

# WholeCellKB: Pathway/Genome Databases for Comprehensive Whole-Cell Models

Jonathan R. Karr<sup>1</sup>, Jayodita C. Sanghvi<sup>2</sup> & Markus W. Covert<sup>2\*</sup>

<sup>1</sup>Graduate Program in Biophysics, Stanford University, 318 Campus Drive West, Stanford CA 94305, USA and

<sup>2</sup>Department of Bioengineering, Stanford University, 318 Campus Drive West, Stanford CA 94305, USA.

\*To whom correspondence should be addressed. Tel: +1 650 7256615; Fax: +1 650 7211409; Email: mcovert@stanford.edu.

WholeCellKB is a collection of free, open-source model organism databases designed specifically to enable comprehensive, dynamic simulations of entire cells and organisms. WholeCellKB provides comprehensive, quantitative descriptions of individual species including:

- Cellular chemical composition,
- Growth medium composition,
- Gene locations, lengths, and directions,
- Transcription unit organization and transcriptional regulation,
- Macromolecule composition,
- Reaction stoichiometry, kinetics, and catalysis, and
- Extensive links and cross-links to all references used to construct each model organism database.

Please see the tutorial <http://wholecellkb.stanford.edu/tutorial> and the about page <http://wholecellkb.stanford.edu/about> for more information about WholeCellKB how to use WholeCellKB as well as how it was designed, implemented, and curated.

This document provides instructions on how to install and use WholeCellKB. Please contact the authors with any questions; updated contact information is available at [wholecellkb.stanford.edu/about](http://wholecellkb.stanford.edu/about).

# Contents

<b>1</b>	<b>Installation and requirements</b>	<b>1</b>
1.1	Download and unpack WholeCellKB . . . . .	1
1.2	Install required packages . . . . .	1
1.3	Configure required packages . . . . .	2
1.4	Configure WholeCellKB . . . . .	3
1.5	Creating and editing model organism databases . . . . .	4
<b>2</b>	<b>Creating and editing model organism databases</b>	<b>5</b>
<b>3</b>	<b>Customizing the WholeCellKB data model and user interface</b>	<b>6</b>
<b>A</b>	<b>License</b>	<b>7</b>

# Chapter 1

## Installation and requirements

This chapter provides instructions on how to install WholeCellKB on CentOS 5.5.

### 1.1 Download and unpack WholeCellKB

1. Download WholeCellKB software from <http://simtk.org/home/wholecell>
2. Unpack to /path/to/WholeCellKB

### 1.2 Install required packages

1. Install Python (2.7.2), Apache (2.2.3), MySQL (5.0.95), GraphViz
2. Install mod\_wsgi to enable Apache to run Django. Following the instructions at <http://code.google.com/p/modwsgi/wiki/QuickInstallationGuide>.

```
wget http://modwsgi.googlecode.com/files/mod_wsgi-3.3.tar.gz
tar xvfz mod_wsgi-3.3.tar.gz
cd mod_wsgi-3.3/
./configure --with-python=/opt/python2.7.2/bin/python
make
make install
```

3. Install c++ compiler

```
yum install gcc-c++
```

4. Install Xapian

```
curl -O http://oligarchy.co.uk/xapian/1.2.10/xapian-core-1.2.10.tar.gz
curl -O http://oligarchy.co.uk/xapian/1.2.10/xapian-bindings-1.2.10.tar.gz
tar xvzf xapian-core-1.2.10.tar.gz
tar xvzf xapian-bindings-1.2.10.tar.gz
cd xapian-core-1.2.10
./configure
make
make install
cd ..
cd xapian-bindings-1.2.10
./configure --PYTHON=/opt/python2.7.2/bin/python
make
make install
```

5. Install Django and other python packages

```
pip install django django-haystack xapian-haystack django-extensions
pip install xhtml2pdf httplib2 nose pisa hashlib huTools BeautifulSoup docutils epydoc
```

6. Install ChemAxon Marvin

- (a) Download Marvin Beans from <https://www.chemaxon.com/download/marvin/for-end-users/>
- (b) Run installer, install to /usr/share/ChemAxon/MarvinBeans-5.10.4

```
./marvinbeans-5.10.4-linux.sh
```

- (c) Add binaries to system path

```
cd /usr/local/bin
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/MarvinSketch MarvinSketch
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/MarvinSpace MarvinSpace
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/MarvinView MarvinView
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/cxcalc cxcalc
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/cxtrain cxtrain
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/evaluate evaluate
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/molconvert molconvert
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/msketch msketch
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/mspace mspace
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/mview mview
ln -s /usr/share/ChemAxon/MarvinBeans-5.10.4/bin/structurecheck structurecheck
```

## 1.3 Configure required packages

1. Configure Apache to use threading. Uncomment the following line in /etc/sysconfig/httpd

```
HTTPD=/usr/sbin/httpd.worker
```

2. Configure Apache to use mod\_wsgi to serve Python scripts. Add the following lines to your apache configuration

```
LoadModule wsgi_module modules/mod_wsgi.so
```

```
WSGIDaemonProcess django process threads=25
```

```
WSGIProcessGroup django
```

```
WSGISocketPrefix /var/run/wsgi
```

```
Alias /url/to/WholeCellKB/static /path/to/WholeCellKB/static
```

```
<Location "/djangoproject/static">
```

```
Order allow,deny
```

```
Allow from all
```

```
</Location>
```

```
WSGIScriptAlias /url/to/WholeCellKB /path/to/WholeCellKB/apache/django.wsgi
```

```
<Directory /path/to/WholeCellKB>
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

3. If you're also using PHP on your server, setup PHP for threading using FastCGI.

- (a) Install FastCGI

