# perfectphyloR: An R package for reconstructing perfect phylogenies

Charith Bhagya Karunarathna and Jinko Graham [*]

**Abstract**

A perfect phylogeny is a rooted binary tree that recursively partitions DNA sequences. The nested partition structures of a perfect phylogeny provide insight into the pattern of ancestry of DNA sequence data. For example, disease sequences may cluster together in a local partition indicating that they arise from a common ancestral haplotype. The availability of an R package that reconstructs perfect phylogenies should therefore be useful to researchers seeking insight into the ancestral structure of their sequence data. We develop an R package `perfectphyloR` to reconstruct the local perfect phylogenies underlying a sample of DNA sequences. Our implementation first partitions the DNA sequences using a classic partitioning algorithm and then uses well known heuristics to refine them further. We here briefly demonstrate the reconstruction process and illustrate the major functionality of the package using worked examples.

## Introduction

A perfect phylogeny is a rooted binary tree that represents a recursive partitioning of a set of objects such as DNA sequences. Their nested partition structures can provide insight into the pattern of ancestry of DNA sequence data. These patterns of ancestry can provide useful information about the location of a disease-influencing variant, for example (Mailund et al., 2006).

We present an R package `perfectphyloR` to reconstruct perfect phylogenies underlying a sample of DNA sequences, at a focal single-nucleotide variant (SNV). We implement the partitioning of DNA sequences using the classic algorithm of Gusfield (1991), and then further partition them using heuristics introduced by Mailund et al. (2006). Specifically, starting from the focal SNV, we expand the neighborhood of compatible SNVs until we find an incompatible SNV. However, when the block of compatible SNVs is smaller than a user-defined minimum size, we expand the block to include incompatible SNVs in order of proximity to the focal SNV. Once we identify the neighborhood of SNVs, we order a set of compatible SNVs from the most ancient to the most recent. We then construct the perfect phylogeny by recursive partitioning on SNVs. The algorithm first partitions on the most ancient SNV, and then recursively moves towards the present, partitioning at each SNV it encounters until either running out of SNVs or until each partition consists of a single sequence.

We first briefly discuss the implementation of the reconstruction of the partitions underlying a sample of DNA sequences. We then illustrate the major functionality of the R package `perfectphyloR` via worked examples.

## Implementation

In this section, we briefly discuss our reconstruction process. This reconstruction process consists of three steps:

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

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

## 1. Create hapMat data object

We first construct a `hapMat` data object with the function `createHapMat()` and use this data object throughout the reconstruction.

To construct a `hapMat` data object, users are required to specify:

- `hapmat`, a matrix of 0's and 1's, with rows representing haplotypes and columns representing SNVs,
- `snvNames`, a vector of names of SNVs labelling the columns of `hapmat`,
- `hapNames`, a vector of names of haplotypes labelling 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 haplotypes and 4 SNVs. 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"
```

## 2. 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 using Gusfield's algorithm for the reconstruction of perfect phylogenies on compatible SNVs, and Mailund et al.'s modifications to include incompatible SNVs. 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 main function `reconstructPP()` of our implementation consists of three major steps:

1. Select a window of SNVs at a given focal SNV:

   Starting at the given focal SNV, this algorithm expands the neighborhood of compatible SNVs according to the Four-Gamete Test (Hudson & Kaplan, 1985) until it finds an incompatible SNV. However, when the neighborhood of compatible SNVs about the focal SNV is too small for the user, following heuristics introduced by Mailund et al. (2006), the algorithm expands the neighborhood by including incompatible SNVs in order of proximity from the focal SNV.

2. Order the SNVs:

   Once the window of SNVs about the focal SNV is selected, following Gusfield (1991), the compatible SNVs are ordered from most ancient to most recent. Then, following Mailund et al. (2006), incompatible SNVs are ordered according to their proximity to the focal SNV.

3. Build the sequence partitions using recursive partitioning on SNVs:

   With the ordered SNVs, we build the perfect phylogeny at the focal SNV using recursive partitioning on the SNVs in the neighborhood.

Then the reconstruction can be performed 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 around the focal SNV in the window of SNVs used to reconstruct the perfect phylogeny (default is one),

as follows:

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

## 3. 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 `mutliPhylo` that contains multiple `phylo` objects of `ape`.

## Examples

In this section, we illustrate how the user can reconstruct the partitions underlying a sample of DNA sequences and employ the major functionality in the package via worked examples.

In addition to the reconstruction, this R package allows the user 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 dendrograms directly. The dCor statistic, HHG statistic, Mantel statistic, and RV coefficient quantify the association between two distance matrices.

We first reconstruct a perfect phylogeny at a focal SNV. We then reconstruct perfect phylogenies across a genomic region of interest. Finally, we show how the user can perform the association tests. Specifically, across a genomic region of interest, we consider association between

- a comparator dendrogram and a list of reconstructed dendrograms,
- a comparator distance matrix and the reconstructed dendrograms, and
- a phenotypic distance matrix and the reconstructed dendrograms.

### 1. Reconstruct a perfect phylogeny at a focal SNV

Once the `hapMat` object is created as mentioned 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 window of SNVs used to reconstruct the perfect phylogeny.
4. `sep`: Character string separator to separate haplotype names for haplotypes that can not be distinguished in the window around the focal point. For example, if haplotypes "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, we can reconstruct a perfect phylogeny at the first SNV of the package's small example dataset, `ex_hapMatSmall_data`, containing 10 haplotypes of 20 SNVs with:

