CVTree User Manual

Version 4

Zhao XU <xuzh@fudan.edu.cn>
T-life Research Center,
Department of Physics, Fudan University.
Shanghai 200433, China.

Lei GAO <leigao@ucr.edu>
Department of Botany and Plant Sciences 3133 Batchelor Hall
University of California, Riverside, 900 University Avenue
Riverside, CA 92521,USA.

Hong LUO <hong.luo@wur.nl>
Laboratory of Bioinformatics,
Wageningen University and Research Centre,
PO Box 8128, 6700 ET Wageningen, The Netherlands.

Ji QI <qij@bx.psu.edu>
Penn State University,
Center for Comparative Genomics and Bioinformatics,
310 Wartik Building, University Park, PA 16802, USA.

Bailin HAO*<hao@mail.itp.ac.cn>
T-life Research Center,
Department of Physics, Fudan University.
Shanghai 200433, China.

Institute of Theoretical Physics,
Academia Sinica Beijing 100080, China.

*Corresponding author

CVTree is an alignment free program which generates a dissimilarity matrix from comparatively large collection of DNA or Amino Acid sequences, preferably whole genome data.

1 Algorithm

The algorithm of CVTree consists of the following steps:

1. Fix the string length K (K∈[3,7] for AA sequence, K∈[3,16] for nucleotide sequence in 32-bit system). Read in the sequence collection of each species separately, count the total number of each K, K-1 and K-2 tuples.

2. Calculate the subtraction value for the i-th K tuple:

$$a_i(a_1a_2\cdots a_K) \equiv \frac{f(a_1a_2\cdots a_K) - f^0(a_1a_2\cdots a_K)}{f^0(a_1a_2\cdots a_K)}$$

   where $f(a_1a_2\cdots a_K)$ is the frequency of K-tuple, $f^0(a_1a_2\cdots a_K)$ is the frequency predicted from that of K-1 and K-2 tuples by using a (K-2)-th Markov assumption.

3. Calculate the pairwise dissimilarity:

$$C(\vec{CV_A}, \vec{CV_B}) = \frac{\sum_{i=1}^{N} A_i \times B_i}{(\sum_{i=1}^{N} A_i^2 \times \sum_{i=1}^{N} B_i^2)^{\frac{1}{2}}}$$

4. Then the user can use this dissimilarity matrix to plot a phylogenetic tree by calling the appropriate program in PHYLIP or MEGA.

For more detailed description see [Qi *et al.*, 2004*b*] and [HAO and QI, 2004].

2 User's Guide

2.1 Installation

The main program was implemented in C++. For most purposes, the C++ program `cvtree` is enough for the end user. However we supply some Perl scripts to treat extremely massive input data (e.g., exceeding several gigabytes). If you encounter "Out of memory" when running CVTree program by `-o` or `-l` option, you can try to use `-c` option instead. Option `-c` will output separated CV files into the given directory, which can be used by `batch_dist.pl` to calculate the final dissimilarity matrix. In using `batch_dist.pl`, you can set the memory limitation through command option. So in general, only GCC 4.x or higher is needed, but if you wish to have additional functions, you will need Perl as well.

The installation is very simple, just type "make" command in the CVTree source directory, you will get a runnable binary file named `cvtree`.

If you want a even simpler way to use CVTree, you can visit our web server at `http://tlife.fudan.edu.cn/cvtree`, which is an improved version of the original CVTree web server published in NAR ([Qi *et al.*, 2004*a*]).

2.2 Examples

- Calculate dissimilarity matrix from single input file.
  Input data file:

  ```
  >SeqID1 | Organism name
  SequenceSequenceSequence...
  >SeqID2 | Organism name
  SequenceSeq...
  ...
  ```

  Note: There can be multiple sequences for one Organism.

  Command(K=10, dna sequence for example):

  ```
  cvtree -s input.fa -k10 -t dna -o PHYLIP_format_matrix
  ```

  or

  ```
  cvtree -s input.fa -k10 -t dna -l MEGA_format_matrix
  ```

- Calculate dissimilarity matrix from multiple files in one directory.
  Input name list file:

  ```
  OrganismTotalCount
  Organism1
  Organism2
  ...
  ```

Input directory content:

```
  Organism1.faa
  Organism2.faa
  ...
```

Note: If you want to process DNA sequence, please make sure the suffix name of each FASTA file is .ffn (e.g., Organism1.ffn).

Command(K=6, protein sequence for example):

```
cvtree -i name_list.txt -d input_dir -k6 -t aa -o
  PHYLIP_format_matrix
```

or

```
cvtree -i name_list.txt -d input_dir -k6 -t aa -l
  MEGA_format_matrix
```

- Calculate dissimilarity matrix under memory limitation.

