



# Regulon Lib

User Manual

2012 iGEM USTC-Software Team

09.2012

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Introduction to 2012 iGEM USTC-Software Team . . . . .	3
1.2	Introduction to Regulon Lib . . . . .	3
1.2.1	Functions . . . . .	3
1.2.2	Database . . . . .	3
<b>2</b>	<b>Beginning Your First Launch</b>	<b>4</b>
<b>3</b>	<b>Searching For GRN Candidates</b>	<b>5</b>
3.1	Example GRN . . . . .	5
3.2	How To Input . . . . .	5
3.3	How To Generate Results . . . . .	6
<b>4</b>	<b>Searching For Regulons</b>	<b>7</b>
4.1	Launching Search Page . . . . .	7
4.2	Entering Regulator Name . . . . .	8
4.3	Entering Regulatee Name . . . . .	8
4.4	Generating Regulatory Relation . . . . .	9
<b>5</b>	<b>View Source Code</b>	<b>10</b>
<b>6</b>	<b>Copyrights &amp; Acknowledgements</b>	<b>10</b>
6.1	Database . . . . .	10
6.2	Copyrights . . . . .	10
6.3	Acknowledgements . . . . .	10

# 1 Introduction

## 1.1 Introduction to 2012 iGEM USTC-Software Team

The 2012 iGEM USTC-software Team is made up of 9 students from different departments and with different backgrounds. Team members are *Jack Kang, Francis Chen, Jaxyon, Physics Xu, Kaishuai Yang, Qi Hu, Litong Jiang, Yunlong Zheng* and *Wenjuan Zhang*. This summer USTC-Software Team enjoy having fun together and facing challenges together.

Our team's main project is concerned with reverse engineering, which is a hot topic in synthetic biology. Based on this idea, from August, our team began to write this clotho app as an aiding tool.

For more information about 2012 iGEM USTC-Software Team, please search in our wiki page:

<http://2012.igem.org/Team:USTC-Software>

For more information about USTC, please visit this website:

<http://www.ustc.edu.cn>

## 1.2 Introduction to Regulon Lib

### 1.2.1 Functions

*Regulon lib* is an clotho app written by 2012 iGEM USTC-Software Team based on java platform. This app includes following functions:

- Searching for proper regulators and regulatees in databases to complete user's GRN.
- Searching for regulatory relation in databases.

### 1.2.2 Database

In this app, the following databases are used:

- **Operon-Operon Regulatory Matrix.** To be specific, in this matrix regulators and regulatees are operons. The regulatory direction is from the operons in the rows to the operons in corresponding columns. Each element of the matrix is their regulation type, which is a number chosen from 1, 0, -1, 2 or -2.
- **Gene-Promoter Regulatory Matrix.** To be specific, in this matrix regulators are genes and regulatees are promoters. The regulatory direction is from the genes in the rows to the promoter in corresponding columns. Each element of the matrix is their regulation type, which is a number chosen from 1, 0, -1, 2 or -2.

The elements in the matrix have their certain meanings:

- 1: only positive regulation.
- 0: no regulation.
- -1: only negative regulation.
- 2: both positive and negative regulation.
- -2: unknown regulation.

The following sections will tell you more specific usage of **Regulon Lib**.

## 2 Beginning Your First Launch

Our app can be built on all the Java IDEs. In this user manual, I take MyEclipse 10.0 for example. The source code can be downloaded from my github repository.<sup>1</sup> When first launching our app, you'll see this main page:

