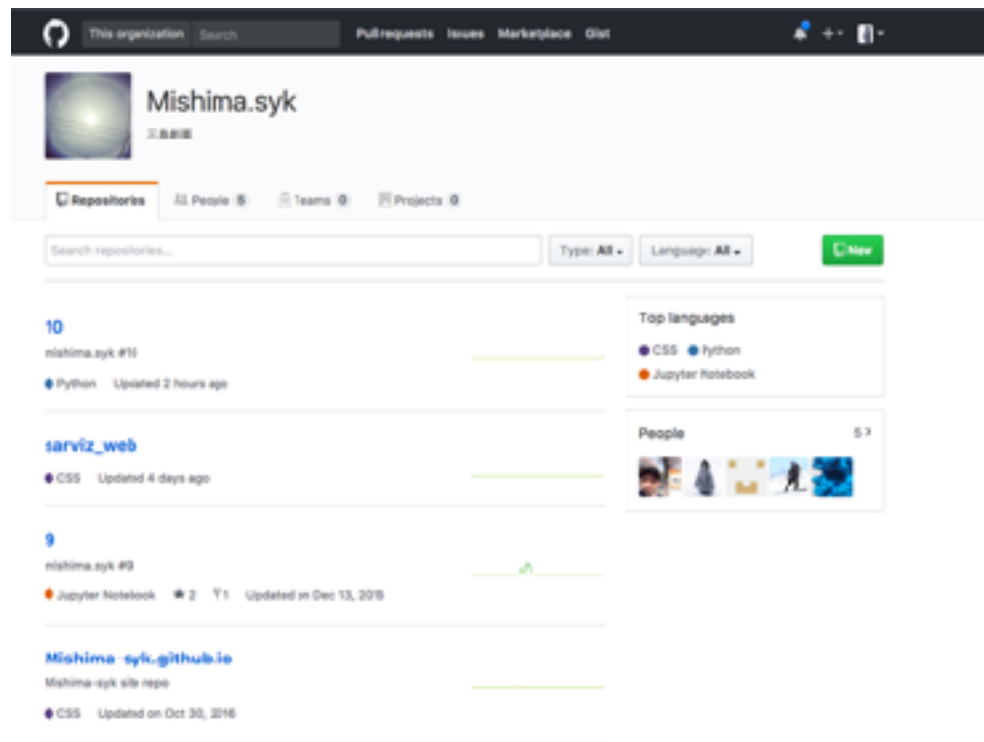


Machine Learning を流れるようにw

Mishima.syk #10
@iwatobipen

祝！

10th anniversary Mishima.syk!



<https://github.com/Mishima-syk>

あとで資料とコードはpushします。

Who am I ?

- Twitter: @iwatobipen
- 業種: Medicinal Chemist
- 興味: Chemoinformatics, Organic Synthesis
- 言語: Japanese, Python, Java Script, R



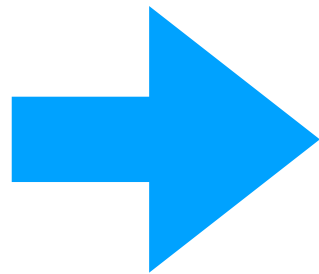
Today's topics

- Use pipeline / machine learning !
- Use Luigi !

