Bioinformática para identificação e anotação de non-conding RNA

Ferramentas para identificação de non-coding RNA

10/08/2022

Alisson G. Chiquitto
Doutorando em Bioinformática
Programa de Pós-Graduação Associado em Bioinformática UFPR/UTFPR-CP

Obter as amostras

Obter datasets

```
wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_
41/gencode.v41.lncRNA_transcripts.fa.gz
```

```
wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_
41/gencode.v41.pc_transcripts.fa.gz
```

```
gunzip -k gencode.v41.lncRNA_transcripts.fa.gz
```

```
gunzip -k gencode.v41.pc_transcripts.fa.gz
```

Amostragem dos datasets

```
seqtk sample -s22 gencode.v41.lncRNA_transcripts.fa 50 >
lncRNA_samples.fa
```

```
seqtk sample -s22 gencode.v41.pc_transcripts.fa 50 > pc_samples.fa
```

Preparar e mesclar amostras

```
seqkit replace -w 0 -p "(.+)" -r "lncRNA#\$1" lncRNA_samples.fa >
lncRNA_samples1.fa

seqkit replace -w 0 -p "(.+)" -r "pc#\$1" pc_samples.fa >
pc_samples1.fa
```

```
cat lncRNA_samples1.fa pc_samples1.fa > samples.fa
```

Contar sequências

```
grep ">" lncRNA_samples.fa | wc -l
grep ">" pc_samples.fa | wc -l
grep ">" samples.fa | wc -l
```

CPC2

http://cpc2.gao-lab.org/

https://github.com/gao-lab/CPC2_standalone

Instalação do CPC2 standalone

As instruções a seguir são para instalação do CPC2 para Python 2.

```
wget http://cpc2.gao-lab.org/data/CPC2-beta.tar.gz
gzip -dc CPC2-beta.tar.gz | tar xf -
cd CPC2-beta
export CPC_HOME="$PWD"
cd libs/libsvm
gzip -dc libsvm-3.18.tar.gz | tar xf -
cd libsvm-3.18
make clean && make
pip install biopython==1.76
```

No site oficial existem instruções para instalação do de uma versão para Python 3.

Execução

```
cd $CPC_HOME && python ./bin/CPC2.py -i input.fa -o output.txt
```

RNAmining

Website oficial: https://rnamining.integrativebioinformatics.me/

Instalação

A seguir estão instruções para a versão standalone do RNAmining.

```
wget
https://gitlab.com/integrativebioinformatics/RNAmining/-/archive/m
aster/RNAmining-master.zip
```

```
unzip RNAmining-master.zip
```

```
cd RNAmining-master/
```

```
pip install xgboost biopython pandas scikit-learn
```

Também existe uma versão baseada em imagem do Docker. Mais detalhes dessa versão podem ser encontrados no website oficial.

Execução

```
python3 rnamining.py -f cod filename -organism_name organism_name
-prediction_type coding_prediction -output_folder output
```

RNAplonc (with Docker)

Docker is a software platform that allows the creation of applications in a controlled environment. The idea for creating an RNAplonc docker is to make the tool readly available, without going to the trouble of configuration and compatibility of its dependencies.

The RNAplonc image was created using Ubuntu 20.04 image, cd-hit-est and txCdsPredict tools compiled in Ubuntu 20.04, python version 3.8.2, openjdk version 1.8.0_252.

The RNAplonc tools were installed on /RNAplonc directory and contain the following files:

- 200nt.pl
- feature_extraction.pl
- seq_test: Citrus_sinensis_lncRNA_GREENC.fasta
- Dockerfile
- RNAplonc.model
- FilterResults.py
- txCdsPredict
- README.txt
- cd-hit-est

- random_selection.pl
- weka.jar

1. Installing docker

You can find how to install docker on Apple, Windows and Linux on the official documentation link below: https://docs.docker.com/get-docker/