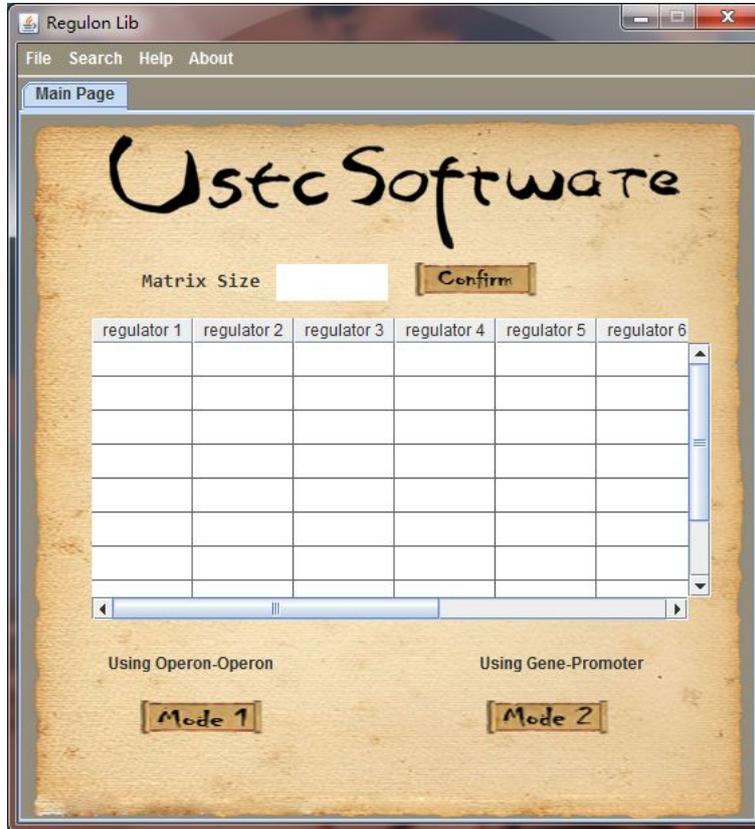


Figure 1: Main Page

There is one text field at the top of the table. Users should first input their matrix size in the text field and then click on button "**Confirm**" to commit. The main table in the middle is for users to input their regulatory matrix. Below the main table, two buttons named "**Mode 1**" and "**Mode 2**" are on your choice. Mode 1 basically uses Operon-Operon database to find the proper results and Mode 2 uses Gene-Promoter database.

---

<sup>1</sup>see "View Source Code" section.

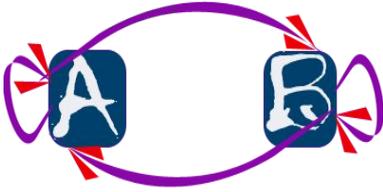
### 3 Searching For GRN Candidates

The function of the main page is for users to search for Genetic Regulatory Network<sup>2</sup>Candidates in the situation that they lack enough information about the nodes in their GRN.

In this section, an simple example will be shown to you about how to use our database to complete your GRN.

#### 3.1 Example GRN

The picture below shows an example regulatory network with abstract nodes A and B. As for their regulatory relations, A activates itself and B, and B both activates itself and A. Then we can figure out A and B's regulatory matrix like this:



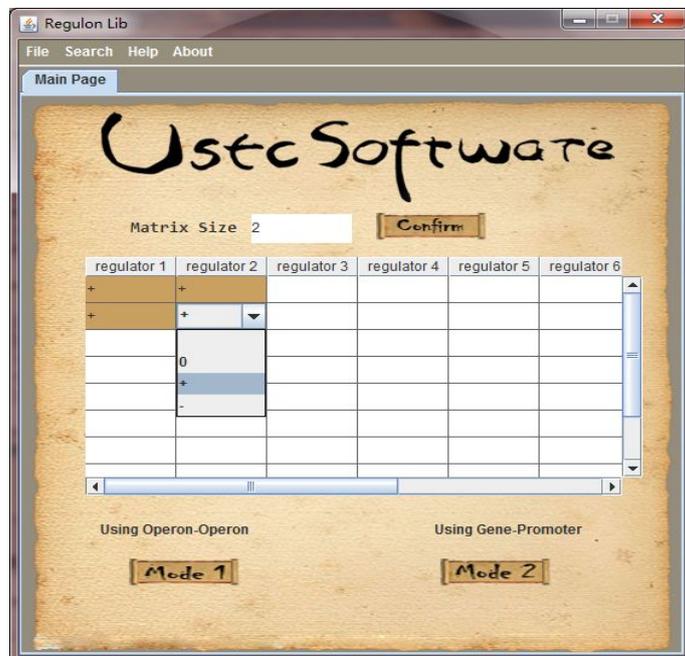
	A	B
A	+	+
B	+	+

#### 3.2 How To Input

Please follow these steps to enter your data:

- (1) Input your matrix size 2 into the text field and press "Confirm".
- (2) A small table with different color will appear within the main table. Input the regulatory matrix in the small table.

The screen shot is pasted on the right side:



<sup>2</sup>Always using "GRN" for short

### 3.3 How To Generate Results

Please follow these steps to generate results:

- (1) You can choose "Mode 1" or "Mode 2" as you like. Mode 1 uses Operon-Operon database and Mode 2 uses Gene-Promoter database.
- (2) If you want to save the results, you can click "File" on the menu bar or click right on the current result tab and choose "save".

Screen shots are pasted below. Picture on the left side is the screen shot for using Mode 1, and on the right side using mode 2. The output of Mode 1 includes *number of choices* and *operons names of each choice*. The output of Mode 2 includes *number of choices*, *promoter names of each choice* and *gene names of each choice*.

