# perfectphyloR: An R package for reconstructing perfect phylogenies

Charith Bhagya Karunarathna and Jinko Graham *

**Abstract**

**Background:** A perfect phylogeny is a rooted binary tree that recursively partitions DNA sequences. The nested partitions of a perfect phylogeny provide insight into the pattern of ancestry of DNA sequence data. For example, sequences may cluster together in a partition indicating that they arise from a common ancestral haplotype.
**Results:** We present an R package `perfectphyloR` to reconstruct the local perfect phylogenies underlying a sample of DNA sequences. The package enables users to associate the reconstructed partitions with a user-defined partition. We describe and demonstrate the major functionality of the package.
**Conclusion:** The `perfectphyloR` package should be of use to researchers seeking insight into the ancestral structure of their sequence data. The reconstructed partitions have many applications, including the mapping of trait-influencing variants.

***Keywords:*** Perfect phylogeny, Ancestry, DNA sequence, Trait-influencing variants

## Background

A perfect phylogeny is a rooted binary tree that represents a recursive partitioning of a set of objects such as DNA sequences. The nested partitions provide insight into the pattern of ancestry of DNA sequences. The partitions have many uses, including applications in mapping trait-influencing variants (Mailund et al., 2006).

We present an R package `perfectphyloR` to reconstruct perfect phylogenies underlying a sample of DNA sequences. The package uses a classic algorithm (Gusfield, 1991) together with heuristics (Mailund et al., 2006) to partition sequences. Briefly, a neighborhood of SNVs is determined about a focal SNV, as will be described. Then, the perfect phylogeny is built by recursive partitioning on SNVs in this neighborhood.

We first discuss the implementation of the reconstruction of the partitions underlying a sample of DNA sequences. We then illustrate the major functionality of the package with worked examples.

## Implementation

In this section, we describe the reconstruction process, which consists of three steps:

1. Create a `hapMat` data object.
2. Reconstruct the perfect phylogeny at a focal SNV.
3. Reconstruct perfect phylogenies across a genomic region.

*Department of Statistics and Actuarial Science, Simon Fraser University, Burnaby, BC, Canada.

## Create hapMat data object

We first create a hapMat data object of SNV sequences to be partitioned with the function createHapMat(). To construct a hapMat data object, users are required to specify:

- hapmat, a matrix of 0's and 1's, with rows representing sequences and columns representing SNVs,
- snvNames, a vector of names of SNVs labelling the columns of hapmat,
- hapNames, a vector of names labelling the sequences in the rows of hapmat,
- posns, a numeric vector specifying the physical locations along the chromosome (in base pairs) of SNVs in the columns of hapmat.

To illustrate, we consider a toy example with 4 sequences comprised of 4 SNVs at positions 1-4 kb. The required hapMat object is created by executing the following command:

```
R> ex_hapMat  <- createHapMat(hapmat = matrix(c(1,1,1,0,
                                                0,0,0,0,
                                                1,1,1,1,
                                                1,0,0,0),byrow = TRUE, ncol = 4),
                              snvNames = c(paste("SNV",1:4,sep = "-")),
                              hapNames = c("h1","h2","h3","h4"),
                              posns = c(1000, 2000, 3000, 4000))
```

The structure of the resulting object of class hapMat is as follows.

```
R> ex_hapMat
   $hapmat
      SNV1 SNV2 SNV3 SNV4
   h1    1    1    1    0
   h2    0    0    0    0
   h3    1    1    1    1
   h4    1    0    0    0

   $posns
   [1] 1000 2000 3000 4000

   attr(,"class")
   [1] "hapMat"
```

## Reconstruct the perfect phylogeny at a focal SNV

With the main function reconstructPP(), the user can reconstruct the perfect phylogeny at a chosen focal SNV. The result is a phylo object to which the user may apply all the tools from the ape package (Paradis et al., 2004) for summarizing the reconstructed partition of sequences.

The function reconstructPP() consists of three major steps:

1. Determine a neighborhood of SNVs around a given focal SNV.
2. Order the SNVs in the neighborhood.
3. Recursively partition sequences based on SNVs in the neighborhood.

