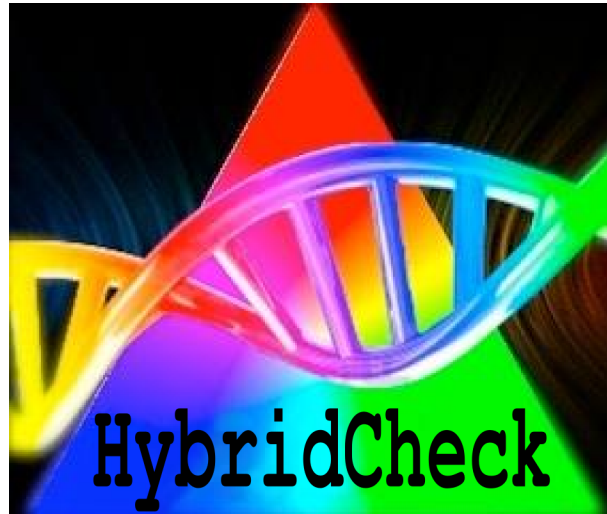# HybridCheck User Manual

Ben J. Ward

February 2015

HybridCheck is a software package to visualise the recombination signal in assembled next generation sequence data, and it can be used to detect recombination, genetic introgression, hybridisation and horizontal gene transfer. It can scan large (multiple kb) contigs and whole genome sequences of three or more individuals. The HybridCheck package has an optional user-friendly graphical interface. HybridCheck conducts ABBA-BABA tests, visualises the mosaic-like genome structure in high quality graphics, and estimates the coalescence times of each recombinant block.

Table of Contents

Introduction

This manual is written for researchers who are not familiar with using R, but who want to use HybridCheck. We have created a Graphical User Interface (GUI) for HybridCheck based on a web-app framework, in addition to launchers for Windows, OS X, and Linux systems. The GUI version of HybridCheck is intended for exploratory analysis of a limited number (up to ~10) of sequences.

When batch-processing many files and analysing many sequences, HybridCheck is best used as an R library in a script. When analyzing many sequences, please refer to the document "Programming with HybridCheck", but note however that in that case the user needs to be familiar with the R syntax.

What is needed before installing HybridCheck

HybridCheck is an R package, and R needs to be installed on the computer before installing HybridCheck. This section of the manual describes how to do this. An alternative is to use HybridCheck bundled inside a Docker container along with R and all the required dependencies. Because a full description of Docker is beyond the scope of this manual, for more info please visit the Docker website (https://www.docker.com), and the DockerHub webpage for HybridCheck (https://hub.docker.com/r/ward9250/dockerized-hybridcheckapp/).

Installing R on Windows

Go to the CRAN web page for R for Windows (http://cran.r-project.org/) and click on the link to get 'base' R and choose the latest version. This will download an installer for Windows. Run the downloaded installer and install R in the default location on the machine.

After this, click on Rtools (http://cran.r-project.org/bin/windows/Rtools/) and download the installer that is applicable to the version of R you chose to download and install. Once downloaded, click and install Rtools according to the default settings.

Installing R on OS X

Go to the CRAN webpage for R for OS X (http://cran.r-project.org/) and click on the latest version of R. Run the downloaded file, keep the

default settings and R.app will be downloaded to the Applications folder.

Installing R on Linux

The process for installing R on a computer running a distribution of the GNU/Linux operating system depends on which distribution is installed. The most commonly used distributions include Ubuntu and Fedora, and for such distributions, there is a CRAN page (http://cran.r-project.org/bin/linux/) with links to instructions for several common GNU/Linux distributions. The install process for a popular distribution like Ubuntu will involve adding the R repository to the package manager, and then using the package manager to install R, however the specifics will be found on the aforementioned CRAN Linux page.

Installing HybridCheck with the launchers for OS X, Linux and Windows

Go to the HybridCheck website and download the appropriate launcher for your operating system.

Installing HybridCheck on OS X

On OS X simply copy the downloaded .app file to the Applications folder. Clicking on the app from Launcher will present you with the option to Run HybridCheck or force an update of HybridCheck and the packages it depends on.

Installing HybridCheck on Linux

For a GNU/Linux distribution, extract the downloaded tar.gz compressed folder anywhere in your user space of your choosing. Contained within the extracted folder are two shell files, one is used to install HybridCheck and its dependencies to your R library and is called Install_HybridCheck.sh. After running this file. Restarting your terminal and running the command "hybridcheck".

Installing HybridCheck on Windows

On Windows, double click the downloaded file and go through the guided installer. Once the installer is finished, go to the 'Start Menu' and look under 'All Programs' where there will be a new HybridCheck section with three executable options:

HybridCheck

Runs the HybridCheck program.

Uninstall

Uninstalls the HybridCheck launcher. R and the R library will still be on the machine.

Update

Forces an update of the HybridCheck library and the packages it depends on.

