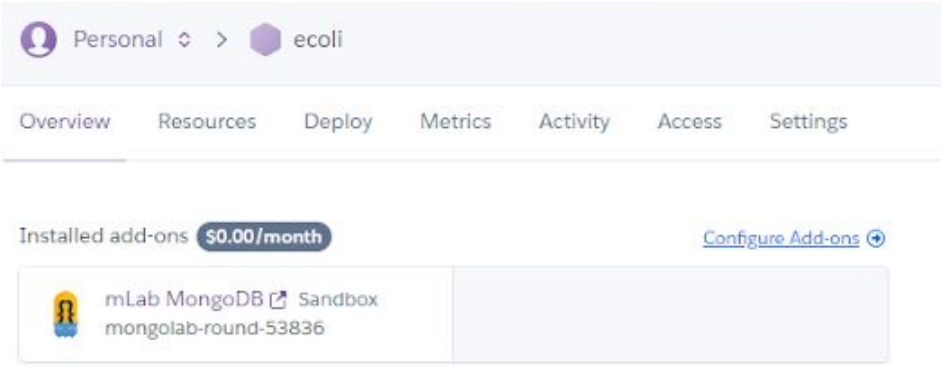
E.coli WholeCell BlueCrystal Setup

Original Setup by ACRC (specifically Dee)

There was an issue we came across while installing Scipy: it was uninstalling numpy 1.14 and reinstalling a newer version that then wasn't operating like it should. OpenBlas also wasn't picking up the gfortran we have installed and needed to set the environment for it to find it.

Onetime setup of MongoDB mLab instance on Heroku

Read Covert Lab's readme for the gist
https://github.com/CovertLab/WholeCellEcoliRelease/blob/master/wholecell/fireworks/README.md
Go to Heroku: heroku.com
Create an account, and add card details to billing (you won't be charged at any point, but you can't create an mLab database without doing this).
Create an ***app***, and call it anything to do with *E.coli*
Go to the mLab Heroku Extension: https://elements.heroku.com/addons/mongolab
Install it to your *E.coli* ***app*.**
Return to https://dashboard.heroku.com/
Click the mLab MongoDB link (see below)



In your mLab database, go to the Users tab, and create and save a new user.
Leave this window open, as we will come back for information stored here later.

Download the *E.coli* whole-cell model

https://github.com/CovertLab/WholeCellEcoliRelease.git

Connect to BlueCrystal using Putty and WinSCP

Use your university username and assigned password (see the BlueCrystal docs on how to change it). Also save your BlueCrystal settings in putty with a keepalive of 5 seconds (see Connection tab)

Preface: The wcEcoli software expects to run with wcEcoli/ as both the current working directory and on the $PYTHONPATH.

Create directories and upload model

```
#Add module:
module add tools/git-2.22.0

#Create directories
mkdir wholecell2
cd wholecell2
mkdir wcEcoli
cd wcEcoli
```

Upload the contents of https://github.com/CovertLab/WholeCellEcoliRelease.git via WinSCP to wcEcoli, but not including the WholeCellEcoliRelease directory folder (effectively renaming WholeCellEcoliRelease to wcEcoli).

Install python tools

```
git clone https://github.com/pyenv/pyenv.git ~/.pyenv
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile
echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n eval "$(pyenv init
-)"\nfi' >> ~/.bash_profile
source ~/.bash_profile

git clone https://github.com/pyenv/pyenv-virtualenv.git $(pyenv
root)/plugins/pyenv-virtualenv
echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bash_profile
git clone https://github.com/pyenv/pyenv-virtualenvwrapper.git $(pyenv
root)/plugins/pyenv-virtualenvwrapper
source ~/.bash_profile
```

Set shell login script to:

- **Add gfortran and git**
- **initialize pyenv and penv virtualenv**

```
vi ~/.bash_profile
#use 'p' to paste copied text, or press 'i' to type out the below

#At the top:
module add languages/gcc-7.1.0
module add tools/git-2.22.0
#At the bottom:
export PYENV_ROOT="$HOME/.pyenv"
```

```
if [ -d "${PYENV_ROOT}" ]; then
    export PATH="${PYENV_ROOT}/bin:${PATH}"
    eval "$(pyenv init -)"
    eval "$(pyenv virtualenv-init -)"
fi
#quit and save: esc then :wq
```