### Determine a neighborhood of SNVs around a given focal SNV

For a given focal SNV, the algorithm finds a neighborhood of SNVs. Starting from the focal SNV, the neighborhood of SNVs that are compatible with the focal SNV is expanded as much as possible on either side of the focal SNV until an incompatible SNV is found. The compatibility of a pair of SNVs is determined by the Four-Gamete Test (Hudson & Kaplan, 1985). For example, under the infinite-sites mutation model and no recombination, if the patterns at two SNVs are 00, 01, 10 and 11, then a mutation must have occurred twice at the same SNV and the two SNVs are said to be incompatible. If the neighborhood of compatible SNVs is smaller than a user-defined minimum size, we include incompatible SNVs in order of their physical proximity to the focal SNV, until the minimum size is reached.

### Order the SNVs in the neighborhood

Once the neighborhood of SNVs is determined, we order the compatible SNVs in the neighborhood from the most ancient to the most recent based on the derived allele frequency. Then, incompatible SNVs are ordered according to their physical proximity to the focal SNV.

### Recursively partition sequences based on SNVs in the neighborhood

The algorithm partitions sequences based on the most ancient compatible SNV in the neighborhood, and then recursively moves towards the most recent compatible SNV. When there are no further compatible SNVs in the neighborhood, the algorithm partitions sequences based on the incompatible SNVs, in order of their physical proximity to the focal SNV. Starting with the most ancient compatible SNV in the neighborhood, the algorithm partitions the sequences based on their carrier status for its derived allele. Then the algorithm jumps to the next-oldest compatible SNV in the neighborhood and continues partitioning (Gusfield, 1991). After considering the compatible SNVs, the algorithm moves to any incompatible SNVs in the neighborhood in order of their physical proximity to the focal SNV (Mailund et al., 2006). This process is repeated until each cluster contains only one sequence or there are no more SNVs to consider in the neighborhood.

A partition can be reconstructed by specifying:

- `hapMat`, a data object of class `hapMat`,
- `focalSNV`, the column number of the focal SNV at which to reconstruct the perfect phylogeny,
- `minWindow`, the minimum number of SNVs in the neighborhood of SNVs used to reconstruct the perfect phylogeny (default is one).

For example, to reconstruct the partition at the second SNV in the `ex_hapMat` example data, we can execute:

```
R> ex_dend <- reconstructPP(hapMat = ex_hapMat, focalSNV = 2,
                            minWindow = 1)
```

## Reconstruct perfect phylogenies across a genomic region

With the function `reconsPPregion()`, users can also reconstruct dendrograms across a genomic region of interest:

```
R> all_rdends <- reconsPPregion(hapMat = ex_hapMat,
                                minWindow = 1)
```

The result is `all_rdends`, an object of class `multiPhylo` that contains multiple `phylo` objects of `ape`.

# Examples

This section gives worked examples illustrating how to reconstruct the partitions underlying a sample of DNA sequences. In addition, we show how to investigate the association between the reconstructed partitions and a user-specified partition. The association statistics we consider include the Rand index (Rand, 1971), the dCor statistic (Székely et al., 2007), the HHG statistic (Heller et al., 2012), the Mantel statistic (Mantel, 1967), and the RV coefficient (Escoufier, 1973). The Rand index quantifies the association between two partitions directly. The dCor statistic, HHG statistic, Mantel statistic, and RV coefficient quantify the association between two distance matrices derived from partitions.

We first reconstruct a perfect phylogeny at a focal SNV. Next, we reconstruct perfect phylogenies across a genomic region. Finally, we show how to visualize and test associations between these reconstructed partitions and

- a comparator partition or dendrogram,
- a comparator distance matrix, and
- a phenotypic distance matrix.

## Reconstruct a perfect phylogeny at a focal SNV

Once the `hapMat` object is created as described in the implementation section, the user can reconstruct a perfect phylogeny at a focal SNV with `reconstructPP()`, by specifying the following four arguments:

1. `hapMat`: A data structure of class `hapMat`, created by `createHapMat()`.
2. `focalSNV`: The column number of the focal SNV at which to reconstruct the perfect phylogeny.
3. `minWindow`: Minimum number of SNVs around the focal SNV in the neighborhood of SNVs used to reconstruct the perfect phylogeny.
4. `sep`: Character string separator to separate sequence names for sequences that can not be distingiushed in the neighborhood around the focal point. For example, if sequences "h1" and "h3" can not be distinguished and `sep = "-"`, then they will be grouped together with the label "h1-h3". The default value is `"-"`.