```
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 window of SNVs used to reconstruct, `rdend` as follows:

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

   [1] 1500 7000
```

To see the haplotypes in the window of SNVs used to reconstruct `rdend`, 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 haplotypes that have the same ancestral and derived alleles at each SNV position: haplotypes 354, 1009 and 2818, and haplotypes 1904, 454, 2931, 2994 and 370. These two groups of haplotypes 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 haplotypes.

## 2. 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 haplotypes of 20 SNVs in `ex_hapMatSmall_data`. We reconstruct perfect phylogenies across this subset of SNVs.

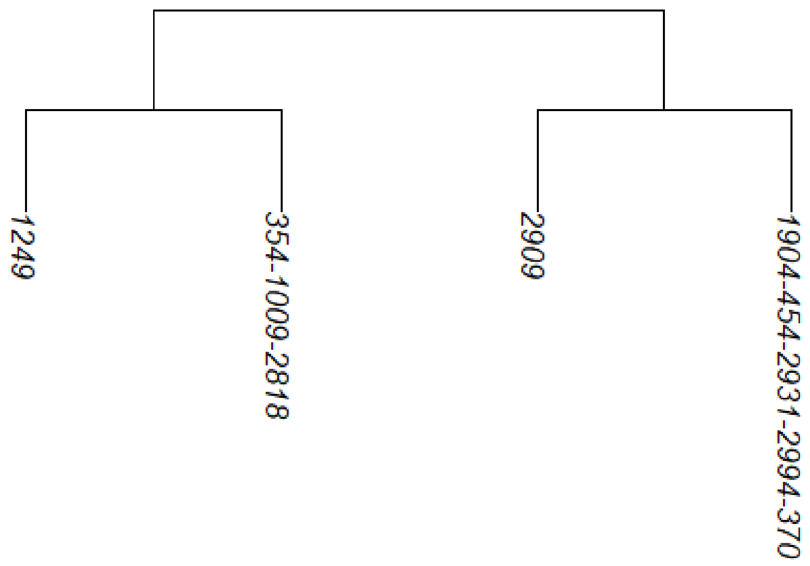

Figure 1: The reconstructed partition structure at the first SNV of ex_hapMatSmall_data.

