

How to use the *BiGGR* package

Anand K. Gavai

April 11, 2011

1 Introduction

The **BiGGR** package provides an interface for the **BiGG** database available on <http://bigg.ucsd.edu/biggy/main.pl>. The users can access this database by requesting for 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 **BiGG** database, perform 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.

Start R and run

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

3 Getting Started

BiGGR provides various functionality, some of the most common functions 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 *hypergraph* framework.

3.1 Download input files

The input files for model creation can be download for 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 exported to be used for the modeling. These data files are also provided in the `data` section of the package as *Metabolites & Reactions* (these are tab delimited text files) for Glycolysis.

3.2 Create Model File

BiGGR provides a unique 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 created in a working area R is run from. 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. 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
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)
```

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("Model.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<-"Model.lim"
uptake<-"glcD"
minimal<-"TRUE"
layout<-"neato"
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 from the *LIM* package. This problem occurs when the constraints 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 from *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.*