For example, consider the example dataset `ex_hapMatSmall_data` with 10 sequences comprised of 20 SNVs. We can reconstruct a perfect phylogeny at the first SNV of `ex_hapMatSmall_data` by executing the following commands:

```
R> # Load the example hapMat data object.
   data(ex_hapMatSmall_data)

   # Reconstruct dendrogram at the first SNV of ex_hapMatSmall_data.
   rdend <- reconstructPP(hapMat = ex_hapMatSmall_data,
                          focalSNV = 1,
                          minWindow = 1,
                          sep = "-")
```

Figure 1 shows the reconstructed dendrogram, `rdend`, at the first SNV of `ex_hapMatSmall_data`. The structure of `rdend` is as follows:

```
R> str(rdend)
   List of 6
    $ edge         : num [1:6, 1:2] 5 6 6 5 7 7 6 1 2 7 ...
    $ Nnode        : int 3
    $ tip.label    : chr [1:4] "1249" "354-1009-2818" "2909" "1904-454-2931-2994-370"
```

```
 $ edge.length  : num [1:6] 6 3 3 4 5 5
 $ node.label   : NULL
 $ snvWinIndices: int [1:2] 1 5
 - attr(*, "class")= chr "phylo"
 - attr(*, "order")= chr "cladewise"
```

The user can extract the positions of the lower and upper limits of the neighborhood of SNVs used to reconstruct `rdend` as follows:

```
R> ex_hapMatSmall_data$posns[rdend$snvWinIndices]

   [1] 1500 7000
```

To see the sequences in the neighborhood of SNVs used for the reconstruction, the user can execute the following command:

```
R> ex_hapMatSmall_data$hapmat[, rdend$snvWinIndices[1]:rdend$snvWinIndices[2]]

         SNV3 SNV4 SNV7 SNV9 SNV14
   1904    0    0    0    0     0
   454     0    0    0    0     0
   1249    1    1    1    1     0
   2931    0    0    0    0     0
   2994    0    0    0    0     0
   2909    0    0    0    0     1
   354     1    1    1    0     0
   1009    1    1    1    0     0
   370     0    0    0    0     0
   2818    1    1    1    0     0
```

As can be seen in the above output, there are two groups of sequences that have the same ancestral and derived alleles at each SNV position: sequences 354, 1009 and 2818, and sequences 1904, 454, 2931, 2994 and 370. These two groups of sequences therefore cannot be distinguished in the reconstructed partition. In Figure 1, we can verify that two tips of the partition are comprised of these two groups of sequences.

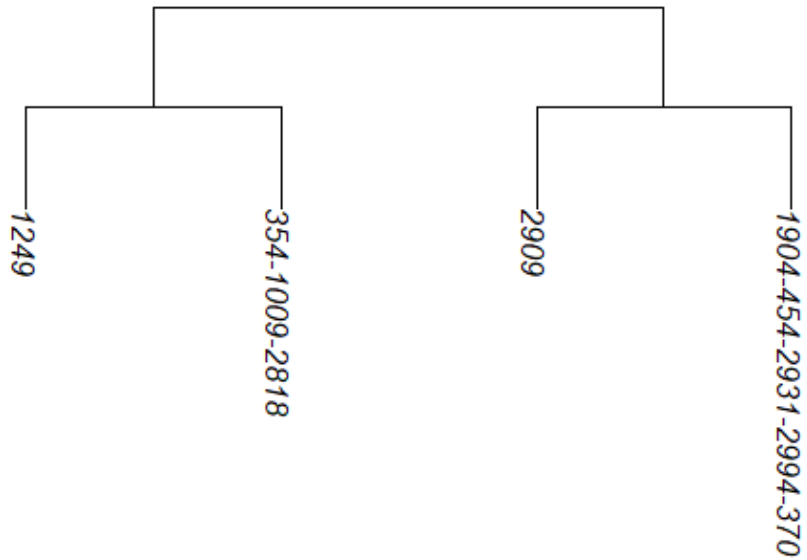


Figure 1: The reconstructed partition at the first SNV of `ex_hapMatSmall_data`.