The bash_profile should look like this:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
module add languages/gcc-7.1.0
module add tools/git-2.22.0
PATH=$PATH:$HOME/bin

export PATH
export PYENV_ROOT="$HOME/.pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
if command -v pyenv 1>/dev/null 2>&1; then
 eval "$(pyenv init -)"
fi
eval "$(pyenv virtualenv-init -)"
export PYENV_ROOT="$HOME/.pyenv"
if [ -d "${PYENV_ROOT}" ]; then
    export PATH="${PYENV_ROOT}/bin:${PATH}"
    eval "$(pyenv init -)"
    eval "$(pyenv virtualenv-init -)"
fi
```

Open a new shell so it runs the updated profile.

Install the required version of Python and enable it as a shared library

```
cd wcEcoli
PYTHON_CONFIGURE_OPTS="--enable-shared" pyenv install 2.7.16
```

Create the "wcEcoli-paper" python virtual environment

```
cd wcEcoli
pyenv local 2.7.16
pyenv virtualenv wcEcoli-paper
pyenv local wcEcoli-paper

pip install --upgrade pip setuptools virtualenv virtualenvwrapper
virtualenv-clone wheel
```

Put the project on the PythonPATH
From within /newhome/jr0904/wholecell2/wcEcoli:

```
pyenv virtualenvwrapper
add2virtualenv .
```

OR:

```
echo 'export PYTHONPATH="$HOME/wcEcoli:$PYTHONPATH"' >> ~/.bash_profile
source ~/.bash_profile
```

Install OpenBLAS 0.3.5

```
cd wholecell2
#Add module:
module add tools/git-2.22.0
git clone https://github.com/xianyi/OpenBLAS
cd OpenBLAS
git checkout v0.3.5
```

Build OpenBLAS *(single line from make to -j 8)*

```
make FC=/cm/shared/languages/GCC-7.1.0/bin/gfortran
CC=/cm/shared/languages/GCC-7.1.0/bin/gcc -j 8
```

Building in parallel
GNU Make 3.80 and above, which is necessary to build GCC, supports building in parallel. To activate this, you can use 'make -j 2' instead of 'make'. You can also specify a bigger number, and in most cases using a value greater than the number of processors in your machine will result in fewer and shorter I/O latency hits, thus improving overall throughput; this is especially true for slow drives and network filesystems.

Make OpenBLAS
Change the username below before running

```
make PREFIX=/newhome/jr0904/wholecell2/install install
```

Create .numpy-site.cfg

Change the username below before creating

```
# Create a file
touch ~/.numpy-site.cfg
vi ~/.numpy-site.cfg
#use 'p' to paste copied text, or press 'i' to type out the below

[openblas]
libraries = openblas
library_dirs = /newhome/jr0904/wholecell2/install/lib
include_dirs = /newhome/jr0904/wholecell2/install/include
runtime_library_dirs = /newhome/jr0904/wholecell2/install/lib

#quit and save esc then :wq
```

Install numpy / scipy / requirements.txt

```
cd wholecell2/wcEcoli
pip install scipy==1.0.1 --no-binary all --force-reinstall
pip install numpy==1.14.5 --no-binary all --force-reinstall
CPATH=/usr/include/glpk CVXOPT_BUILD_GLPK=1 pip install -r
requirements.txt --no-binary numpy,scipy,cvxopt
pyenv rehash
```

Test numpy / scipy installation

```
python runscripts/debug/summarize_environment.py
```

Should see a version of this:

lapack_opt_info:
libraries = ['openblas', 'openblas']
library_dirs = ['/usr/local/opt/openblas/lib']
define_macros = [('HAVE_CBLAS', None)]
language = c

If not expected result:

```
vi ~/.numpy-site.cfg
#see appendix for expected result
```

Then:

