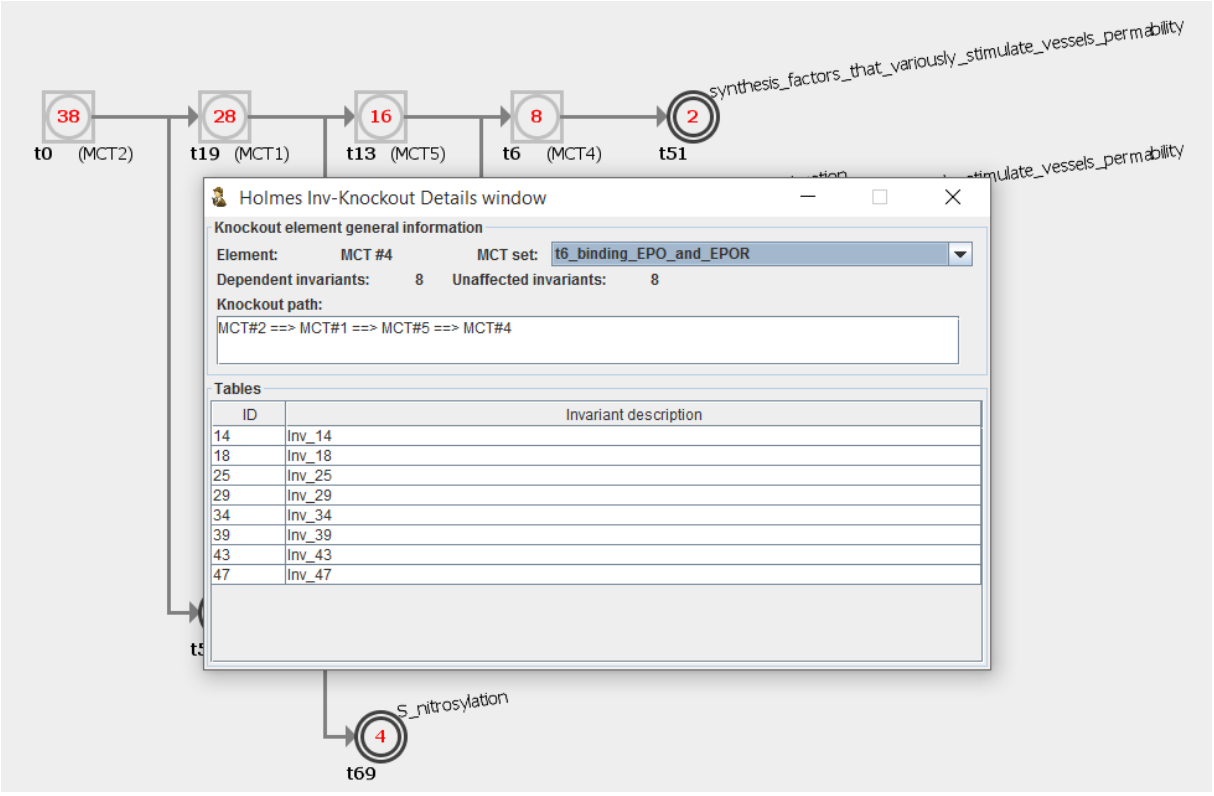
This analysis in general is the opposite idea of MCS. Here we answer a question *what* will be knocked out, if some specific transition will be disabled – based on a t-invariants set.

- **Obj. Reaction** - selection of a transition to be disabled (for which a knockout impact will be drawn).
- **Show full names** – when checked, full names of transition will be used.
- **Contracted** – choosing this option will group transitions into MCT sets.

Buttons:

-  - this button will generate the knockout map.
-  - this button will show knockout impact for every transition as a percentage odf disabled transitions and invariants
-  - for selected transition, some data will be provided:
  - list of all transitions knocked out due to disabling of t_x,
  - list of all transition which together with t_x belong to the same MCT set,
  - list of all t-invariants (ID + description if exists) disabled by t_x knockout.
-  - this will color the structure of the net:
  - selected t_x – red color,
  - transition knocked out by t_x – black color,
  - transition from the same MCT as t_x – blue color.
-  - *single-transition knockout* results from MonaLisa program can be loaded by this button (into Holmes memory).
-  - similar as above, but the file will be used to activate Knockout tab in the Section 6 of the main window of Holmes.

When LMB is clicked on some node of the map, additional data in a separate window will be provided.

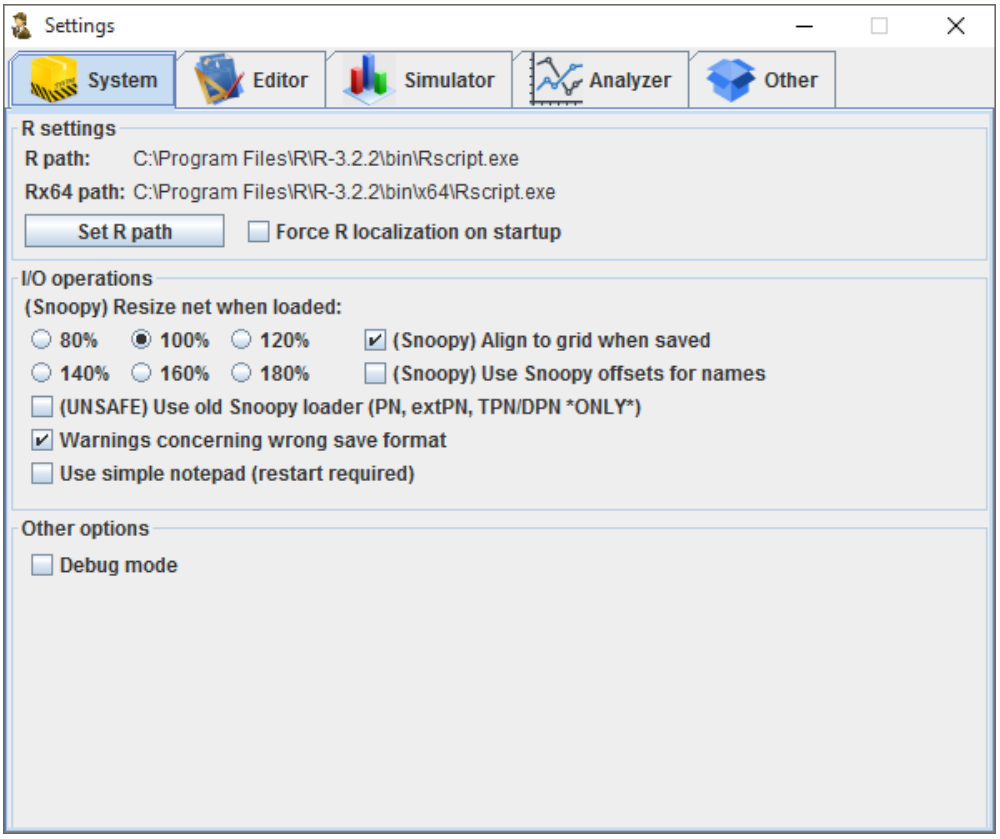

38
t0 (MCT2)
28
t19 (MCT1)
16
t13 (MCT5)
8
t6 (MCT4)
2
t51
synthesis_factors_that_variously_stimulate_vessels_permability
Holmes Inv-Knockout Details window
Knockout element general information
Element: MCT #4
MCT set: t6_binding_EPO_and_EPOR
Dependent invariants: 8
Unaffected invariants: 8
Knockout path:
MCT#2 ==> MCT#1 ==> MCT#5 ==> MCT#4
Tables
ID
Invariant description
14 Inv_14
18 Inv_18
25 Inv_25
29 Inv_29
34 Inv_34
39 Inv_39
43 Inv_43
47 Inv_47
S_nitrosylation
4
t69

8. Other options

8.1 Properties

Properties window is available from menu **Windows** -> **Properties** (**Ctrl+W**).

Many of the option from this window have been already explained, this chapter serves as a summary.

