1. rxncon

The rxncon framework requires Python 3.5 or 3.6. Make sure you have one of these Python versions installed. Anaconda (https://www.continuum.io/downloads) provides an easy way to install the most current Python version. With Python installed and up to date, you are ready to install rxncon:

Under Windows:

1. Open the console and type “pip install rxncon”.[1] The default installation folder will depend on your Python installation. With Anaconda, the rxncon folder appears in [user]/Anaconda3/lib/Site-packages. The files you will need to call appears in [user]/Anaconda3/Scripts.
2. To test the installation, navigate the console to the folder with the scripts and type “python rxncon2bngl.py”.[2] Expect a string “Usage: rxncon2bngl.py [OPTIONS] EXCEL_FILE” and an error message “Error: Missing argument “excel_file”.

Under OS X:

1. Open the console and type ”pip install rxncon”. The default installation folder will depend on your Python installation. With Anaconda, the rxncon folder appears in [user]/Anaconda3/lib/python3.6/Site-packages. The files you will need to call appears in [user]/Anaconda3/bin.
2. To test the installation, navigate the console to the folder with the scripts and type “python rxncon2bngl.py”.[3] Expect a string “Usage: rxncon2bngl.py [OPTIONS] EXCEL_FILE” and an error message “Error: Missing argument “excel_file”.

Under Linux:

1. Make sure you have PIP installed. If not, use your package manager to install it. E.g., on debian-based systems type “sudo apt install python3-pip”.
2. Open a terminal and type “pip3 install rxncon --user”. This installs into $HOME/.local, the executables are in $HOME/.local/bin.
3. To get easy access to the rxncon scripts, you can update your PATH environment variable to include this directory: put something like “export PATH=$HOME/.local/bin:$PATH” into your .bashrc.
4. To test the installation, type “rxncon2bngl.py”.[4] Expect a string “Usage: rxncon2bngl.py [OPTIONS] EXCEL_FILE” and an error message “Error: Missing argument “excel_file”.

2. Visualisation tools

For model visualisation, we will use Cytoscape, which can be downloaded from cytoscape.org.

3. Boolean simulation tools

---

[1] On some computers, the installation of Pyeda fails for unknown reasons. In this case, try typing “pip install --no-cache-dir rxncon”.

[2] In this case, by typing ”cd Anaconda3/Scripts”.

[3] In this case, by typing ”cd Anaconda3/bin”.

[4] This only works with the path set. Without this, type “=$HOME/.local/bin/rxncon2bngl.py”.

The logical simulation of rxncon networks uses BoolNet, an R package. To use these tools:

1. (Optional) Download and install R-studio (https://www.rstudio.com).[5]
2. Make sure you have R installed. R can be installed through Anaconda, by opening the console and typing: “conda install –c r r-essentials”.[6]
3. The BoolNet package can be installed from R. In the console, type “R” to enter the R environment. Then type “install.packages("BoolNet")” and select the download server.

4. Rule based simulation tools

The rule based model export uses the BioNetGen language. To simulate these models, we need BioNetGen and NFsim. To set these up:

1. Make sure PERL is installed.[7]
2. Download NFsim from http://michaelsneddon.net/nfsim/download/, and extract the content to a suitable folder. This includes a binary for Windows , Mac and Linux.
3. To test the installation, open the console and navigate to the NFsim folder. Type “perl BNGL.pl –v”. Expect “BioNetGen version 2.2”.

[5] Depending on installation method, this may or may not come with R. If not, the location of R must be set in the dialogue box. If R is installed from Anaconda, the windows path would be “[user]/Anaconda3/R”.
[6] If the conda command is not recognised, try to close and reopen the console.
[7] For example by typing ”perl -v”.