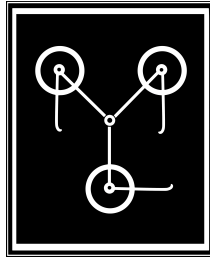


Flux Capacitor Toolkit for Systems Biology



User Manual

Daniel Ortiz Martínez
daniel.ortiz.researcher@gmail.com

November 2017

CONTENTS

1	Introduction	1
1.1	Toolkit Features	1
1.2	Distribution Details	1
1.3	Current Status	2
1.4	Documentation and Support	2
2	Installation	3
2.1	Basic Installation Procedure	3
2.2	Third Party Software	4
2.3	Add Flux Capacitor to the System PATH	5
3	User Guide	7
3.1	Toolkit Overview	7
3.1.1	Main Functionality	7
3.1.2	Bioinformatic Pipeline	8
3.2	Main Tools	8
3.3	Step by Step Pipeline Execution	10
	Bibliography	11

CHAPTER 1

INTRODUCTION

Flux Capacitor is an open source software toolkit for systems biology and flux balance analysis. Flux Capacitor is focused on the application of FBA to study metabolism.

1.1 Toolkit Features

The toolkit includes the following features:

- FBA implementation for metabolic models in SBML format.
- Implementation of tissue specific FBA (Shlomi et al. [2008](#)).
- Flux Variability Analysis (FVA) implementation.
- Techniques to accelerate FVA calculations including parallel execution.
- Automated techniques to generate metabolic network representations.
- Metabolic network reduction methods.

1.2 Distribution Details

Flux Capacitor has been coded using C, C++, Python, R and shell scripting. Flux Capacitor is known to compile on Unix-like and Windows (using Cygwin) systems. As future work we plan to port the code to other platforms. See Section [1.4](#) section of this file if you experience problems during compilation.

It is released under the GNU Lesser General Public License (LGPL)^a.

^a<http://www.gnu.org/copyleft/lgpl.html>

1.3 Current Status

The Flux Capacitor toolkit is under development. Basic usage instructions are being added. In addition to this, there are some toolkit extensions currently in preparation:

- Incorporate interactive Python mode (currently the toolkit offers a command-line interface).
- Enable use of alternative mathematical solvers for FBA and FVA (current version only supports CPLEX).

1.4 Documentation and Support

Project documentation is being developed. Such documentation include:

- Flux Capacitor website^b.
- The Flux Capacitor manual (`flux_capacitor_manual.pdf` under the `doc` directory).
- Daniel Ortiz' MSc thesis in bioinformatics^c: the thesis explains the theoretical foundations of the functionality implemented in the toolkit and uses it to report results.
- MSc thesis slides^d: provide a summary of the above mentioned MSc thesis.

If you need additional help, you can:

- use the github issue tracker^e.
- send an e-mail to the author^f.

^b<http://daormar.github.io/flux-capacitor/>

^chttps://daormar.github.io/flux-capacitor/docsupport/dortiz_bio_msc_thesis.pdf

^dhttps://daormar.github.io/flux-capacitor/docsupport/dortiz_bio_msc_thesis_slides.pdf

^e<https://github.com/daormar/flux-capacitor/issues>

^fdaniel.ortiz.researcher@gmail.com

CHAPTER 2

INSTALLATION

2.1 Basic Installation Procedure

The code of the Flux Capacitor toolkit is hosted on [github](https://github.com/daormar/flux-capacitor)^a. To install Flux Capacitor, first you need to install the autotools (autoconf, autoconf-archive, automake and libtool packages in Ubuntu). If you are planning to use Flux Capacitor on a Windows platform, you also need to install the Cygwin environment^b. Alternatively, Flux Capacitor can also be installed on Mac OS X systems using MacPorts^c.

On the other hand, Flux Capacitor can be combined with third party software so as to enable extended functionality, see more information in Section [2.2](#).

Once the autotools are available (as well as other required software such as Cygwin, MacPorts), the user can proceed with the installation of Flux Capacitor by following the next sequence of steps:

1. Obtain the package using git:

```
$ git clone https://github.com/daormar/flux-capacitor.git
```

Additionally, Flux Capacitor can be downloaded in a zip file^d.