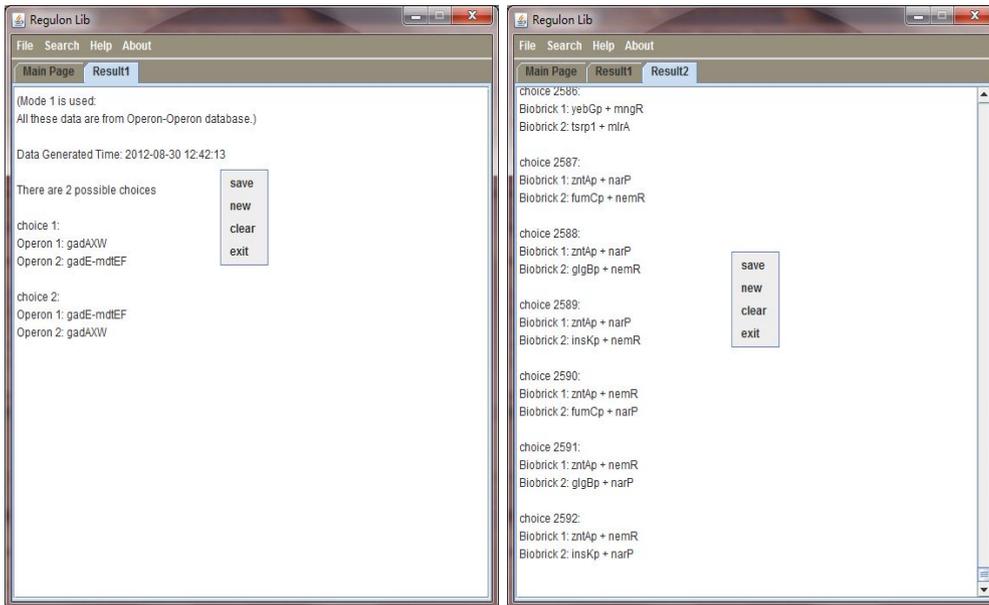


Figure: Using Mode 1

Figure: Using Mode 2

## 4 Searching For Regulons

Function for this page:

- Finding relations between the regulator and regulatee you will use in your experiments.
- Finding proper regulator candidates if you have regulatee but cannot figure out which regulator to use.
- Finding proper regulatee candidates if you have regulator but cannot figure out which regulatee to use.

### 4.1 Launching Search Page

Click "Search" on the menu bar in the main page and choose one search page as you like. In this section I will take "Search In Operon-Operon" for example. The screen shot is pasted below.

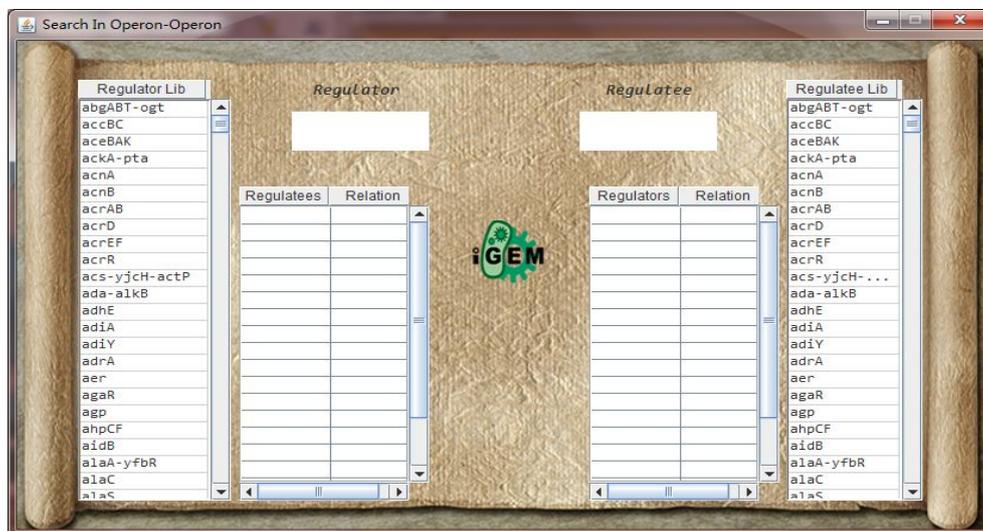


Figure 2: Searching Page

These are the meanings of each components in search page:

- (1) **Regulator:** This text field is for you to enter the regulator name. A Label named "Regulator" is on its above side.
- (2) **Regulatee:** This text field is for you to enter the regulatee name. A Label named "Regulatee" is on its above side.
- (3) **Regulator Table:** The left table of the page. It includes all the regulators in Operon-Operon database.
- (4) **Regulatee Table:** The right table of the page. It includes all the regulatees in Operon-Operon database.
- (5) **Regulator Candidates Table:** The table below the regulatee name text field. If regulatee name is not null, this table will contain all the related regulators of this certain regulatee.

- (6) **Regulatee Candidates Table:** The table below the regulator name text field. If regulatee name is not null, this table will contain all the related regulatees of this certain regulator.
- (7) **Search Button:** The button with an Icon of iGEM logo. If the regulator name and regulatee name are not null, you can press this button to generate regulatory relation between the regulator and the regulatee.

