Working with metabolic models in KBase + Jupyter Notebooks

**Notes on this version of instructions**

Written by Kat Beilsmith in January 2020.

These are instructions to set up a Jupyter Notebook in which you can download KBase objects (such as metabolic models) and work with them using python packages (such as cobrapy).

These instructions were written while setting up a KBase and Jupyter Notebook environment on a MacBook Pro running macOS Big Sur 11.0.1 with no previous installations of the required software.

**Prerequisites**

MacPorts will be used to install unix packages on the Mac. Follow the instructions below to install it if needed.

<https://www.macports.org/install.php>

1. **Install XCode and XCode Command Line Tools.** XCode can be installed through the Mac App Store and Command Line Developer Tools can be installed on recent OS versions by running this command in the Terminal: xcode-select –install
2. **Agree to Xcode license in Terminal:** sudo xcodebuild -license
3. **Download the MacPorts package (.pkg) installer** for your version of Mac from the link above. Double click on the .pkg and follow the instructions.

**Set up a KBase workspace**

1. **Make an account.** Go to <https://www.kbase.us/> and click “Sign Up.” To learn how to use Narratives and Apps and how to store data in KBase, read the documentation here: <https://docs.kbase.us/> or go to this site and register for the **January 27** webinar “Introduction to KBase”: <https://www.kbase.us/learn/>
2. **Make a Narrative and upload data.** Copy this public tutorial Narrative: <https://narrative.kbase.us/narrative/18302> and save your copy. Read through the Narrative and run the Apps to generate a metabolic model. Note the name of the model you want to practice with and the number identifying the narrative, which you can find in the link (for example, my copy has ID 81169): <https://narrative.kbase.us/narrative/81169>
3. **Request a developer token.** In order to import data from your narrative into a notebook, you will need to have a token set up for your KBase account. Apply for a token here using the same email address and username that you used to make your KBase account: <https://accounts.kbase.us/index.php?tpl=request_identity.tpl> Enter Chris Henry when prompted for a sponsor. Let me know when you have applied. I will need to reach out to someone with KBase and get token authorization added to your developer account. Once that has happened, you will be able to generate a developer token by logging into your regular KBase account as described in the next step.
4. In the left menu of your KBase Dashboard, go to “Account.” Click on the tab for “Developer Token.” You can generate a token here. The token will be a long alphanumeric string-- make sure you copy it down somewhere. Periodically your token will expire and you can return to this page to generate a new one.

**Install python**

You may already work with python on your computer. You need to upgrade to python 3 if you haven’t already. (For reasons we will get to later, install 3.7.9, which is not the latest version.)

You need to know where python is installed and make sure that you can access it in the terminal while in the directory where you want to work with your notebooks.

To check the location of python, open the terminal and type:

which python

To check the version of python, type:

python

Python 3.7.9 (default, Aug 31 2020, 07:22:35)

[Clang 10.0.0 ] :: Anaconda, Inc. on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>

If you don’t have python installed already, I suggest installing it with conda, available here:

<https://docs.conda.io/projects/conda/en/latest/user-guide/install/>

Under “Regular installation” on this page, follow the links for “macOS” 🡪 “Anaconda installer for macOS” to download the Anaconda installer. Double click the downloaded .pkg file, and follow the instructions.

Activate the environment: conda activate

The file .zshrc should be in your home folder but may be hidden. Use command + shift + . to see the hidden files in Finder.

The contents of the file .zshrc will run every time a new shell is open so the root conda environment (base) will show in the terminal prompt. The contents of the folder should look something like what is shown below:

**(base) kbeilsmith@CSI0360715 ~ %** cat ~/.zshrc

# >>> conda initialize >>>  
# !! Contents within this block are managed by 'conda init' !!  
\_\_conda\_setup="$('/Users/kbeilsmith/opt/anaconda3/bin/conda' 'shell.zsh' 'hook' 2> /dev/null)"  
if [ $? -eq 0 ]; then  
 eval "$\_\_conda\_setup"  
else  
 if [ -f "/Users/kbeilsmith/opt/anaconda3/etc/profile.d/conda.sh" ]; then  
 . "/Users/kbeilsmith/opt/anaconda3/etc/profile.d/conda.sh"  
 else  
 export PATH="/Users/kbeilsmith/opt/anaconda3/bin:$PATH"  
 fi  
fi  
unset \_\_conda\_setup  
# <<< conda initialize <<<