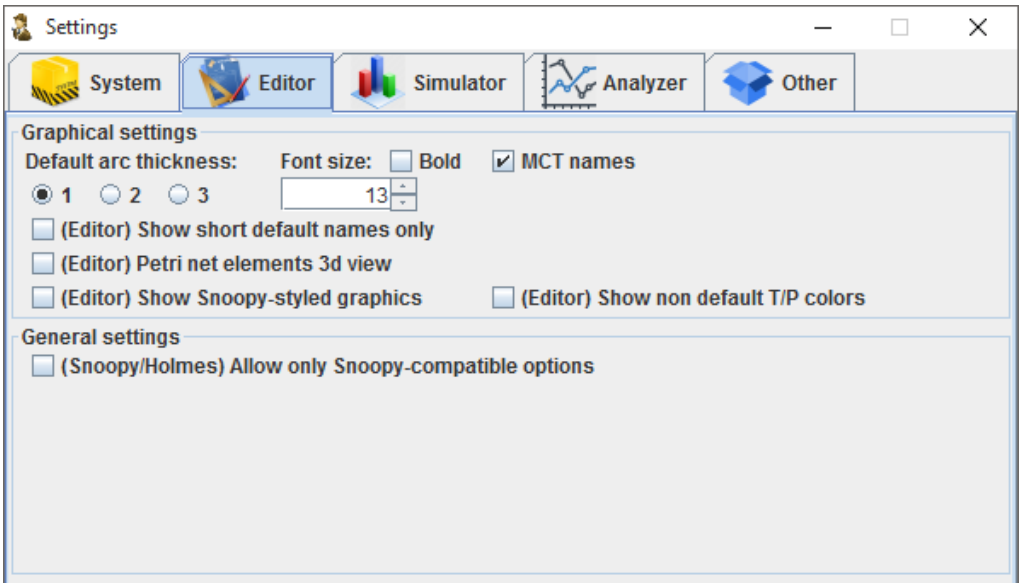


System tab.

- **R settings**
  - **R path** – path for Rscript.exe file necessary for clusters calculations.
  - **Rx64 path** – set automatically assuming the R path is chosen correctly and the 64bit R program exists.
  - **Set R path** - here the user can update the path for Rscript.exe (**32-bits)**
  - **Force R localication on startup** – when this option is turned on, Holmes will warn on startup when the path to R is invalid – this can happen when R environment is updated to the newer version
- **I/O operations**

- **(Snoopy) Resize net when loaded 80% - 180%** - choosing any option other than 100% will resize the net when a file from Snoopy is imported
- **(Snoopy) Align to grid when saved** – align the net elements to 20p grid
- **(Snoopy) Use Snoopy offsets for names** – when turned on, names of places and transition will have location depending on the data from a Snoopy net file, otherwise they will be centered below places/transitions symbols.
- **(UNSAFE) Use old Snoopy loader (PN, extPN, TPN/DPN *ONLY*)** – using old Snoopy loading algorithm. For debug purposes only.
- **Warnings concerning wrong save format** – when enabled, Holmes will warn when user tries to save a net into a Snoopy format which is not suitable to store all the net data.
- **Use simple notepad (restart required)** – turning this on will use simpler version of a notepad (debug).

- **Other options**
  - **Debug mode** – debug mode, for test and development purposes.

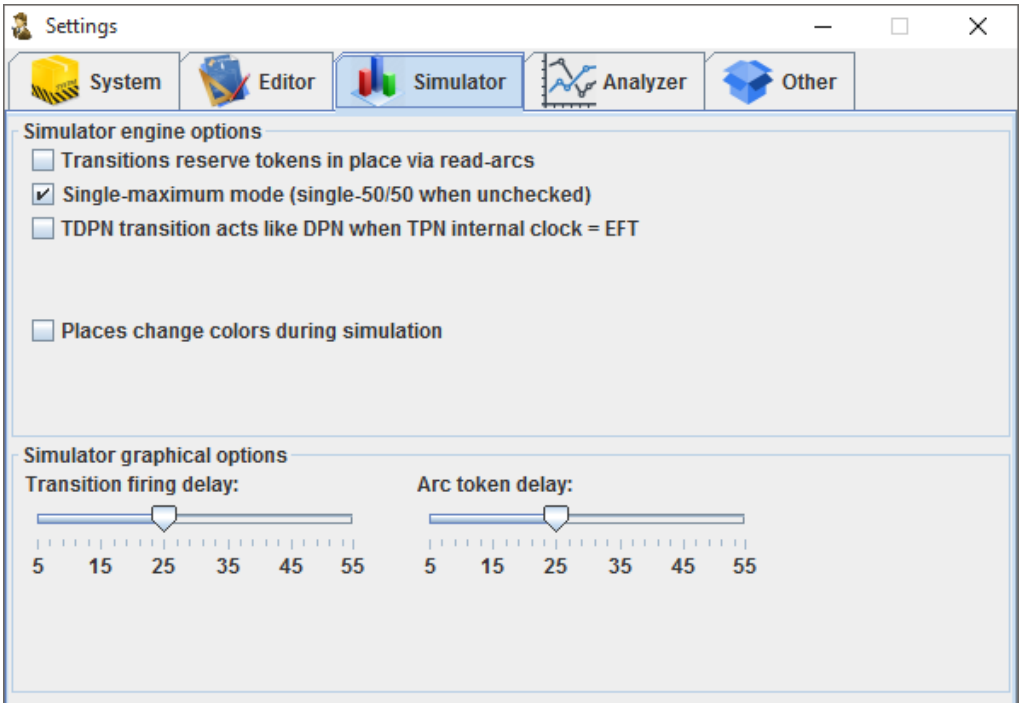


Editor tap:

- **Graphical settings**
  - **Default arc thickness : 1,2,3** – default thickness of arc (in pixels)
  - **Font size** – size for names of places and transitions
  - **Bold** – bold font if enabled
  - **MCT names** – when showing MCT set on a net structure their name will or will not be shown depending on checking this option
  - **(Editor) Show short default names only** – when enabled, Holmes will show short form of places and transition names, e.g. p0, t34, etc. Old names will be still safely stored in a project file, only not shown in the editor.
  - **(Editor) Petri net elements 3d view** – activate shadows of net elements

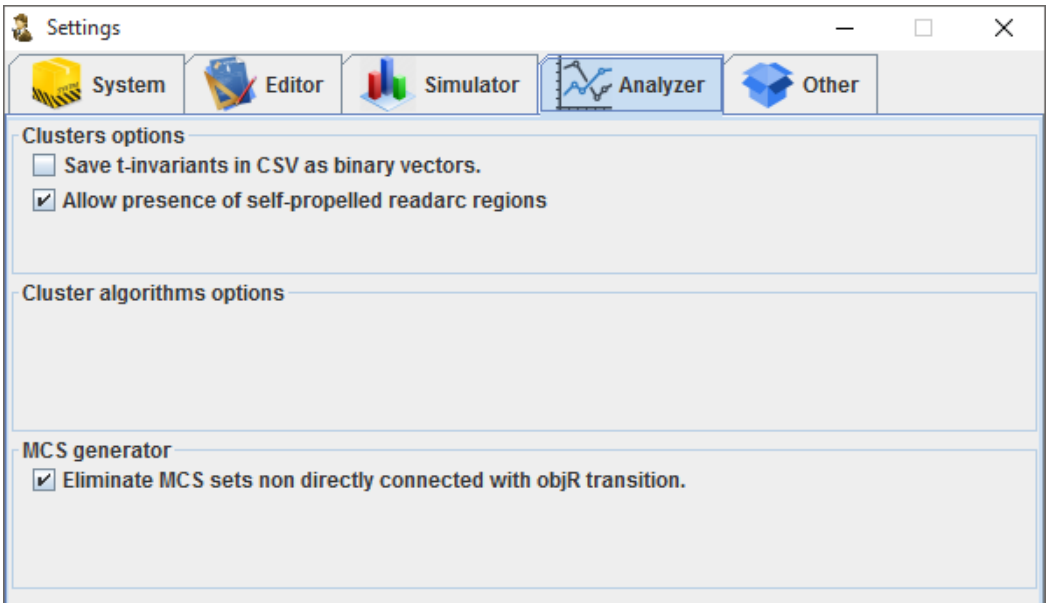
- **(Editor) Show Snoopy-styled graphics** – when enabled, colors assigned to places and transitions in Snoopy will be used when Snoopy net is imported.
- **(Editor) Show non default T/P colors** – allows using non-standard colors for net elements

- **General settings**
  - **(Snoopy/Holmes) Allow only Snoopy-compatible options** – when turned on, non all hierarchical net options will be available.
  - **Use meta-arcs compression for metanodes** – test option.