## Reconstruct perfect phylogenies across a genomic region

With `reconsPPregion()`, the user can reconstruct perfect phylogenies at each possible focal SNV in a `hapMat` data object. In the following example, we consider the 10 sequences with 20 SNVs in `ex_hapMatSmall_data`. We reconstruct perfect phylogenies across the 20 SNVs.

```
R> # Reconstruct partitions across the region.
   rdends <- reconsPPregion(hapMat = ex_hapMatSmall_data,
                            minWindow = 1)
```

`rdends` is an `ape` multiphylo object. The reconstructed partition at the first focal SNV in `ex_hapMatSmall_data` is the first `phylo` object in `rdends`:

```
R> str(rdends[[1]])
   List of 6
    $ edge         : num [1:6, 1:2] 5 6 6 5 7 7 6 1 2 7 ...
    $ Nnode        : int 3
    $ tip.label    : chr [1:4] "1249" "354-1009-2818" "2909" "1904-454-2931-2994-370"
    $ edge.length  : num [1:6] 6 3 3 4 5 5
    $ node.label   : NULL
    $ snvWinIndices: int [1:2] 1 5
    - attr(*, "class")= chr "phylo"
    - attr(*, "order")= chr "cladewise"
```

## Associate reconstructed partitions to a comparator dendrogram

The function `testDendAssoRI()` uses the Rand Index to investigate the association between a comparator dendrogram or partition and multiple reconstructed dendrograms or partitions across a genomic region. The function has five key arguments:

1. `rdend`: An `ape` multiphylo object of reconstructed dendrograms at each focal SNV.
2. `cdend`: An `ape` phylo object of the comparator dendrogram.
3. `hapMat`: An object of class `hapMat` containing SNV sequences.
4. `k`: An integer that specifies the number of clusters that the dendrogram should be cut into. The default is `k = 2`. Clusters are defined by starting from the root of the dendrogram, moving towards the tips and cutting across when the appropriate number of clusters is reached.
5. `nperm`: Number of permutations for the test of any association across the genomic region. The default is `nperm = 0`; i.e., association will not be tested.

To illustrate, we use the example dataset `ex_hapMat_data` with 200 sequences and 2747 SNVs. We plot the Rand index values summarizing the association between the comparator dendrogram at SNV position 975 kilobase pairs (kbp) and the reconstructed dendrogram at each SNV position across the 2 Mbp genomic region.

```
R> # Comparator true dendrogram at 975 kbp.
   data(tdend)

   # hapMat data object.
   data(ex_hapMat_data)

   # Reconstruct dendrograms across the region.
   allrdends <- reconsPPregion(hapMat = ex_hapMat_data,
                               minWindow = 1)


   # Rand index profile based on 6 clusters.
   RI_profile <- testDendAssoRI(rdend = allrdends, cdend = tdend,
                                k = 6, hapMat = ex_hapMat_data,
                                nperm = 1000, xlab ="SNV positions (bp)",
                                ylab = "Rand indices",
                                main = "Association Profile")

    # Omnibus P value for overall associaion.
    RI_profile$OmPval
    [1] 0.000999001
```

Figure 2 shows the association profile between a comparator true dendrogram, `tdend` at position 975 kbp and a list of reconstructed dendrograms across the genomic region of `ex_hapMat_data`. As can be seen from the Figure 2, the strongest association is around the SNV position 975 kbp, as expected, with the value of 0.9. According to the omnibus $p$-value (`RI_profile$OmPval`), the association across the genomic region is significant ($P \approx 0.001$).

