

# Package ‘fastreeR’

August 13, 2025

**Type** Package

**Title** Phylogenetic, Distance and Other Calculations on VCF and Fasta Files

**Version** 1.99.4

**biocViews** Phylogenetics, Metagenomics, Clustering

**Description** Calculate distances, build phylogenetic trees or perform hierarchical clustering between the samples of a VCF or FASTA file. Functions are implemented in Java-11 and called via rJava. Parallel implementation that operates directly on the VCF or FASTA file for fast execution.

**License** GPL-3

**Encoding** UTF-8

**Language** en-US

**LazyData** false

**Depends** R (>= 4.4)

**Imports** ape, data.table, dynamicTreeCut, methods, R.utils, rJava, stats, stringr, utils

**SystemRequirements** Java (>= 11)

**RoxygenNote** 7.3.2

**URL** <https://github.com/gkanogiannis/fastreeR>,  
<https://github.com/gkanogiannis/BioInfoJava-Utills>

**BugReports** <https://github.com/gkanogiannis/fastreeR/issues>

**Suggests** BiocFileCache, BiocStyle, graphics, knitr, memuse, rmarkdown, spelling, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/fastreeR>

**git\_branch** devel

**git\_last\_commit** 815fd6f

**git\_last\_commit\_date** 2025-08-08

**Repository** Bioconductor 3.22

**Date/Publication** 2025-08-13

**Author** Anestis Gkanogiannis [aut, cre] (ORCID:  
    <<https://orcid.org/0000-0002-6441-0688>>)  
**Maintainer** Anestis Gkanogiannis <[anestis@gkanogiannis.com](mailto:anestis@gkanogiannis.com)>

Contents

|                            |           |
|----------------------------|-----------|
| fastreeR-package . . . . . | 2         |
| dist2clusters . . . . .    | 3         |
| dist2tree . . . . .        | 4         |
| fasta2dist . . . . .       | 5         |
| tree2clusters . . . . .    | 6         |
| vcf2clusters . . . . .     | 7         |
| vcf2dist . . . . .         | 8         |
| vcf2istats . . . . .       | 10        |
| vcf2tree . . . . .         | 11        |
| <b>Index</b>               | <b>13</b> |

---

|                  |                                                                                       |
|------------------|---------------------------------------------------------------------------------------|
| fastreeR-package | <i>fastreeR: Phylogenetic, Distance and Other Calculations on VCF and Fasta Files</i> |
|------------------|---------------------------------------------------------------------------------------|

---

Description

Calculate distances, build phylogenetic trees or perform hierarchical clustering between the samples of a VCF or FASTA file. Functions are implemented in Java-11 and called via rJava. Parallel implementation that operates directly on the VCF or FASTA file for fast execution.

Author(s)

**Maintainer:** Anestis Gkanogiannis <[anestis@gkanogiannis.com](mailto:anestis@gkanogiannis.com)> (**ORCID**)

See Also

Useful links:

- <https://github.com/gkanogiannis/fastreeR>
- <https://github.com/gkanogiannis/BioInfoJava-Utils>
- Report bugs at <https://github.com/gkanogiannis/fastreeR/issues>

---

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| dist2clusters | <i>Perform Hierarchical Clustering and tree pruning on a distance matrix</i> |
|---------------|------------------------------------------------------------------------------|

---

## Description

Performs Hierarchical Clustering on a distance matrix (i.e. calculated with [vcf2dist](#) or [fasta2dist](#)) and generates a phylogenetic tree with agglomerative Neighbor Joining method (complete linkage) (as in [dist2tree](#)). The phylogenetic tree is then pruned with [cutreeDynamic](#) to get clusters (as in [tree2clusters](#)).