```
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"
```

## 3. Association between a comparator dendrogram and a list of reconstructed dendrograms

With the function testDendAssoRI(), the user can perform the Rand index test to assess the association between a comparator dendrogram, and a list of reconstructed dendrograms across a genomic region of interest. testDendAssoRI() 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 haplotypes.

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 of interest. The default is `nperm = 0`; i.e., association will not be tested.

To illustrate, we use the example dataset `ex_hapMat_data` with 200 haplotypes and 2747 SNVs. We plot the Rand index values summarizing the association between the comparator dendrogram at SNV position 975 kilobase pair (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 and the reconstructed dendrogram at the same position is around 0.9. According to the omnibus P value (`RI_profile$OmPval`), the association across the genomic region is significant ($P \approx 0.001$).

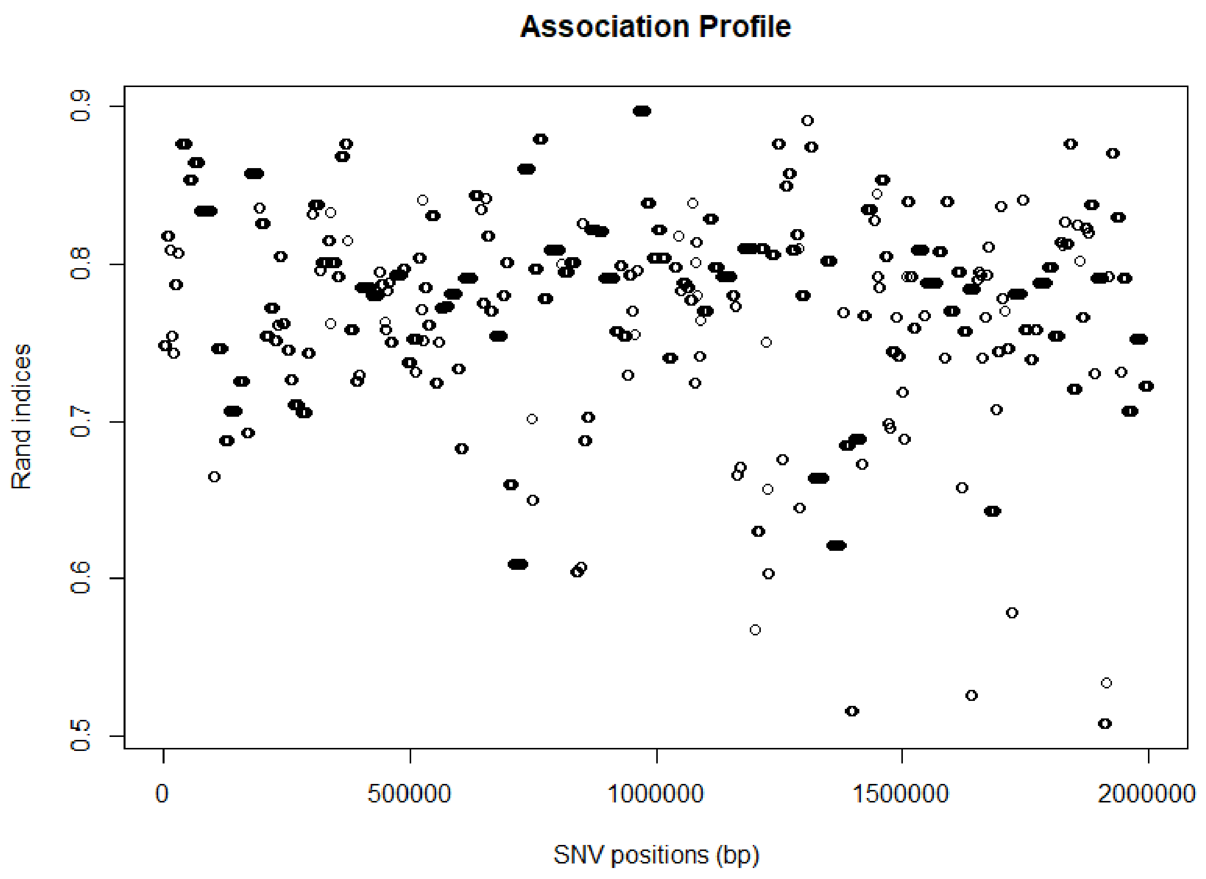

Figure 2: Rand indices associating a comparator true dendrogram at position 975 kbp and a list of reconstructed dendrograms across the genomic region.

## 4. Association between a comparator distance matrix and the reconstructed dendrograms

With the function `testAssoDist()`, the user can test the association between a comparator distance matrix, based on any pairwise distance measure, and the reconstructed dendrograms across a genomic region of interest. The association statistics available in the function are the dCor statistic, HHG statistic, Mantel statistic, and RV coefficient. The function `testAssoDist()` 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 haplotypes of a comparator dendrogram).
