CVTree
(Standalone Version)
User's Manual

Guanghong Zuo and Bailin HAO

June 28, 2018

CVTree stands for **Composition Vector Tree** which is the implementation of an alignment-free algorithm to generate a dissimilarity matrix from comparatively large collection of DNA or Amino Acid sequences, preferably whole-genome data, for phylogenetic studies.

There are two versions of the program:

1. **CVTree Web Server** which has been published twice in the Web Server Issues of *Nucleic Acids Research*, [Qi *et al.*, 2004*a*] and [Xu and Hao, 2009]. The lastest CVTree Web Server, CVTree3 Web Server [Zuo and Hao, 2015], have two identical but independent installations:
   - Fudan University, Shanghai:
     `http://tlife.fudan.edu.cn/cvtree3/`
   - Beijing Institute of Genomics, Beijing:
     `http://cvtree.cn`
     `http://bigd.big.ac.cn/cvtree`
2. **CVTree Standalone Version** which is provided to those who are interested in the intermediate results, e.g., the collection of all CVs, or deal with extremely huge datasets of their own. We provide also a few options and scripts that were not available in the Web Server versions.

1 The Algorithm

The algorithm of CVTree consists of the following steps:

1. Fix a string length K ($K \in [3, 7]$ for Amino Acid sequences and $K \in [3, 16]$ for nucleotide sequences in 32-bit system). Read in the sequence collection of each species separately. Count the number of all K, $K-1$ and $K-2$ tuples for a species. A *raw* Composition Vector (CV) of dimension 4^K or 20^K is formed by putting the counts of K-tuples in lexicographic order.

2. Calculate the subtraction score 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, [Qi *et al.*, 2004*b*]. All components of the *raw* CV is replaced by its subtraction score to yield a renormalized CV.

3. Using the renormalized CVs to calculate the pairwise dissimilarity between two species:

$$d(A, B) = (1 - C(\vec{CV}_A, \vec{CV}_B))/2,$$

   where

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

4. Then obtain the phylogenetic tree (Newick Format) based on this dissimilarity matrix by Neighbor Joint method.

For more detailed description of the algorithm please consult [Qi *et al.*, 2004*b*].

2 The Installation

2.1 Preparation

- cmake $\geq$ 2.6
- g++ $\geq$ 4.8 or other compiler supporting C++11 standard
- compiler with support openmp for parallel
- Library: libz, netcdf, netcdf_cpp

2.2 Compiling

Unzip or checkout the source files, Obtained the compiling option by cmake

- unzip the package file and change into it
- mkdir build and change into it
- cmake ../src or add some options you wanted, e.g.:
  -DCMAKE_INSTALL_PREFIX=/usr/local
- make
- make install

3 Programs and Command-Line Options

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" warning 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 `bdist`(or `batch_dist.pl`) to calculate the final dissimilarity matrix.

1. cv – Generate CV files from input data

```
cv [ -I faa ]          input genome file directory, defaut: faa
   [ -i list ]         input species list, defaut: list
   [ -k '3 4 5 6 7' ]  values of k, defaut: N = 3 4 5 6 7
   [ -g faa ]          the type of genome file, defaut: faa
   [ -O cv ]           output cv directory, defaut: cv
   [ -S 0/1]           whethe do the subtract, defaut: 1
   [ -h ]              disply this information
```

2. tree – Generate newick tree based on CV files

```
tree
 [ -o dist.matrix ]   Output distance matrix,
                      defaut: dist.matrix
 [ -I extdir ]        Directory of extend cv files,
                      defaut: cv
```

```
[ -i infile ]        Extend cv file list,
                     defaut: no extend cv used
[ -s cv6.gz ]        Suffix of cv file,
                     defaut: cv6.gz
[ -E orgdir  ]       Directory of the orginal cv files
[ -m indist.matrix ] Input distance matrix,
                     default: no input matrix used
[ -e orglist ]       Name of selected genomes,
                     which are in input distance matrix
[ -n ndxlist ]       Index of selected genomes,
                     which are input distance matrix
                     The index file used first!
[ -t taxfile ]       input taxonomy information
[ -T ]               Do not output taxonomy information
[ -M <N> ]           Runing memory size as G roughly,
                     default 80% physical memory
[ -C ]               Force use the netcdf compress distance matrix
[ -h ]               disply this information
```

3. cvdump – convect the binary cv file to acsii file

```
cvdump  -i <cvfile>        input file name
 [ -g faa|ffn ]      the type of genome file, defaut: faa
 [ -n ]              output the number code, default: the letters
 [ -h ]              disply this information
```

4. runCVTree.sh (in example folder) – Easy script to obtain the result.

4 Citing CVTree in a Publication

Please cite:

1. Ji Qi, Bin Wang, Bailin Hao (2004), Whole proteome prokaryote phylogeny without sequence alignment: a K-string composition approach, *Journal of Molecular Evolution*, **58**: 1 – 11.
2. Guanghong Zuo and Bailin Hao (2015) CVTree3 Web Server for Whole-genome-based and Alignment-free Prokaryotic Phylogeny and Taxonomy. *Genomics Proteomics Bioinformatics,* **13**: 321–331.

5 Version History and Contributors

Since 2002 the CVTree software has undergone many revisions. A few major versions were:

1. Web Server CVTree 1.0 was written by Ji Qi, Hong Luo and Bailin Hao
2. Most 3.x versions of Standalone CVTree was written by Lei Gao; Ver. 3.9.6 was written by Ji Qi.
3. Web Server CVTree 2.0 was written by Zhao Xu and Bailin Hao
4. Standalone CVTree 4.4 was written by Zhao Xu
5. Web Server CVTree 3.0 was written by Guanghong Zuo and Bailin Hao
6. Standalone CVTree 5.0 was written by Guanghong Zuo

References

[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. (2004*b*) Whole proteome prokaryote phylogeny without sequence alignment: a k-string composition approach. *Journal of Molecular Evolution,* **58**, 1–11.

[Xu and Hao, 2009] Xu,Z. and Hao,B. (2009) Cvtree update: a newly designed phylogenetic study platform using composition vectors and whole genomes. *Nucleic Acids Research,* **37 Web Server Issue**, W174–W178.

[Zuo and Hao, 2015] Zuo,G. and Hao,B. (2015) CVTree3 Web Server for Whole-genome-based and Alignment-free Prokaryotic Phylogeny and Taxonomy. *Genomics Proteomics Bioinformatics,* **13**, 321–331.