Simulator tab:

- **Simulator engine options**
  - **Transitions reserve tokens in place via read-arcs** – see chapter 6.2.1.1.
  - **Single-maximum mode (single-50/50 when unchecked)** – see chapter 6.2.1
  - **TDPN transition acts like DPN when TPN internal clock = EFT** – see chapter 4.1.3.3
  - **Places change colors during simulation** – see chapter 6.1.2
- **Simulator graphical options**
  - **Transition firing / Arc token delay** – see chapter 6.1.3



Analyzer tab:

- **Save t-invariants in CSV as binary vectors** – if enabled, all t-invariants will be stored in files as a 0-1 binary data files. For specific clustering options only (simulate the Binary distance metric).
- **Allow presence of self-propelled readarc regions** – feasible t-invariants will not be generated for some specific t-invariants, which can provide its own tokens (cyclic invariants),
- **Eliminate MCS sets non directly connected with objR transition** – experimental option for reducing MCS sets depending on the net structure and arcs direction.

9. Net comparison

This section focus on Petri nets comparison module. It is available from menu **Analysis** -> **Net Comparison** (**Ctrl+?**). It contains several comparison techniques. In process participate network model from Workspace and second one loaded from file by proper button in module.

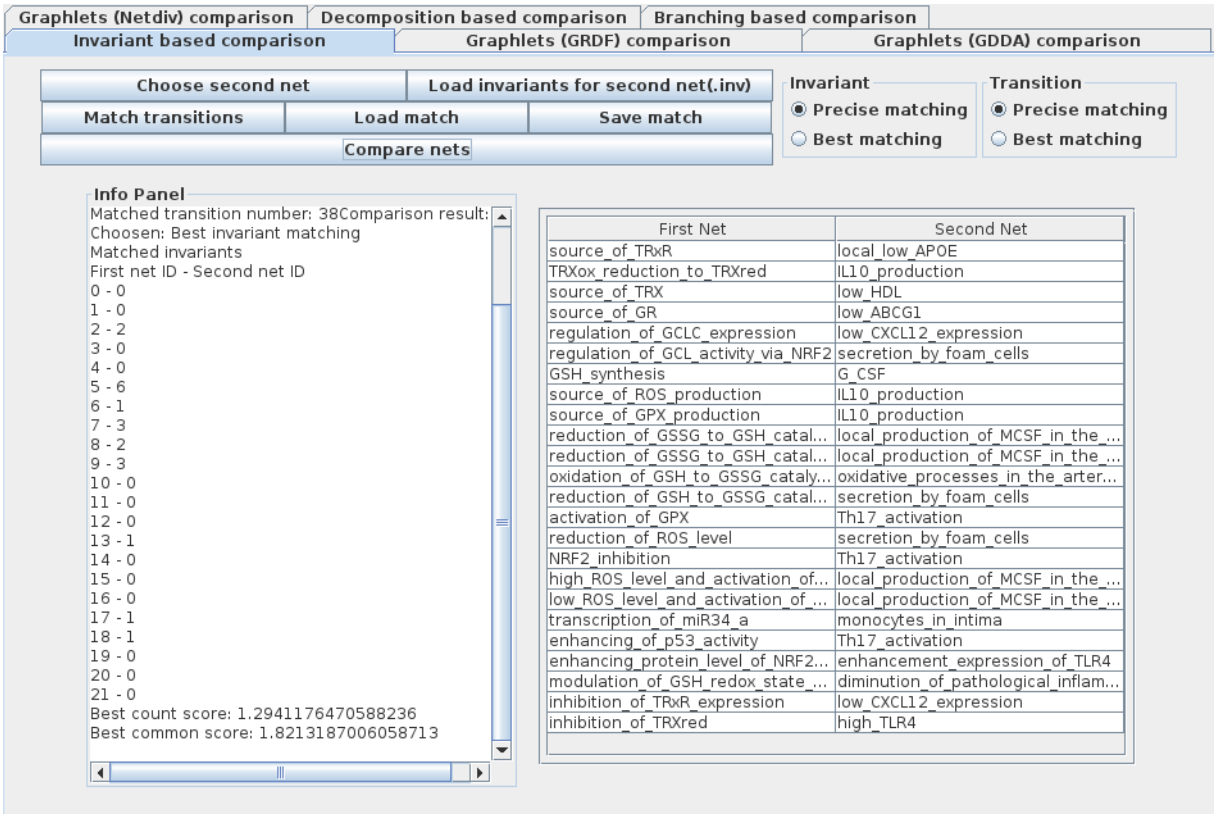
1. Invariants based comparison
2. Branching nodes based
3. Decomposition based
4. Graphlet Relative Distribution Frequency
5. Graphlet Degree Distribution Agreement
6. netDis (Graphlet based)

9.1 Invariant based comparison

First comparison method ever created for Petri nets has been proposed in “Petri nets for modelling metabolic pathways: a survey in 2010”. It is composed of two sub-methods. First one is based on Enzyme Numbers while second on t-invariants. The later has been extended and implemented in this module as independent method. It is based on simple concept of matching two t-invariants from compared nets.

First step is recognizing which transitions can be called identical or similar. User can choose between automatic method based on Levenstein Distance will match nodes, or load prepared matching from file.

Second step is the proper comparison process that matches t-invariants. Depending on used option, it will only match identical invariants or it will find invariants will highest similarity of transitions.



First Net	Second Net
source_of_TRxR	local_low_APOE
TRXox_reduction_to_TRXred	IL10_production
source_of_TRX	low_HDL
source_of_GR	low_ABCG1
regulation_of_GCLC_expression	low_CXCL12_expression
regulation_of_GCL_activity_via_NRF2	secretion_by_foam_cells
GSH_synthesis	G_CSF
source_of_ROS_production	IL10_production
source_of_GPX_production	IL10_production
reduction_of_GSSG_to_GSH_catal...	local_production_of_MCSF_in_the_...
reduction_of_GSSG_to_GSH_catal...	local_production_of_MCSF_in_the_...
oxidation_of_GSH_to_GSSG_cataly...	oxidative_processes_in_the_arter...
reduction_of_GSH_to_GSSG_catal...	secretion_by_foam_cells
activation_of_GPX	Th17_activation
reduction_of_ROS_level	secretion_by_foam_cells
NRF2_inhibition	Th17_activation
high_ROS_level_and_activation_of...	local_production_of_MCSF_in_the_...
low_ROS_level_and_activation_of_...	local_production_of_MCSF_in_the_...
transcription_of_miR34_a	monocytes_in_intima
enhancing_of_p53_activity	Th17_activation
enhancing_protein_level_of_NRF2...	enhancement_expression_of_TLR4
modulation_of_GSH_redox_state_...	diminution_of_pathological_inflam...
inhibition_of_TRxR_expression	low_CXCL12_expression
inhibition_of_TRXred	high_TLR4

Matching results for transition labels are visible on table on the right side of panel.

Available buttons are as follows:

- **Choose second net** – choosing second net used in comparison process.
- **Load invariants for second net** – choosing previously generated invariants for the second net.
- **Match transitions** – choosing this option will match transitions using chosen label criterion.
- **Load match** - choosing this option will allow to load transition matching from file.
- **Save match** - choosing this option will allow to save matching to file.
- **Compare nets** - choosing this button will start comparison algorithm.

Type of invariant match:

- **Precise matching** – only invariants with the same size and all transitions are matched will be matched.
- **Best matching** – invariants with the highest number of matched transition labels will be matched.

Type of transition matching

- **Precise matching** - only transitions with identical labels will be matched(string distance equal to 0).

- **Best matching** - transitions witch smallest string distance will be matched.

Results of comparison are presented in **info panel** with IDs of matched invariants and similarity result based on Sorensen index. It is presented as a list of pairs where first element is an ID of invariant from first net and second is an invariant ID from second net.

9.2 Decomposition based comparison

9.2.1 Decomposition

The second comparison method presented in this section uses structures acquired by process of Petri net decomposition. Petri net model is partitioned into set of disjoint subnets that overlap on only on specific type of node. In literature there also exists decomposition that do not follow disjoint restriction. Decomposition of both types were implemented in separate module of Holmes.