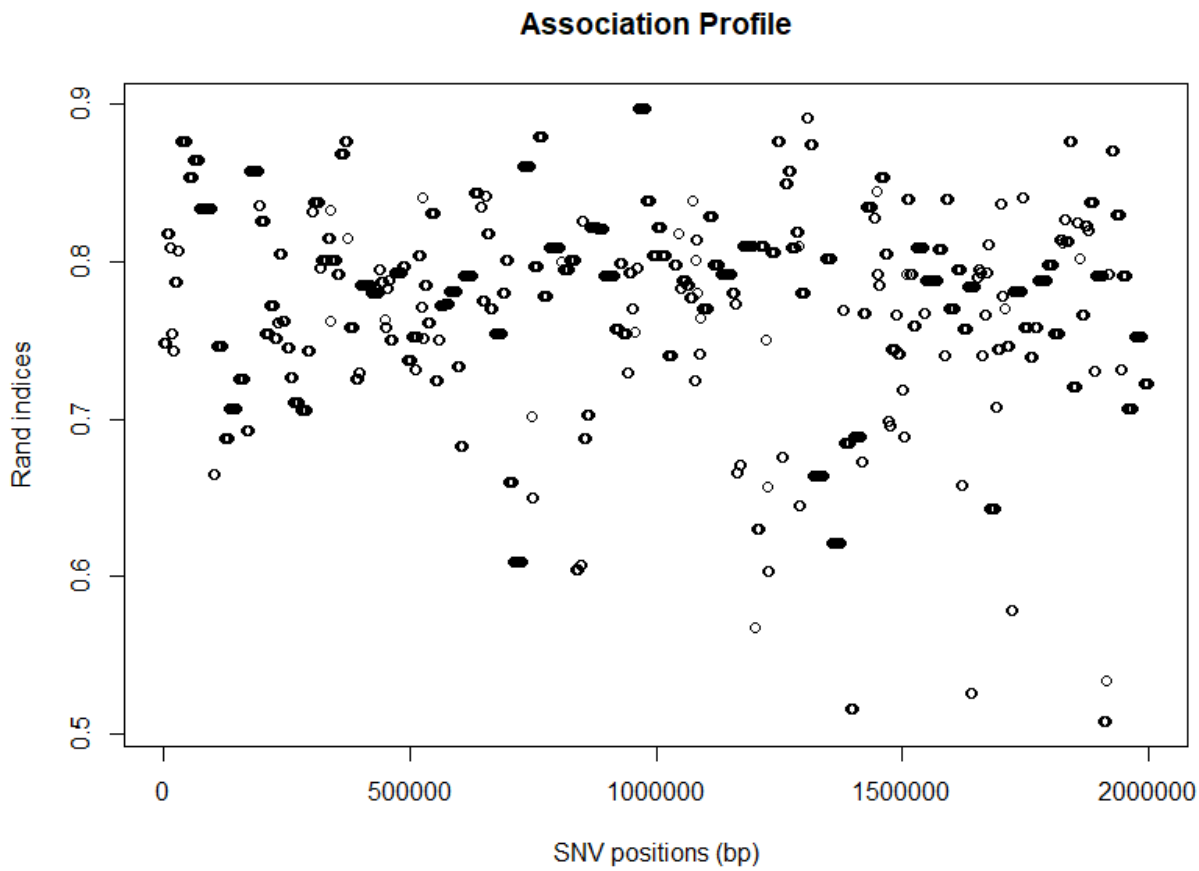


Figure 2: Rand indices associating a comparator true dendrogram at position 975 kbp and reconstructed dendrograms across the genomic region.

## Associate reconstructed partitions to a comparator distance matrix

The function `testAssoDist()` investigates the association between a comparator distance matrix and multiple reconstructed dendrograms across a genomic region. The association statistics available in the function are the dCor statistic, HHG statistic, Mantel statistic, and RV coefficient. The function has the following five key arguments:

1. `rdend`: An `ape` multiphylo object of reconstructed dendrograms at each focal SNV.
2. `cdmat`: A comparator matrix of pairwise distances (e.g. pairwise distances between sequences of a comparator dendrogram).
3. `method`: A character string specifying one of `"dCor"`, `"HHG"`, `"Mantel"` or `"RV"` for the dCor, HHG, Mantel or RV statistics, respectively.
4. `hapMat`: An object of class `hapMat` containing SNV sequences.
5. `nperm`: Number of permutations for the omnibus test of any association across the genomic region. The default is `nperm = 0`; i.e., association will not be tested.

To illustrate, we plot the dCor statistics summarizing the association between a comparator distance matrix, `cdmat`, and the reconstructed dendrograms across the genomic region of the example dataset `ex_hapMat_data`.