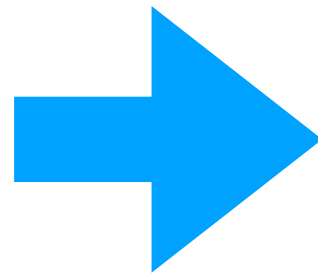
早速

Machine Learningの流れを考えます

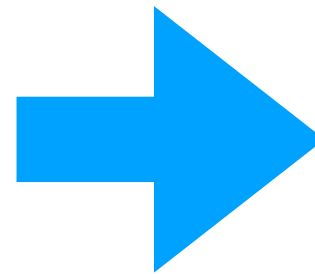
Data Retrieve



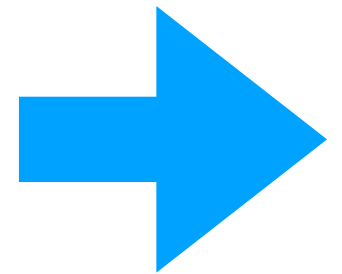
Split Data
Train / Test



Model Build

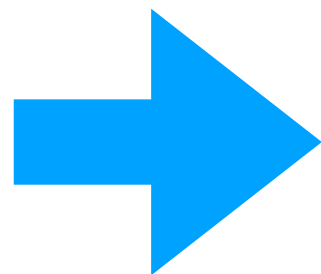


Predict



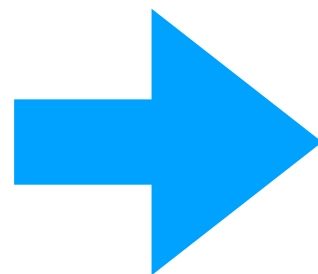
開発初期だとコードの断片を作ってとりあえず動かしたい

Data Retrieve



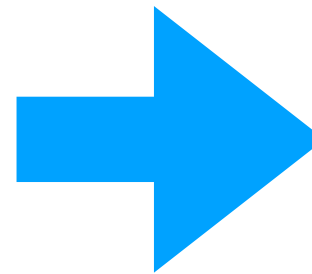
fetch.py

Split Data
Train / Test



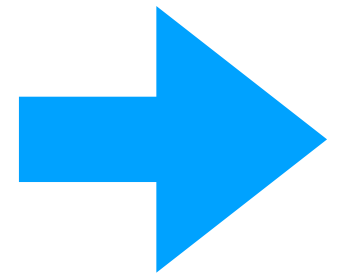
splitdata.py

Model Build