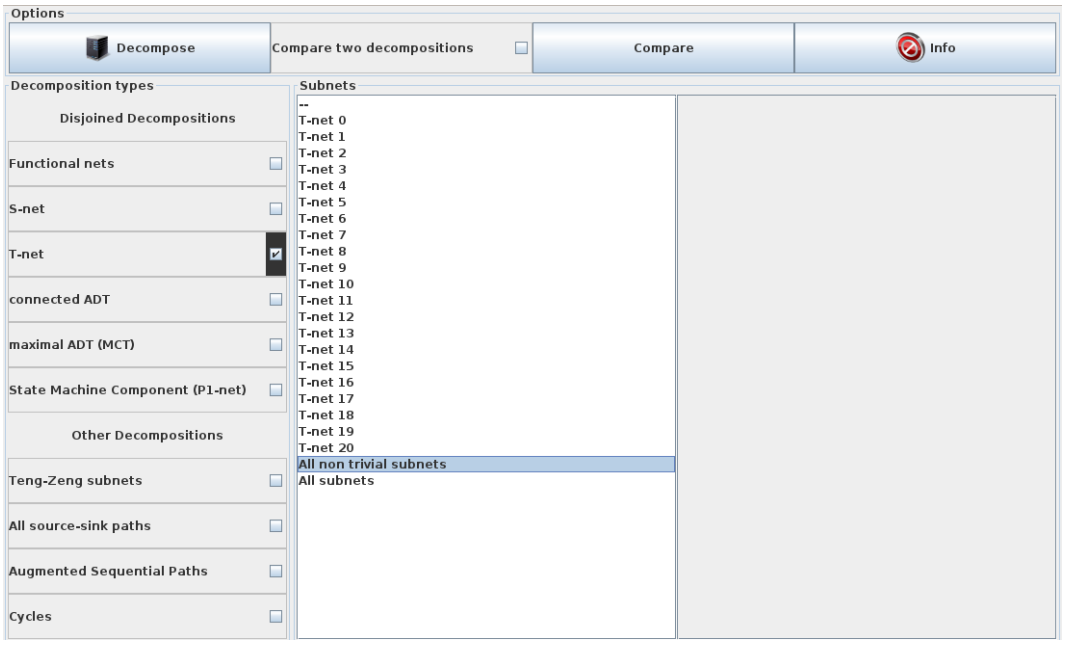
In application version 1.1 ADT decomposition is used in comparison process. More types of decompositions are planned to be added to this type of comparison.

Decomposition window is available from menu **Analysis** -> **Decomposition** (**Ctrl+D**).

Buttons are as follows:

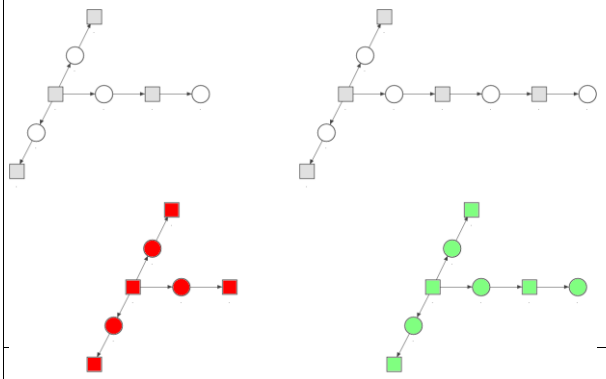
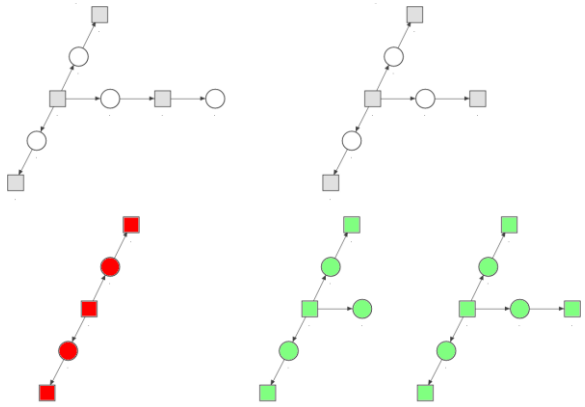
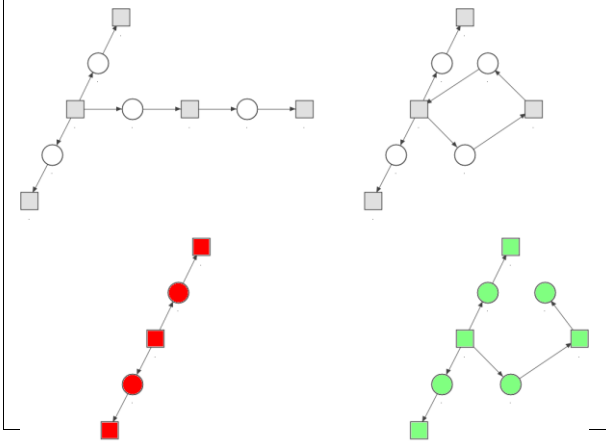
- **Decompose** – decompose net to chosen subnets type
- **Compare two decomposition** – allows to choose two types of decomposition and compare subnets from both variants.
- **Compare** – available only for specific decompositions, opens comparison module window.
- **Info** - open windows with detail information about available decompositions

N.2.2 Maximal common structure variants



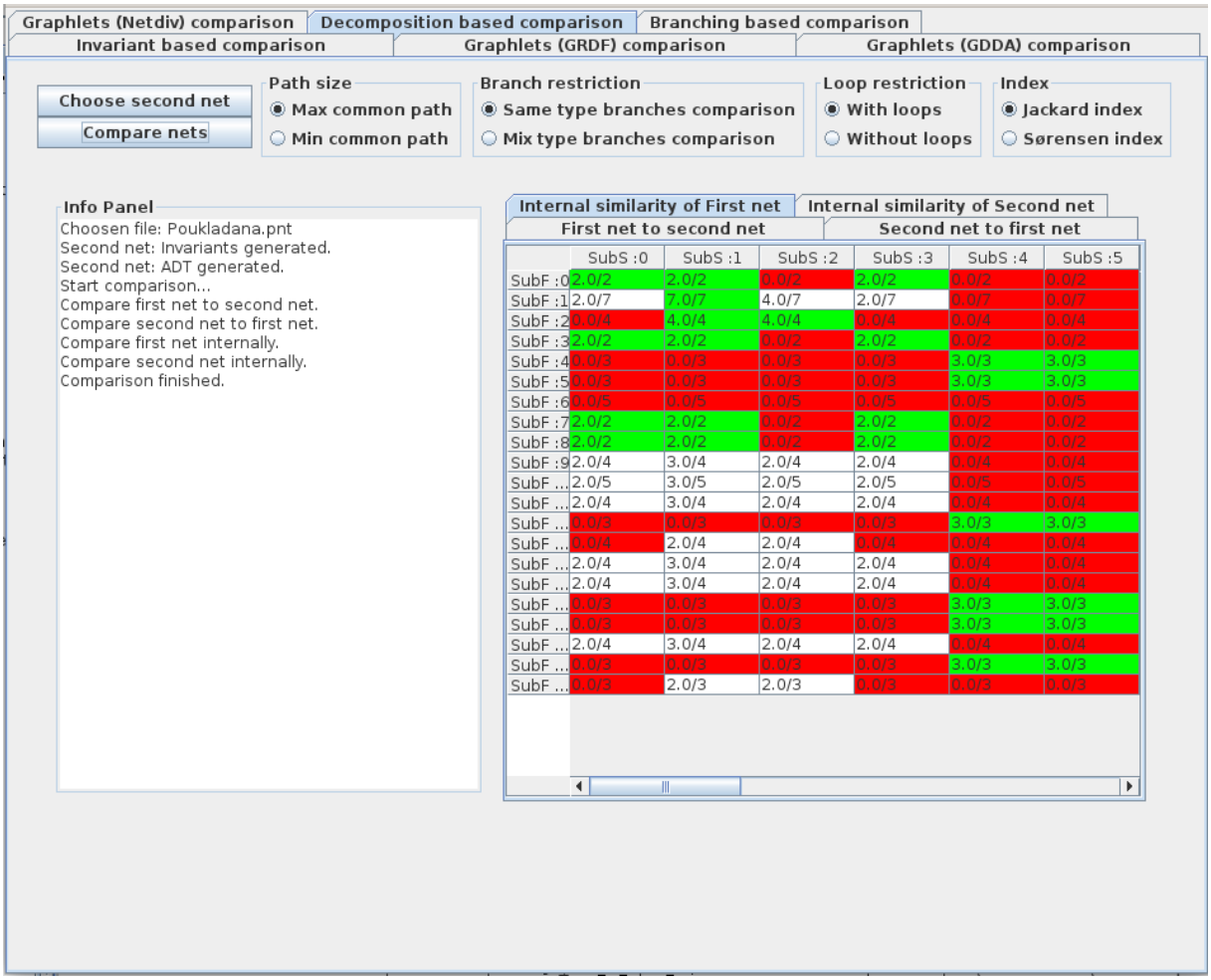
It was observed that Maximal Common Subgraph is not always the most desirable structure as a result of comparison process. In context of subnets like t-net, ADT or MCT, it can be observed that non trivial structures are built around set of branching transitions. Those nodes are responsible for flow of tokens in network and represent higher information value than non-branching transitions.