2. `cd` to the directory containing the package's source code and type `./reconf`.
3. Type `./configure` to configure the package.
4. Type `make` to compile the package.

^a<https://github.com/daormar/flux-capacitor/>

^b<https://www.cygwin.com/>

^c<https://www.macports.org/>

^d<https://github.com/daormar/flux-capacitor/archive/master.zip>

5. Type `make install` to install the programs and any data files and documentation.
6. You can remove the program binaries and object files from the source code directory by typing `make clean`.

By default the files are installed under the `/usr/local` directory (or similar, depending of the OS you use); however, since Step 5 requires root privileges, another directory can be specified during Step 3 by typing:

```
$ configure --prefix=<absolute-installation-path>
```

For example, if `user1` wants to install the Flux Capacitor package in the directory `/home/user1/flux-capacitor`, the sequence of commands to execute should be the following:

```
$ make clean # This is recommended if the package has already been built
$ ./reconf
$ configure --prefix=/home/user1/flux-capacitor
$ make
$ make install
```

The installation process also creates three directories with additional information:

- **`${PREFIX}/share/flux-capacitor/cfg_templates`**: contains configuration files to be used with different Flux Capacitor utilities (see Chapter 3 for more details).
- **`${PREFIX}/share/flux-capacitor/doc`**: contains the documentation of Flux Capacitor, which currently consists in the Flux Capacitor manual (`flux_capacitor_manual.pdf`).

IMPORTANT NOTE: if Flux Capacitor is being installed in a PBS cluster (a cluster providing `qsub` and other related tools), it is important that the `configure` script is executed in the main cluster node, so as to properly detect the cluster configuration (do not execute it in an interactive session).

2.2 Third Party Software

Flux Capacitor internally uses CPLEX^e as a mathematical solver to obtain the solutions required by FBA and FVA procedures. Therefore, users also need to install this package to be able to access most of the functionality of the toolkit.

^e<https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

2.3 Add Flux Capacitor to the System PATH

To end the installation process, it might be useful to add Flux Capacitor to the system PATH. This will allow us to easily execute commands provided in the package without the necessity of providing the whole Flux Capacitor installation path.

For this purpose, we can execute the following commands:

```
$ FCAP_HOME_DIR=<absolute-installation-path>
$ export PATH=$PATH:${FCAP_HOME_DIR}/bin
```

These variable definitions can be added to the `.bashrc` user profile file, so as to define them automatically whenever a new interactive shell session is started.

CHAPTER 3

USER GUIDE

This chapter provides usage information for the Flux Capacitor toolkit. Chapter content is organized as follows: first, a toolkit overview is given in Section 3.1, including the provided functionality as well as a general vision of the implemented bioinformatic pipeline. Next, the main software tools implementing the toolkit functionality are explained in Section 3.2. Finally, a detailed description of how to execute the bioinformatic pipeline provided by the toolkit is given in Section 3.3.

To better understand the technical background of the toolkit, it is highly recommended that the reader consults the work presented in (Ortiz-Martínez 2016).

3.1 Toolkit Overview

In this section the main functionality of Flux Capacitor as well the bioinformatic pipeline it executes are introduced.

3.1.1 Main Functionality

Flux Capacitor is focused on the application of flux balance analysis (FBA) (Fell and Small 1986) techniques to study metabolism. The toolkit is currently under development. Below there is a list of its main functionalities:

- **FBA:** the toolkit implements FBA for metabolic models given in SBML format, maximizing the biomass function and returning the flux values that correspond to the optimal solution.
- **FVA:** a parallel version of the so-called flux variability analysis (FVA) procedure (Mahadevan and Schilling 2003) is included. Flux Capacitor also incorporates the techniques proposed in (Gudmundsson and Thiele 2010) to accelerate the calculations.
- **Tissue-specific FBA:** the package provides an implementation of the tissue-specific FBA procedure proposed by Shlomi et al. 2008. The application of tissue-specific FBA requires the generation of lists of lowly and highly expressed reactions. The

procedure required to obtain such lists depends on whether the gene expression data comes from a microarray experiment or from an RNA-Seq experiment. An overview of the steps required for both procedures is explained in (Ortiz-Martínez 2016). The two procedures are implemented in Flux Capacitor.