model.py

Predict



predict.py

Run !



fetch.py



splitdata.py



model.py



predict.py

```
$python fetch.py  
$python splitdata.py  
$python model.py  
$python predict.py
```

とか

```
if __name__ == '__main__':  
    fetch()  
    splitdata()  
    model()  
    predict()
```

But.....

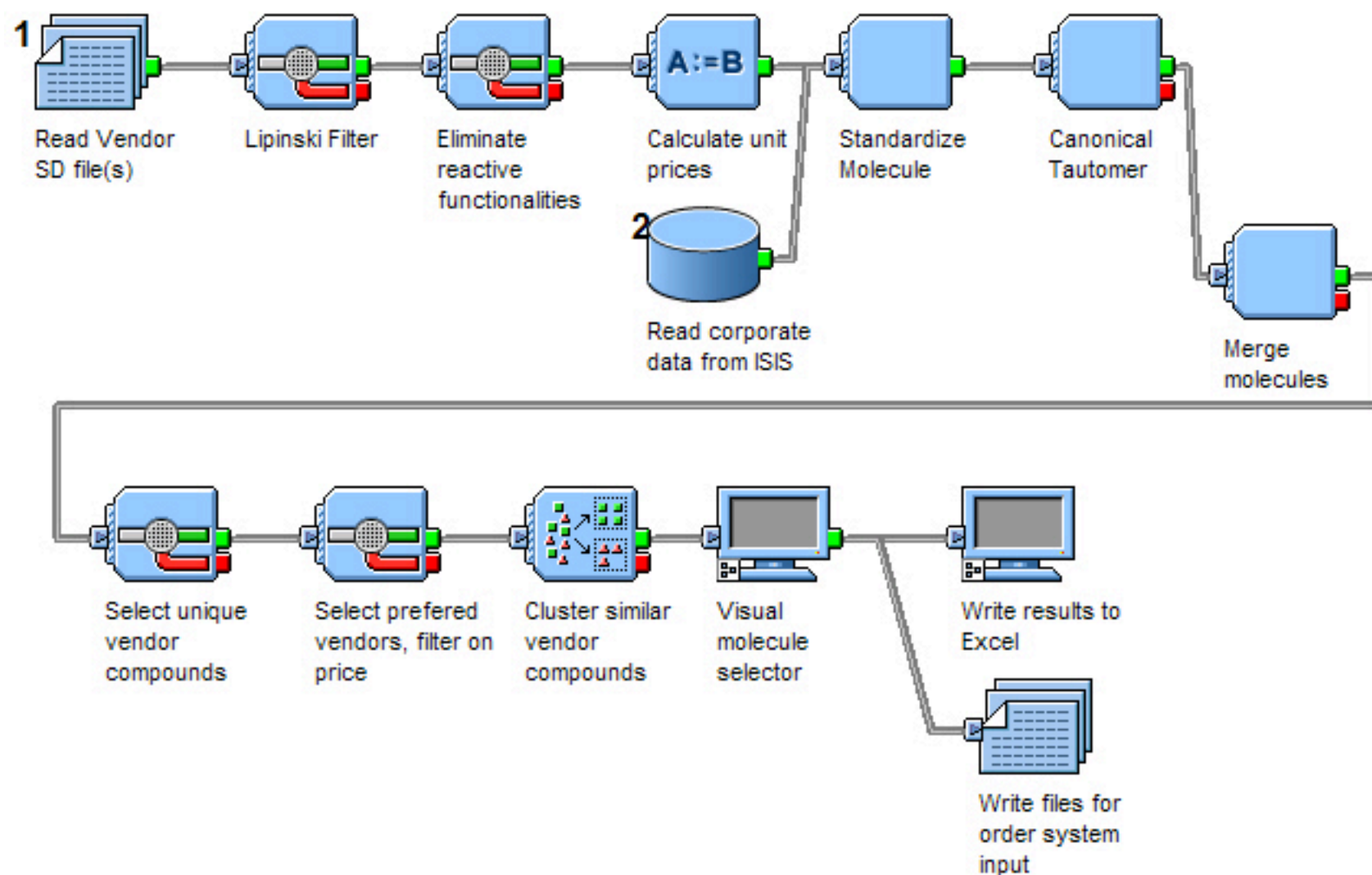
デバックがしにくい、、、
途中までうまく行っていたのに最初からやり直し、書き直し
全部まとめるとそれはそれで面倒なこともある。

```
$python fetch.py  
$python splitdata.py  
$python model.py  
$python predict.py
```

