ETH zürich

Analyzing openBIS datasets with Jupyter & MATLAB

Henry Lütcke
Scientific IT Services, ETH Zurich

openBIS Training, 24.06.2020

Slides / materials: http://u.ethz.ch/O5RXC

Why Jupyter notebooks?

Why Jupyter notebooks?

- Jupyter notebooks combine documentation, code, input and output generated by the code (e.g. graphs, plots, images, videos)

TOOLBOX · 30 OCTOBER 2018

Why Jupyter is data scientists' computational notebook of choice

An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.

Jeffrey M. Perkel

Why Jupyter notebooks?

- Jupyter notebooks combine documentation, code, input and output generated by the code (e.g. graphs, plots, images, videos)
- Useful for interactive / exploratory data analysis and reproducibility
- > 40 programming languages supported (**Ju**lia, **Pyt**hon, **R**, Scala, Matlab, etc)
  - Different programming languages can be mixed
- Easily share code, documentation and results
- It can even act as a modern lab notebook
- Modern, web-based UI

Jupyter notebook examples

Cancer genomics analysis

ucsd-ccbb / jupyter-genomics

Watch 14 | Star 43 | Fork 18

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Security | Insights

Branch: master ▾ jupyter-genomics / notebooks / networkAnalysis / TCGA_clustering_OV / TCGA_clustering_OV.ipynb

Find file | Copy path

brinrosenthal added network analysis notebooks 8a5d87e on 10 Jun 2016

1 contributor

2.62 MB Download | History

Clustering methods applied to TCGA Ovarian Cancer Coexpression Matrix

Author: Brin Rosenthal (sbrosenthal@ucsd.edu)

April 15, 2016

- We will provide a first look into which clustering methods work best on TCGA coexpression matrices, starting with Ovarian Cancer.
- We will visualize these clustering results.
- Clustering methods tested include:
  - Modularity maximization (Louvain)
  - Affinity propagation
  - DBSCAN
  - Hierarchical clustering
- The Ovarian Cancer TCGA Co-expression matrix may be found here https://ucsd-ccb-data-analysis.s3.amazonaws.com/Brin/ccbb_jupyter_genomics/network_analysis/brin_OV_clustering_TCGA/OV.tsv

Plot the clusters in network form

- Spring embedded layout for node positions (more strongly connected nodes are positioned closer together)
- Node colors encode cluster membership found from the Louvain modularity maximization algorithm

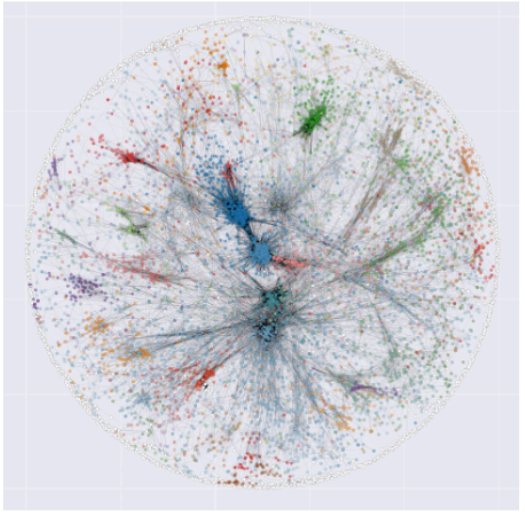
In [1]:

```
# plot the network
import matplotlib.colorbar as cb
import seaborn as sns

vmin=None
vmax=None

cmap = 'Paired'

pos = nx.spring_layout(Gtemp,k=.03)

fig,ax=plt.subplots(figsize=(50,40))


# draw small community nodes as white
partition = pd.Series(partition)
par_VC = partition.value_counts()
groupL5 = list(par_VC[par_VC<5].index)
groupG5 = list(par_VC[par_VC>=5].index)



# select out nodes in small communities
nodes_w = []
[nodes_w.extend(list(partition[partition==i].index)) for i in groupL5]

# now select large community nodes
nodes_c = []
[nodes_c.extend(list(partition[partition==i].index)) for i in groupG5]

GL5 = nx.subgraph(Gtemp,nodes_w)
GG5 = nx.subgraph(Gtemp,
```

Run the above code block to generate network figure

- Static image included here to conserve space

```
# rename partitions for
group_map = dict(zip(grou
par_rename = []
[par_rename.append(group_

cols = par_rename
cols = pd.Series(cols,inc
#cols[nodes_c]=partition
#cols = list(cols)

nx.draw_networkx_nodes(G

nodes_col = nx.draw_netwo

=.5,

nx.draw_networkx_edges(G
plt.grid('off')
```



Options for running Jupyter notebooks

Options for running Jupyter

- Local installation on your computer
- Dedicated JupyterHub server (e.g. running on virtual machine in the cloud)
- Public cloud-based offerings
  - MyBinder: https://mybinder.org/
  - Google cloud: https://colab.research.google.com/notebooks
  - EGI Notebooks: https://notebooks.egi.eu/hub/login
- To get started
  - https://jupyter.org/try

