How to use the *BiGGR* package

Anand K. Gavai

January 31, 2013

1 Introduction

The main purpose of this package is to analyze metabolic systems and measure the biochemical reactions in metabolic networks in order to calculate the rates of these reactions. The BiGGR package provides an interface for the BiGG database which stores reconstructions of metabolic networks available on http://bigg.ucsd.edu/bigg/main.pl. The users can access this database by requesting a username and a password from the maintainer. The BiGG system provides metabolic reconstructions on Humans, M. barkeri, S. cerevisiae, H. pylori,E. coli and S. aureus. These reconstruction consists of genes, metabolites, reactions, proteins that are identified and connected with each other to form a network structure. This package provides various functions interfacing to the BiGG database, and performing flux balance analysis (FBA) by importing selected reactions and pathways. Other key functions included in this package allows users to create metabolic models for computation, linear optimization routines, likelihood distribution of possible flux distributions based upon limited amount of measurement data. Finally, the package allows visualization using hypergraph framework.

2 Installation

The following sections describe the installation process of BiGGR on various platforms.

2.1 Linux and Mac OS X

The package BiGGR depends on an independent installation of rsbml package which depends on *LibSBML* (use *LibSBML* version 3.4.1) and *pkg-config* (must be available on Linux distribution). If *pkg-config* is installed in a non-standard directory, set the two environment variables PKG_CONFIG_PATH and PKG_CONFIG. If *LibSBML* is installed in a nonstandard directory, set the environment variables LIBSBML3_CFLAGS and LIBSBML3_LIBS, for example, if *LibSBML* is installed in your home directory:

```
$ export LIBSBML3_CFLAGS="-I$HOME/include"
$ export LIBSBML3_LIBS="-L$HOME/lib -lxml2 -lz -lm -lbz2 -lsbml -lstdc++"
```

For installation of `rsbml` start `R` and run

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("rsbml",depend=TRUE,type="source")
```

2.2 Windows

In general `BiGGR` works without any additional effort on Windows platform with the precompiled version of rsbml available. To install this version,start `R` and run

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("rsbml")
```

By default the package *rsbml* is installed in

```
C:\Program Files\R\R-x.xx\library
```

2.2.1 Rgraphviz

The fastest way to troubleshoot Rgraphviz configuration is to first install the `Rgraphviz` in a regular way such as

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("Rgraphviz",depend=TRUE)
```

then load the library in the way

```
> library("Rgraphviz")
```

the resulting error will show which Graphviz version is compatible with this package, this can be found at http://www.graphviz.org/pub/graphviz/stable/"OPERATING SYSTEM". After installation of this Graphviz version. The `PATH` settings can then be adjusted by going to Start->Control Panel->System->Advanced system settings->Environment Variables->System Variables, and editing the PATH settings for the Graphviz bin directory.

3 Getting Started

`BiGGR` provides various functionalities. Some of the most common features used in the simulation process are described in this section. The general framework when using this package consists of the following steps:

- Download metabolites and reactions from the site of the BiGG database for a specific organism and compartment.

- Create a model file from the input files downloaded from above.
- Run simulations to estimate fluxes &
- Finally visualize the results using the *hypergraph* framework.

3.1 Download input files

The input files for model creation can be downloaded in *SBML* format from the `BiGG` website at (http://bigg.ucsd.edu/) by selecting an organism, compartment and pathway of interest (the users of `BiGG` database must have a prior registration). Reactions can be searched and a reaction list can be exported. Similarly metabolite can be searched for specific pathways and metabolite list can be exported. In this way metabolite list and reaction list for pathways of interest can be exported to be used within `BiGGR`. These lists are used within the `createBiggModel` function as input parameters, discussed in the next section. In addition an example of *Glycolysis* is provided in the `data` folder of the package.

3.2 Create Model File

`BiGGR` provides a framework which allows a user to create model files automatically which in reality can be a cumbersome process when creating an *in-silico* model in real time. The function for this in `BiGGR` is `createBiggModel` which takes the reactions, metabolites, maximize or minimize objective function, uptake rates for specific reaction,constraints range of constraints and externals (list of external metabolites) into account to create a model file on the fly. This model file will be created in the working area of `R`. Details on each parameter is available in the help sections of the respective functions of the package. The data structure for this format has been adopted from the work of Soetaert et al[1]. The model file created in this way is called as *Model.lim* is ready for simulations as well as can be edited by the user to make additional changes if necessary.

For example:

