User guide: magnum (v1.0)

Daniel Marbach

October 5, 2015

Table of contents

1. Synopsis

Magnum is a java tool implementing the methods described in the paper:

- **Cell type-specific regulatory circuits reveal variable modular perturbations across complex diseases**. Marbach D, Lamparter D, Quon G, Kellis M, Kutalik Z, Bergmann S. *Submitted.*

Download

- Standalone tool and documentation: http://regulatorycircuits.org
- Source code on GitHub: https://github.com/marbach/magnum

Features

1. **Compute network properties**: diffusion kernels, clustering coefficients, betweenness centrality, shortest path lengths, closeness centrality
2. **Perform network operations**: union
3. **Network-based analysis of GWAS**: connectivity enrichment analysis

Magnum is under active development, more features are currently being added.

Requirements

- Java 5 or later: https://www.java.com/en/download/help/index_installing.xml

Support

- **Computational Biology Group, University of Lausanne, Switzerland**
- Swiss Institute of Bioinformatics
- Broad Institute
- MIT

Quick start

- Open the console, go to the magnum directory, run:
  ```
  java -jar magnum-1.0.jar –help
  ```
- See also the step-by-step tutorial (Section 3).

Contact

- Daniel Marbach
  daniel.marb…@gmail.com (fill in the missing letters)

2. Introduction

Magnum is a java command-line tool. As such, it can be used on any operating system (**Mac OS X**, **Linux**, **Windows**, etc.) where java is available.

Requirements

- Java 5 or later: https://www.java.com/en/download/help/index_installing.xml

Installation

- Download the zip file with the standalone tool from: http://regulatorycircuits.org
- Uncompress the zip file and you're ready to go

Getting started. First, let's test whether you have java installed and which version it is. Open the terminal / console and enter:

- `java -version`

Magnum is compatible with Java 1.5 or later, earlier versions have not been tested.

Next, check if you can run magnum. From the command line, change to the magnum directory and launch magnum with no option or `--help`, which will display the usage information:

- `cd path/to/magnum`
- `java -jar magnum-1.0.jar --help`

Java options. Two relevant java options are:

- `-Xmx<memory>`
  Increase the maximum amount of memory that the java virtual machine can use. E.g., `-Xmx8g` for 8GB maximum memory. The required memory depends on the network size and the functions that are performed. If the maximum memory is set too low, java will abort with the following error:
  ```
  exception in thread "main" java.lang.OutOfMemoryError
  ```
- `-ea`
  Enables assertions, which are debugging tests built into the code. This will slow down performance marginally, but give you extra confidence that everything is correct. You can safely run magnum without this option.

See the java documentation for further details.

Usage. Magnum has three modes: (1) compute network properties, (2) perform network operations, and (3) connectivity enrichment analysis. The mode is selected with `--mode`, for example:

- `java -jar magnum-1.0.jar --mode 3`

Note that in this user guide we assume that the reader is already familiar with the methods described in our paper (Marbach et al., submitted).

The remainder of this user guide is organized as follows. First, we present a step-by-step tutorial. Next, we describe the file formats (Section 4) and the options of magnum in detail. We distinguish between general options (Section 5), which apply to more than one mode, and options that are specific to one of the three modes (Sections 6-8).

3. Step-by-step tutorial

If you haven't done so already, follow the instructions in the previous section to install magnum. Here we give a few examples of how magnum can be run. See Section 4-7 for details on each option or run:

- `java -jar magnum-1.0.jar --help`

For this tutorial, we consider the regulatory network of vascular smooth muscle cells and the GWAS of age-related macular degeneration of neovascular type (see Fig. 6d of the paper). The corresponding input data is included in the directory `tutorial_data` within the magnum directory.

3.1. Connectivity enrichment analysis

First, we evaluate whether genes that are perturbed by disease-associated genetic variants are more densely interconnected in the network than expected (connectivity enrichment analysis). Besides the network, we need gene scores as input, which summarize the GWAS SNP p-values at the level of genes. Gene scores have been computed using Pascal (pathway scoring algorithm). The Pascal tool and documentation are available at:

- http://www2.unil.ch/cbg/index.php?title=Pascal

Given the network and gene score files, the whole connectivity enrichment analysis including computation of random-walk network kernels can be performed using the following command (as a single line):

Warning: you may have to remove line-breaks when copy-pasting commands from the PDF, we recommend to first paste into a plain text editor and verify the command.

- `java -Xmx6g -ea -jar magnum-1.0.jar --mode 3 --permut 10000 --scores tutorial_data/macular_degeneration_neovascular.txt --net tutorial_data/smooth_muscle_cells_-_umbilical_vein.txt.gz`

Runtime is about 10 minutes on a high-end laptop. The enrichment p-value is printed on the console, it is significant ($p=10^{-4}$). Thus, the neovascular type of advanced macular degeneration indeed shows connectivity enrichment in vascular smooth muscle cells (cf. Fig. 6d). The full results are written to two files:

- `macular_degeneration_neovascular--smooth_muscle_cells_-_umbilical_vein_4stepKernel_alpha2.0_weighted.AUC.txt.gz`
- `macular_degeneration_neovascular--smooth_muscle_cells_-_umbilical_vein_4stepKernel_alpha2.0_weighted.txt.gz`

The files contain the AUCs of the actual data and the 10,000 permutations and the enrichment curves, respectively. See the provided R scripts in the directory `R_plots` for an example of how these results can be loaded and visualized.

