CNVSelectR Example

Peter Chi

5/7/2021

Preliminaries

This document demonstrates how to use the CNVSelectR package to take input files and run the method on them. First, the package can be downloaded as a zip file from this github repository, and then in RStudio, go to Tools → Install Packages... from Package Archive File and point it to the zip file that was just downloaded.

Then we load the package:

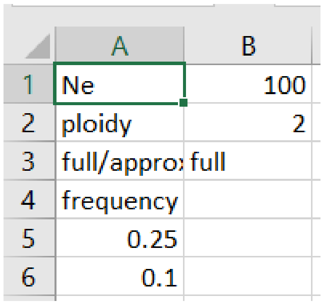
```
library(CNVSelectR)
```

```
## Loading required package: Matrix

## Loading required package: seqinr

## Warning: package 'seqinr' was built under R version 4.0.5

## Loading required package: readr

## Loading required package: knitr
```

Requirements

Two input files are required:

- CSV file containing two columns, as shown below:



	A	B
1	Ne	100
2	ploidy	2
3	full/appro:	full
4	frequency	
5	0.25	
6	0.1	

Figure 1: example csv file

- txt file containing aligned sequences in FASTA format

It is assumed that these files are in your current working directory.

In the CSV file:

- Ne refers to an estimate of the organismal effective population size. Larger Ne values will require more RAM and take longer to run. In the current implementation, values up to 1,000 are possible.

- Ploidy refers to the organismal ploidy, typically 1 for bacteria and 2 for eukaryotes. A model for values of ploidy > 2 has not been implemented, and this `R` implementation currently only allows for diploid organisms, with the haploid version still in progress.
- Full/approximate refers to the choice between applying the full Moran model or one of several approximations. This option will be updated in the future as only the full model is currently available. An approximate model, once implemented, will allow for larger Ne values.
- Below the frequency label, please list the frequency of the duplicate in the population of interest. For each duplicate examined, include 2 sequences in order in a FASTA file saved in text format.

The FASTA file should contain aligned codon (nucleotide) sequences representing the protein coding sequence for each copy of the gene duplicate.

Examples of each are provided in the `examples` directory of this github repository as a guide. Below, we use them to demonstrate the method.

Running the method

Now, to generate the null model and obtain confidence intervals and p-values for each duplicate pair, we run:

```
test_out <- CNVSelect_test("cnv_sample_file_1.csv", "cnv_sample_file_2.txt")
```

The raw output from this function looks as follows:

```
test_out
```

```
## $freqs
## [1] 0.25 0.10
##
## $dS
##        wt1/dup1   wt2/dup2
## [1,] 0.02265236 0.01784061
##
## $CIlower
## [1] 0.005 0.005
##
## $CIupper
## [1] 0.110 0.095
##
## $p_val
## [1] 0.0002111328 0.0361783105
```

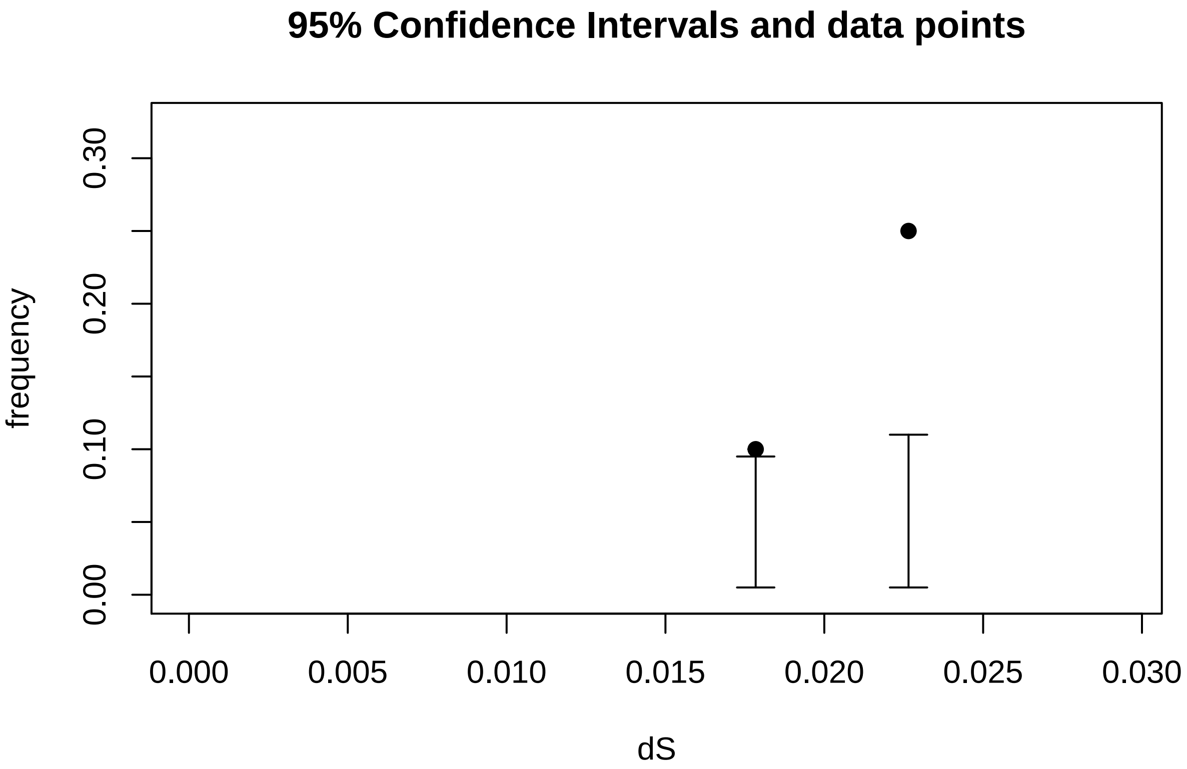
Creating summary output

We can create a summary table and plot as follows:

```
CNVSelect_summary(test_out)
```

	dS	frequency	95% CI	p-value
wt1/dup1	0.0227	0.25	(0.005, 0.11)	0.000211
wt2/dup2	0.0178	0.1	(0.005, 0.095)	0.0362

```
CNVSelect_plot(test_out)
```


95% Confidence Intervals and data points
frequency
0.00
0.10
0.20
0.30
0.000
0.005
0.010
0.015
0.020
0.025
0.030
dS