Help File for compareHap()

R version: 3.4.2

Required Packages: sangerseqR, stringi

R code found in file FinalcompareHapAndDependencies.R

R code in Folder: Shared Folder/ Alicia Williams/ R

Data called: newCombinedNoGapAllAdded7021.csv

Data found in folder: Shared Folder/ Alicia Williams/ R/ Data20180226

Contents:

How to use compareHap()

Description of functions

Suggestions for analyzing homozygous sequences

Suggestions for analyzing the cloned out haplotypes

Fuzzy match

**How to Use compareHap()**

**Overview:**

1. Load packages sangerseqR, stringi
2. Build a dataframe in the right format.
3. Run all the code in FinalcompareHapAndDependencies.R so the functions are in your R environment
4. Get path to chromatogram file you want to use
5. Run compareHap with the path and dataframe as arguments
6. **Loading Packages**

The main packages are sangerseqR, and stringi but they also depend on other packages so it should also load Biostrings, BiocGenerics, parallel, S4vectors, Iranges, Xvector, and stats4 automatically with it.

Suggestion on how to load sangerseqR

<https://www.bioconductor.org/packages/devel/bioc/html/sangerseqR.html> source("https://bioconductor.org/biocLite.R")

biocLite("sangerseqR")

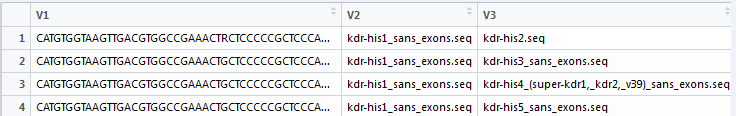
To work most likely need R later version than 3.0.2

1. **Building a dataframe**

The compareHap function requires a database of sequences that you want to search through for a match. Format of database must be:

Sequence of added haplotypes (string), name of one haplotype (string), name of second haplotype (string)

Ex:



Note: can also contain many other columns of information after V3 if desired but won’t be used

Would suggest also adding columns V4 for sequence of haplotype 1 as string and column v5 for sequence of haplotype 2 so that if in future you want to look more in depth at the matches it is easy to locate that information

The file newCombinedNoGapAllAdded7021.csv contain the most recent database for this project. Should import this file into R environment before running compareHap if you want to use this database.

For help on how to generate a completely new database reference ExampleOfGeneratingDatabase.R

1. **Run the code into the R environment**

File: FinalcompareHapAndDependencies.R

Specifically you will need run the functions compareHap(), clipIndex2(), addPriSeq3(), addBases2(), MODcheckmasteist4(), and overlapIsTrue()

1. **File Path**

Often when compareHap doesn’t work it is because the file path is not leading to the correct file or is not leading to a file at all

Type should be string so you need to have quotation marks around file and it needs a .ab1 at the end

Ex "C:/Users/amw346/Desktop/aabys11May16-3F kdrFL-R7 s kdrFL.ab1"

1. **Run compareHap()**

Function can now be used!

Example:

compareHap("C:/Users/amw346/Desktop/aabys11May16-3F kdrFL-R7 s kdrFL.ab1", newCombinedNoGapAllAdded7021)

Output will be the dataframe of matches see below for more detail

Expected Run time: around 22 minutes for 7000 entries in database

**Description of Functions**

**compareHap(file, master)**

Overview: will take an input chromatogram and return the pairs of possible haplotypes that could be added together to make the input

Format of inputs:

File: a file path. Type: string. Must put quotation marks around file path so that it is read as a string.

EX: "C:/Users/amw346/Desktop/aabys11May16-3F kdrFL-R7 s kdrFL.ab1"

Master: A database containing the sequences you want to compare file to. Type: data.frame.

Format of database must be:

Sequence of added haplotypes (string), name of one haplotype (string), name of second haplotype (string)

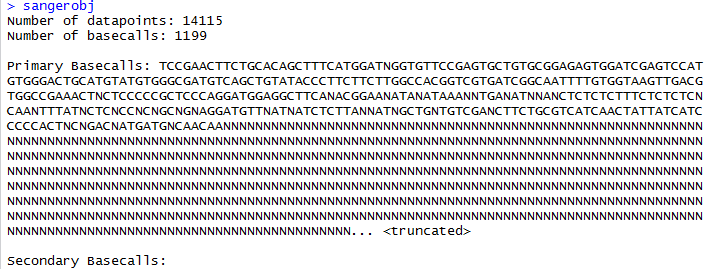
Currently arbitrarily cuts out the first 20 bases from input file. Easily can change this by modifying line 27 and 28.