  1. Generate CV files(K=5, protein sequence for example)

     ```
     cvtree -i name_list.txt -d input_dir/ -k5 -taa \
                             -c out_cv_dir/
     ```

     or

     ```
     cvtree -s input.fa -k5 -taa -c out_cv_dir/
     ```

  2. Calculate dissimilarity matrix(set memory limit to 1.5G for example)

     ```
     batch_dist.pl 1.5 name_list.txt out_cv_dir/ \
                  output_PHYLIP_format_matrix
     ```

     Note: User can get name_list.txt from input.fa(the single input file) by issuing command "get_name_list.pl"

  3. To convert PHYLIP dissimilarity matrix format to MEGA matrix format, use:

     ```
     phylip2mega.pl PHYLIP_format_matrix \
                    output_MEGA_format_matrix
     ```

2.3 Options

1. cvtree – Generate dissimilarity matrix or CV files from input data

```
 cvtree -i list.file -d seq.dir/ \
        -k num -t dna/aa -o out.dist.file
VERSION: 4.0
  -i  str   input list file name, contains
            sequences' name(without .faa/.ffn)
  -d  str   input sequences' dir
  -s  str   single input mutil-FASTA file
            (will ignore -i and -d)
  -c  str   output CV files' dir
  -C  str   output TXT CV files' dir(for human read)
  -k  num   k-tuple's length, AA range: [3,7],
            DNA range: [3,16]
  -t  str   sequence type: dna or aa
  -S        Turn on Subtraction(default is off)
  -o  str   output distance matrix filename(PHYLIP matrix)
  -l  str   output distance matrix filename
            (MEGA lower-left matrix)
  -q        be quiet
  -h        print help info
```

2. batch_dist.pl – Generate dissimilarity matrix from CV files

```
 ./batch_dist.pl mem_limit(GByte) new_list.file \
         new_cv.dir out_matrix.file  \
         [exists_list.file exists_cv.dir exists_matrix.file]
  (please give 4 parameters or >=7 paramters.)
VERSION 4.0
```

3. dist – Calculate dissimilarity value from CV files

```
  dist -d db_file1.cv,db_file2.cv \
          q_file1.cv q_file2.cv > distance
VERSION: 4.0
  -n  num   number of query CVs, and load query CV from stdin
  -d  str   load database CV, seperate by ','
  -c        calculate distance between database CVs
  -i        input two file(db_file and q_file) from stdin
```

```
          note: -s,-d,-c,-n will be ignored
-s        output f1-f2_k.xls file for statistics
-f  str   statistics file's name, used with -i
          (coz -s will be disabled)
-t  str   sequence type: dna or aa (affect -s/-f)
-h        print help info
examples:
    dist -d file1.cv file2.cv
    cat file1.cv file2.cv | dist -i
    cat qfile1 qfile2 | dist -d dfile1,dfile2 -n2
```

4. phylip2mega.pl – Convert PHYLIP distance matrix to MEGA distance matrix

```
phylip2mega.pl PHYLIP.matrix MEGA.matrix
```

5. get_name_list.pl – Generate name_list.txt form single input data file

```
get_name_list.pl single_input.fa name_list.txt
```

References

[Bai-Lin and Lei, 2008] Bai-Lin,H.A.O. and Lei,G.A.O. (2008) Prokaryotic branch of the tree of life: a composition vector approach. *Journal of Systematics and Evolution,* **46**, 258–262.

[Gao and Qi, 2007] Gao,L. and Qi,J. (2007) Whole genome molecular phylogeny of large dsdna viruses using composition vector method. *BMC Evolutionary Biology,* **7**, 41.

[Gao *et al.*, 2007] Gao,L., Qi,J., Sun,J.D. and Hao,B.L. (2007) Prokaryote phylogeny meets taxonomy: an exhaustive comparison of composition vector trees with systematic bacteriology. *Science in China Series C: Life Sciences,* **50**, 587–599.

[GAO *et al.*, 2003] GAO,L., QI,J., WEI,H., SUN,Y. and HAO,B. (2003) Molecular phylogeny of coronaviruses including human sars-cov. *Chinese Science Bulletin,* **48**, 1170–1174.

[HAO and QI, 2004] HAO,B. and QI,J. (2004) Prokaryote phylogeny without sequence alignment: from avoidance signature to composition distance. *Journal of Bioinformatics and Computational Biology,* **2**, 1–19.

[Qi *et al.*, 2004*a*] Qi,J., Luo,H. and Hao,B. (2004*a*) Cvtree: a phylogenetic tree reconstruction tool based on whole genomes. *Nucleic acids research,* **32**, W45–7. PMID: 15215347.

[Qi *et al.*, 2004*b*] Qi,J., Wang,B. and Hao,B.L. (2004*b*) Whole proteome prokaryote phylogeny without sequence alignment: a k-string composition approach. *Journal of Molecular Evolution,* **58**, 1–11.

[Yu *et al.*, 2005] Yu,Z.G., Zhou,L.Q., Anh,V.V., Chu,K.H., Long,S.C. and Deng,J.Q. (2005) Phylogeny of prokaryotes and chloroplasts revealed by a simple composition approach on all protein sequences from complete genomes without sequence alignment. *Journal of Molecular Evolution,* **60**, 538–545. PMID: 15883888.