- **Statistical testing:** Flux Capacitor allows to apply statistical hypothesis tests for case/-control samples. In particular, the t -test and the Mann-Whitney's U -test can be executed.
- **Network visualization:** the package allows to generate automated graphical representations of metabolic networks in SBML format. For this purpose, the open-source graph visualization tool called Graphviz (Gansner and North 2000) is used.
- **Network reduction:** Flux Capacitor includes an implementation of the *NetworkReducer* algorithm proposed by Erdrich et al. 2015 as well as a fast version of it specifically implemented in this toolkit (Ortiz-Martínez 2016).

3.1.2 Bioinformatic Pipeline

The toolkit functionality introduced in previous section is used to execute a whole bioinformatic pipeline whose purpose is to study cancer metabolism by means of systems biology techniques.

Figure 3.1 shows a diagram representing the different steps that compose the pipeline. For each step the diagram also incorporates information regarding the methods that are applied as well the external software that is required (if any).

3.2 Main Tools

The functionality of Flux Capacitor is provided by means of a set of tools executing modular tasks. Next, we provide a list of the most important of such tools, briefly describing the input parameters they expect as well as their dependencies with other software (if any):

- **extract_sbml_model_info:** extracts information from a metabolic model in SBML format. The program takes as input a file in SBML format and generates a list of text files with varied information (reaction and metabolite names, stoichiometric matrix, etc.). It is implemented in R and requires the `sybilSBML` library.
- **auto_fba:** automates an FBA procedure. The tool receives as input the name of the SBML file containing the metabolic model and the type of optimization to be computed: biomass function or tissue-specific. If tissue-specific FBA is to be applied, then the program requires transcriptomic information, that can be provided as a set of CEL files for microarray data or as a file with RNA-Seq counts. `auto_fba` is implemented as a UNIX shell script.
- **auto_fva:** automates a whole FVA procedure. The program takes as input the prefix of the files in lp format representing the initial FBA problem to be solved (they are