```
yum install libtool httpd-devel apr-devel apr
```

```
cd /opt
```

```
wget http://www.fastcgi.com/dist/mod_fastcgi-current.tar.gz
```

- ```
tar -zxvf mod_fastcgi-current.tar.gz
cd mod_fastcgi-2.4.6/
cp Makefile.AP2 Makefile
make top_dir=/usr/lib64/httpd
make install top_dir=/usr/lib64/httpd
```
- (b) Create file `/etc/httpd/conf.d/mod_fastcgi.conf` with line  
`LoadModule fastcgi_module modules/mod_fastcgi.so`
- (c) Disable old PHP configuration  
`mv /etc/httpd/conf.d/php.conf /etc/httpd/conf.d/php.conf.disable`
- (d) Create FastCGI configuration file `/var/www/cgi-bin/php.fcgi` with contents
- ```
#!/bin/bash
# Shell Script To Run PHP5 using mod_fastcgi under Apache 2.x
# Tested under Red Hat Enterprise Linux / CentOS 5.x
### Set PATH ###
PHP_CGI=/usr/bin/php-cgi
PHP_FCGI_CHILDREN=4
PHP_FCGI_MAX_REQUESTS=1000
### no editing below ###
export PHP_FCGI_CHILDREN
export PHP_FCGI_MAX_REQUESTS
exec $PHP_CGI
```
- (e) Make FastCGI configuration executable  
`chmod +x /var/www/cgi-bin/php.fcgi`
- (f) In `/etc/httpd/conf/httpd.conf` under the `<Directory "/>` tag add lines  
`AddHandler php5-fastcgi .php`  
`Action php5-fastcgi /cgi-bin/php.fcgi`

## 1.4 Configure WholeCellKB

1. Edit WholeCellKB settings. Edit the following variables in the file `/path/to/WholeCellKB/settings.py`
  - `ROOT_URL = '/url/to/WholeCellKB'`
  - `ADMINS`
  - `DATABASES`: edit host, user, password, schema
2. Create MySQL database for WholeCellKB.

```
mysql -h <mysql_host> -u <mysql_user> --password=<mysql_pass> <<MYSQL
CREATE DATABASE <mysql_schema>
DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_unicode_ci
MYSQL
```

3. Install WholeCell MySQL schema

```
cd /path/to/WholeCellKB
./manage.py syncdb
```

4. Initialize full-text search index

```
cd /path/to/WholeCellKB
chown -R apache:apache .
./manage.py rebuild_index
chmod -R ug+w templates/search/
chmod -R ug+w xapian_index/
```

5. Setup cron job to periodically update full text index. Run `crontab -e` and the following line

```
0 */1 * * * cd /path/to/WholeCellKB; ./updateIndex.sh
```

## 1.5 Creating and editing model organism databases

Visit `/url/to/WholeCellKB` to begin viewing and editing a model organism database. See Chapter 2 for more information.

## Chapter 2

# Creating and editing model organism databases

To create your first PDGB:

1. Visit `/url/to/WholeCellKB`
2. Login. (Optionally edit your user profile)
3. Click the "+" button at the bottom right corner to create your first model organism database. Fill out the subsequent form and save.
4. To add data to your new model organism database
  - Web form method:
    - (a) Browse the to the object type
    - (b) Click the "+" button at the bottom right
    - (c) Fill out the web form and save
  - Batch import method:
    - (a) Export an Excel template using the "download" page
    - (b) Edit the Excel template and save
    - (c) Upload the Excel workbook using the "import" page
5. Use the browse and search features to begin viewing your model organism database

## Chapter 3

# Customizing the WholeCellKB data model and user interface

Customizing the WholeCellKB data model and user interface is easy, and requires minimal programming and no knowledge of SQL. The following documentation is very brief. Please contact the authors for further help customizing the data model.

Briefly, the WholeCellKB data model is defined by the Python file `/path/to/WholeCellKB/public/models.py`. The data model contains two kinds of classes: (1) main classes derived from the `Entry` class and (2) supplementary classes derived from the `EntryData` class which represent relationships among entries, encapsulate related data, and represent multiple-valued (array) properties of entries.

Both types of classes support a subset of the Django model syntax (see <http://www.djangoproject.com> for more information). First, the classes only support 9 kinds of fields: `BooleanField`, `CharField`, `FloatField`, `ForeignKey`, `ManyToManyField`, `IntegerField`, `PositiveIntegerField`, and `TextField`. Second, `OneToOneField` properties can only be used to specify inheritance relationships of `Entry` subclasses. Third, `EntryData` subclasses can only define `ForeignKey` relationships to `Entry` subclasses.

To edit an existing class, simply edit the definitions of its properties. To add a table to the data model, simply define a new class in the `models.py` file. To custom how a property is displayed in the detail view add a function with the name `get_as_html_<propname>` which returns an HTML description of how that property should be displayed. To customize the in order in which properties are displayed in each table's detail view edit the `fieldsets` meta property. Similarly, edit the `field_list` meta property to customize the order in which fields are displayed in each table's edit and Excel views. Finally, edit the `facet_fields` meta property to customize the faceted search provided with each table's list view. Edit each class' `clean` and `validate_unique` meta methods to define custom model validation.



# Appendix A

## License

Copyright (c) 2012, Stanford University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.