

Computer Practical – 2017

Use of R for the calculation and analysis of Tajima's D

Purpose and Scope

Using the DNAsp program you have already investigated sequence polymorphism in six *P. falciparum* genes using data from a Kenyan population. There are, however, over 5000 genes spread across the entire genome and analysing each of these using DNAsp would be a laborious process. In this exercise we will therefore utilise genome wide SNP data from a 2008 Gambian dataset (Amambua-Ngwa et al. 2012 PLOS Genetics). This data was aligned to the *P. falciparum* 3D7 reference genome using the same principles you learnt during week 1 of the module with SNPs subsequently called and filtered from the BAM files which were generated.

Loading the data

After opening R studio the first thing you need to do is to tell it which folder you will be working from by setting the working directory. For this exercise you'll be working from the folder called *R-intro* which is located on the D drive. To set the working directory use the following command:

```
setwd("D:/R-intro")
```

You can check which directory you are in by typing the following command into the console:

```
getwd()
```

We're also going to check the required files are present using:

```
list.files()
```

This should list files called *tjddata.txt*, *tjdfunctions.r* and *standardised_ihs.txt*.

The *tjdfunctions.r* file contains the custom functions we are going to be using in this exercise, if you wish to see how custom functions are written you can open this file in notepad or wordpad to see some examples. To use these functions we need to tell R to load these into memory, making them available for later. This is done with the command:

```
source("tjdfunctions.r")
```

As we have already set the working directory R will automatically look there for the *tjdfunctions.r* file, if it was in a different folder we would need to tell R that. After loading the file you should see a list of functions appear in the workspace window in the top right of R studio.

In addition to the functions we also need to load the data from *tjddata.txt* into a variable called *snpdata* using the command:

```
snpdata <- read.table("tjddata.txt", header=T)
```

Have a look at the *snpdata* variable using some of the commands you learnt in the introduction to R.

Each row in *snpdata* represents a single SNP and the first 12 columns represent annotation information for that SNP. Can you work out what each of these columns represent? What do columns 13 onwards contain?

How many SNPs are present in this dataset? How many samples are there?

As you have seen when using a function we typically type the command `functionname(argument)` where the argument may be a variable, such as *snpdata* or a file such as *data.txt* (not all functions need an argument however).

Using `head(snpdata)` you should have been able to view the first few rows of *snpdata*, which is a type of variable we encountered in the introduction to R called a `data.frame`. It is organised in a manner very similar to an excel spreadsheet with rows and columns. Remember that we can navigate through a subset of the data using `['row number', 'column number']` or `['row name', 'column name']`, for example:

```
snpdata[30,9]
```

```
snpdata[30,"gene"]
```

will both bring up the data from row 30, column 9 as column 9 has been named "gene"

If we want to get a range of rows or columns we can use the a colon (:) to specify a range of rows or columns, so

```
snpdata[5:20,]
```

would return rows 5 through to 20 (inclusive) of *snpdata*.

What are the limitations of retrieving data in this way?

Using this method try and find the position of the gene *PF3D7_0800600*

Can you find the gene using one of the approaches from the introduction to R?

Manipulating the data

Just as we can view data by using square brackets [] we can copy the results to a new variable, which we're going to do by creating two new variables, the first of which is called *datalegend*:

```
datalegend <- snpdata[,1:12]
```

As we haven't specified any rows before the comma *datalegend* now contains every row from columns 1 to 12 of *snpdata*.

Create a second variable called *datagenotypes* and then copy columns 13 to 64 of *snpdata* into it.

Check it by comparing the contents of *datagenotypes* to *snpdata*.

How many columns are there in *datagenotypes*? What is the first column name of *datagenotypes*?

Before we can calculate our Tajima's D scores we need to convert the data into a more usable format, which we do using the following two commands:

```
genos <- converts(datagenotypes, as.character(datalegend[,3]))
```

```
genos <- apply(genos, 2, as.numeric)
```

Examine the *genos* variable and compare it to the *datagenotypes* and *datalegend* variables, can you work out what these two commands are doing?

