SQUAD simulation results analysis using the SQUADD package

Martial Sankar

May 3, 2016

1 SQUADD: the SQUAD add-on

Logical modeling is increasingly popular method in system biology to explain and predict the behavior of a complex process (differentiation, cell growth, division etc...) through their regulatory and signalling mechanisms. The SQUAD software (Standardized Qualitative Dynamical Systems) is dedicated to the analysis of Boolean logical models [**?**]. It was the first tool to combine the traditional Boolean approach and a continuous system. This novel feature permits the dynamical analysis of regulatory network in the absence of kinetic data [**?**], [**?**], [**?**]. In order to improve this approach, we provide the R package SQUADD, the SQUAD add-on. This extension permits: (i) to simplify the visualization of the initial SQUAD simulation output, (ii) to generate qualitative predictions of the component change between two conditions and (iii) to assess the coherence of a network model using PCA analysis.

2 Example of SQUAD Simulation

1. SQUAD software can be downloaded at `http://www.enfin.org/wiki/doku.php?id=enfin:squad:start`.
2. Install and open it.
3. Create a network model or use the xml file, containing a sample model *complete_model_511.xml* for the following steps. Load it in SQUAD.
4. To visualize the network, click on the View tab.
5. Then Advance, then Perturbator.
6. Click on Edit Perturbator to set the pertubation on your model or load the provided pertubation file, *pert_samples_m511.prt* if you use the sample model.
7. Click Initialize, then click run to run simulations with perturbations.
8. When the simulation window appears, press Ctrl+T to open the result table and save it (Ctrl+S) into a folder. When naming files, **add an index number in the file name** (i.e. *1_lof.txt*, *2_rescue.txt* etc...). This folder

should contain six files, one for each perturbated model simulation (for the sample network, folder and files can be found in the zip file, *simRes.zip*, or in the data folder of the SQUADD package).

3 Example of SQUADD analysis

```
> library(SQUADD)
```

The package was built using S4 classes and methods. Before launching a function, the user should call the constructor to construct an instance of the Class *SquadSimResServiceImpl*.

The constructor is: *simResService()*. Do not forget to use the *help()* function to obtain R help.

To obtain a table of fitted values, construct the object of Class *SquadSimResServiceImpl*:

```
> # construct instance of SquadSimResServiceImpl
> fpath <- system.file("extdata", package="SQUADD")
> folder <- file.path(fpath,"data_IAA")
> sim <- simResService (folder=folder,
+                           time= 45,
+                           ncolor=5,
+                         legend= c("ARF(a)", "ARF(i)", "AR_Genes", "Aux/IAA",
+                         "BES1/BZR1","BIN2","BR","BRI1BAK1","BRR_Genes","BRX","BR_Biosynt
+                           method="lowess")
```

3.1 Display the simulation matrix

The *plotSimMatrix* method permits to analyse the individual node's behavior during each perturbated condition. The shape of the response curve reflects the network topologies and the initial SQUAD parameters (Figure 1). The user can select the node to analyse by filling the *selectNode* attribute of the object.

```
> sim["selectNode"] <-c("DO","IAA_Biosynthesis","BR_Biosynthesis", "IAA", "BR")
> plotSimMatrix(sim)
```

```
/private/tmp/RtmpCaR5DG/Rinst9af1f07564e/SQUADD/extdata/data_IAA/brx_1_normal.txt
/private/tmp/RtmpCaR5DG/Rinst9af1f07564e/SQUADD/extdata/data_IAA/brx_2_brxlof.txt
/private/tmp/RtmpCaR5DG/Rinst9af1f07564e/SQUADD/extdata/data_IAA/brx_3_brrescue.txt
/private/tmp/RtmpCaR5DG/Rinst9af1f07564e/SQUADD/extdata/data_IAA/brx_4_brxarfi.txt
/private/tmp/RtmpCaR5DG/Rinst9af1f07564e/SQUADD/extdata/data_IAA/brx_5_brxarfiRescue.txt
/private/tmp/RtmpCaR5DG/Rinst9af1f07564e/SQUADD/extdata/data_IAA/brx_6_gain.txt
```