or

```
if __name__ == '__main__':  
    fetch()  
    splitdata()  
    model()  
    predict()
```


でPipelineすっきりさせたいのです



俺の出番か！



What is Luigi ?

- A Python Framework for data flow definition and execution.
- 音楽配信サービスを展開するSpotifyが開発
- All Python
- Pipでインストールできます



Luigiの基本構成

```
# task.py
import luigi
class FirstTask( luigi.Task ):
    def requires( self ):
        return []
    def output( self ):
        return luigi.LocalTarget( "hogehoge.txt" )
    def run( self ):
        with self.output().open( 'w' ) as output:
            " do something "
            output.write( 'hogehoge' )

class SecondTask( luigi.Task ):
    def requires( self ):
        return FirstTask()
    def output( self ):
        return luigi.LocalTarget( "hogehoge2.txt" )
    def run( self ):
        " do something "
        output.write( 'hogehoge2' )

if __name__ == "__main__":
    luigi.run()
```

requires, output, runセット

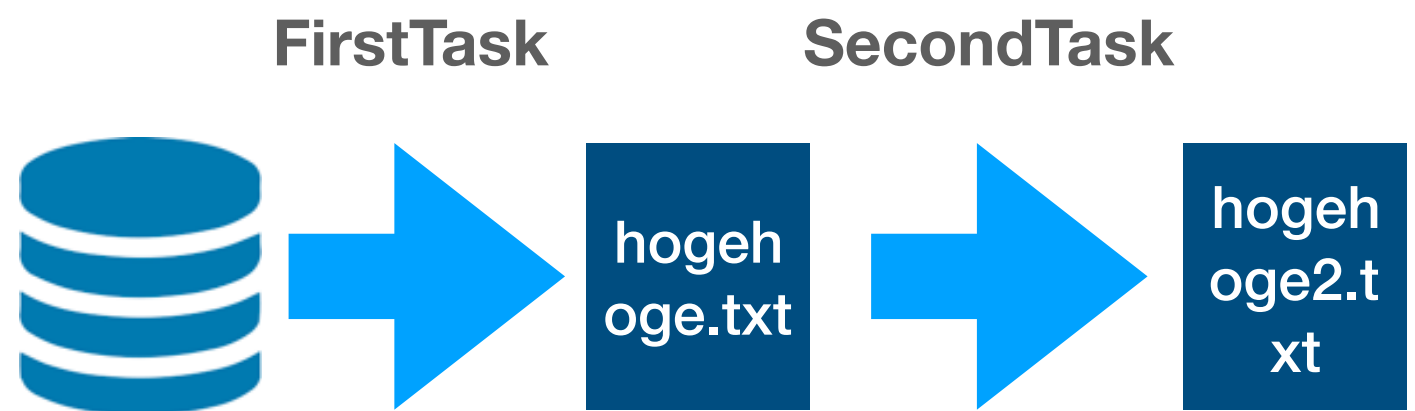
requires : 依存するタスク

output : タスクの出力

=> pickleは永続化に向かない

run : 実際の実行部分

=> withで書くこと推奨



Run the Task

```
$ python task.py SecondTask --local-scheduler
```

とりあえずローカルで実行するのなら

—local-schedulerを指定するといい。

Central-schedulerを使うとタスクの実行状況などが可視化される（後で実際に動かします）。

FirstTask => SecondTaskのように実行される。

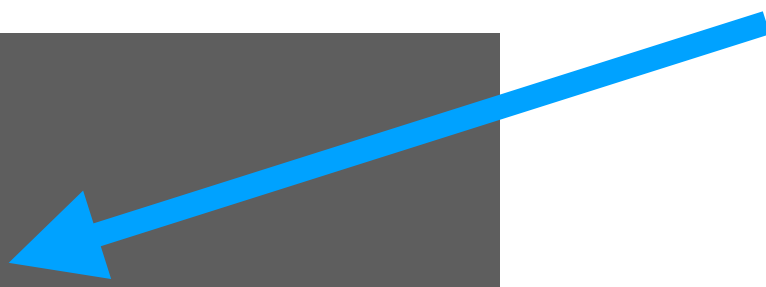
FistTaskと指定すればそこで止めることも可能です。

Parameter handling

```
$ python task.py Task --local-scheduler --param hogehoge
```

```
# task.py
import luigi
class Task( luigi.Task ):
    param = luigi.Parameter()
    def requires( self ):
        return []
    def output( self ):
        return luigi.LocalTarget( "hogehoge.txt" )
    def run( self ):
        with self.output( ).open( 'w' ) as output:
            " do something "
            param = self.param
            output.write( 'hogehoge' )

if __name__ == "__main__":
    luigi.run()
```



実際にやってみる

- Retrieve data from Local ChEMBL DB*ここは前もって設定が必要です。僕の環境のverが22ですがそこはスルーしてください。
- Vectorize Data
- Train Test Split
- Model Build
- Model Test / Prediction

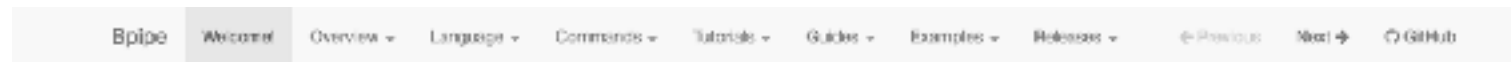
Requirements

- RDKit
- Psycopg2
- Scikit-learn
- Luigi
- Pandas

Go to Code !!!!

おまけ 1 For Bioinformatics

<http://docs.bpipe.org/Examples/RNASeqCorset/>
RNAseqの解析事例 => データでかいとビビってやってない



Welcome to Bpipe

Welcome to Bpipe build pending

Bpipe provides a platform for running big bioinformatics jobs that consist of a series of processing stages - known as 'pipelines'.

- June 18th, 2017 - New! [Bpipe 0.9.9.4](#) released!
- [Download latest, all](#)
- [Documentation](#)
- [Mailing List](#) (Google Group)

Bpipe has been published in [Biominformatics](#). If you use Bpipe, please cite:

Sadeeln S, Papa B & Oakleaf A, *Bpipe: A Tool for Running and Managing Bioinformatics Pipelines*, *Bioinformatics*

Why Bpipe?

Many people working with bioinformatics data end up running jobs as shell scripts. While this makes running them easy it has a lot of limitations. For example, when scripts fail half way through it is often hard to tell where, or why they failed, and even harder to restart the job from the point of failure. There is no accessible log of the commands executed or a sensible picture of overall output to ensure it is possible to later on confused with good files. Modifying modifications in multiple places, or running on incorrect data. Bpipe tries to solve these problems by providing a platform for running and managing bioinformatics pipelines. In fact, your Bpipe pipeline is a shell script.