## Usage

```
dist2clusters(
  inputDist,
  cutHeight = NULL,
  minClusterSize = 1,
  extra = TRUE,
  verbose = FALSE
)
```

## Arguments

|                |                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| inputDist      | Input distances file location (generated with <a href="#">vcf2dist</a> or <a href="#">fasta2dist</a> ). File can be gzip compressed. Or a <a href="#">dist</a> distances object. |
| cutHeight      | Define at which height to cut tree. Default automatically defined.                                                                                                               |
| minClusterSize | Minimum size of clusters. Default 1.                                                                                                                                             |
| extra          | Boolean whether to use extra parameters for the <a href="#">cutreeDynamic</a> .                                                                                                  |
| verbose        | Logical. If TRUE, enables verbose output from the Java backend.                                                                                                                  |

## Value

A list of :

- [character](#) vector of the generated phylogenetic tree in Newick format
- [character](#) vector of the clusters. Each row contains data for a cluster, separated by space. The id of the cluster, the size of the cluster (number of elements) and the names of its elements, Cluster id 0 contains all the objects not assigned to a cluster (singletons). Example clusters output :

```
0  3  Sample1  Sample2  Sample3
1  3  Sample4  Sample5  Sample6
2  2  Sample7  Sample8
3  2  Sample9  Sample0
```

## Author(s)

Anestis Gkanogiannis, <[anestis@gkanogiannis.com](mailto:anestis@gkanogiannis.com)>

## References

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utils>

## Examples

```
my.clust <- dist2clusters(
  inputDist =
    system.file("extdata", "samples.vcf.dist.gz", package = "fastreeR"),
  verbose = TRUE
)
```

---

|           |                                                                     |
|-----------|---------------------------------------------------------------------|
| dist2tree | <i>Generate phylogenetic tree from samples of a distance matrix</i> |
|-----------|---------------------------------------------------------------------|

---

## Description

Performs Hierarchical Clustering on a distance matrix (i.e. calculated with [vcf2dist](#) or [fasta2dist](#)) and generates a phylogenetic tree with agglomerative Neighbor Joining method (complete linkage).

## Usage

```
dist2tree(inputDist, verbose = FALSE)
```

## Arguments

|           |                                                                                                                                                                                  |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| inputDist | Input distances file location (generated with <a href="#">vcf2dist</a> or <a href="#">fasta2dist</a> ). File can be gzip compressed. Or a <a href="#">dist</a> distances object. |
| verbose   | Logical. If TRUE, enables verbose output from the Java backend.                                                                                                                  |

## Value

A [character](#) vector of the generated phylogenetic tree in Newick format.

## Author(s)

Anestis Gkanogiannis, <[anestis@gkanogiannis.com](mailto:anestis@gkanogiannis.com)>

## References

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utils>

## Examples

```
my.tree <- dist2tree(
  inputDist =
    system.file("extdata", "samples.vcf.dist.gz", package = "fastreeR"),
  verbose = TRUE
)
```

---

fasta2dist*Calculate distances between sequences of a FASTA file*

---

### Description

This function calculates a d2\_S type dissimilarity measurement between the n sequences (which can represent samples) of a FASTA file. See [doi:10.1186/s1285901611863](https://doi.org/10.1186/s1285901611863) for more details.

### Usage

```
fasta2dist(  
  ...,  
  outputFile = NULL,  
  threads = 2,  
  kmer = 6,  
  normalize = FALSE,  
  compress = TRUE,  
  verbose = FALSE  
)
```

### Arguments

|            |                                                                 |
|------------|-----------------------------------------------------------------|
| ...        | Input fasta files locations (uncompressed or gzip compressed).  |
| outputFile | Output distances file location.                                 |
| threads    | Number of java threads to use.                                  |
| kmer       | Kmer length to use for analyzing fasta sequences.               |
| normalize  | Normalize on sequences length.                                  |
| compress   | Compress output (adds .gz extension).                           |
| verbose    | Logical. If TRUE, enables verbose output from the Java backend. |

### Value

A [dist](#) distances object of the calculation.

### Author(s)

Anestis Gkanogiannis, <[anestis@gkanogiannis.com](mailto:anestis@gkanogiannis.com)>

### References

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utills>

### Examples

```
my.dist <- fasta2dist(  
  inputfile = system.file("extdata", "samples.fasta.gz",  
    package = "fastreeR"  
  )  
)
```

---

|               |                                                                                |
|---------------|--------------------------------------------------------------------------------|
| tree2clusters | <i>Perform Hierarchical Clustering and tree pruning on a phylogenetic tree</i> |
|---------------|--------------------------------------------------------------------------------|

---

## Description

The phylogenetic tree is pruned with `cutreeDynamic` to get clusters.

