Covidor

Release 0.1.0

Thomas Cokelaer

Feb 09, 2022

CONTENTS

Python version Python 3.7, 3.8, 3.9

Source See http://github.com/sequana/covidor.

Issues Please fill a report on github

Platform This is currently only available for Linux distribution with bash shell (contributions are welcome to port the tool on MacOSX and other platforms)

CHAPTER

ONE

OVERVIEW

```
pip install covidor
```

1.1 third-party tools

kraken2 and art_illumina are required for the simulation and analysis.

```
pip install damona
# for kraken2 only:
damona install kraken:2.0.9
# to install kraken2 and art_illumina altogether, there are both provided in the sequana_
↪tools package:
damona install sequana_tools
```

1.2 Installation

For Users:

```
pip install covidor
```

For developers:

```
git clone git@github.com:sequana/covidor.git
cd covidor
pip install .
```

1.3 Testing

pytest -v ./tests

CHAPTER

TWO

USER GUIDE AND REFERENCE

2.1 User Guide

Table of Contents

2.1.1 Getting help

Coming soon

2.2 References

covidor.kmers

covidor.distance

covidor.main

2.2.1 kmer module

class KmerOverlap(*kmer_len=35*)

from covidor import KmerOverlap ko = KmerOverlap() for fasta in fastas:

ko.enumerate_unique_kmers(fasta)

names = self.kmer_dict.keys() ko.get_list_specific_kmers(name)

enumerate_unique_kmers(*fasta_files*)

populate kmers for each fasta

get_list_specific_kmers(*name*)

Returns kmer specific (unique) to a fasta

get_proportion_kmer_specific(*name*)
Gives same results as get_specific_kmers but normalised and for a given name

get_proportion_reads_specific(*name*, *N=100*, *read_length=75*, *paired=False*)
import glob fastas = glob.glob("databases/genomes/*fasta") kmer_dict = get_all_kmers() expected_percentage = get_proportion_reads_specific(kmer_dict, "NAME_ONE_FASTA")

get_specific_kmers()
Return dataframe with number of unique kmers in each fasta

2.2.2 distance module

class Distance

compute_pdistance(*alignment*, *mode='spike'*)
Compute distance between genomes on the spike gene

This takes as input the ouput of **mafft** tool on a set of input fasta file:

```
cat *fasta > in.fasta
mafft --auto in.fasta > alignment.fasta
```

Then:

```
from covidor.distance import Distance
d = Distance()
d.compute_pdistance("alignment.out")
```

distance is computed as the number of bases that are different along 2 sequences, normalised by the length (and multiplied by 100). Each sequence is identified by the starting and ending 20-bases long sequences.

plot(*alignment*, *mode='spike'*)

```
d = Distance()
d.plot("alignment.out")
```

2.2.3 Covidor standalone (script module)

covidor

This is the main entry point

```
covidor [OPTIONS] COMMAND [ARGS]...
```

Options

--version
Show the version and exit.

analyse

Example of mix of wuhan and beta (PE):

covidor --file1 4520_S4_L001_R1_001.fastq.gz --file2 4520_S4_L001_R2_001.fastq.gz

Example of omicron (SE):

fastq-dump SRR17673561 covidor analyse --file1 SRR17673561.fastq --db databases/default/

```
covidor analyse [OPTIONS]
```

Options

--file1 <file1>
Required

--file2 <file2>

--tag <tag>

--db <db>

--factor <factor>

stats

```
covidor stats [OPTIONS]
```

Options

--tag <tag>

--rl <rl>

--paired

--factor <factor>

PYTHON MODULE INDEX

C

INDEX

Symbols

C

D

E

G

K

M

P