```
pip uninstall scipy
pip uninstall numpy
# Go back to install numpy/scipy step above
```

Check installations

```
pip list > installed.txt
#See if updated numpy 1.14.5 to 1.14.6
```

Test Theano

```
python
import theano
theano.config.blas.ldflags
#It should print something like
#'-L/usr/local/opt/openblas/lib -lopenblas -lopenblas'
```

Check Matplotlib issue
Issue #161: The matplotlib rendering "backend" might need to be changed from what its installer configures.

After installing or updating matplotlib, test if it can import pyplot:

```
cd wcEcoli/docs/
pyenv
#pyenv should load
python
import matplotlib.pyplot
matplotlib.get_backend()
#check that it returns 'agg'
```

If raises an import error or gives TkAgg instead of agg:
Change the username below before running

```
cd
/newhome/jr0904/.pyenv/versions/wcEcoli-paper/lib/python2.7/site-package
s
#Find the file:
site-packages/matplotlib/mpl-data/matplotlibrc
#Comment out the line backend : TkAgg or backend : macosx
# Then rerun the import pyplot test above
```

Compile the project's native code

```
cd wcEcoli
make clean compile
```

Run tests

```
nosetests
```

If the unit tests fail with an error message saying the loader can't load .../pyenv/versions/.../lib/libpython2.7.a, that means you didn't successfully --enable-shared when installing python. Go back to that step, run PYTHON_CONFIGURE_OPTS= "--enable-shared" pyenv install 2.7.16 --force, and repeat all the steps after it.

If tests complete = You have installed the model correctly!

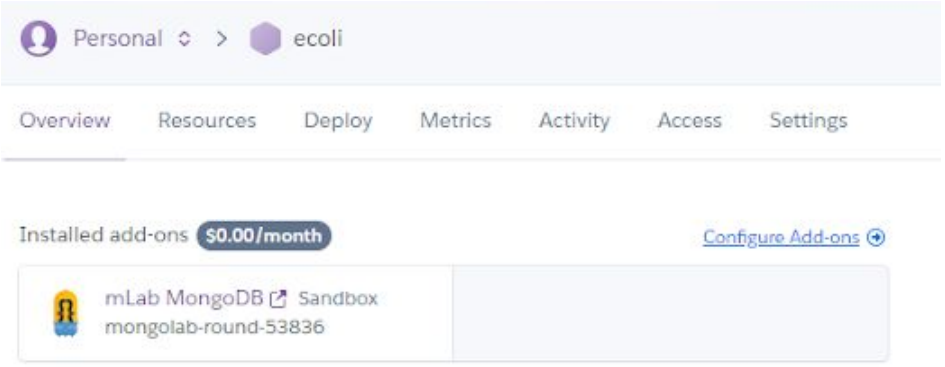
Now to connect the *E.coli* whole-cell model to your mLab MongoDB database!

Download and create your own version of Fireworks files

Using your user information:

1. FireWorksBox1.sh
    a. Paths
    b. Variables
    c. LAUNCHPAD_FILE location
    d. Comment out python path lines
    e. Update sleep value
2. FireWorksDisplay.sh
3. My_fworker.yaml
4. my_qadapter.yaml

Using information from your mLab database page (the window we left open)



5. my_launchpad.yaml

Upload Fireworks files to wholecell2/wcEcoli

Using WinSCP to

```
/newhome/jr0904/wholecell2/wcEcoli/wholecell/fireworks
```

Create directories in /fireworks:

```
mkdir logs
cd logs
mkdir launchpad
mkdir qadapter
```

Initialise Launchpad

In /fireworks:

```
lpad init
```

This recreates my_launchpad.yaml in the new format that Fireworks wants. Copy and paste the data you collected for my_launchpad.yaml into the terminal as you are asked for it.
This previously caused issues as my older format file was working, but apparently not for Jake and Sophie.

Run FireWorksBox1
In /fireworks:

