User Manual

Minimum SNPs

Table of Contents

[Getting started 1](#_Toc493239839)

[Importing file 1](#_Toc493239840)

[Processing the allelic profiles and identify profiles to be excluded 1](#_Toc493239841)

[Using % mode 2](#_Toc493239842)

[Using D mode 2](#_Toc493239843)

# Getting started

To get started, first download the package from <https://github.com/ludwigHoon/PRT452-BINFO->. After that, import the library in the R-programming environment by the command: library(minSNP).

# Importing file

The first thing that a user probably wants to do first is to import the fasta file which contains a list of allelic profiles into the programming environment. This can be achieved in the following steps:

1. Change directory to the location of the fasta file in the R-programming environment (option under files).
2. Import SeqinR library, by the command: library(seqinr).
3. Read the fasta file, by using the command: VariableA<-read.fasta(‘filename’).

# Processing the allelic profiles and identify profiles to be excluded

Before doing any other operation, it is prudent for the user to first process the allelic profiles, in order to make sure that the file is correct and that the allelic profiles don’t contain any allelic with deletion as the algorithms in the package does not take into account of the SNPs with deletion and may return incorrect result or fail to execute. In order to process the allelic profiles, the following steps can be taken:

1. After reading the allelic profiles file and setting that in a VariableA, flagAllele function can be used to return a list of allelic profiles that is shorter than the others. I.e. with command: flagAllele(VariableA), the flagged allelic profiles will be shown if there is any.
2. The function processAllele can then be used to processed the allelic profiles and ignore those allelic profiles which is shorter. I.e. with the command: VariableA <- processAllele(VariableA). The VaraibleA contains only those allelic profiles with normal length.

# Using % mode

In order to calculate the percentage of difference of a SNP of an allelic profile as compared to the others, similar.percent function can be used. The command steps are as followed:

1. Read the fasta file that contain the list of allelic profiles to be analysed (i.e. the allelic profiles are now in VariableA) into the R-programming environment.
2. Use the function similar.percent to calculate the percentage and assign the result to a variable. I.e. with the command: Result\_Variable <- similar.percent(VariableA, ‘Targeted\_Allelic\_Profile’).
3. Pass the result variable to present.percent to process the result so that it shows only (1) predefined number of results, and (2) results with the minimum percentage of difference. I.e. with the command: present.percent(Result\_Variable, <minimum percent>, <number of result>).

# Using D mode

In order to find the pair of SNPs that can be used to type the different sequence type, along with the Simpson’s index, similar.simpson function can be used. The command steps are as followed:

1. Read the fasta file that contain the list of allelic profiles to be analysed (i.e. the allelic profiles are now in VariableA) into the R-programming environment.
2. Use the function similar.simpson to find the top SNPs pair and assign the result to a variable. I.e. with the command: Result<- similar.simpson (VariableA, <level>, NULL, <included>, <excluded>), where level is number of SNPs returned (e.g. 2, returns 2 SNPs, 3 return 3); included is the position(s) of SNP that is forced into consideration; excluded is the position(s) of SNP that is forced out of consideration.

\*Note: If there are more than 1 positions to be included/excluded, <included>/<excluded> can be replaced with a vector of numeric, i.e. c(position1, position2,…..).

1. The result can be viewed by typing the result variable into the environment. Alternatively, what the users might want to see is the sequence types that are defined by the specific SNPs selected. In this case, the user can pass the result variable into present.simpson. I.e. with the command: present.simpson(VariableA, Result).