2. Downloading the docker image and installing.

```
docker pull lopesandrecosta/rnaplonc
```

3. RNAplonc usage

3.1 Change directory to the path of the database using cd command

This step is needed, because we are binding the database folder with a virtual folder called /app.

```
cd rnaploncepb
```

3.2 - 200nt.pl (Optional)

200nt.pl is a perl script that verifies that the sequences on the fasta files have more than 200 nucleotides.

```
sudo docker run -it --rm -v "$(pwd):/app" --user $(id -u):$(id -g)
lopesandrecosta/rnaplonc perl RNAplonc/200nt.pl app/samples.fa
```

Output: 200nt will output a fasta file with "_" at the end, eg file_fasta.

3.3 - CD-HIT-EST (Optional)

CD-HIT-EST (Cluster nucleotide sequences) is an executable that removes the sequences with a given similarity X. On this work we use 80%.

```
sudo docker run -it --rm \
-v "$(pwd):/app" \
--user $(id -u):$(id -g) \
lopesandrecosta/rnaplonc RNAplonc/cd-hit-est \
-i app/samples_.fasta -o app/cd_hit_est_result.fasta -c 0.8
```

where

-i = Name of the output file from step 2

-o = Output file name

-c = Percentage cut used of 80% similarity

3.4 txCdsPredict (Mandatory)

txCdsPredict is an executable used to find Open Reading Frames in the Dataset. It will use the output file from 3.3.

```
sudo docker run -it --rm \
-v "$(pwd):/app" \
--user $(id -u):$(id -g) \
lopesandrecosta/rnaplonc RNAplonc/txCdsPredict \
app/cd_hit_est_result.fasta app/tx_cds_result.cds
```

where

cd_hit_est_result.fasta = Name of the output from step 3.3 (CD-HIT-EST)

tx_cds_result = Name of the output file from step 3.4 - (txCdsPredict)

3.5 feature_extraction.pl (Mandatory)

The argument for this step is both output files from 3.3 and 3.4.

```
sudo docker run -it --rm \
-v "$(pwd):/app" \
--user $(id -u):$(id -g) \
lopesandrecosta/rnaplonc perl RNAplonc/feature_extraction.pl \
app/cd_hit_est_result.fasta app/tx_cds_result.cds > features.arff
```

The output of this step is a .arff file.

3.6 RNAplonc.model execution on weka (Mandatory)

The argument needed for the weka execition is the resulting file from 3.5

```
sudo docker run -it --rm \
-v "$(pwd):/app" \
--user $(id -u):$(id -g) \
lopesandrecosta/rnaplonc java -cp RNAplonc/weka.jar
weka.classifiers.trees.REPTree \
-l RNAplonc/RNAplonc.model -T app/features.arff \
-p 0 > classification_result.txt
```

The output is classification_result.txt

3.7 Python Script to put back the sequence names back into the final result (Optional)

This step will use the classification_result.txt from the step 3.4 and 3.6.

```
sudo docker run -it --rm \
-v "$(pwd):/app" \
--user $(id -u):$(id -g) \
lopesandrecosta/rnaplonc python3 RNAplonc/FilterResults.py \
-c app/tx_cds_result.cds -r app/classification_result.txt \
-o app/final_result.txt
```

Tips

Run shell

```
docker run -it --rm -v "$(pwd):/app" lopesandrecosta/rnaplonc bash
```

Exemplos para processamento dos resultados

Remover linhas desnecessárias

```
tail -n +2 cpc2.txt > cpc2-data.txt
```

```
tail -n +6 rnamining.txt > rnamining-data.txt
```

Obter apenas linhas com sequências lncRNA

```
awk 'BEGIN { OFS = ";" } {print (substr($1,0,6)=="lncRNA") ?
"lncRNA" : "pc", $7, $1}' cpc2-data.txt > cpc2-parsed.csv
```

```
awk 'BEGIN { OFS = ";" } {print (substr($1,0,6)=="lncRNA") ?
"lncRNA" : "pc", $2, $1}' rnamining-data.txt >
rnamining-parsed.csv
```