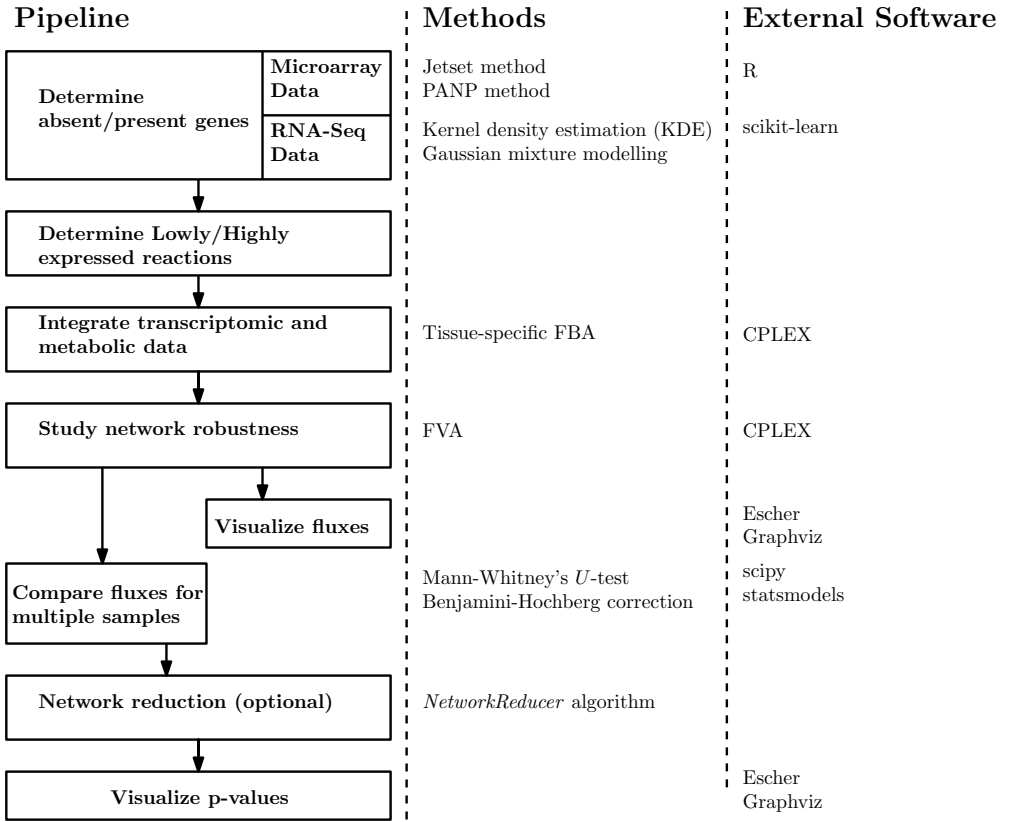


Figure 3.1: Overview of the bioinformatic pipeline implemented by the Flux Capacitor toolkit.

obtained by means of the `auto_fba` tool). In addition to this, `auto_fva` also takes additional parameters to control process efficiency. This tool is implemented as a UNIX shell script.

- **test_samples:** performs statistical tests for a set of samples classified into cases and controls. The tool expects as input a CSV file with the sample data and another one with the phenotype data. `test_samples` is a Python program using the `scipy` and the `statsmodels` modules.
- **correct_pvalues:** corrects a set of p-values using the Benjamini-Hochberg procedure. It receives as input a file with p-values generated by means of `test_samples` and the value of α . The tool is written in Python and uses the `statsmodels` module.
- **plot_metab_network:** generates files in Graphviz format representing metabolic networks. Such files can later be converted to graphics files in different formats. The tool takes as input the plot type to be generated, the prefix of a series of files represent-

ing the metabolic network generated with the `extract_sbml_model_info` tool, a file containing the identifiers of the reactions to be included in the plot, another file with data about the reactions (e.g. flux values, p-values) and optionally, a list of identifiers of external metabolites. `plot_metab_network` is written in Python.

- **network_reducer**: reduces the number of elements of a metabolic network. It is designed to work with the output of the `auto_fba` tool. `network_reducer` is a UNIX shell script.

All of the tools included in the package can display help messages describing their expected input parameters.

3.3 Step by Step Pipeline Execution

CURRENTLY IN PREPARATION

BIBLIOGRAPHY

- Erdrich, P., R. Steuer, and S. Klamt (2015). “An algorithm for the reduction of genome-scale metabolic network models to meaningful core models”. In: *BMC Syst Biol* 9, p. 48.
- Fell, D. A. and J. R. Small (1986). “Fat synthesis in adipose tissue. An examination of stoichiometric constraints”. In: *Biochem. J.* 238.3, pp. 781–786.
- Gansner, E. R. and S. C. North (2000). “An open graph visualization system and its applications to software engineering”. In: *SOFTWARE - PRACTICE AND EXPERIENCE* 30.11, pp. 1203–1233.
- Gudmundsson, S. and I. Thiele (2010). “Computationally efficient flux variability analysis”. In: *BMC Bioinformatics* 11, p. 489.
- Mahadevan, R. and C. H. Schilling (2003). “The effects of alternate optimal solutions in constraint-based genome-scale metabolic models”. In: *Metab. Eng.* 5.4, pp. 264–276.
- Ortiz-Martínez, Daniel (2016). “Systems Biology Strategies to Study Cancer Metabolism”. MA thesis. Universitat de València. URL: https://daormar.github.io/flux-capacitor/docsupport/dortiz_bio_msc_thesis.pdf.
- Shlomi, T., M. N. Cabili, M. J. Herrgård, B. Ø. Palsson, and Eytan Ruppin (2008). “Network-based prediction of human tissue-specific metabolism”. In: *Nat. Biotechnol.* 26(9), pp. 1003–10.