As we need to calculate Tajima's D for each individual gene we also need to extract a list of the gene names using the following two commands:

```
genenames <- names(table(datalegend[, "gene"]))
```

```
genenames <- genenames[genenames != "-"]
```

How many genes are listed in the *genenames* variable?

You can use the `length()` functions to check this.

Why might there be less genes listed than the 5772 that are present in the reference sequence?

Calculating Tajima's D

We're now ready to calculate Tajima's D for our dataset, before we do that have a closer look at the `tajmad1` function we're going to be using by clicking on it in the workspace window. This should bring up multiple lines of code, which would be a lot to type out each time we wished to calculate Tajima's D. By placing this code into a function we are able to run it simply by entering `tajmad1(arguments)` into the console.

To calculate Tajima's D you need to enter the following two lines of code:

```
tajimascores <- NULL

for (i in genenames){tajimascores <- rbind(tajimascores, tajmad1(genos,
  datalegend, i, 3, 1))}
```

which will calculate the Tajima's D scores for each gene and store them in the variable called `tajimascores`.

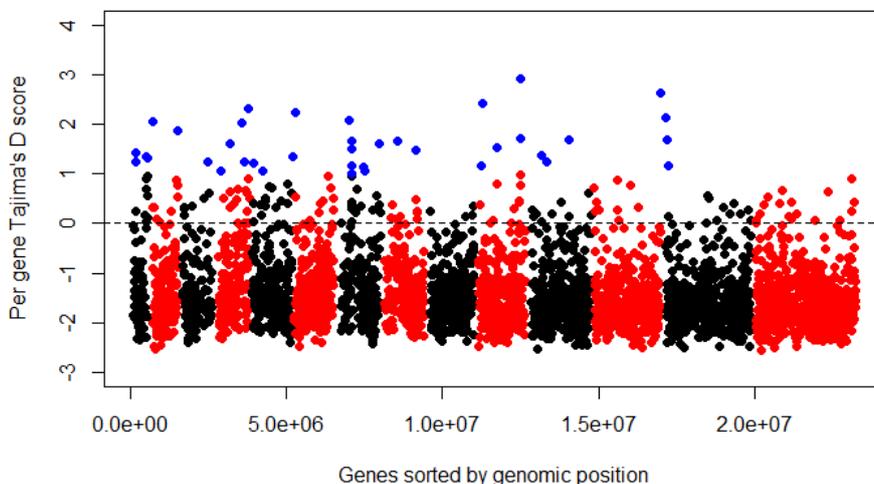
How many genes was Tajima's D calculated for?

Plotting the data

One of the biggest strengths of R is its ability to plot data, which we're going to do using a custom version of the plot function called `plotted`:

```
plotted(tajimascores, 1)
```

This will plot the Tajima's D score for each gene on the Y axis while the X axis indicates the position of the gene in the genome. For this visualisation we have coloured each chromosome in alternating colours (black and red) while genes with a score of above 1 are coloured in blue. It should look like this:



We now know that the majority of genes have a negative Tajima's D score and only a small number have a score above 1. The negative scores across much of the genome is due to the presence of an excess of rare alleles compared to that expected under a neutral model of evolution. In malaria this was caused by a historical population expansion, with the rare SNPs having entered the population subsequently.

In order to find out which genes have scores above 1 we can use the *which()* function, demonstrated below. Here we are using it to say "which rows in the *tajimasd* column of *tajimascores* are greater than or equal to 1" then using that to copy the data into a new variable, called *highscores* (For the purpose of this module you don't need to understand exactly how *which* works, just that it is possible to select data in this manner).

```
highscores <- tajimascores[which(tajimascores["tajimasd"] >= 1),c(1:3,6)]
```

How many genes have a Tajima's D score of ≥ 1 ?

What is the gene with the highest scoring Tajima's D score? What is its function? (the website www.plasmodb.org will be useful here).

Genes with high Tajima's D scores are predicted to be under balancing selection, indicating that there is an excess number of alleles with intermediate frequencies at these loci.

What sort of genes in malaria might be subject to this type of selection?

What processes might drive balancing selection? Do you think the top scoring gene fits this model?