Build B cell lineage trees

Kenneth B. Hoehn

2022-10-04

Contents

Dowser offers multiple ways to build B cell phylogenetic trees. These differ by the method used to estimate tree topology and branch lengths (e.g. maximum parsimony and maximum likelihood) and implementation (IgPhyML, PHYLIP, or R packages ape and phangorn).

Before trees can be built, B cell sequences must be separated into clonal clusters, and had their clonal germline sequences reconstructed. Default settings assume input data is in AIRR TSV format, though column names may be specified using function arguments.

Format clones

Before trees can be built, data must be formatted into a data table of AIRR clone objects. This is accomplished using the formatClones function. This function will:

1. Change non-nucleotide characters to `N` characters.
2. By default, collapse sequences that are either identical or differ only by ambiguous characters.
3. Sequences will not be collapsed if they differ by columns specified in the `traits` option, or if the `collapse` option is set to `FALSE`.
4. Include data columns specified by `num_fields` or `text_fields`.
5. Remove uninformative sequence sites in which all sequences have `N` characters.

The output of this function is a tibble in which each row is a clone, ordered by the number of sequences. The column `data` contains airrClone objects with the clonal sequence alignments. Other columns contain information about the clone, and can be specified using the `columns` argument.

```
library(dowser)

# load example AIRR tsv data
data(ExampleAirr)

# Subset data for this example
ExampleAirr = ExampleAirr[ExampleAirr$clone_id %in% c("3170", "3184"),]
ExampleAirr$subject_id = "Subject_1"
```

```
# Process example data using default settings
clones = formatClones(ExampleAirr)

print(clones)
```

```
## # A tibble: 2 x 4
##   clone_id data       locus  seqs
##      <dbl> <list>     <chr> <int>
## 1     3170 <airrClon> N        13
## 2     3184 <airrClon> N        12
```

```
# Process example data keeping samples from different times
# distinct, adding duplicate_count among collapsed sequences,
# and show the sample_id within each clone in the tibble.
clones = formatClones(ExampleAirr, traits=c("sample_id","c_call"),
    num_fields=c("duplicate_count"), columns=c("subject_id"))

print(clones)
```

```
## # A tibble: 2 x 5
##   clone_id data       locus  seqs subject_id
##      <dbl> <list>     <chr> <int> <chr>
## 1     3170 <airrClon> N        13 Subject_1
## 2     3184 <airrClon> N        12 Subject_1
```

Build maximum parsimony trees

A common way to build B cell lineage trees is the find the tree topology that minimizes the number of mutations needed along the tree (i.e. is the most parsimonious). Branch lengths can then be estimated as the number of mutations per site between each node in the tree.

Maximum parsimony trees can be built with the `getTrees` function, which by default uses the `pratchet` maximum parsimony function in the `phangorn` phylogenetics package.

The output is the same tibble as the input, but with a `trees` column containing an R `ape::phylo` object for each clone.

```
# Build trees using the pratchet maximum parsimony function in phangnorn
clones = getTrees(clones, nproc=1)

print(clones)
## A tibble: 2 x 6
#  clone_id data       locus  seqs subject_id trees
#     <dbl> <list>     <chr> <int> <chr>      <list>
#1     3170 <airrClon> N        13 Subject_1  <phylo>
#2     3184 <airrClon> N        12 Subject_1  <phylo>
```

Maximum parsimony trees can also be built using the PHYLIP function `dnapars`. To do this, the `build` option needs to be set as `dnapars` and the path to the `dnapars` executable needs to be specified in the `exec` option.

PHYLIP download site

```
# Build trees using dnapars.
# exec here is set to dnapars position in the Docker image.
clones = getTrees(clones, build="dnapars", exec="/usr/local/bin/dnapars", nproc=1)

clones
## A tibble: 2 x 6
#  clone_id data       locus  seqs subject_id trees
#     <dbl> <list>     <chr> <int> <chr>      <list>
#1     3170 <airrClon> N        13 Subject_1  <phylo>
#2     3184 <airrClon> N        12 Subject_1  <phylo>
```

Build maximum likelihood trees

A common way to build B cell lineage trees is the find the tree topology and branch lengths that maximize the likelihood of the sequence data given a substitution model.

Standard maximum likelihood trees can also be built with the `getTrees` function, which if specified the `optim.pml` function in the `phangorn` phylogenetics package.

Maximum likelihood trees can also be built using the PHYLIP function `dnaml`. To do this, the `build` option needs to be set as `dnaml` and the path to the `dnaml` executable needs to be specified in the `exec` option.

```
# Build trees using the optim.pml maximum likelihood function in phangnorn
clones = getTrees(clones, build="pml")

print(clones)
## A tibble: 2 x 6
#  clone_id data       locus  seqs subject_id trees
#     <dbl> <list>     <chr> <int> <chr>      <list>
#1     3170 <airrClon> N        13 Subject_1  <phylo>
#2     3184 <airrClon> N        12 Subject_1  <phylo>
```

Build trees using dnaml instead, which like `dnapars` is also distributed as part of `PHYLIP`.

```
# Build trees using dnaml.
# exec here is set to dnaml position in the Docker image.
clones = getTrees(clones, build="dnaml", exec="/usr/local/bin/dnaml")

clones
# A tibble: 2 x 6
#  clone_id data       locus  seqs subject_id trees
#     <dbl> <list>     <chr> <int> <chr>      <list>
#1     3170 <airrClon> N        13 Subject_1  <phylo>
#2     3184 <airrClon> N        12 Subject_1  <phylo>
```

Build IgPhyML B cell trees

B cell somatic hypermutation violates important assumptions in most phylogenetic models. IgPhyML implements models that incorporate SHM hotspot and coldspot motifs. To build trees using IgPhyML, specify the build option appropriately and pass the location of the IgPhyML executable. The returns

object will also include a `parameters` column, which will contain the HLP19 model parameters estimated from IgPhyML.

Note: This function is slower than other maximum likelihood and parsimony approaches.

IgPhyML documentation/download site

```
# Build trees using dnapars.
# exec here is set to dnapars position in the Docker image.
clones = getTrees(clones, build="igphyml",
    exec="/usr/local/share/igphyml/src/igphyml", nproc=1)

print(clones)
## A tibble: 2 x 7
#  clone_id data       locus  seqs subject_id trees        parameters
#     <dbl> <list>     <chr> <int> <chr>      <named list> <named list>
#1     3170 <airrClon> N        13 Subject_1  <phylo>      <named list [13]>
#2     3184 <airrClon> N        12 Subject_1  <phylo>      <named list [13]>


clones$parameters[[1]]$omega_mle
#[1] 0.5286
```