fastcov - Fast Covariant Mutation Detector v1.02

Introduction

Single genetic mutation always brings along with a set of compensatory mutations, therefore multiple changes commonly occur in the biological sequences, which play crucial roles to maintain the conformational and functional stability. Although there are a lot of methods to detect single mutation or covariant pairs, it is still a great challenge to explore the non-synchronous multiple changes at different sites in the sequences.

Here we developed a novel algorithm, named Fastcov, to identify multiple correlated changes of biological sequences, by using an independent pair model followed by a tandem model of site-residue elements, based on a inter-restriction thinking. The results showed that Fastcov has excellent performances on harvesting co-pairs and detecting multiple covariant patterns. By 10-fold cross-validation in different scales of datasets, the characteristic patterns successfully classified the sequences into their target groups with an accuracy of 98% above. Moreover, it demonstrated the multiple covariant patterns represented co-evolutionary modes, corresponding to the phylogeny tree, and it provided new understanding of the structural stability of protein during the evolution. In contrast to other methods, Fastcov, as an original algorithm, provides not only a reliable and effective approach to harvest covariant pairs of site-residues, but also more powerful functions including multiple covariance detection and sequence classification.

Installation

Visit download page (http://yanlilab.github.io/fastcov/download/).

fastcov is implemented in Golang (https://golang.org/) programming language, executable binary files for most popular operating system are freely available.

Just download executable file, uncompress it with `tar -zxvf *.tar.gz` command, and then run it in command-line interface, no any dependency are needed.

You can also add the directory of the executable file to environment variable PATH, so you can run fastcov anywhere.

1. For windows, the simplest way is copy it to `C:\WINDOWS\system32`.
2. For Linux, type:

   ```
   chmod a+x /PATH/OF/FASTCOV/fastcov
   echo export PATH=\$PATH:/PATH/OF/FASTCOV >> ~/.bashrc
   ```

   or simply copy it to `/usr/local/bin`

Usage

```
Name:
  fastcov V1.02 -- Fast Covariant Mutation Detector
  http://yanlilab.github.io/fastcov

Authors:
  Yan Li   <liyan.com@gmail.com>
  Wei Shen <shenwei356@gmail.com>

Usage:
  fastcov [options] inputfile

Available Options:
  -p FLOAT                  minimum pairing purity of two sites [0.7]
  -d FLOAT                  minimum associated degree [0.7]
  -r FLOAT                  minimum matching ratio of to the pattern [0.45]
  -n INT                    minimum residue number at each site [5]
  -c FLOAT                  minimum proportion of any sequence identical to the
                            consensus [0.33]
  -o STRING                 prefix of output files [inputfile]
  -j INT                    CPU number [CPU number of your computer]
  -h, --help                show this help message

Copyright:
  Copyright © 2015-2016, All Rights Reserved
  This software is free to distribute for academic research.
```

Positional arguments

- `inputfile` should be aligned protein sequences in FASTA format file, produced by multi sequence alignment softwares.
  Case is not sensitive.

  One-seq-per-line format could be converted to FASTA format by

  ```
  for f in *.aln; do cat -n $f | awk '{print ">"$1"\n"$2}' > $f.fas; done
  ```

Options

Main algorithm parameters

- `-p` defines the minimum pairing purity of two sites. Default is 0.7.
- `-d` defined the minimum associated degree of one group of covariant mutation elements. Default is 0.7.
- `-r` defines the minimum matching ratio of to the pattern at clustering stage. Default is 0.45.

Sequences filter criteria

- `-n` is the minimum residue number at each site. Default value is 5.
- `-c` is the minimum proportion of any sequence identical to the consensus. Default value is 0.33, i.e. the number of residues identical to the that of the same position of consensus sequences should be at least one third of the length of consensus.

Sequences that fail to reach this criteria will be discarded.

Output

- `-o` defines the prefix of output files, default value is the same as input file. e.g, for a input file `test.fa`, output files will be:

```
test.aligned.fa.pairs
test.aligned.fa.clusters
test.aligned.fa.patterns
test.aligned.fa.seq2patterns
```

Performance

- `-j` is the number of CPU. fastcov detects your computer and set the default value with the maximum CPU number. The bigger the value is, the faster fastcov runs.

Examples

Taking `examples/ABCD_RT_M.aligned.fas` for example.

Quik run:

```
fastcov ABCD_RT_M.aligned.fas
```

Terminal Output:

```
Input: ABCD_RT_M.aligned.fas

Step 1/5: Reading sequences


Done

Step 2/5: Searching candidate sites

Done

Step 3/5: Searching independent pairs
21115 / 21115
[================================================================================
=====] 100.00 % 28s

Covariant site pairs saved to file: ABCD_RT_M.aligned.fas.pairs

Done

Step 4/5: Searching covariant patterns
52 / 52
[================================================================================
=============] 100.00 % 0

Covariant patterns saved to file: ABCD_RT_M.aligned.fas.patterns

Done

Step 5/5: Clustering by covariant patterns
Covariant patterns assigned to sequences: ABCD_RT_M.aligned.fas.seq2patterns
Sequences clustered by covariant patterns: ABCD_RT_M.aligned.fas.clusters
```

The most time-consuming stage is step 3, so we add a process bar.

Output files:

```
ABCD_RT_M.aligned.fas.pairs            # covariant pairs information
ABCD_RT_M.aligned.fas.patterns         # covariant patterns
ABCD_RT_M.aligned.fas.clusters         # sequence clusters by covariant
patterns
ABCD_RT_M.aligned.fas.seq2patterns     # covariant patterns of every sequence
```

Note: For windows user, please use a modern text editor to view the result files. Notepad is not recommended, Notepad++ (https://notepad-plus-plus.org/) is a better choice.

More examples (https://github.com/yanlilab/fastcov/tree/master/examples)

Errors and Solutions

1. No input file given. Please feed fastcov a aligned amino acids sequences in FASTA format.

```
 $ fastcov
 [Error] no input file (aligned amino acids sequences in FASTA format)
given.
 type "fastcov -h" for help
```

2. Input file is not aligned.

```
 [Error] sequence length not equal: 343 (AB014392_Pol-C) != 344.
 input file should be aligned amino acids sequences in FASTA format
```

3. Illegal characters in sequence. FASTA parsing module of fastcov strictly check the sequences, you may check input sequence according according to the IUPAC nucleotide code (http://www.bioinformatics.org/sms2/iupac.html). It may also be caused by unmatch of sequence type (PROTEIN) and actual sequence type (DNA) in FASTA file.

```
 Input: test.fa

 Step 1/5: Reading sequences
 error when reading AB014367_Pol-C: invalid Protein sequence:
AB014367_Pol-C
```

FAQ

Please don't hesitate to email us.

Q: What a mess when opening the result files!

A: Microsoft Windows user may open the result files by Notepad provided by the Operating system.
Please choose another moder text editor like [Notepad++ (https://notepad-plus-plus.org/)](https://notepad-plus-plus.org/).

Authors

Yan Li [liyan.com@gmail.com (mailto:liyan.com@gmail.com)](mailto:liyan.com@gmail.com), Wei Shen [shenwei356@gmail.com (mailto:shenwei356@gmail.com)](mailto:shenwei356@gmail.com)

Copyright