```
chmod u+x FireWorksBox_1.sh
./FireWorksBox_1.sh
```

This will ask if it is okay to rewrite workflows in Launchpad, run a load of code, then submit workflows to the Launchpad, then start a cycle of submitting jobs to BlueCrystal and sleeping between commands.
The model is now running!

Advice for Running the Model
Leave the terminal running the qlaunch command open. You can close it once the simulation finishes.

- If the connection is closed at any point, don't worry. Open a new terminal window, connect to BlueCrystal, navigate to Fireworks folder and run:

```
qlaunch -r -l my_launchpad.yaml -w my_fworker.yaml -q my_qadapter.yaml
rapidfire --nlaunches infinite --sleep 300 --maxjobs_queue 100
```

Open a second terminal window and monitor supercomputer jobs using:
qstat -u [username]
You can also open a third terminal window and use ./FireWorksDisplay to check how the workflows/fireworks are progressing. This script runs the command: lpad webgui.

You will receive a lot of emails while the workflow is running. The best of knowing when the simulations are complete is when the emails stop coming. If any errors are thrown, the emails will display exit code:1 and tell you the location of the error file.

You will see **output from Fireworks** in:
/fireworks/BLOCK_NNNN
/fireworks/logs/…
You will see **model output** in:
wholcell2/wcEcoli/[folder named date-time+description of the simulation in *.sh file]

After a run, to check if any of the fireworks failed and to see how the simulations went, I use:

```
### Check Fizzled (from fireworks folder)
lpad get_fws -s FIZZLED -d all > NAME_fizzled.txt
### Check Launchpad (from fireworks folder)
lpad report -c fws -i months -n 2 > NAME_report_fws.txt
lpad report -c wflows -i months -n 2 > NAME_report_wflows.txt
lpad report -c launches -i months -n 2 > NAME_report_launches.txt
lpad introspect > NAME_introspection.txt
### Rerun Fireworks
lpad rerun_fws -s FIZZLED
qlaunch -r -l my_launchpad.yaml -w my_fworker.yaml -q my_qadapter.yaml
rapidfire --nlaunches infinite --sleep 30 --maxjobs_queue 100
```

Fixing E.coli whole-cell model

1. MLab no longer available, numerous issues with MongoDB Atlas
    a. Host MLab MongoDB addon on Heroku
        i. https://elements.heroku.com/addons/mongolab
    b. Follow https://github.com/CovertLab/WholeCellEcoliRelease/blob/master/wholecell/fireworks/README.md
        i. Ignore advice / script for producing .yaml files
2. Creating PBS queue based my_qadapter.yaml that loads CommonAdaptor PBS template
    a. https://materialsproject.github.io/fireworks/qadapter_programming.html
    b. Do not create custom template
    c. Just use '_fw_q_type: PBS'
3. Missing arg needed passing to fw_queue.py
    a. LAUNCHPAD_FILE='/newhome/jr0904/wholecell/WholeCellEcoliRelease/wholecell/fireworks/my_launchpad.yaml'
        i. Line 71 of /runscripts/fw_queue.py
        ii. Use standard args from /runscripts/paper/paper_runs.sh
        iii. DESC="SET B 9 gens 8 seeds shift to plus AA without growth noise with D period" \ VARIANT="nutrientTimeSeries" FIRST_VARIANT_INDEX=2 LAST_VARIANT_INDEX=2 \ SINGLE_DAUGHTERS=1 N_GENS=9 N_INIT_SIMS=8 \ MASS_DISTRIBUTION=1 GROWTH_RATE_NOISE=0 D_PERIOD_DIVISION=1 \ python runscripts/fw_queue.py
4. Missing args needed passing to qlaunch
    a. -l my_launchpad.yaml
    b. -w my_fworker.yaml
    c. -q my_qadapter.yaml
        i. Use standard command from /runscripts/paper/paper_runs.sh:
        ii. qlaunch -r rapidfire --nlaunches infinite --sleep 5 --maxjobs_queue 100
5. Qlaunch submitting jobs to queue on BlueCrystal, but jobs erroring after a few seconds = Python Path problem
    a. Error code:
        i. ImportError: No module named wholecell.fireworks.firetasks.initRawData
    b. Failed fixes:
        i. __init__.py in WholeCellEcoliRelease
            1. This is an issue with the GitHub release I believe.
        ii. Export directory to Python PATH
            1. python -c "import sys; print('\n'.join(sys.path))"
            2. Export PYTHONPATH='/newhome/jr0904/wholecell/WholeCellEcoliRelease'
    c. Successful fix:
        i. From within /newhome/jr0904/wholecell/wcEcoli:
            1. pyenv virtualenvwrapper

        2. add2virtualenv .
    d. Successful fix References
        i. https://stackoverflow.com/a/17963979
        ii. https://virtualenvwrapper.readthedocs.io/en/latest/command_ref.html?highlight=add2virtualenv
        iii. https://github.com/pyenv/pyenv-virtualenvwrapper

6. Numpy Issues:
    a. Error codes:
        i. ImportError: libgfortran.so.4: cannot open shared object file: No such file or directory
            1. cd /newhome/jr0904/.pyenv/versions/wcEcoli-paper/lib/python2.7/site-packages/numpy/linalg/lapack_lite.so
            2. ldd lapack_lite.so > /newhome/jr0904/lapack_lite.txt
                a. Can see is installed:
                    i. libgfortran.so.4 => /cm/shared/languages/GCC-7.1.0/lib64/libgfortran.so.4 (0x00002aaaaca29000)
        ii. ImportError: cannot import name scimath
        iii. ImportError: Importing the multiarray numpy extension module failed. Most likely you are trying to import a failed build of numpy. If you're working with a numpy git repo, try `git clean -xdf` (removes all files not under version control). Otherwise reinstall numpy.
            1. NumPy was not picking up the version of gfortran installed. This is similar to the OpenBlas issues ACRC were having
    b. Failed fixes:
        i. Reinstall numpy manually
            1. https://hunseblog.wordpress.com/2014/09/15/installing-numpy-and-openblas/
            2. https://leemendelowitz.github.io/blog/installing-numpy-with-openblas.html
            3. python setup.py install --force
        ii. Reinstalling numpy and scipy manually
    c. Successful fix: (from Dee again, via an IT services ticket)
        i. Add to ~/.bash_profile
            1. module add tools/git-2.22.0
            2. module add languages/gcc-7.1.0

Appendix: ~/.numpy-site.cfg

