CLEVER 1.0 (Cluster Expansion Version 1.0) is an open-source package that can be used to convert any user-provided energy function into a linear combination of sequence-based terms.

(http://web.mit.edu/biology/keating)

version: 1.0 (Feb 5, 2010)

correspondence:

Amy Keating (keating@mit.edu)

Major Author:

Seungsoo Hahn (hahnss@kaist.ac.kr, coding source and writing docs)

Minor Author:

Orr Ashenberg (writing docs and comment)

Other contributor:

Gevorg Grigoryan (comment)

CLEVER have been developed to carry out cluster expansion method for rapid evaluation of protein energy or other sequence-related properties. The cluster expansion method was first introduced in the paper, Zhou(2005), and used in various papers, i.e. Grigoryan(2006,2009), Apgar(2009), and Hahn(2010). Please, use Hahn(2010) as a reference for this program

References

Hahn(2010), J Comput Chem 2010

Grigoryan(2009), Nature 458, 859, 2009

Apgar(2009), J Comput Chem 30, 2402, 2009

Grigoryan(2006), PLoS Comput Biol 2, e63, 2006

Zhou(2005), Phys Rev Lett 95, 148103, 2005

Software

GenSeqs: generating training/testing sequences from a design file

CETrFile: obtaining CE training results from a sequence/energy file

CEEnergy: evaluating CE energies of new sequences using CE training results

Code was compiled with g++3.4.6 version.

Installation

Note: these installation instructions are written for Unix users, but Windows versions of the GenSeqs, CETrFile, and CEEnergy binary files are provided in clever1.0/compiled/clever-1.0-windows.zip.

1. Run

unzip clever_package.zip

The resulting directory structure is

clever1.0/

compiled/

docs/

docs/webhelp1.0/

source/

test/

tutorial/

2. Obtain executable files

a. using compiled files

Change directory to clever1.0/compiled and run

tar xzvf clever-1.0-linux-binary.tar.gz

This unzips the binary files GenSeqs, CETrFile, and CEEnergy.

b. using source codes

Change directory to clever1.0/source and type “make”

This generates GenSeqs, CETrFile, and CEEnergy.

3. Copy the binary files (GenSeqs, CETrFile, and CEEnergy) to an appropriate directory.

File formats

[Design file]

This file contains information on design-sites and list of clusters (protein structural information). Design files are used to generate sequences for training or testing, and for training of the cluster expansion.

Design-sites information starts with the line “#design_start” and ends with “#design_end”. Between these two reserved strings, design sites should be listed as a line per site. The line consists of a number, starting from 1 and increasing by 1 for each additional design site (the program will ignore the design site numbers what user designate and assign numbers from 1 with increasing by 1 manner. Thus clusters should be described through this number rule), followed by a list of amino acids allowed at the design site (A-Z, capital letters only). Design site information can also be used to generate a random training set by independently drawing residues from the allowed residues at each design site.

Example) design-sites information format

#design_start

```
1    A L I V T M N R Q Y K
2    A E Q K R L N S H T D
3    A E Q K R L N S H T D
#design_end
```

Structural information starts with "#cluster_start" and ends with "#cluster_end". Structural information is written using the clusters: point, pair, and higher-order clusters. Here, all point clusters should be listed, but pair and higher-order clusters can be selected by a user according to some criteria such as mutual information between design sites or structural proximity. Following examples show how to list clusters, the two possible pair clusters, and the single possible triplet cluster for a protein with three design sites.

Example 1) listing simple clusters.

If there are three design sites then cluster information can be listed as following,

```
#cluster_start
  1
  2
  3
  1  2
  1  3
  2  3
  1  2  3
#cluster_end
```

Where first three lines denote point clusters, next two lines denote pair clusters, and the last line denotes triplet cluster.

Example 2) design file incorporating symmetry

In coiled-coil case, all design sites designated by the same heptad position can be approximated to have the same molecular orbital so that those positions generate similar molecular interaction energy. This structural repeating pattern makes the CE approximation more efficient. If we just design 'a','d','e', and 'g' positions among heptad positions then the cluster information of dimeric coiled coils can be described as,

```
#cluster_start
  1; 5;   9; 11; 15; 19
  2; 6; 10; 12; 16; 20
  3; 7; 13; 17
  4; 8; 14; 18
  1 2; 5   6;   9 10; 11 12; 15 16; 19 20
  1 3; 5   7; 11 13; 15 17
  1 4; 5   8; 11 14; 15 18
```