First, we compute the pairwise distances between sequences based on the comparator true dendrogram at SNV position 975 kbp. These pairwise distances are computed with the function `rdistMatrix()`, available in the package. The `rdistMatrix()` function uses the rankings of the nested partitions in the dendrogram to calculate rank-based distances between the sequences. However, users can provide any distance measures of interest for `cdmat`. We then plot the dCor statistic summarizing the association between the rank-based distance matrix for the reconstructed dendrograms at each SNV position and the comparator distance matrix at SNV position 975 kbp (Figure 3).

```
R> # Comparator true dendrogram at SNV position 975 kbp.
   data(tdend)

   # hapMat data object.
   data(ex_hapMat_data)

   # Compute rank-based distances between haplotypes based on the
   # comparator true dendrogram (tdend) using the function, rdistMatrix().

   dendDmat = perfectphyloR::rdistMatrix(tdend)

   # dCor profile comparing the association between distance matrix of
   # true dendrogram (comparator dendrogram) and all reconstructed
   # dendrogram across the genomic region.
   dCor_profile <- testAssoDist(cdmat = tdendDmat, rdend = allrdends,
                                method = "dCor", hapMat = ex_hapMat_data,
                                nperm = 1000,
                                xlab = "SNV positions(bp)",
                                ylab = "dCor Statistics",
                                main = "Association Profile")

   # Omnibus P value for overall association.
   dCor_profile$OmPval
   [1] 0.000999001
```

In Figure 3, we can clearly see the strongest association around the SNV position 975 kbp, and the association across the genomic region is significant ($P \approx 0.001$), as expected.

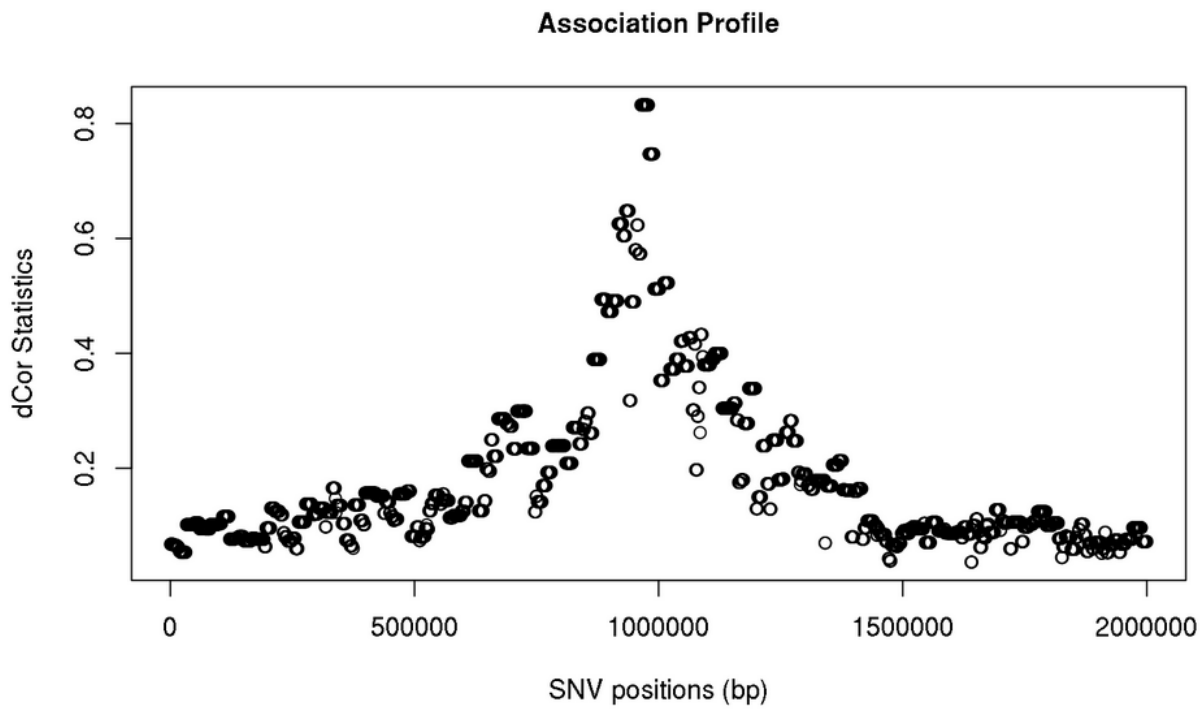


Figure 3: Associations between a comparator distance matrix from the true dendrogram at position 975 kbp and the reconstructed dendrograms across the genomic region.