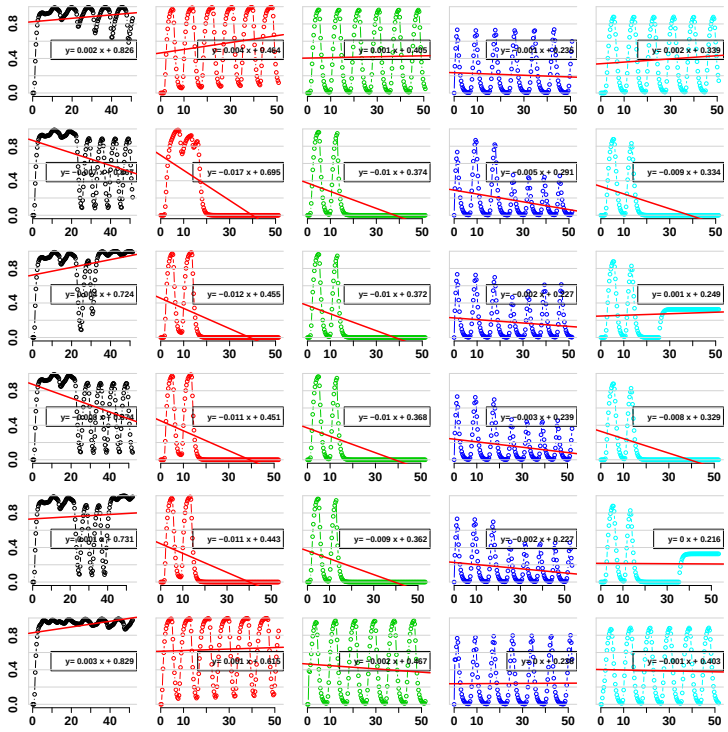

Figure 1: Simulation matrix for the selected five nodes, DO, IAA Biosynthesis, BR Biosynthesis, IAA and BR (respectively in black, red, green, blue, cyan). The columns represent the nodes, the rows represent the conditions. Each cell shows the specific response curve for the corresponding conditions. The red line is the least-square fitted line. It indicates the tendancy of the response.

3.2 Obtain the matrix of fitted values

The *getFittedTable* method permits to obtain a matrix of the interpolated value of the response curve at a user-defined time point. The interpolation can be done either linearly or locally (lowess). Users can choose one of them by filling the *method* attribut of an object of Class *SquadSimResServiceImpl*. In this example, the values were interpolated using the lowess method at t=45 (see section **??**).

```
> tab <- getFittedTable(sim)
```

3.3 Display the prediction heatmap

Biological experiments generate more and more quantitative data. However, classical logical models are prone to reflect the qualitative nature of a system, but fail to predict the outcome of biological experiments that yield quantitative data. SQUAD was the first software to assess this issue. It was built around a standardized method [**?**] that translate a Boolean model into a continuous one. The SQUADD package through the *plotPredMap* method takes full advantage of this novel feature. It generates a heatmap of predictions of the component's activation state change between two experiments. A ratio is therefore calculated between the interpolated values of the current condition (numerator) and a reference condition (denominator). The latter is chosen by setting the *indexDeno* attribut. In this example we chose *indexDeno*=1 (see section **??**), which corresponds to the index of the first result file *brx_1_normal.txt* (Figure 2).

```
> plotPredMap(sim)
```