Running HybridCheck on your machine with the launcher and Graphical User Interface (GUI)

In Windows, you must click on the HybridCheck option in the 'All Programs' menu. In OS X, you must run the app from Launcher and choose the Run HybridCheck option. In a GNU/Linux distribution, you can type 'hybridcheck' in a terminal window.

In all cases the launcher will check that HybridCheck, and all packages it depends on, are installed in the system's R library. Then the launcher will start R, load the HybridCheck package, fetch the latest version of the GUI, and open it in your systems default Internet browser.

Using HybridCheck with the GUI to analyse sequences

The HybridCheck GUI will open on the first out of five GUI sections; each section is dedicated to an analysis step:

Sequence Data

This first tab is the Sequence Data tab with one button to load in a FASTA formatted data file (Figure 1). Click the button and select a FASTA formatted alignment file from your computer to load.
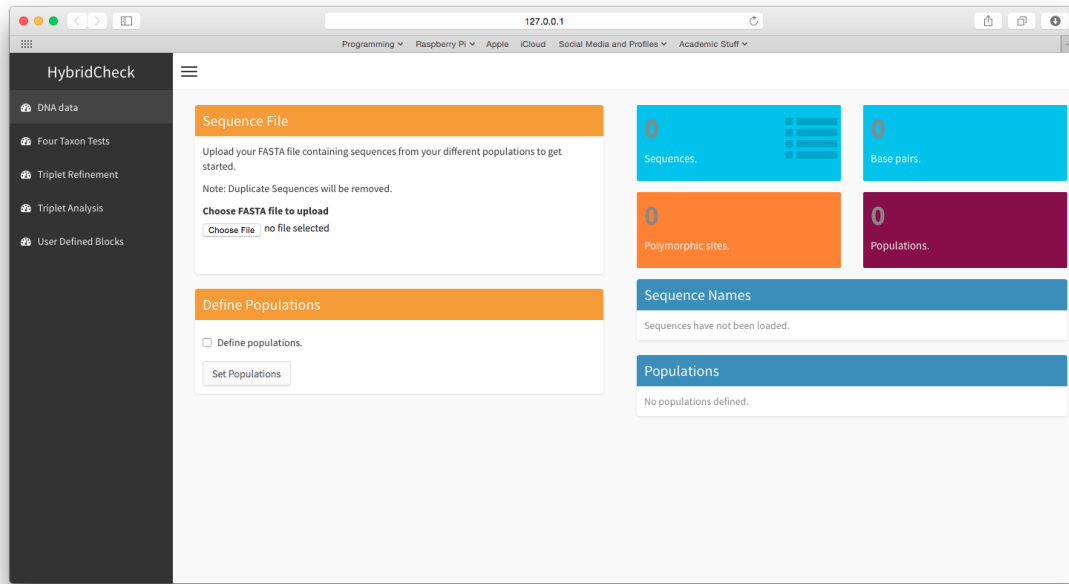
Figure 1. The DNA sequence file input screen.

After loading the DNA sequence data, the screen updates and displays some summary statistics about the sequences that have been loaded in. In the example (Figure 2), 10 aligned sequences have been loaded. By default, HybridCheck assumes every sequence is from a separate population (it calls these "unnamed_1", "unnamed_2" and so on). You can specify populations for your sequences using the "Define Populations" box (Figure 2): to set some populations, enter the number of populations and for each population, enter a name, and the sequences that make up that population. In the example (Figure 2) four populations are defined called 'Pop1', 'Pop2', 'Pop3', and 'Pop4'.
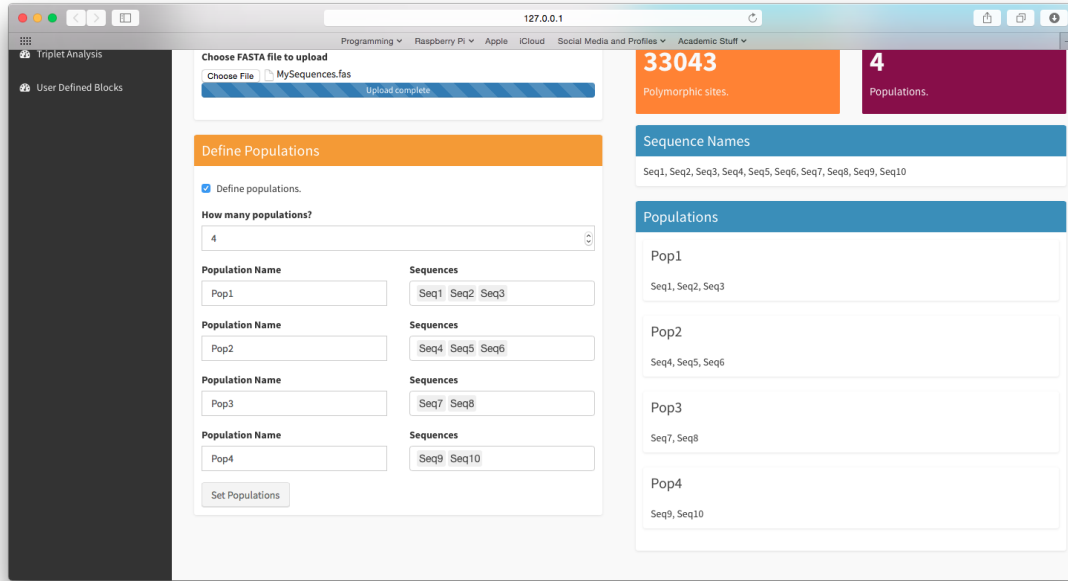
Figure 2. HybridCheck with sequences loaded and populations defined.