## Usage

```
tree2clusters(
  treeStr,
  treeDistances = NULL,
  treeLabels = NULL,
  cutHeight = NULL,
  minClusterSize = 1,
  extra = TRUE,
  verbose = FALSE
)
```

## Arguments

|                |                                                                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------|
| treeStr        | A <code>character</code> vector of a phylogenetic tree in Newick format                                                               |
| treeDistances  | numeric <code>matrix</code> of distances, that were used to generate the tree. If NULL, it will be inferred from tree branch lengths. |
| treeLabels     | A <code>character</code> vector of tree leaf labels.                                                                                  |
| cutHeight      | Define at which height to cut tree. Default automatically defined.                                                                    |
| minClusterSize | Minimum size of clusters. Default 1.                                                                                                  |
| extra          | Boolean whether to use extra parameters for the <code>cutreeDynamic</code> .                                                          |
| verbose        | Logical. If TRUE, enables verbose output from the Java backend.                                                                       |

## Value

- `character` vector of the clusters. Each row contains data for a cluster, separated by space. The id of the cluster, the size of the cluster (number of elements) and the names of its elements, Cluster id 0 contains all the objects not assigned to a cluster (singletons). Example clusters output :

```
0  3  Sample1  Sample2  Sample3
1  3  Sample4  Sample5  Sample6
2  2  Sample7  Sample8
3  2  Sample9  Sample0
```

## Author(s)

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

## References

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utills>

## Examples

```
my.clust <- tree2clusters(
  treeStr = dist2tree(
    inputDist = system.file("extdata", "samples.vcf.dist.gz",
      package = "fasttreeR"
    )
  ),
  verbose = TRUE
)
```

---

|              |                                                                                |
|--------------|--------------------------------------------------------------------------------|
| vcf2clusters | <i>Perform Hierarchical Clustering and tree pruning on samples of VCF file</i> |
|--------------|--------------------------------------------------------------------------------|

---

## Description

Performs Hierarchical Clustering on a distance matrix calculated as in [vcf2dist](#) and generates a phylogenetic tree with agglomerative Neighbor Joining method (complete linkage) (as in [dist2tree](#)). The phylogenetic tree is then pruned with [cutreeDynamic](#) to get clusters (as in [tree2clusters](#)).

## Usage

```
vcf2clusters(
  inputFile,
  threads = 2,
  cutHeight = NULL,
  minClusterSize = 1,
  extra = TRUE,
  verbose = FALSE
)
```

## Arguments

|                |                                                                                 |
|----------------|---------------------------------------------------------------------------------|
| inputFile      | Input vcf file location (uncompressed or gzip compressed).                      |
| threads        | Number of java threads to use.                                                  |
| cutHeight      | Define at which height to cut tree. Default automatically defined.              |
| minClusterSize | Minimum size of clusters. Default 1.                                            |
| extra          | Boolean whether to use extra parameters for the <a href="#">cutreeDynamic</a> . |
| verbose        | Logical. If TRUE, enables verbose output from the Java backend.                 |

## Details

Biallelic or multiallelic (maximum 7 alternate alleles) SNP and/or INDEL variants are considered, phased or not. Some VCF encoding examples are:

- heterozygous variants : 1/0 or 0/1 or 0/2 or 1|0 or 0|1 or 0|2
- homozygous to the reference allele variants : 0/0 or 0|0
- homozygous to the first alternate allele variants : 1/1 or 1|1

If there are  $n$  samples and  $m$  variants, an  $n \times n$  zero-diagonal symmetric distance matrix is calculated. The calculated cosine type distance  $(1 - \text{cosine\_similarity})/2$  is in the range  $[0,1]$  where value 0 means completely identical samples (cosine is 1), value 0.5 means perpendicular samples (cosine is 0) and value 1 means completely opposite samples (cosine is -1).

The calculation is performed by a Java back-end implementation, that supports multi-core CPU utilization and can be demanding in terms of memory resources. By default a JVM is launched with a maximum memory allocation of 512 MB. When this amount is not sufficient, the user needs to reserve additional memory resources, before loading the package, by updating the value of the `java.parameters` option. For example in order to allocate 4GB of RAM, the user needs to issue `options(java.parameters="-Xmx4g")` before `library(fastreeR)`.