3. `method`: For association, use `dCor`, `HHG`, `Mantel` and `RV` for the dCor, HHG, Mantel and RV statistics, respectively.
4. `hapMat`: An object of class `hapMat` containing SNV haplotypes.
5. `nperm`: Number of permutations for the test of any association across the genomic region of interest. 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 package's example dataset `ex_hapMat_data`.

We first compute the pairwise distances between haplotypes based on the comparator true dendrogram at SNV position 975 kbp. These pairwise distances are computed using the function `rdistMatrix()` which is available in the package. The `rdistMatrix()` function uses the rankings of the nested partitions in the dendrogram to calculate rank-based distances between haplotypes. 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.

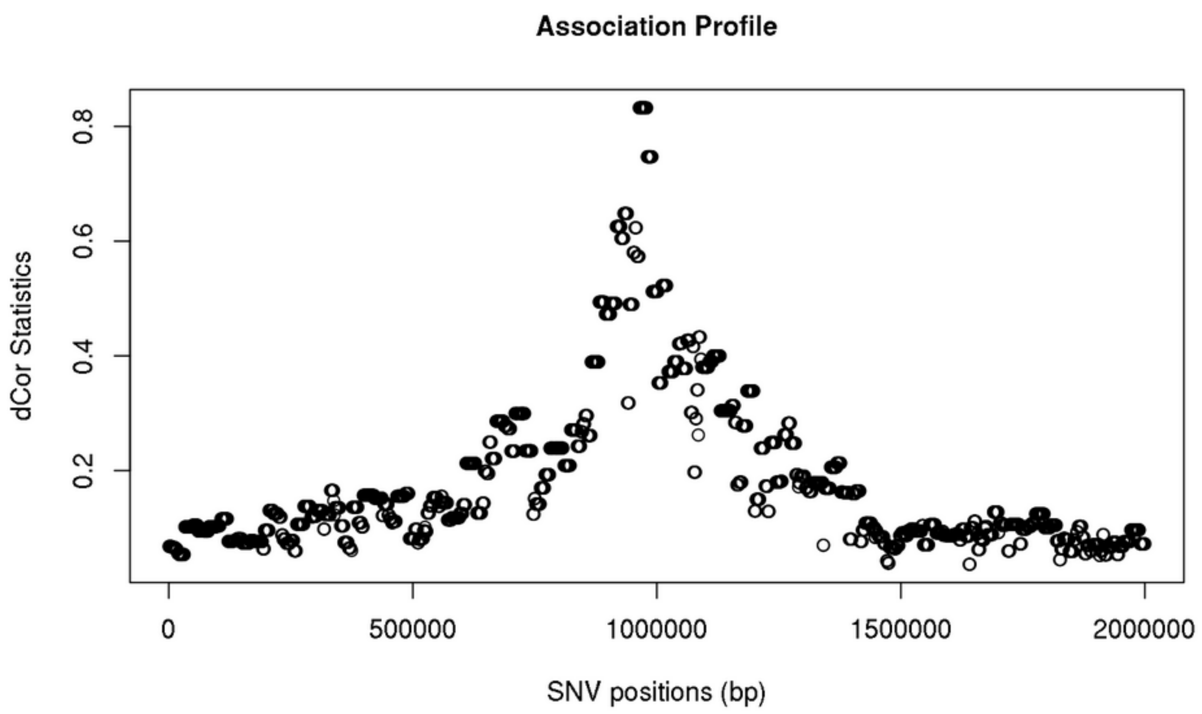

Figure 3: dCor associations between a comparator distance matrix from the true dendrogram at position 975 kbp and the reconstructed dendrograms across the genomic region.

## 5. Association between a phenotypic distance matrix and the reconstructed dendrograms.

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 disease phenotype data and distances are described in Karunarathna & Graham (2018). These distances are contained in the data object `phenoDist`. Binary disease status was assigned based on causal SNVs from the middle subregion of 950 - 1050 kbp (risk region) within a 2 Mbp genomic region of interest.