```
Metabolites<- "Metabolites.txt"
Reactions<- "Reactions.txt"
maximize<-"R_PYK"   #the reactions to be optimized
equation_var<-"R_HEX1"  #the initial task
equation_value<-1   #start value for the reaction
constraint<-"[0,1000]"  # Range of possible fluxes
externals<-c("glcD","pyr")   # provide externals from the processes
createBiggModel(Metabolites,Reactions,maximize,equation_var,
                equation_value,constraint,externals)
```

The model file created in this way may contain *externals*, these could be manually edited as described in the troubleshooting section below. Functions such as `possibleExternals` and `pruneBiggModel` are provided within the package to identify *externals* and edit the model.For simplicity reasons a manually curated *Glycolysis.LIM* file is provided in the data section is already provided in the examples section of the package that could be used here.

3.3 Running simulations to estimate fluxes

`BiGGR` uses Linear Inverse Models for estimating the fluxes as provided by the `LIM` package from Bioconductor. All the functionality of this package can be used in this framework. The interfacing for some of the key functions provided by `BiGGR` is `getRates` which takes the model file as an input parameter to estimate fluxes which finds the solution where one linear function (i.e. the sum of unknowns) is either minimized (a ”cost” function) or maximized (a ”profit” function). Most of the simulation functionality provided by `BiGGR` is through extensive use of `LIM` package. The *Model.lim* is created in user specified location.

```
    rates<-getRates("Glycolysis.LIM")
```

3.4 Finally visualize the results using *hypergraph* framework.

`BiGGR` provides visualization possibility using `hypergraphs`. The function used for this type of visualization are model2hyperdraw. This function takes parameters like modelFile, uptake (metabolites), a logical value minimal which specifies if the visualization should display external metabolites as well and layout which is a layout scheme as provided by Graphviz (http://www.graphviz.org) as input parameters. The graph produced will be a visual display of metabolites connected with each other using `hyperedges` where the edges represent reaction enzymes. The thickness of the edges represent the flow of these fluxes. The edges are colored blue if the flux estimates are positive else they are red.

```
modelFile<-"Glycolysis.LIM"
uptake<-"glcD"
minimal<-"TRUE"
layout<-"neato" # Layout structure from Graphviz.
levels<-7 # value
#to determine thickness of edges in a graph.
model2hyperdraw(modelFile,uptake,minimal,levels,layout)
```

4 Troubleshooting *BiGGR*

Model building is an iterative process and requires careful selection of parameters and arguments. Though `BiGGR` provides a generic framework and automates model building steps, most packages like this are not immune to bugs and problems which are encountered during this process. Some of the most common problems and solutions are described as follows:

- *Infeasible solution*: This problem is encountered when using the *linp* method form the `LIM` package. This problem occurs when the constrains provided for the model are conflicting with the objective function. The reactions embedded in the model file may be incorrect. The other solution is to check the uptake rates for the *inequalities* which may be incorrect.

- *External* : The *createBiGGModel* & *pruneBiGGModel* functions form `BiGGR` provide generic functionality to create model file. The model file created using this functions takes into account all possible externals from the model. However, identifying externals for an entire model can be contextual and depending on the problem at hand. For this reasons the model file created can as well be hand curated. The *Setup* function from the *LIM* package accepts the model file as a parameter and in case of errors being generated, missing externals can be manually edited in the model file under the externals tag.

  NOTE: The externals added in the externals tag must be unique and the corresponding externals (if present) in the compounds tag must be removed.

References

[1] Karline Soetaert and Dick van Oevelen, *Modeling food web interactions in benthic deep-sea ecosystems*. Oceanography, 22, 2009.