Prediction of RNA and DNA binding sites: weekly report

Pandu Raharja [1,2] Julian Schmidt [1,2]

[1]Technische Universität München

[2]Ludwig-Maximilians-Universität München

December 8, 2016

Outline

- What we did:
  - Julian: xRNA binding prediction
  - Pandu: xRNA binding site prediction
- Problems.

As always:

- Scripts and results could be found in: https://github.com/raharjaliu/BIFers

Part I: xRNA binding prediction

Part II: xRNA binding site prediction

Intro

- We're a bit late onto the game due to illness and conference.
- Extracted features could be found in `/mnt/project/pp2_1516/xrna_raharjaschmidt/machine_learning/{dna_big.arff,rna_big.arff}`.

Extraction of Features on ppDNA2

- `query.disis not found.`
- `Q84ZU4:  no significant pfam hit, using default values...`
- `P03206` **(same warning)**

Extraction of Features on ppRNA2

- `query.disis not found.`
- `Processing P17574... ParseError: "It seems that we have a situation now. The expected amount of columns is 22, found: %s!" % len(tokens)`
- `P24264` **(same ParseError)**
- `P67876: no significant pfam hit, using default values...`
- `P0C206` **(same warning)**
- `P0C8P8` **(same warning)**
- `P07243` **(same warning)**
- `Q57817` **(same warning)**
- `P04891` **(same warning)**
- `P18683` **(same warning)**

Machine Learning Sorcery

Some considerations:

- Stack choice: not a big fan of Java but features are contained in ARFF already:
  $\rightarrow$ Weka binding of Python?
- Easy model (SVN, DT/RF etc) vs. sophisticated model (deep representation learning and all the new fancy things coming out of NIPS/ICML 201X):
  $\rightarrow$ Implementation **will** be constrained by Weka (and time).
- Single model vs. ensemble learning (with boosting etc).
- Time constraint: exams and works.

Some notes regarding Weka on Py

- Seems to be possible:
  http://www.cs.waikato.ac.nz/~eibe/WEKA_Ecosystem.pdf
- Requires python-weka-wrapper and cos's:
  `$sudo apt-get install python-pip python-numpy python-dev python-imaging python-matplotlib python-pygraphviz imagemagick`
  `$sudo pip install javabridge python-weka-wrapper`
  $\rightarrow$ are all these installed on the server?
- Actually requires **starting-up of JVM on Python**.

Example (1)

- Starting the JVM from Python:
  ```
  import weka.core.jvm as jvm jvm.start()
  ```
- Getting help:
  help(jvm.start)
- Loading and printing some data in ARFF format:
  ```
  from weka.core.converters import Loader
  l = Loader("weka.core.converters.ArffLoader")
  d = l.load_file("weka-3-7-11/data/iris.arff")
  d.set_class_index(d.num_attributes() - 1)
  print(d)
  ```

Example (2)

- Building and printing a decision tree:

```
from weka.classifiers import Classifier
c = Classifier("weka.classifiers.trees.J48")
c.build_classifier(d)
print(c)
```

- Evaluating classifier using cross-validation:

```
from weka.classifiers import Evaluation
from weka.core.classes import Random
e = Evaluation(d)
e.crossvalidate_model(c, d, 10, Random(1))
print(e.percent_correct())
print(e.to_summary())
print(e.to_class_details())
```

Some notes regarding prediction

- What constitutes a good predictor?
  - which scoring function to optimize?
    $\rightarrow F_\beta$, recall, coverage?
- Relaxed or rigid definition of binding residue?
  –!!!– vs –!–!– vs ––!!– vs –!!!!
- etc (what is the meaning of life?)

References