## Associate reconstructed partitions to a phenotypic distance matrix

To illustrate another application of the function `testAssoDist()`, we perform the RV test of association between a phenotypic distance matrix as the `cdmat` argument and the reconstructed dendrograms across the genomic region of `ex_hapMat_data`. The phenotype data and distances are described in (Karunarathna & Graham, 2018) and are contained in the data object `phenoDist`. Binary phenotype status was assigned based on causal SNVs from a causal subregion defined from 950 - 1050 kbp within the 2-Mbp genomic region.

```
R> # Phenotypic distances.
   data(phenoDist)
   # RV profile.
   RV_profile <- testAssoDist(cdmat = phenoDist,
                              rdend = allrdends,
                              method = "RV",
                              hapMat = ex_hapMat_data, nperm = 1000,
                              xlab = "SNV positions (bp)",
                              ylab = "RV coefficients",
                              main = "Association Profile")

   # Indicate the risk region where the causal SNVs are located.
   abline(v = 950000); abline(v = 1050000)

   # Omnibus P value for overall association.
   RV_profile$OmPval
   [1] 0.4955045
```

Figure 4 shows the resulting association profile between the phenotypic distances and the reconstructed dendrograms across the genomic region in `ex_hapMat_data`. The vertical lines indicate the causal subregion of 950 - 1050 kbp. The strongest association is close to the causal subregion. However, in this example, the association across the genomic region is not significant ($P \approx 0.5$).