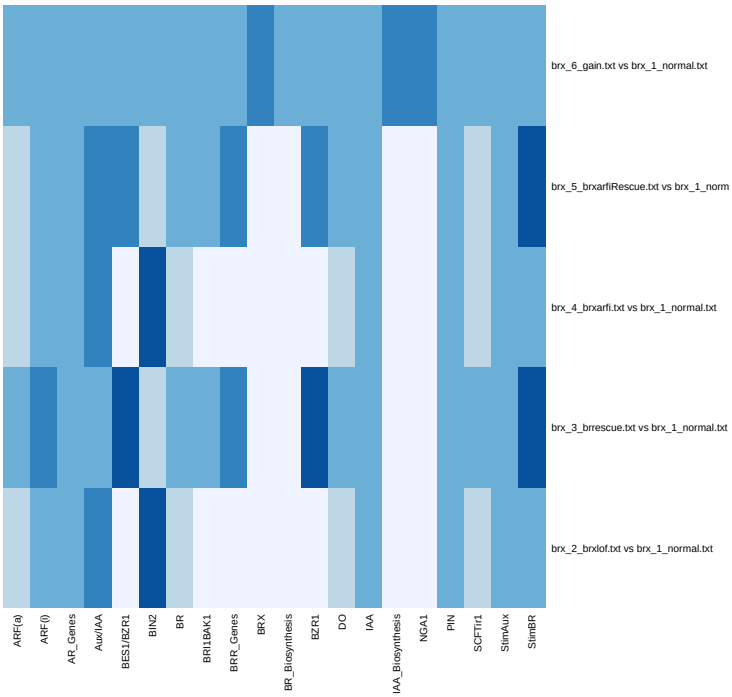

Figure 2: Prediction heatmap. The rows represent the ratios between conditions, the columns represent the network components. The blue intensities (from white to dark blue) show the activation patterns (respectively, from under-activated to over-activated).

3.4 Display the correlation circle

We suggest to use a principal component analysis (PCA) to assess the coherance of the prediction supplied by one model. PCA can be visualized as a correlation circle, which can be displayed by applying the *plotCC* method of the Class *SquadSimResServiceImpl* (Figure 3).

```
> # Fill the field conditionList of the object sim
> sim["conditionList"] <- c("Normal", "brxlof", "BRrescue","brxarfilof","brxarfiBRrescue",
> plotCC(sim)
```

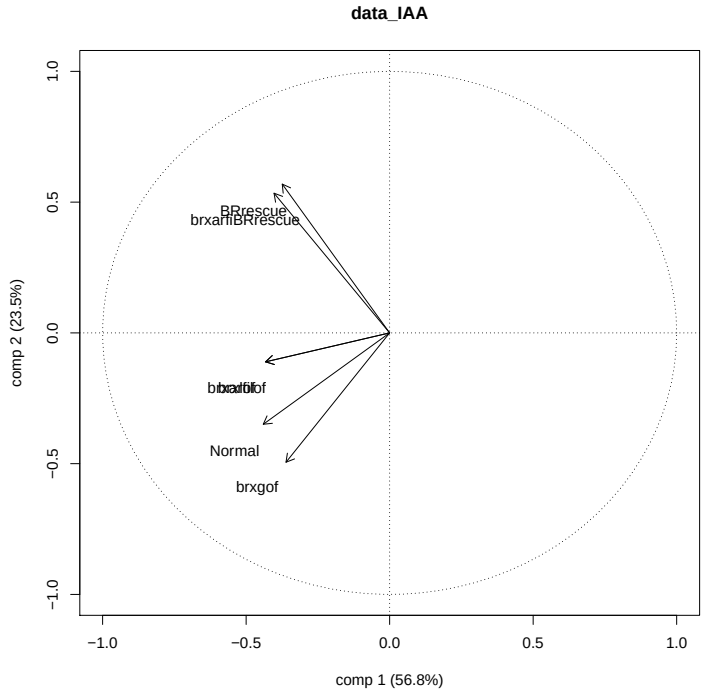

Figure 3: Correlation circle. The vectors represent the variables. The variables correspond to the perturbated conditions (Normal (wildtype), brxlof (brx loss of function), BRrescue (BR rescue the brx loss of function phenotype), brxarfilof (double loss of function mutant brx arfi), brxarfiRescue (BR rescue the double mutant phenotype), brxgain (BRX gain of function)). The first two PCs explained 80.3% of the variation. The angle between the vectors indicate the correlation between the variables. The closer two vectors are, the better the correlation is. In this figure, the angle between the wild type condition and the the BRX loss of function conditions (brxlof and brxarfilof) are very close meaning that the correlation exists between the two variables. This is contradictory with biological findings , where loss of function mutants display shorter root [**?**]. Moreover, the angle between the normal and the rescue conditions is more than 90 degrees, meaning no correlation. These results demonstrate that the model coherence is weak, meaning that the model still needs to be refined (change interaction sign, remove or add nodes etc. . .).

4 Session info

```
> sessionInfo()

R version 3.3.0 RC (2016-04-26 r70550)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] SQUADD_1.22.0

loaded via a namespace (and not attached):
[1] tools_3.3.0        RColorBrewer_1.1-2
```

5 Perspectives

* better access to the plot function parameters
* Improve plotCC: Title to define, label of vector to localize
* Improve the simulation matrix: set row names (conditions) and column names (selected nodes).

6 Reference