Interpretation of structures build around branching transitions is strongly connected to context of composed net. Depending on situation some characteristics are preferable that others. To cover such situations three options are available before beginning comparison process.

Common structure restrictions	Information
	Case: Path size When ... 1) Minimal common length (common arcs) 2) Maximal common length (common nodes)
	Case: Branch restriction When ... 1) Paths of different types will not be matched 2) Path of different types will be matched A) Path length minimal B) Path length maximal
	Case: Loop restriction Path that represents a loop will be compared only with other loops or with others non loop paths. 1) Only loop to loop comparison 2) Loop to non-loop path allowed

9.2.3 Comparison

Common structure variants from previous subsection can be chosen using specific radio buttons at the top on the main panel.



Buttons are as follows:

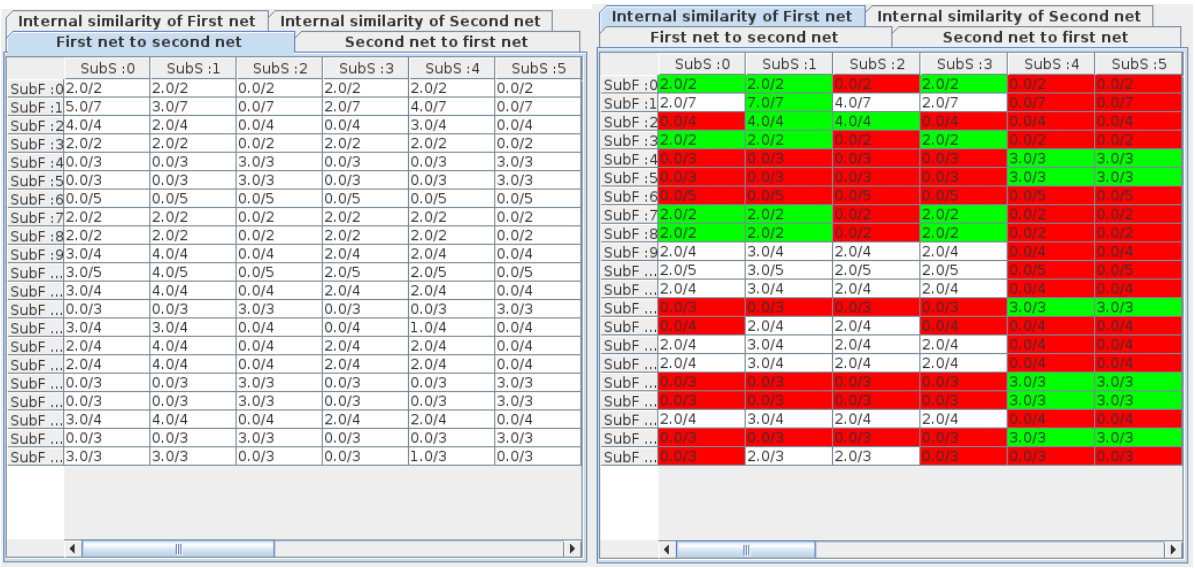
- **Choose second net** – choosing second net used in comparison process.
- **Compare nets** – choosing this button will start comparison algorithm.

Options are as follows:

- **Path size:**
  - Max common path
  - Min common path
- **Branch restrictions:**
  - Same type branches comparison
  - Mix type branches comparison
- **Loop restriction:**
  - With loops
  - Without loops
- **Index:**

- Jackard Index
- Sorensen Index

Results are shown on four matrix presented in four separate tabs. Two of them represents similarity of subnets from the same net (**Internal similarity of … net**), while rest two represents to comparison perspectives first net to second and and second net to first.

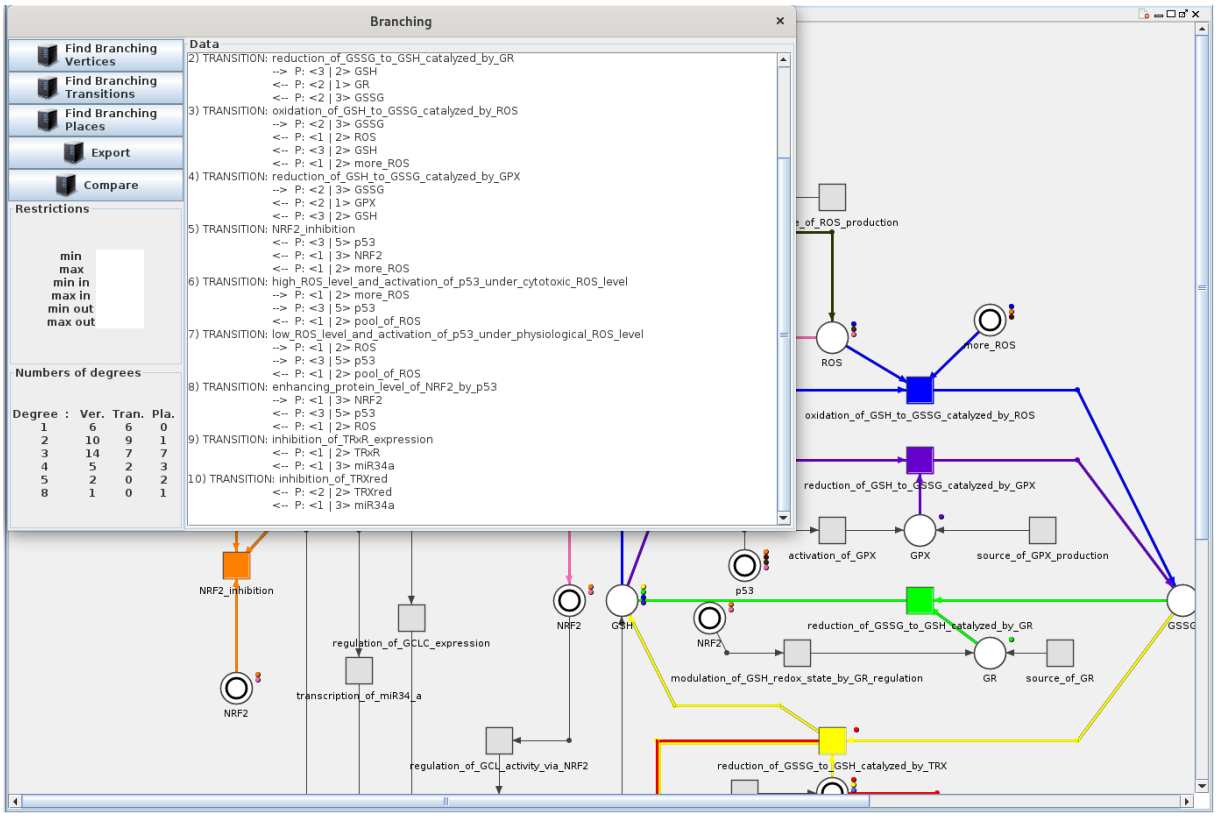


Internal similarity tabs, cells with color green represents situation when two subnets are isomorphic(dark green), or it is included in the other(light green). Their allow for more accurate interpretation of acquired results.

9.3 Branching based comparison

9.3.1 Branching vertices

Branching comparison method is based on finding nodes that have at least two incoming arcs or at least two outgoing arcs. Each branching node is observed through a set of other branching nodes with whom it is connected directly(by arc) or by path. Such nodes are called an endpoint of specific branch node.



Buttons are as follows:

- **Find branching vertices** – finds and colours vertices that fulfils branching restrictions.
- **Find branching transitions** – finds and colours transitions that fulfils branching restrictions.
- **Find branching places** - finds and colours places that fulfils branching restrictions.
- **Export** – export results to file.
- **Compare** – shows branching comparison tab of comparison window.