## Value

A list of :

- `dist` distances object.
- `character` vector of the generated phylogenetic tree in Newick format
- `character` vector of the clusters. Each row contains data for a cluster, separated by space. The id of the cluster, the size of the cluster (number of elements) and the names of its elements. Cluster id 0 contains all the objects not assigned to a cluster (singletons). Example clusters output :

|   |   |         |         |         |
|---|---|---------|---------|---------|
| 0 | 3 | Sample1 | Sample2 | Sample3 |
| 1 | 3 | Sample4 | Sample5 | Sample6 |
| 2 | 2 | Sample7 | Sample8 |         |
| 3 | 2 | Sample9 | Sample0 |         |

## Author(s)

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

## References

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utills>

## Examples

```
my.clust <- vcf2clusters(
  inputFile = system.file("extdata", "samples.vcf.gz",
    package = "fastreeR"
  )
)
```

---

vcf2dist

---

*Calculate distances between samples of a VCF file*


---

## Description

This function calculates a cosine type dissimilarity measurement between the  $n$  samples of a VCF file.

## Usage

```
vcf2dist(
  inputFile,
  outputFile = NULL,
  threads = 2,
  compress = FALSE,
  verbose = FALSE
)
```

## Arguments

|            |                                                                 |
|------------|-----------------------------------------------------------------|
| inputFile  | Input vcf file location (uncompressed or gzip compressed).      |
| outputFile | Output distances file location.                                 |
| threads    | Number of java threads to use.                                  |
| compress   | Compress output (adds .gz extension).                           |
| verbose    | Logical. If TRUE, enables verbose output from the Java backend. |

## Details

Biallelic or multiallelic (maximum 7 alternate alleles) SNP and/or INDEL variants are considered, phased or not. Some VCF encoding examples are:

- heterozygous variants : 1/0 or 0/1 or 0/2 or 1|0 or 0|1 or 0|2
- homozygous to the reference allele variants : 0/0 or 0|0
- homozygous to the first alternate allele variants : 1/1 or 1|1

If there are  $n$  samples and  $m$  variants, an  $n \times n$  zero-diagonal symmetric distance matrix is calculated. The calculated cosine type distance  $(1 - \text{cosine\_similarity})/2$  is in the range  $[0,1]$  where value 0 means completely identical samples (cosine is 1), value 0.5 means perpendicular samples (cosine is 0) and value 1 means completely opposite samples (cosine is -1).

The calculation is performed by a Java backend implementation, that supports multi-core CPU utilization and can be demanding in terms of memory resources. By default a JVM is launched with a maximum memory allocation of 512 MB. When this amount is not sufficient, the user needs to reserve additional memory resources, before loading the package, by updating the value of the `java.parameters` option. For example in order to allocate 4GB of RAM, the user needs to issue `options(java.parameters="-Xmx4g")` before `library(fastreeR)`.

Output file, if provided, will contain  $n+1$  lines. The first line contains the number  $n$  of samples and number  $m$  of variants, separated by space. Each of the subsequent  $n$  lines contains  $n+1$  values, separated by space. The first value of each line is a sample name and the rest  $n$  values are the calculated distances of this sample to all the samples. Example output file of the distances of 3 samples calculated from 1000 variants:

```
3 1000
Sample1 0.0 0.5 0.2
Sample2 0.5 0.0 0.9
Sample3 0.2 0.9 0.0
```

## Value

A `dist` distances object of the calculation.

**Author(s)**

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

**References**

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utills>

**Examples**

```
my.dist <- vcf2dist(
  inputFile = system.file("extdata", "samples.vcf.gz",
    package = "fastreeR"
  )
)
```

---

vcf2istats

---

*Calculate various per sample statistics from a VCF file*


---

**Description**

Only biallelic SNPs are considered. For each sample, the following statistics are calculated :