For more information about Bpipe, see the

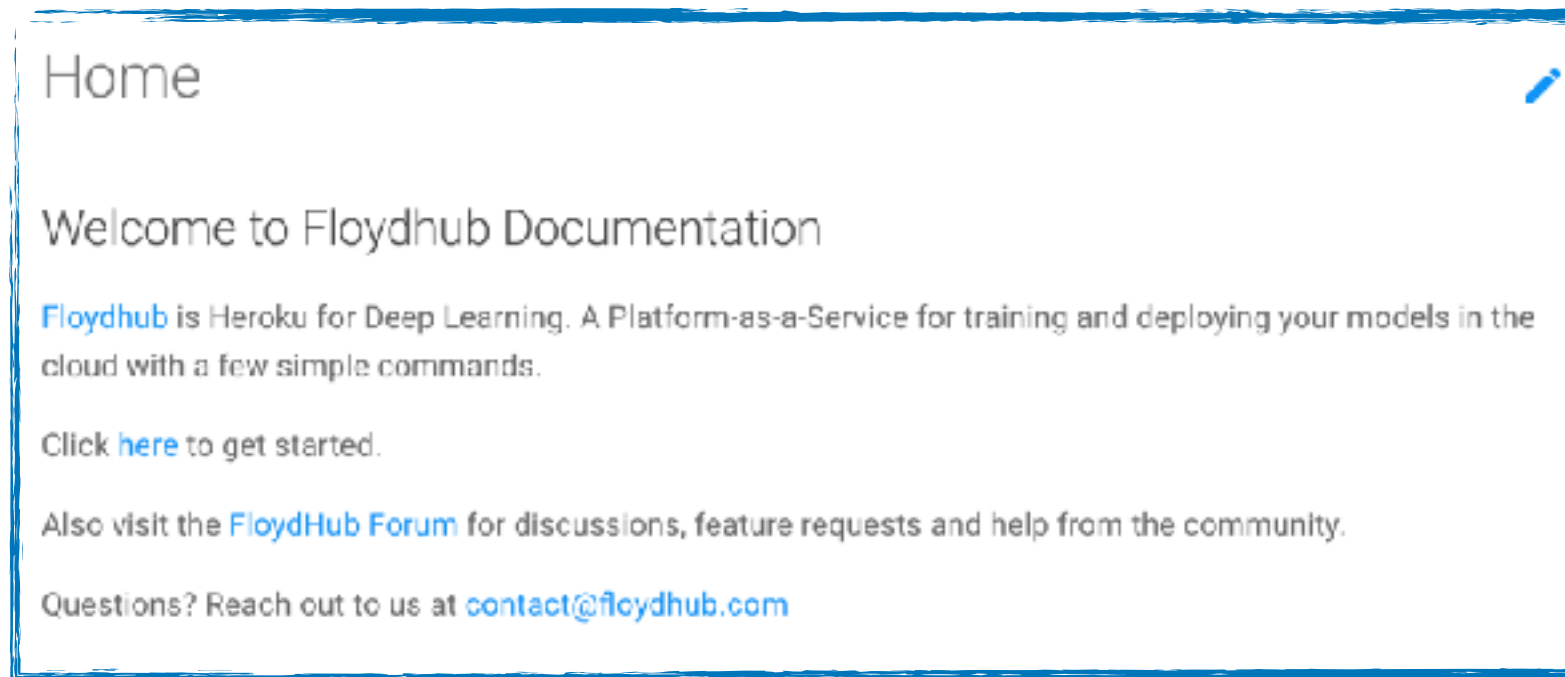
Feature Comparison Table

Tool	GUI	Command Line (**)	Audit Trail	Built in Cluster Support	Workflow Sharing	Online Data Source Integration	Need Programming Knowledge?	Easy Shell Script Portability
Bpipe	No	Yes	Yes	Yes	No	No	No	Yes
Ruffus	No	Yes	Yes	No	No	No	Yes	No
Galaxy	Yes	No	Yes	Yes	Yes	Yes	No	No
Taverna	Yes	No	Yes	Yes	Yes	Yes	No	No
Pegasus	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Bpipeぱっと見便利そう

おまけ 2 For Deep Learning

- Floydのアカウント作ってみんなでGPU使おうぜ！
- Deep learningの置けるHerokuとのこと。
- アカウント無料のプランでGPUが使えます。AWS+DL環境



Thank you!

https://www.youtube.com/watch?v=Ny2X_WNxrB4

- ググると日本語の情報も結構あります。また、上のプレゼンはわかりやすくて良かったです。