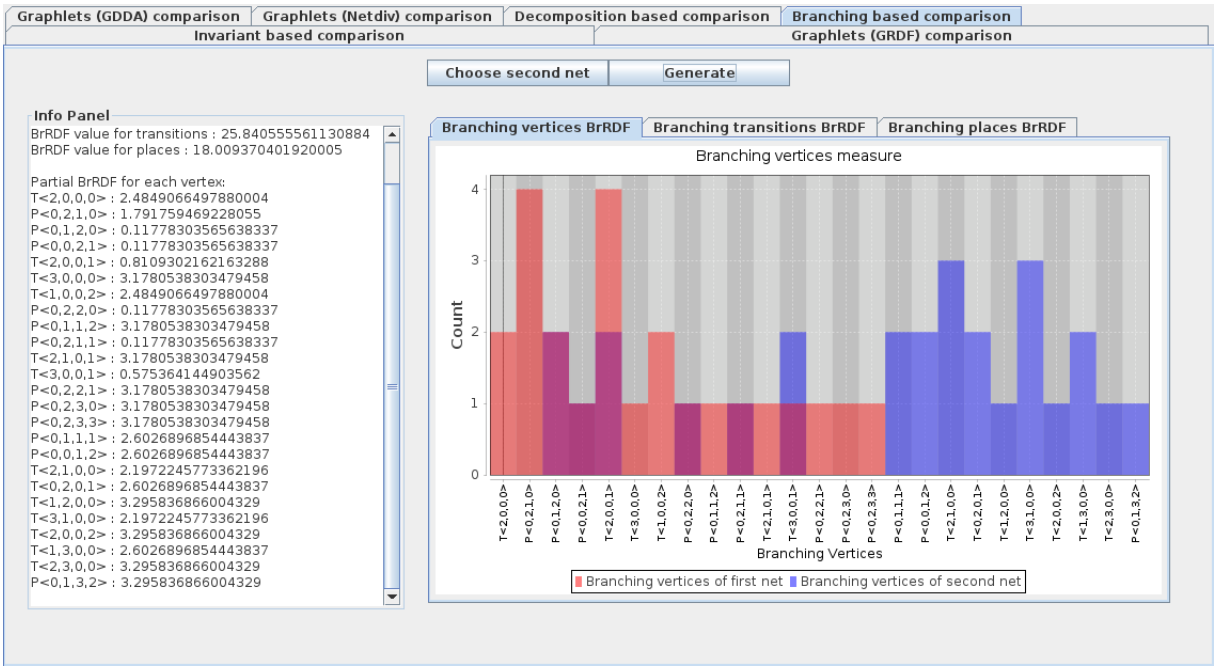
Restrictions are as follows:

- **min** – minimal degree of node
- **max** – maximal degree of node
- **min in** - minimal incoming degree of node
- **max in** - maximal incoming degree of node
- **min out** - minimal outgoing degree of node
- **max out** – maximal outgoing degree of node

Numbers of degrees shows statistic for each degree with recognition of node type.

9.3.2 Branching based comparison

From t-invariant perspective branching transitions represents an complex sub processes inside invariant while branching places represents a point of connection/disconnection with other t-invariants.



Buttons are as follows:

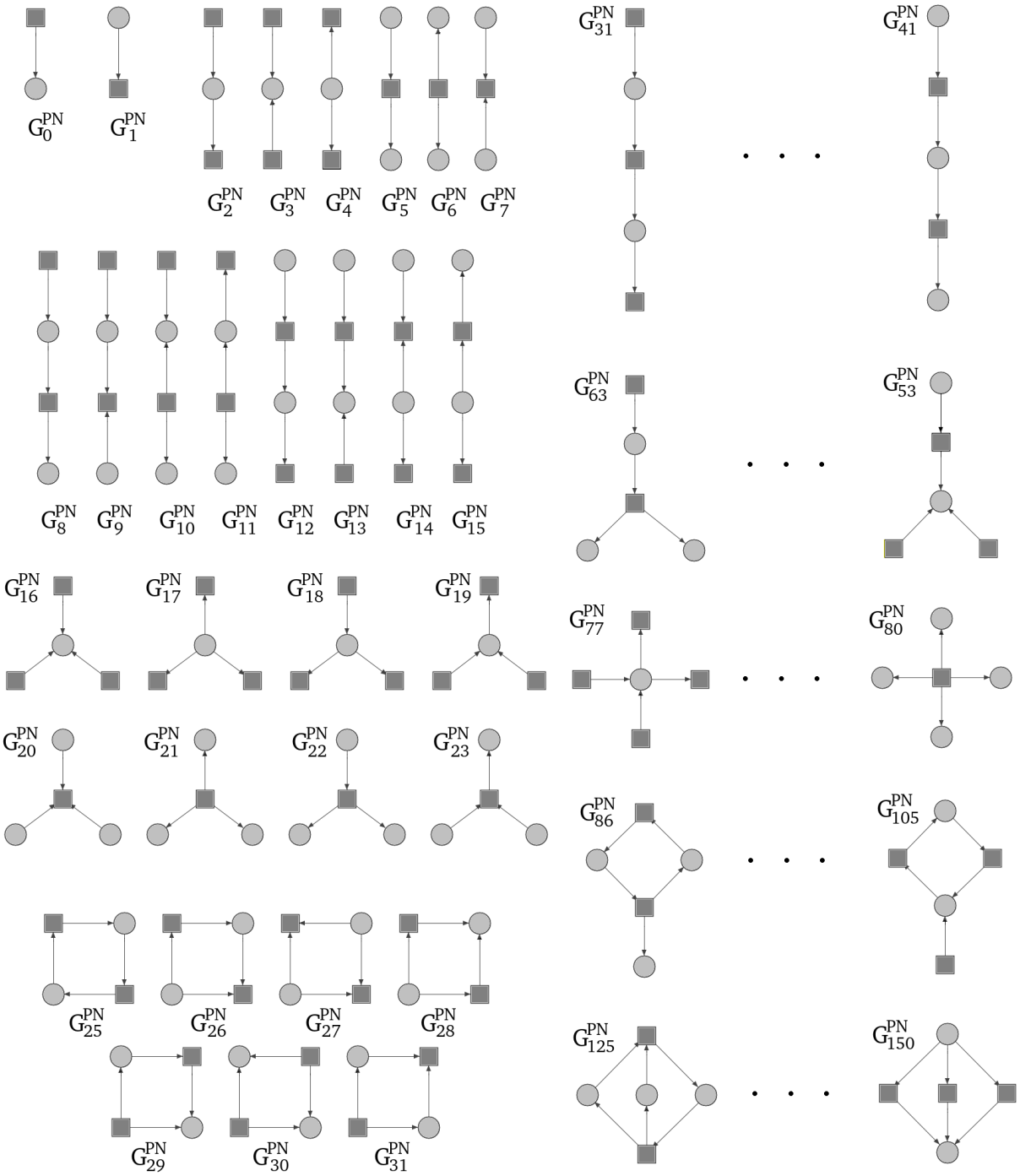
- **Choose second net** – choosing second net used in comparison process.
- **Generate**– choosing this button will start comparison algorithm.

Diagram shows relation between branching vertices existing in compared nets.

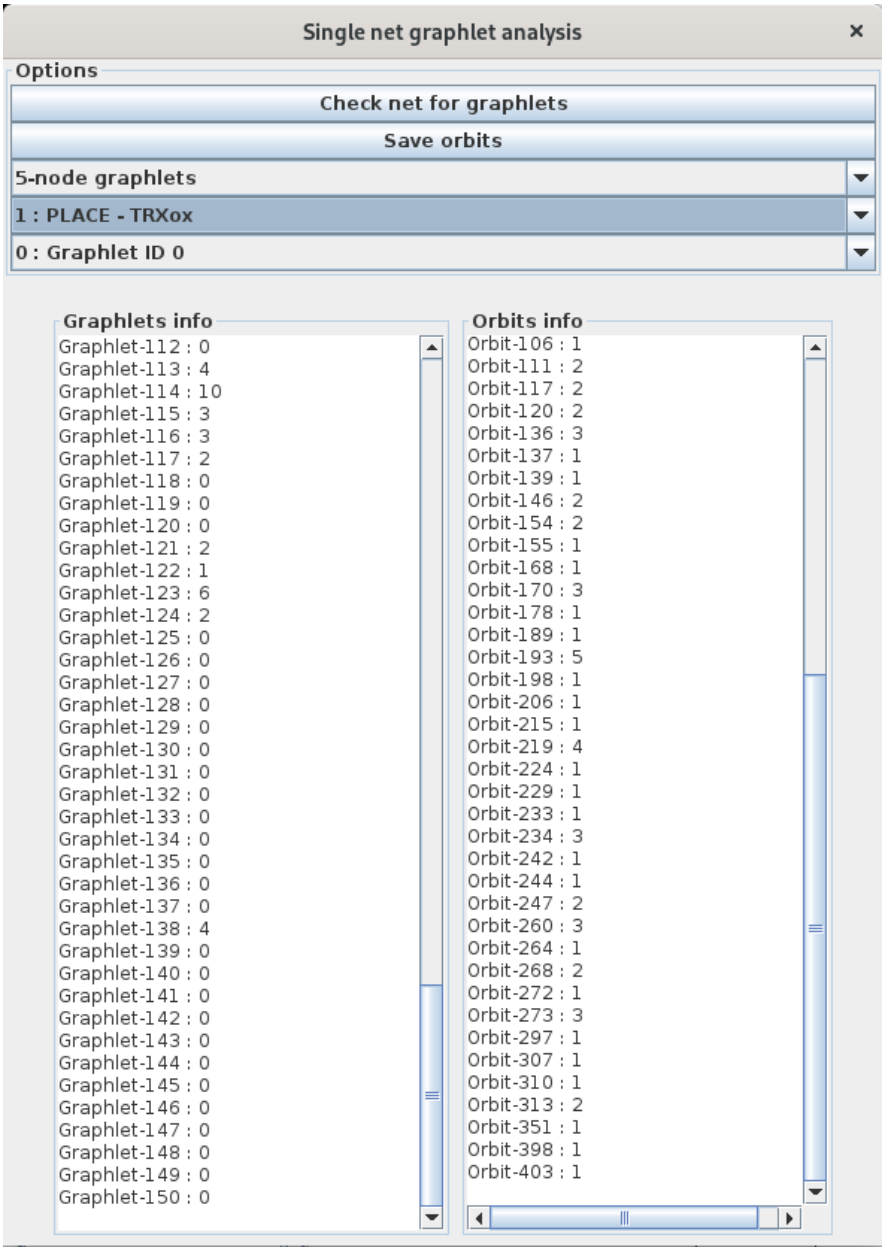
9.4 Graphlets comparison (GRDF)