- INDIV : Sample name
- N\_SITES : Total number of SNPs
- N\_HET : Number of SNPs with heterozygous call (0/1 or 0|1 or 1/0 or 1|0)
- N\_ALT : Number of SNPs with alternate homozygous call (1/1 or 1|1)
- N\_REF : Number of SNPs with reference homozygous call (0/0 or 0|0)
- N\_MISS : Number of SNPs with missing call (./. or .|.)
- P\_HET : Percentage of heterozygous calls
- P\_ALT : Percentage of alternate homozygous calls
- P\_REF : Percentage of reference homozygous calls
- P\_MISS : Percentage of missing calls (missing rate)

**Usage**

```
vcf2istats(inputFile, outputFile = NULL)
```

**Arguments**

|            |                                                            |
|------------|------------------------------------------------------------|
| inputFile  | Input vcf file location (uncompressed or gzip compressed). |
| outputFile | Output samples statistics file location.                   |

**Value**

A [data.frame](#) of sample statistics.

**Author(s)**

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

## References

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utills>

## Examples

```
my.istats <- vcf2istats(
  inputFile =
    system.file("extdata", "samples.vcf.gz", package = "fastreeR")
)
```

---

|          |                                                              |
|----------|--------------------------------------------------------------|
| vcf2tree | <i>Generate phylogenetic tree from samples of a VCF file</i> |
|----------|--------------------------------------------------------------|

---

## Description

This function calculates a distance matrix between the samples of a VCF file as in [vcf2dist](#) and performs Hierarchical Clustering on this distance matrix as in [dist2tree](#). A phylogenetic tree is calculated with agglomerative Neighbor Joining method (complete linkage).

## Usage

```
vcf2tree(inputFile, threads = 2, verbose = FALSE)
```

## Arguments

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| inputFile | Input vcf file location (uncompressed or gzip compressed).      |
| threads   | Number of java threads to use.                                  |
| verbose   | Logical. If TRUE, enables verbose output from the Java backend. |

## Details

Biallelic or multiallelic (maximum 7 alternate alleles) SNP and/or INDEL variants are considered, phased or not. Some VCF encoding examples are:

- heterozygous variants : 1/0 or 0/1 or 0/2 or 1|0 or 0|1 or 0|2
- homozygous to the reference allele variants : 0/0 or 0|0
- homozygous to the first alternate allele variants : 1/1 or 1|1

If there are  $n$  samples and  $m$  variants, an  $n \times n$  zero-diagonal symmetric distance matrix is calculated. The calculated cosine type distance  $(1 - \text{cosine\_similarity})/2$  is in the range  $[0,1]$  where value 0 means completely identical samples (cosine is 1), value 0.5 means perpendicular samples (cosine is 0) and value 1 means completely opposite samples (cosine is -1).

The calculation is performed by a Java backend implementation, that supports multi-core CPU utilization and can be demanding in terms of memory resources. By default a JVM is launched with a maximum memory allocation of 512 MB. When this amount is not sufficient, the user needs to reserve additional memory resources, before loading the package, by updating the value of the `java.parameters` option. For example in order to allocate 4GB of RAM, the user needs to issue `options(java.parameters="-Xmx4g")` before `library(fastreeR)`.

**Value**

A **character** vector of the generated phylogenetic tree in Newick format.

**Author(s)**

Anestis Gkanogiannis, <anestis@gkanogiannis.com>

**References**

Java implementation: <https://github.com/gkanogiannis/BioInfoJava-Utils>

**Examples**

```
my.tree <- vcf2tree(  
  inputFile = system.file("extdata", "samples.vcf.gz",  
    package = "fastreeR"  
  )  
)
```

# Index

## **\* internal**

fasttreeR-package, [2](#)

character, [3](#), [4](#), [6](#), [8](#), [12](#)

cutreeDynamic, [3](#), [6](#), [7](#)

data.frame, [10](#)

dist, [3–5](#), [8](#), [9](#)

dist2clusters, [3](#)

dist2tree, [3](#), [4](#), [7](#), [11](#)

fasta2dist, [3](#), [4](#), [5](#)

fasttreeR (fasttreeR-package), [2](#)

fasttreeR-package, [2](#)

matrix, [6](#)

tree2clusters, [3](#), [6](#), [7](#)

vcf2clusters, [7](#)

vcf2dist, [3](#), [4](#), [7](#), [8](#), [11](#)

vcf2istats, [10](#)

vcf2tree, [11](#)