Local installation of Jupyter

- Option 1: Anaconda platform
  - Download / install Anaconda for your OS: https://www.anaconda.com/distribution/
  - Installs Jupyter, Python, R (+ >1’500 other packages)
  - Start notebook from Anaconda launcher
- Option 2: Python only installation
  - Download / install Python for your OS: https://www.python.org/downloads/
  - Open the command line and type:

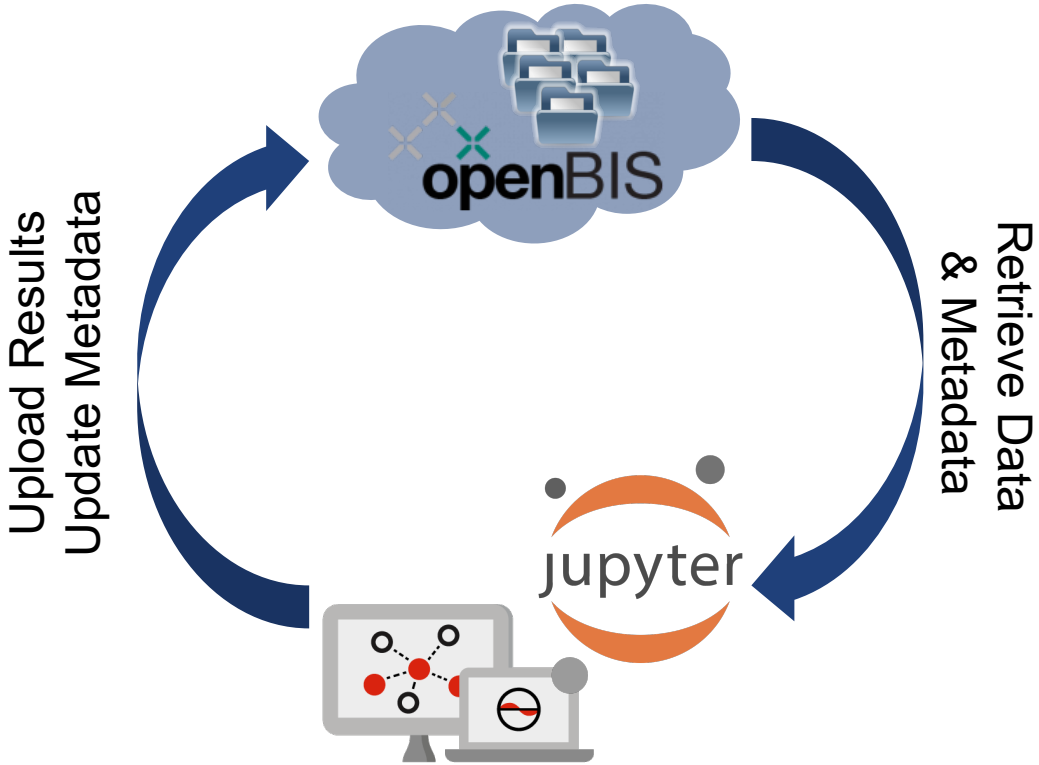
    pip install --upgrade pip

    pip install --upgrade ipython jupyter
  - Open the command line and type: *jupyter notebook*