#cluster_end

If the clusters are related by symmetry and the user would like the corresponding cluster functions in that symmetry class to be trained with the same ECI, then those clusters should be listed on the same line and separated by semicolons. In the dimeric coiled coil, a basic unit of symmetry is the heptad and in the given example design file, design site 1 refers to 'a' in heptad 1 and design site 5 refers to 'a' in the next heptad over. By listing their single clusters as,

1

5

the training program will treat these two clusters separately and define separate cluster functions and ECIs. However because of the symmetry between design sites 1 and 5, it makes sense to treat the interactions these clusters make as identical and this can be enforced by listing these single clusters as,

1; 5

For more detailed considerations in choosing design sites and choosing clusters, please see Grigoryan G, Zhou F, Lustig SR, Ceder G, Morgan D, et al. 2006 Ultra-Fast Evaluation of Protein Energies Directly from Sequence. PLoS Comput Biol 2(6): e63.

Coiled-coil design file)

#design_start

1 A L I V T M N R Q Y K

2 A E Q K R L N S H T D

3 A E Q K R L I H V T D

4 A L V I T N K R M C E

5 A L I V T M N R Q Y K

6 A E Q K R L N S H T D

7 A E Q K R L I H V T D

8 A L V I T N K R M C E

9 A L I V T M N R Q Y K

10 A E Q K R L N S H T D

11 A E Q K R L I H V T D

12 A L V I T N K R M C E

13 A L I V T M N R Q Y K

14 A E Q K R L N S H T D

15 A E Q K R L I H V T D

16 A L V I T N K R M C E

17 A L I V T M N R Q Y K

18 A E Q K R L N S H T D

19 A E Q K R L I H V T D

20 A L V I T N K R M C E

#design_end

#cluster_start

1; 5; 9; 11; 15; 19

2; 6; 10; 12; 16; 20

3; 7; 13; 17

4; 8; 14; 18

1 2; 5 6; 9 10; 11 12; 15 16; 19 20

1 3; 5 7; 11 13; 15 17

1 4; 5 8; 11 14; 15 18

1 5; 5 9; 11 15; 15 19

2 3; 6 7; 12 13; 16 17

2 4; 6 8; 12 14; 16 18

2 5; 6 9; 12 15; 16 19

2 6; 6 10; 12 16; 16 20

3 4; 7 8; 13 14; 17 18

3 5; 7 9; 13 15; 17 19

3 6; 7 10; 13 16; 17 20

3 7; 13 17

4 5; 8 9; 14 15; 18 19

4 6; 8 10; 14 16; 18 20

4 7; 14 17

4 8; 14 18

1 11; 5 15; 9 19

1 12; 5 16; 9 20; 11 2; 15 6; 19 10

1 13; 5 17; 11 3; 15 7

1 14; 5 18; 11 4; 15 8

1 15; 5 19; 11 5; 15 9

2 12; 6 16; 10 20

2 13; 6 17; 12 3; 16 7

2 14; 6 18; 12 4; 16 8

2 15; 6 19; 12 5; 16 9

2 16; 6 20; 12 6; 16 10

3 13; 7 17;

3 14; 7 18; 13 4; 17 8

3 15; 7 19; 13 5; 17 9

3 16; 7 20; 13 6; 17 10

3 17; 13 7

4 14; 8 18;

4 15; 8 19; 14 5; 18 9

4 16; 8 20; 14 6; 18 10

4 17; 14 7

4 18; 14 8

#cluster_end

[Sequence file]

Each sequence file lists a set of sequences that can be used to either train an energy estimator using cluster expansion, or to test an already trained energy estimator. Sequence files can be created manually, or by using a design file and the GenSeqs module. To train an energy estimator using CE (with CETrFile), the energies of the associated sequences must be calculated beforehand and placed in the sequence file. To evaluate the energies of new sequences, input the sequence file to the CEEnergy module.

The sequence file contains a list of {energy, sequence} data. A line in this file contains a sequence and its corresponding protein energy (or other sequence-related property). Each line starts with the protein energy, and then the amino acids at each design site in the sequence separated by a space character. Number of amino acids and their order should correspond to the number of design sites and their order. If the sequence is to be evaluated as part of a test set, set the energy to 0. For guidelines on training set size versus number of clusters to be included in training, see Hahn et al. 2010.