3.2. Loading settings files

Instead of specifying files and parameters as command-line options, you can also define them in a settings file. This is often more convenient and also ensures that results can always be reproduced using the same settings file. You can manually edit settings files in any text editor or export them from the magnum app. We included an example settings file that runs the same analysis as in the example above. To start the following command (as a single line):

- `java -Xmx6g -ea -jar magnum-1.0.jar --set tutorial_data/example.settings.txt`

3.3. Computing network kernels

You can also use magnum to compute random-walk network kernels (this is done automatically when running the connectivity enrichment analysis). The following command computes and exports the kernel for the smooth muscle cell network:

- `java -Xmx6g -ea -jar magnum_v1.0.jar --mode 1 --pstep --weighted --net tutorial_data/smooth_muscle_cells_-_umbilical_vein.txt.gz`

Runtime is about five minutes on a high-end laptop. The majority of the time is spent writing the output file, which is about 900 MB big (the kernel is a square matrix with 13K rows and columns).

4. File formats

All files used by magnum are text files (.txt). Some input and output files are compressed using gzip (.txt.gz).

Networks can be given either as text files (.txt) or gzipped text files (.txt.gz). The format is automatically detected. Each line specifies one edge. Columns are separated by tabs (tab separated format). Unweighted networks have two columns, weighted networks have three columns:

- Column 1: The first node (directed networks: the regulator)
- Column 2: The second node (directed networks: the target)
- Column 3 (only for weighted networks): the edge weight

Undirected vs. directed. The format is the same for undirected and directed networks. By default, magnum considers networks to be undirected. For directed networks, the option `--directed` has to be specified.

Unweighted vs. weighted. By default, magnum considers networks to be unweighted (even if they have three columns)! For weighted networks, the option `--weighted` has to be specified.

The format of the remaining input files should be self-evident from the examples provided in the directory `tutorial_data`.

5. Options

5.1. General options

Options described in this subsection apply to more than one mode.

`--help`	Display help
`--verbose`	Use verbose output
--mode <int>	Select the mode (REQUIRED): 1 = Compute network properties (diffusion kernels, paths, clustering coefficients) 2 = Perform network operations (union) 3 = Connectivity enrichment analysis
--seed <int>	Random number generator seed (default: 42; current time: -1)
--outdir <dir>	Output directory (default: working directory)

--netdir <dir>	Directory of input networks (default: working directory)
--net <file>	Input network filename
--directed	Input network is directed (default: undirected)
--weighted	Input network is weighted (default: unweighted). Edge weights must be given in the third column (see Sect. 4).
--noself	Remove self-loops from input network
--cutoff <value>	Remove edges with weight < cutoff from input network

5.2. Network properties (mode 1)

Select `--mode 1` in combination with the following options to compute network properties. Multiple options can be selected together to compute different network properties in a single run.

Note: for some properties results depend on whether the input network is directed (`--directed`) and/or weighted (`--weighted`). The default is undirected and unweighted. Properties that allow for directed and/or weighted networks are indicated in the table.

--pstep	P-step random walk kernel (Smola & Kondor, 2003; allows for weighted networks)
--nsteps <int>	Number of steps for p-step random walk kernel (default: 4)
--degree	Node degree (for directed networks, also in and outdegree)
--betweenness	Node betweenness centrality (allows for directed networks)
--clustcoeff	Node clustering coefficient (allows for directed networks)
--shortestpath	Shortest path lengths and closeness centrality

5.3. Network operations (mode 2)

Select `--mode 2` in combination with the following options to perform network operations. Currently the graph union is the only implemented network operation.

--union	Union (max edge weight) of all networks in the network directory (see option: --netdir <dir>)

5.4. Network connectivity enrichment (mode 3)

Select `--mode 3` in combination with the following options to perform network connectivity enrichment analysis as described in the paper.

--genes <file>	Custom gene coordinates
--scores <file>	The GWAS gene scores (REQUIRED)
--cmatrix <file>	The connectivity matrix (e.g., diffusion kernel; REQUIRED)
--excl <file>	Custom list of genes to be excluded
--neighbors <X>	Ignore connectivity between genes with distance < X mega-bases. Default: 1 (mega-base).
--bins <int>	The number of bins for within-degree permutation (default: 100)
--permut <int>	Number of permutations to compute empirical p-values (default: 10,000)
--curve <X>	Compute curves only for the top part of the ranked gene list (reduces runtime). Default: 0.2 (top 20%).