```
multiprocessing 0.70a1
multiprocessing.cpu_count(): 16

numpy 1.14.6
lapack_opt_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
blas_opt_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
openblas_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
blis_info:
  NOT AVAILABLE
openblas_lapack_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
lapack_mkl_info:
  NOT AVAILABLE
blas_mkl_info:
  NOT AVAILABLE

os
os.environ:
  'BOOST_NUMPY_LIB': --
  'HOME': '/newhome/jr0904'
  'LIBRARY_PATH': --
  'PI_HOME': --
  'PYENV_ROOT': '/newhome/jr0904/.pyenv'
  'PYTHONPATH': --
  'SHERLOCK': --
os.getcwd(): /newhome/jr0904/wholecell2/wcEcoli
```

os.uname(): ('Linux', 'newblue2', '2.6.32-642.6.2.el6.x86_64', '#1 SMP Tue Oct 25 15:06:33 CDT 2016', 'x86_64')

scipy 1.0.1
lapack_opt_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
blas_opt_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
openblas_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
blis_info:
  NOT AVAILABLE
openblas_lapack_info:
    libraries = ['openblas', 'openblas']
    library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
    define_macros = [('HAVE_CBLAS', None)]
    language = c
    runtime_library_dirs = ['/newhome/jr0904/wholecell2/install/lib']
lapack_mkl_info:
  NOT AVAILABLE
blas_mkl_info:
  NOT AVAILABLE

sys
sys.platform: linux2
sys.prefix: /newhome/jr0904/.pyenv/versions/wcEcoli-paper
sys.version: 2.7.16 (default, Dec 10 2019, 14:35:24)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-3)]
sys.api_version: 1013
sys.version_info: sys.version_info(major=2, minor=7, micro=16, releaselevel='final', serial=0)