```
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 distance matrix and the reconstructed dendrograms across the genomic region in `ex_hapMat_data`. The vertical lines indicate the middle subregion of 950 - 1050 kbp. The strongest association is close to the risk region. However, in this example, the association across the genomic region is not significant ($P \approx 0.5$).

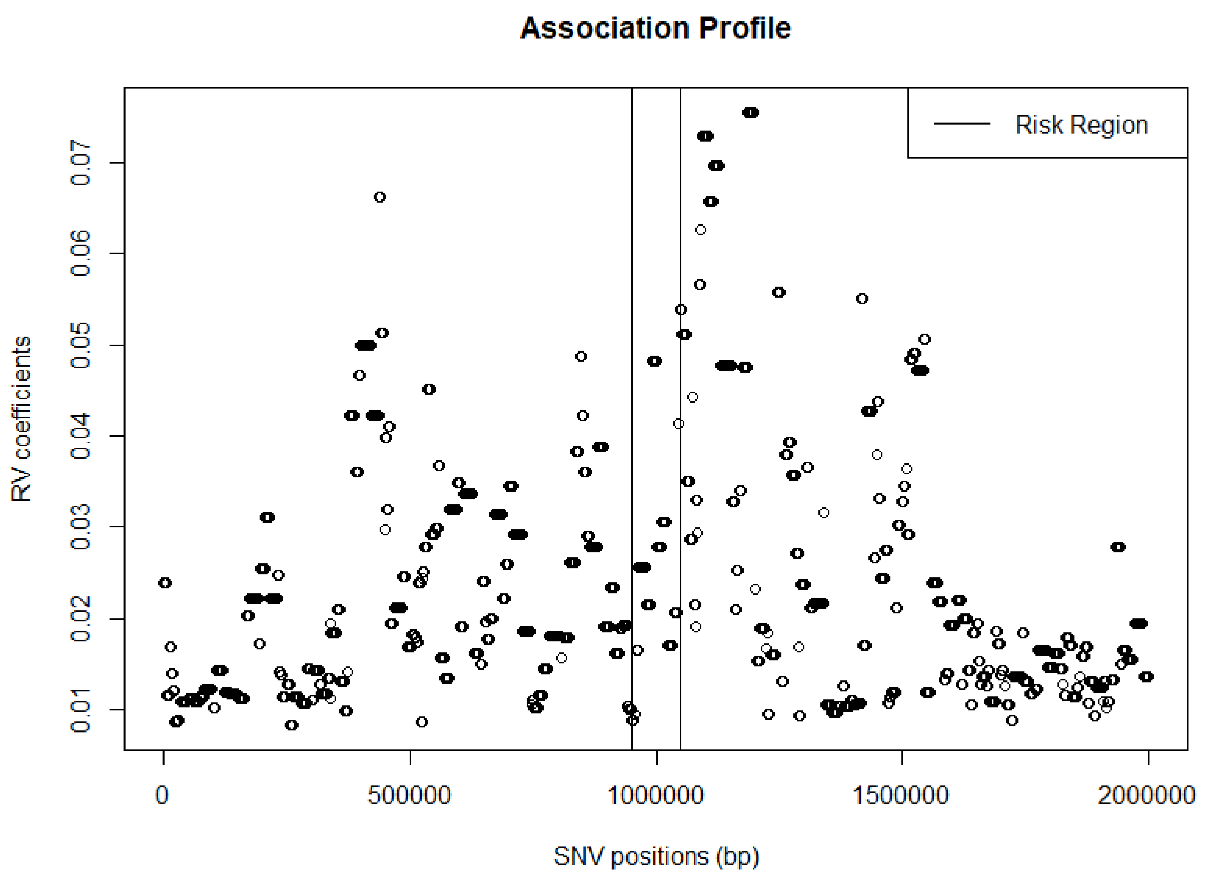

Figure 4: RV associations between the phenotypic distance matrix and the reconstructed dendrograms across the genomic region.

## Timing

The following table 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 2747 SNVs with the 200 haplotypes in `ex_hapMat_data` data set 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 |

## Acknowledgements

We thank Christina Nieuwoudt, for helpful discussions 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.