## 4.2 Entering Regulator Name

There are two ways for you to enter the regulator name:

- 1 Enter the regulator name in the regulator name text field.
- 2 Choose your regulator from regulator table or regulator candidates table.

In the demo, I take the second choice, the following picture is what you will get if you click on Operon "*ada-alkB*". The regulatee candidates of "*ada-alkB*" will appear in the regulatee candidates table in the meantime.



Figure 3: Search Regulator

## 4.3 Entering Regulatee Name

There are also two ways for you to enter your regulator name:

- 1 Enter the regulatee name in the regulatee name text field.
- 2 Choose your regulatee from regulatee table or regulatee candidates table.

In the demo, I continue using the example of the above section. You can choose one of the regulatees from the regulatee candidates table of Operon "*ada-alkB*". For example, the Operon "*aidB*" is chosen. Then "*aidB*" will appear in the regulatee name text field and its related regulators will appear in the regulator candidates table in the meantime.



Figure 4: Search Regulatee

#### 4.4 Generating Regulatory Relation

If the regulator name and the regulatee name are not null, press **Search Button** the regulatory relation will appear between two text fields.

In the demo, I still use examples stated above. After clicking on the Icon, a cartoon picture appears indicating positive regulatory relation between "ada-alkB" and "aidB". For other types of regulatory relations, different cartoon pictures will be shown. You can have a try.

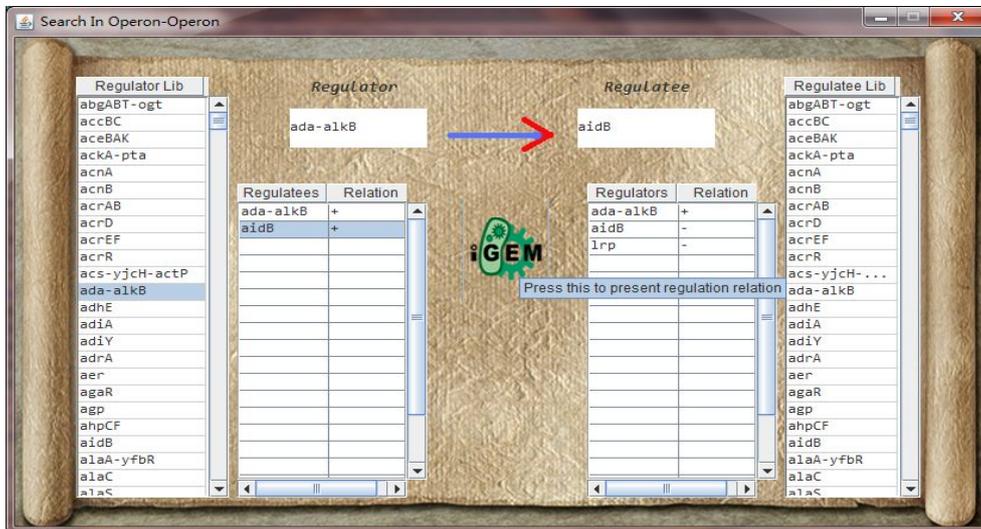


Figure 5: Generate Regulatory Regulation

## 5 View Source Code

As *Regulon Lib* is an open source program, its source code can be viewed on github. You can download all the source codes in this github repository: <http://github.com/kfrancischen/src>

Some of the classes are trivial and should not be included in your project if you want to rebuild *Regulon Lib*. These are some brief introductions for the classes.

- **main\_GUI**: class for the welcome page.
- **Search\_Page**: class for the search page.
- **Operon\_Operon**: class for Mode 1.
- **Gene\_Promoter**: class for Mode 2.
- **LinkLabel**: class extended from JLabel, using for a specific label.

All these classes are packed in "**USTC.Clotho\_App**" package. The private variables are always at the bottom of the codes.

All the source codes and this user manual are written by **Francis Chen**. The images are made by **Physics Xu**. They are members of the 2012 iGEM USTC-Software Team.

If you have problems, please contact me: [kfrancischen@gmail.com](mailto:kfrancischen@gmail.com)

## 6 Copyrights & Acknowledgements

### 6.1 Database

The Operon-Operon database and the Gene-Promoter database are generated based on database from Regulon DB: <http://regulondb.ccg.unam.mx>

If you want to use this database, please read User Licence first.

### 6.2 Copyrights

This clotho app is developed by 2012 iGEM USTC-Software Team. The copyright is reserved to this team.

### 6.3 Acknowledgements

This year our project is sponsored mainly by Life Science Department. We also acknowledge our gratitude to SCGY, School of Physical Science, School of Engineering Science and School of Information Science and Technology.

Personally, I own great thanks to Jenhan Tao, who is a UC Berkeley student and also a clotho developer. He helps us solve a lot of problems about coding and connecting to clotho.