Example) sequence file

-197.000 S L T S K T N F A A A S I S L V K S E I

-213.400 K T Q Y K T E S T S S T Q T A N F A E A

-198.300 M S Q I S K T L M K A T V V D N L D V M

Modules

GenSeqs ####

GenSeqs uses a given design file to randomly generate a set of sequences by independently drawing residues from the list of allowed amino acids at each design site. This can be used to either generate a training set or a test set. If generating a test set, a useful option is -o which allows the user to generate a test set that does not share sequences with the training set. Note, the same seed is used for each random

generation of sequences.

Usage :

GenSeqs [OPTIONS]

ex) GenSeqs -n 3000 -d design.file -s sequence.file

GenSeqs -n 3000 -d design.file -s sequence.file -o sequence.old

REQUIRED OPTIONS

-n [number] : number of sequences to be randomly generated

-d [design file] : design file containing design information (number of design sites and list of possible amino acids at each site) and structural information (clusters)

OTHER OPTIONS

-o [old sequence] : old sequence file, additional sequences different from these old sequences are generated and can be used as a test set containing no training set sequences.

-s [sequence file] : result file containing newly generated sequences. If this option is omitted then the generated sequences print on a standard output device.

-h : display this help and exit

-h design : display an example for a design file

CETrFile

CETrFile trains an energy estimator using the cluster expansion method starting from both a design file and a sequence file that includes associated energies (or other sequence-related property). To remove parts of sequence space not well described by the CE method, such sequence groups may be identified and eliminated based on their having cluster functions with large normalized heterogeneity ratios (NH-ratio).

Usage :

CETrFile [OPTIONS]

ex) CETrFile -d design.file -s sequence.file -r training.result

CETrFile -d design.file -s sequence.file -r training.result -n 3000 -b 5 -l output.file

REQUIRED OPTIONS

-d [design file] : design file containing design information (number of design sites and list of possible amino acids at each site) and structural information (clusters)

-s [sequence file] : training set containing sequences and their associated energies

OTHER OPTIONS

-r [training result] : binary results file containing energy estimator trained using CE method (default file name: trainresults.eci)

-l [output file]: log file containing various information generated during training process. If this option is omitted then the log results print on standard output device.

-n [number of sequences] : number of sequences from training set to use, starting from beginning of sequence file (default : number of sequences in sequence file)

-b [d parameter] : a parameter to adjust the stringency of CF selection. Increasing the value of this option lowers the acceptance ratio for selecting new CFs, so that the total number of selected CFs decreases (default 0.0).

-e [number] : number of sequence groups to be eliminated by eliminating sequences containing the cluster functions with maximum NH-ratio (default 0)

-m [number] : minimum number of sequences for a CF

If a training set does not contain this number of sequences for a given CF, then the CF and its associated sequences are excluded from being selected. This value must be 3 or larger (default 5).

-t [mode] : determine how to deal with higher-order CFs (triplet) (default 1)

mode=0 use all triplets listed in design file

mode=1 use all triplets listed in design file and use triplets made from selected pair CFs (if all three possible pair CFs exist between the three sites, include the triplet CF for those three sites).

-v : print version information

-h : display this help and exit

-h design : display an example design file

-h sequence : display an example sequence file

CEEnergy

CEEnergy calculates energies of sequences using an energy estimator previously trained with the cluster expansion method. The number of amino acids in a provided sequence should be equal to the number of design sites.

Usage :

CEEnergy [OPTIONS]

ex) CEEnergy -l output.file -r training.result -s sequence.file

CEEnergy -l output.file -r training.result -u “C D E A …”

REQUIRED OPTIONS

-r [training result] : results file containing energy estimator trained using CE method (or trained

CE)

Either of the options below, “s” or “u”, must be designated.

-s [sequence file] : set of sequences for which to evaluate energy.

Length of sequence should be equal to the number of design sites in trained CE.

-u “single sequence” : user can input a single sequence instead of a sequence file.

Length of sequence should be equal to the number of design sites in trained CE.

OTHER OPTIONS

-l [output file] : file name for standard output

-k : this option executes a routine which checks whether the sequence contains any of the eliminated CFs. If the sequence contains any eliminated CF/CFs, then sequence energy is marked as “NONE” (default is to skip this checking routine).

-i [num] : print energy information

num=0: print only predicted energy (CeEn) (default)

1: print real energy (ReEn, if in the sequence file the user listed the structure-derived or experimentally-derived energy) and predicted energy (CeEn)

2: print real energy (ReEn), predicted energy (CeEn), energy difference (EnDiff=ReEn-CeEn), and scaled energy difference (ScEnDiff)

-p : turn off rapid calculation mode (reducing memory requirement)

-c : print sequence information

-d : print design information

-h : display this help and exit

-h sequence : display an example sequence file