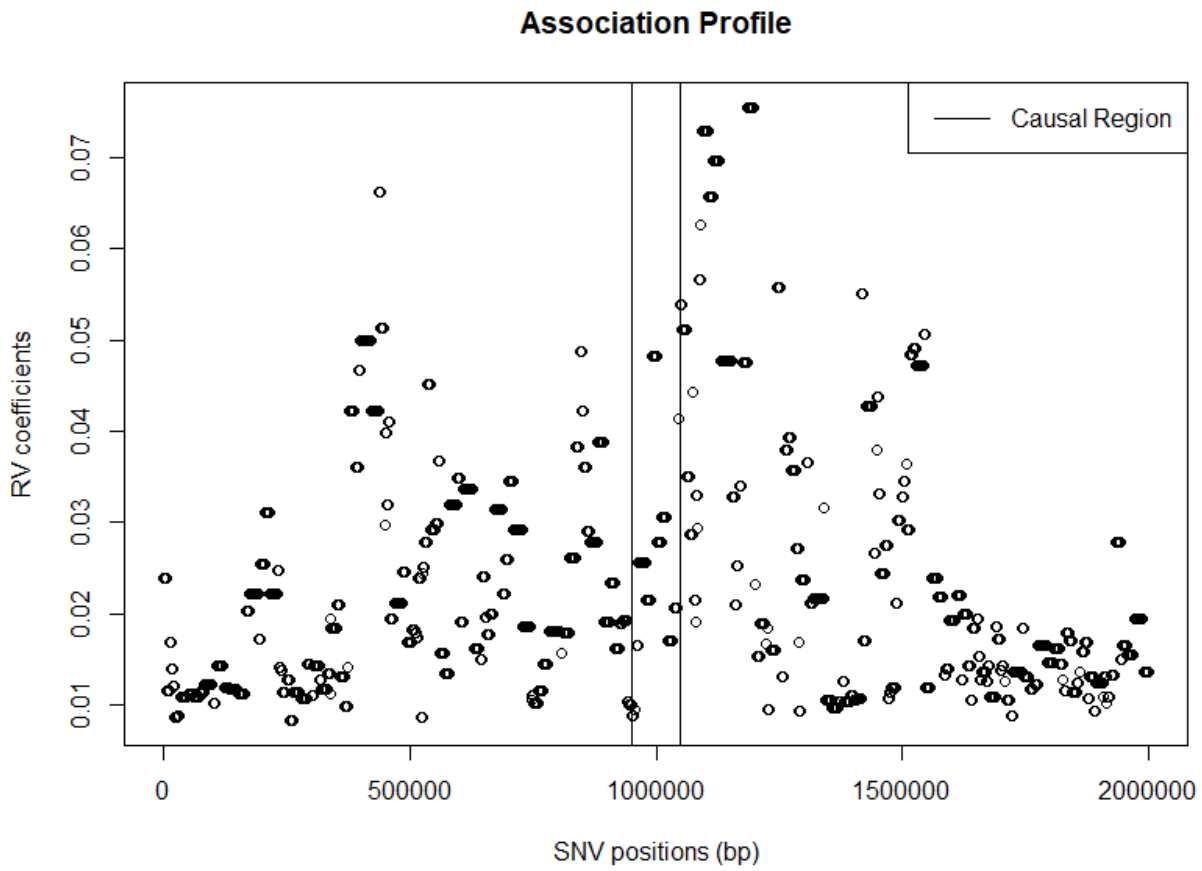


Figure 4: Associations between the phenotypic distance matrix and the reconstructed dendrograms across the genomic region.

# Timing

Table 1 shows the computation times of the package's major functions on an Intel E5-2683 v4 at 2.1 GHz with 20 GB of RAM. These computation times are for the 200 sequences comprised of 2747 SNVs in the example data `ex_hapMat_data` that is included in the package.

Table 1: Computation times of the major functions of the package `perfectphyloR`

| Function | Computation Time (minutes) | |
|---|---|---|
| | **No permutation** | **1000 permutations** |
| `reconstructPP()` | Few seconds | NA |
| `reconsPPregion()` | 11.0 | NA |
| `testDendAssoRI()` | 1.0 | 40.0 |
| `testAssoDist()` | 0.33 | 236.0 |

## Conclusion

The R package `perfectphyloR` provides functions to reconstruct a perfect phylogeny at a user-given focal SNV and perfect phylogenies across a genomic region of interest. The package also computes, tests and displays association measures based on the reconstructed partitions in a genomic region. The reconstructed partitions are useful to researchers seeking insight into the ancestral structure of DNA sequences. For example, associating the reconstructed partitions with a trait can help to localise trait-influencing variants in association studies. `perfectphyloR` can be freely downloaded from the Comprehensive R Archive Network (CRAN) or from https://github.com/cbhagya/perfectphyloR/.

## Availability and requirements

Project name: perfectphyloR
Project home page: https://cran.r-project.org/web/packages/perfectphyloR/index.html
Operating system(s): Windows, Linux, OS X
Programming language: R
Other requirements: R 3.4.0 or newer
License: GPL-2, GPL-3
Any restrictions to use by non-academics: none

The package `perfectphyloR` can be installed from CRAN using `install.packages("perfectphyloR")`. The local zip file can be installed using R Studio by selecting the install package(s) from local zip files.

## List of abbreviations

DNA: Deoxyribonucleic Acid; SNV: Single Nucleotide Variant;
dCor: Distance Correlation; RI: Rand Index;
RV: R-Vector, a vector version of standard $r$ correlation;
GHz: Giga Hertz; GB: Gigabyte; RAM: Random Access Memory;
CRAN: Comprehensive R Archive Network

## Acknowledgements

We thank Christina Nieuwoudt and Kelly Burkett for helpful discussions and comments, and the Department of Statistics and Actuarial Science at Simon Fraser University for its generous support. This research was funded in part by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

# References


Escoufier, Y. (1973). Le traitement des variables vectorielles. *Biometrics*, 751–760.

Gusfield, D. (1991). Efficient algorithms for inferring evolutionary trees. *Networks*, *21*(1), 19–28.

Heller, R., Heller, Y., & Gorfine, M. (2012). A consistent multivariate test of association based on ranks of distances. *Biometrika*, *100*(2), 503–510.

Hudson, R. R., & Kaplan, N. L. (1985). Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, *111*, 147-164.

Karunarathna, C. B., & Graham, J. (2018). Using gene genealogies to localize rare variants associated with complex traits in diploid populations. *Human Heredity*, *83*(1), 30–39.

Mailund, T., Besenbacher, S., & Schierup, M. H. (2006). Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics*, *7*(1), 454.

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer research*, *27*(2 Part 1), 209-220.

Paradis, E., Claude, J., & Strimmer, K. (2004). APE: Analyses of Phylogenetics and Evolution in R language. *Bioinformatics*, *20*(2), 289–290.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*(336), 846–850.

Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, *35*(6), 2769–2794.