Combining openBIS & Jupyter notebooks

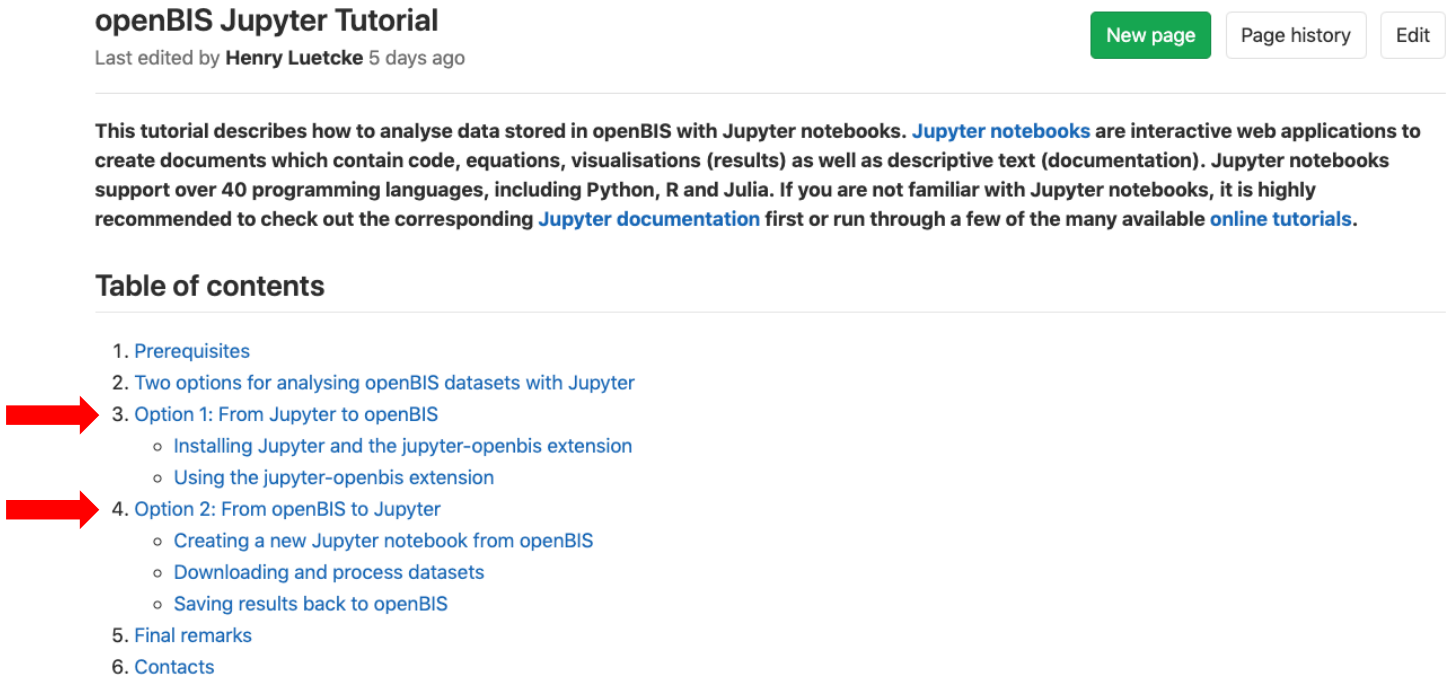
openBIS & Jupyter

- openBIS: data management + ELN-LIMS
- Jupyter: interactive, reproducible data analysis
- openBIS + Jupyter: full provenance tracking of data, analysis and results



openBIS & Jupyter

- openBIS: data management + ELN-LIMS
- Jupyter: interactive, reproducible data analysis
- openBIS + Jupyter: full provenance tracking of data, analysis and results
- Detailed tutorial: http://u.ethz.ch/R70DW



openBIS Jupyter Tutorial

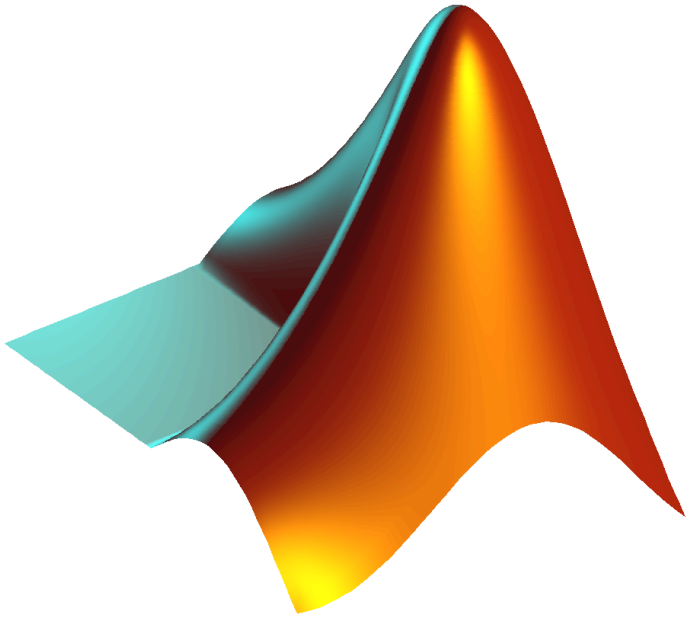
Last edited by **Henry Luetcke** 5 days ago

New page | Page history | Edit

This tutorial describes how to analyse data stored in openBIS with Jupyter notebooks. Jupyter notebooks are interactive web applications to create documents which contain code, equations, visualisations (results) as well as descriptive text (documentation). Jupyter notebooks support over 40 programming languages, including Python, R and Julia. If you are not familiar with Jupyter notebooks, it is highly recommended to check out the corresponding Jupyter documentation first or run through a few of the many available online tutorials.

Table of contents

1. Prerequisites
2. Two options for analysing openBIS datasets with Jupyter
3. Option 1: From Jupyter to openBIS
   - Installing Jupyter and the jupyter-openbis extension
   - Using the jupyter-openbis extension
4. Option 2: From openBIS to Jupyter
   - Creating a new Jupyter notebook from openBIS
   - Downloading and process datasets
   - Saving results back to openBIS
5. Final remarks
6. Contacts



Using openBIS with MATLAB

openBIS & MATLAB

- For experienced programmers: access native openBIS Java API from MATLAB: https://wiki-bsse.ethz.ch/display/openBISDoc/openBIS+V3+API
- MATLAB toolbox for openBIS: https://sissource.ethz.ch/hluetcke/matlab-openbis