Check the location of the python installation:

**(base) kbeilsmith@CSI0360715 ~ %** which python

/Users/kbeilsmith/opt/anaconda3/bin/python

**Install python packages**

The necessary python modules can be installed with pip. With python > 3.4, pip should already be installed. Check with: python -m pip --version

Use pip to install the following modules. Later, installation for cobra may throw and error if you need a specific version (for example, version 5.12.3 of pyqt5). Use the —ignore option for cobra because otherwise it will attempt to uninstall and reinstall some dependencies and the installation will not finish.

pip install configobj

pip install optlang

pip install pyqt5==5.12.3

pip install pyqtwebengine==5.12.1

pip install cobra --ignore-installed

**Install cobrakbase**

Git should be installed with XCode. If not, enter this command and there should be a prompt to install it:

git --version

Then, use git to install cobrakbase:

git clone https://github.com/Fxe/cobrakbase

cd cobrakbase

pip install . --ignore-installed

Here is an error message that may be encountered along the way, but doesn’t cause any issues with the installation:

ERROR: After October 2020 you may experience errors when installing or updating packages. This is because pip will change the way that it resolves dependency conflicts.

We recommend you use --use-feature=2020-resolver to test your packages with the new resolver before it becomes the default.

**Install CPLEX**

Performing flux balance analysis (fba) with a metabolic model will require a solver for linear programming problems. This is how to obtain and install a solver called CPLEX.

Apply for a free IBM academic license here:

<https://www.ibm.com/products/ilog-cplex-optimization-studio/pricing>

Then, navigate to the page below and download the latest version of CPLEX.

IBM Academic Initiative 🡪 Data Science 🡪 Software 🡪 ILOG CPLEX Optimization Studio 🡪 Download

<https://content-eu-7.content-cms.com/b73a5759-c6a6-4033-ab6b-d9d4f9a6d65b/dxsites/151914d1-03d2-48fe-97d9-d21166848e65/academic/technology/data-science>

Use the HTTP download and check the box for this version: IBM ILOG CPLEX Optimization Studio V20.10 for OSX (CC8AVML ).

Unzip the download folder and double click on the Installer to proceed.

During installation, CPLEX will give this prompt:

If you want to use the CPLEX or CP Optimizer engines through their Python APIs, you need to tell Python where to find them.

python /Applications/CPLEX\_Studio1210/python/setup.py install

The latest version of CPLEX may not work with the latest version of python.

Check the python versions that your installed CPLEX works with here after installation:

/Applications/CPLEX\_Studio1210/cplex/python

For example, this directory for CPLEX 12\_10 has folders for python 3.6 and 3.7 within, so it will not work with python 3.8. You may have to downgrade python. And if you downgrade python, you have to repeat the installation of CPLEX. To downgrade python:

conda install python=3.7.9

Test to make sure you can load the module.

python

>>> import cplex

**Install jupyter notebook**

pip install notebook

Open terminal and type:

jupyter notebook

A web window will open showing your home directory. To open a new notebook, navigate to location where you want the notebook. From the dropdown menu on the right select “New” 🡪 “Python 3 Notebook.”

For the cobrakbase.KBaseAPI() commands to work, you need to have the token for the KBase workspace being accessed in the notebook stored in your user folder at this location: /.kbase/token.

Make the directory .kbase (mkdir ~/.kbase) in your home folder and generate a file storing the developer token you get from KBase. In the command below, <token> will be replaced with the long alphanumeric string that is your token.

echo **<token>** > ~/.kbase/token

Now that this is working, you can open up the notebook KBase\_Cobrapy\_Integration.ipynb. Select and run the blocks of code to import python modules and convert a model from the KBase Narrative to cobrapy. To select each block of code, click in the cell and hold Ctrl + A. Then press “Run” in the menu at the top of the screen.

The commands below import the KBase model to a cobrapy model. You will have to insert the number for your own KBase narrative and the name of your model.

**kbmodel = kbase\_api.get\_from\_ws('** <name of model>**',** <number identifying narrative>**)**

Example, using my copy of the tutorial narrative linked earlier:

**kbmodel = kbase\_api.get\_from\_ws('**Shewanella\_oneidensis\_MR-1\_Gapfilled\_Lactate**',** 81169**)**

Now you will be able to work with the model in cobrapy, consulting the documentation and tutorials for that software. <https://opencobra.github.io/cobrapy/>