Guião 1 – Welcome to Bioinformatics

Duração: 2 semanas

Objectivos programáticos:
- Revisão de Python
- Acesso programático ao UniProt
- Utilização da Biblioteca Scikit-learn (Machine Learning)

Descrição da tarefa:
Desenvolver um classificador que possibilite a identificação de proteínas com a função molecular Atividade Catalítica (GO:0003824).
Dada uma lista de identificadores de proteínas humanas envolvidas na Atividade Catalítica obter programaticamente a sua sequência, extrair *features*, e treinar um classificador.

PARTE 1

1. Acesso programático (Python) à base de dados UniProt.

O seguinte código permite a obtenção da sequência de aminoácidos no formato FASTA para a proteína P53 (Uniprot: P04637).

```
>>>import urllib

>>>link = "http://www.uniprot.org/uniprot/P04637.fasta"
>>>f = urllib.urlopen(link)
>>>myfile = f.read()
>>>print myfile
```

Resultado no formato FASTA:

```
>sp|P04637|P53_HUMAN Cellular tumor antigen p53 OS=Homo sapiens
GN=TP53 PE=1 SV=4
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGP
DEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAK
SVTCTYSPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHE
RCSDSDGLAPPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNS
SCMGGMNRRPILTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELP
PGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPG
GSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD
```

2. Deve alterar o código anterior de forma a obter a sequência de aminoácidos para todas as proteínas anotadas na função molecular Atividade Catalítica. Link: http://www.uniprot.org/uniprot/?sort=&desc=&compress=no&query=organism:%22Homo%20sapiens%20(Human)%20[9606]%22%20AND%20proteome:up000005640%20go:3824&fil=&force=no&format=list

PARTE 2

1. Preparação dos dados, extração das *features* e obtenção do dataset de treino/validação.

2. A título de exemplo é utilizado o dataset Iris disponível em (http://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html).
iris.data-> matriz em que cada linha representa uma amostra e cada coluna uma *feature*;
iris.target-> vector em que cada elemento representa uma classe

```
>>>import numpy as np
>>>from sklearn import cross_validation
>>>from sklearn import datasets
>>>from sklearn import svm

>>>iris = datasets.load_iris()
>>>iris.data
array([[ 5.1,  3.5,  1.4,  0.2],
       [ 4.9,  3. ,  1.4,  0.2],
       [ 4.7,  3.2,  1.3,  0.2],
       [ 4.6,  3.1,  1.5,  0.2],
       [ 5. ,  3.6,  1.4,  0.2]
(...)
>>>iris.target
array([0, 0, 0,(…) 1, 1, 1, (…) 2, 2, 2, (…)])
```

2. Deve preparar uma estrutura de dados semelhante à anterior em que cada linha corresponde a uma proteína e cada coluna a uma *feature*. Numa fase inicial as *features* são obtidas através de uma contagem simples de aminoácidos. O vector Target deve ser boleano em que **1** representa "Atividade Catalítica" e **0** representa "Outra".

3. Altere o seu código de forma a considerar as restantes proteínas Humanas.
http://www.uniprot.org/uniprot/?sort=&desc=&compress=no&query=proteome:UP000005640&fil=&force=no&format=list

PARTE 3

1. Treino e validação do classificador. O código seguinte ilustra a utilização do Scikit-learn para o dataset de exemplo:

```
(...)
>>>X_train, X_test, y_train, y_test = cross_validation.
train_test_split(iris.data, iris.target, test_size=0.4,
random_state=0)
>>>clf = svm.SVC(kernel='linear', C=1).fit(X_train, y_train)
>>>clf.score(X_test, y_test)
```

2. Exploração de resultados. Calcule as métricas de avaliação da *cross-validation*; e curva ROC.

3. Explore estratégias alternativas de extração de *features*.