Output: A dataframe of the names of the haplotypes that can be added together to get the input file

**clipIndex2(sangerobject)**

Overview: Will take in a sanger obj and return the index of the first instance of three NNNs.

Input: a sanger object. From package sangerseqR.

Example:

Output: Index of first occurrence of three Ns. Type: integer

Potential problems:

If you have a seq that starts with three N’s or has three Ns iway sooner than desired in the beginning somewhere then clipIndex2 will cause problems ex: ANNNACT

Solution: manually set index in compareHap to reasonable number

**addPriSeq3(pri, sec)**

Overview: inputs two strings that should already be of same length and outputs a string that is the combined version according to IUPAC codes

Input:

Pri: sequence 1 to be added. Type: string.

Sec: sequence 2 to be added. Type: string.

If the two strings are not the same length it will return an error.

**addbases2(a,b)**

Overview: takes in two characters and adds them together according to IUPAC codes

Input: both are character strings

Output: one character string accoding to IUPAC codes

Note: only allows the first character to be A,C,T,G, or N because the specification for makeBasecalls indicates that the primary call will always be the highest base in the window and the secondary calls may be two or ambiguous. addBases will therefore concatenate “input strings invalid” whenever this does not happen to alert you to check either the index or the primary and secondary calls

**MODcheckMasterList4(newseq, master)**

Overview: takes in a string and the database and will output a dataframe of matches for the string based on the input dataframe.

Criterion for beign considered a match: One is completely inside the other. They are exactly the same. The beginning of one matches completely the end of another. (overlapping)

Input:

Newseq = the sequence to be tested against the input database. Type: string

Master: the database. Type: data.frame. must be in the form specified above

Will print out the index and the message “match found” every time it finds a match because this made it easier to figure out what was going on in the output files but this can easily be removed by taking out the print statements on line 162 and 163

Output:a dataframe of matches in the form:

|  |  |  |
| --- | --- | --- |
| Sequence from input database that matched. Type: string | Name of haplotype 1 | Name of haplotype 2 |

**OverlapIsTrue(long, short)**

Function called in modcheckmasterlist4() to check if the two sequences match by an overlapping portion.

Input: the two sequences to be checked. Type: string

Warning: If you get the error: ‘Width NA’s’ it is probably coming from the pairwise alignment in this function. Reasons could possibly be index in MODcheckmasterlist4 are off and you are asking it to align something with value NA

Output: Returns True (type: Boolean) if they do match by overlapping

**Suggestions for analyzing homozygous chromatograms**

Instead of running through the entire database you can change the range in MODcheckmasterList4() to 6094:7021 so that it only runs through the homozygous sequences. Located on line 149 file FinalcompareHapAndDependencies.R Run the entire function MODcheckmasterList4() into the environment with this modification and then use compareHap normally on the chromatogram

**Suggestions for analyzing the cloned out haplotypes**

Trying to read in the chromatogram with compareHap may lead to problems depending on when the first NNN is located for the function clip2index() and may result in weird cut offs or other errors. Might be better to copy the sequence as a string into R and then call MODcheckmasterList7(“sequence”, newCombinedNoGapAllAdded). Running just this by itself will return the matches. Function located in FinalcompareHapAndDependencies.R

Example:

MD17\_18Jul17\_26M\_col1seq= "GAATTTCACCGACTTCATGCACAGCTTCATGATTGTGTTCCGAGTGCTGTGCGGAGAGTGGATCGAGTCCATGTGGGACTGTATGTATGTGGGCGATGTCAGCTGTATACCCTTCTTCTTGGCCACGGTCGTGATCGGCAATTTTGTGGTAAGTTGACGTGGCCGAAACTGCTCCCCCGCTCCCAGGATGGAGGCTTCTGATGGCCAATTAAAAAAAATTAAATCAACCTCTCTCTTTCTCTCTCTCTCAACTTTATTCCGTCCATCCGTTGCAGGTTCTTAATCTTTTCTTAGCTTTGCTTTTGTCCAACTTCGGTTCATCTAGTTTATCAGCCCC"

MODcheckMasterList7(MD17\_18Jul17\_26M\_col1seq,newCombinedNoGapAllAdded)

**Fuzzy match**

To allow certain deviation from the input sequence you can modify compareHap to use MODcheckmasterlist5() instead of MODcheckermasterlist4(). Located in FinalcompareHapAndDependencies.R at the bottom, this function has a key line change which can be modified to suit the deviation criteria:

found = (sum@nmismatch < 2) & (sum@ninsertion[2] ==0 ) & (sum@ndeletion[2] == 0)

Then run compareHap normally.