9.4.1 Graphlets in Petri nets

Graphlets represents a family of small non-isomorphic structures (build on 2-5 nodes). Metric based on them are dedicated for large networks where standard comparison approach is not efficient.



Graphlet window is available from menu **Analysis** -> **Graphlets** (**Ctrl+?**).

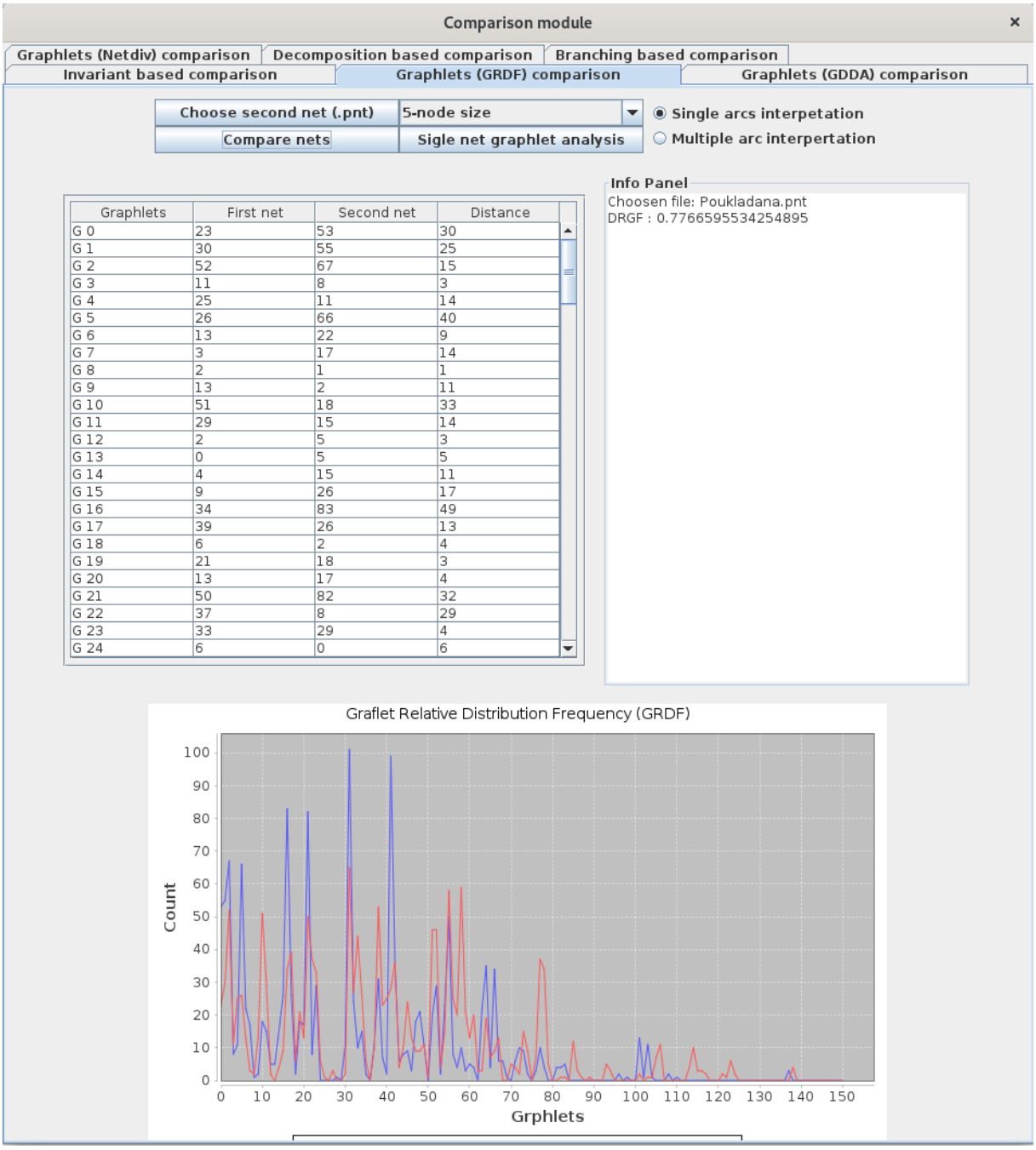


Buttons are as follows:

- **Check net for graphlets** – will find all graphlets with their orbits in network from workspace.
- **Graphlet size**– choosing node size of used graphlets
- **Graphlet info** – will show number of found graphlets of each type
- **Graphlet (ComboBox)** – will show and colour chosen graphlet on workspace.
- **Choose node** – will show found orbits for selected node
- **Save orbits** – will save results to file.

9.4.2 Graphlets Relative Distribution Frequency

Graphlet Relative Distribution Frequency was the first metric based on graphlet created for net comparison. It focus on finding all graphlet in both of compared nets and calculate partial distance for each of graphlets.



Buttons are as follows:

- **Choose second net** – choosing second net used in comparison process.
- **Compare nets** – choosing this button will start comparison algorithm.

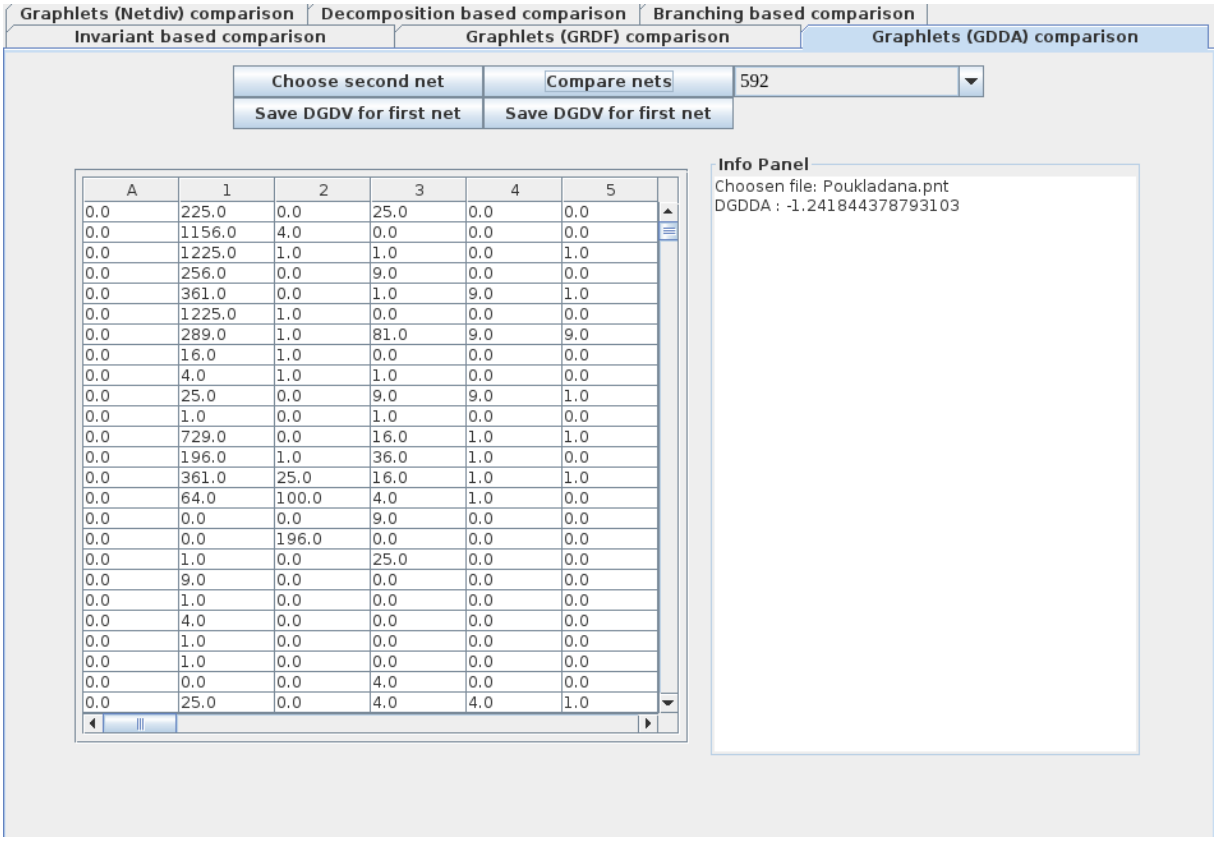
- **Graphlet size** - choosing size/number of graphlets used for comparison
- **Graphlet analysis for single net –** opens window with analysis options for single network

Multiarc interpretation:

- **Single arc interpretation** – weight arcs will be interpreted as single arc
- **Multiple arc interpretation** – weight arcs will be interpreted as multiple arcs

9.5 Graphlets comparison (GDDA)

Graphlet Degree Distribution Agreement Was Proposed as an alternative for GRDF metric. It is based on concept of orbits and counting their occurrence in compared networks.



A	1	2	3	4	5
0.0	225.0	0.0	25.0	0.0	0.0
0.0	1156.0	4.0	0.0	0.0	0.0
0.0	1225.0	1.0	1.0	0.0	1.0
0.0	256.0	0.0	9.0	0.0	0.0
0.0	361.0	0.0	1.0	9.0	1.0
0.0	1225.0	1.0	0.0	0.0	0.0
0.0	289.0	1.0	81.0	9.0	9.0
0.0	16.0	1.0	0.0	0.0	0.0
0.0	4.0	1.0	1.0	0.0	0.0
0.0	25.0	0.0	9.0	9.0	1.0
0.0	1.0	0.0	1.0	0.0	0.0
0.0	729.0	0.0	16.0	1.0	1.0
0.0	196.0	1.0	36.0	1.0	0.0
0.0	361.0	25.0	16.0	1.0	1.0
0.0	64.0	100.0	4.0	1.0	0.0
0.0	0.0	0.0	9.0	0.0	0.0
0.0	0.0	196.0	0.0	0.0	0.0
0.0	1.0	0.0	25.0	0.0	0.0
0.0	9.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0
0.0	4.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	4.0	0.0	0.0
0.0	25.0	0.0	4.0	4.0	1.0

Buttons are as follows:

- **Choose second net** – choosing second net used in comparison process.
- **Compare nets** – choosing this button will start comparison algorithm.
- **Graphlet size –** choosing size/number of graphlets used for comparison
- **Graphlet analysis for single net** -

Multiarc interpretation:

- **Single arc interpretation** – weight arcs will be interpreted as single arc
- **Multiple arc interpretation** – weight arcs will be interpreted as multiple arcss

Table on left represents an GDD matrix before normalization to GDDA value.

10. Changes

Holmes 1.0, 01.01.2017

Holmes 1.1. 20.03.2022

Manual version: 28.03.2022

11 Summary

Correspondence address:

marcin.radom@put.poznan.pl

References

Andrzejewski, H., Chabelski, P. i Szawulak, B. (2013). Zintegrowany system do tworzenia, symulacji i analizy sieci Petriego. Poznań, Polska: Politechnika Poznańska.

Balazki, i. P. i Einloft, J. (2014, December 19). *MonaLisa -- Visualization and analysis of functional modules in biochemical networks.* Pobrano Wrzesień 23, 2015 z lokalizacji http://www.biomedcentral.com/content/supplementary/s12859-015-0596-y-s2.pdf

Ballerstein, K., von Kamp, A., Klamt, S. i Haus, U.-U. (2012). Minimal cut sets in metabolic network are elementary modes in a dual network. *Bioinformatics, 28*(3), strony 381-387.

Caliński, T. i Harabasz, J. (1974). A Dendrite Method for Cluster Analysis. *Communications in Statistics, 3*(1), strony 1-27.

Colom, J. M. i Silva, M. (1991). Convex geometry and semiflows in P/T nets: a comparative study of algorithms for computation of minimal P-semiflows. *Proceedings on Advances in Petri nets* (strony 79-112). New York: Springer-Verlag.

Einloft, J., Ackermann, J., Nothen, J. i Koch, I. (2013). MonaLisa—visualization and analysis of functional modules in biochemical networks. *Bioinformatics, 29*(11), strony 1469-1470.

Formanowicz, D., Sackmann, A., Kozak, A., Błażewicz, J. i Formanowicz, P. (2011). Some aspects of the anemia of chronic disorders modeled and analyzed by petri net based approach. *Bioprocess and Biosystems Engineering, 34*(5), strony 581-95.

Fourier, J. B. (1826). Solution d'une Question Particuliere du Calcul des Inegalites. W *Oeuvres II* (strony 317-328). Paris: Gauthier-Villars.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry, 81*(25), strony 2340-2361.

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics, 115*, str. 1716.

Górecki, T. (2011). Podstawy statystyki z przykładami w R. Wydawnictwo BTC.

Grunwald, S., Speer, A., Ackermann, J. i Koch, I. (2008). Petri net modelling of gene regulation of the Duchenne muscular dystrophy. *BioSystems, 92*, strony 189-205.

Heiner, M., Richter, R. i Schwarick, M. (2008). Snoopy: a tool to design and animate/simulate graph-based formalisms. *Simutools '08 Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*.

Heiner, M., Schwarick, M. i Wegener, J. (2015). Charlie – an extensible Petri net analysis tool. *Proc. PETRI NETS*.

Hofestädt, R. i Thelen, S. (1998). Quantitative modeling of biochemical networks. *In Silico Biology, 1*(1), strony 39-53.

Klamt, S. i Gilles, E. D. (2004). Minimal cut sets in biochemical reaction networks. *Bioinformatics, 20*(2), strony 226-234.

Law, C.-F., Gwee, B.-H. i Chang, J. S. (2006). Optimized Algorithm for Computing Invariants of Ordinary Petri Nets. *Computer and Information Science, 5th IERR/ACIS International Conference.*

Law, C.-F., Gwee, B.-H. i Chang, J. S. (2007). Fast and memory-efficient invariant computation of ordinary Petri nets. *Computers & Digital Techniques, 1*(5), strony 612-624.

Marinescu, D. C., Beaven, M. i Stansifer, R. (1991). *A Parallel Algorithm for Computing Invariants of Petri Net Models.* Department of Computer Science, Purdue University.

Motzkin, T. (1936). *Beitrage zur Theorie der Linearen Ungleichungen, Ph.D. Thesis.* University of Zurich.

Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE, 77*(4), strony 541-580.

Petri, C. A. (1962). Kommunikation mit automaten. Bonn: Institut fur Instrumentelle Mathematik.

Popova-Zeugmann, L. (2013). *Time and Petri Nets.* Springer.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, strony 53–65.

Sackmann, A., Heiner, M. i Koch, I. (2006). Application of Petri net based analysis techniques to signal transduction pathways. *BMC Bioinformatics, 7*(482).

Sokal, R. i Michener, C. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin, 38*, 1409–1438.

Starke, P. (1992). *Integrated Net Analyzer.* Pobrano 4 17, 2015 z lokalizacji http://www2.informatik.hu-berlin.de/~starke/ina.html

Szawulak, B. (2014). Zintegrowany system do tworzenia, symulacji i analizy czasowych sieci sieci Petriego. Poznań: Politechnika Poznańska.

Valk, R. (1978). Self-Modifying Nets, a Natural Extension of Petri Nets. *Proceedings of the Fifth Colloquium on Automata, Languages and Programming*, strony 464-476.

Żurawski, K. (2014). Przygotowanie i analiza sieci Petriego dla procesu naprawdy DNA Base Excision Repair. *(promotor: dr inż. Marcin Radom)*. Polska: Wydział Informatyki Politechniki Poznańskiej.