Perform Four Taxon Tests

After you have loaded the sequence data and defined the populations, you can either go straight to scanning sequence triplets, or perform some Four Taxon Tests (Figure 3). Four taxon tests compute D values and $\hat{f}_d$ values for sets of four populations or sequences. These values quantify the balance of ABBA and BABA sites, and the proportion of the genome that is consistent with a scenario introgression, respectively. For a test a population must be denoted P1, P2, P3, and P4, such that the relationship (((P1, P2), P3), P4) is assumed. In other words, P1 and P2 coalesced first, followed by P3 and finally P4, which is the most ancestral sequence. Two $\hat{f}_d$ statistics are calculated; one expresses the proportion of the genome consistent with a scenario of complete introgression between taxa P2 and P3, the other expresses the same value, but for a scenario of complete introgression between taxa P1 and P3.

For a given set of four taxa, HybridCheck will calculate D and $\hat{f}_d$ statistics for the entire length of sequences (hereafter referred to as the *global statistics*), HybridCheck then uses a leave-one-out jack-knife procedure in which the sequences of the four taxa in the test are

7

divided into non-overlapping segments. For each segment D and $\hat{f}_d$ scores are worked out and the global statistics are recalculated whilst excluding that segment, producing pseudo-estimates that are used to estimate the three global statistics of interest. This procedure also results in a Z score for the three global statistics. Thus, for every test HybridCheck provides the user with the three global statistics (D and the two $\hat{f}_d$ values), jack-knife estimates of those global statistics, and the same statistics but calculated for each segment, allowing the user to see changes in the stats across contigs, and the influence of individual segments on the global statistics.

HybridCheck allows you to specify the population combinations to test manually, alternatively the checkbox "Automatically generate" (Figure 4) causes HybridCheck to generate every non-redundant combination of four populations, and then assign the four populations to P1, P2, P3, and P4 based on their distances. After the combinations have been generated, you can run the four-taxon tests: the UI will display a box to select which tests you want to run – set to "ALL" by default. In addition, you can specify either a number of segments, or the length of segments to use for the jack-knife procedure. Run the four-taxon tests by clicking the "Run Tests" button (Figure 4), and you can select which test results to show on the screen (Figure 5), by default all will be printed. It may take some time for the results to be displayed. These results can be useful for refining triplets to be scanned for recombination signal, as described in the subsequent sections.
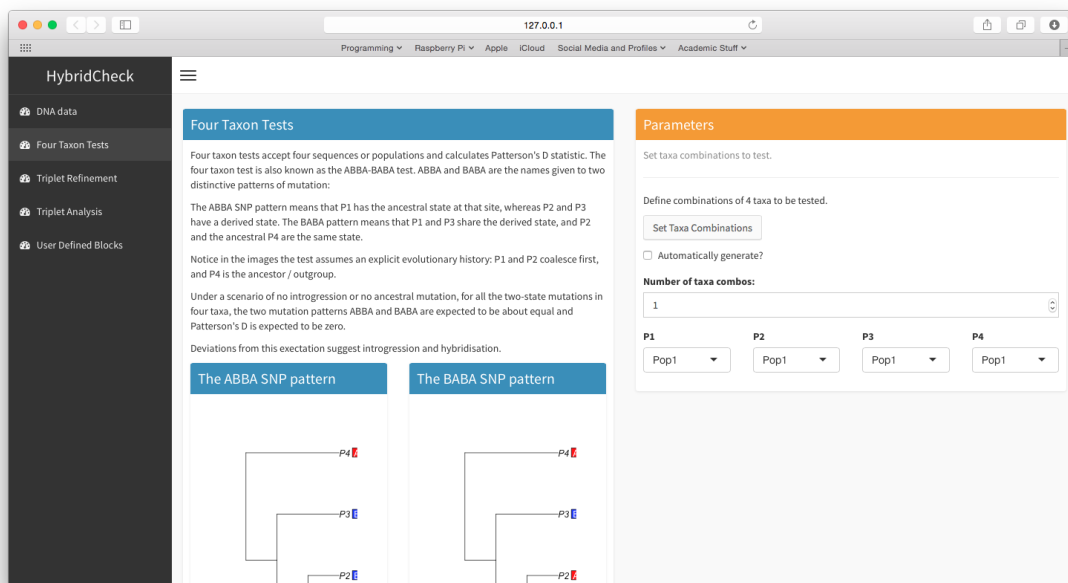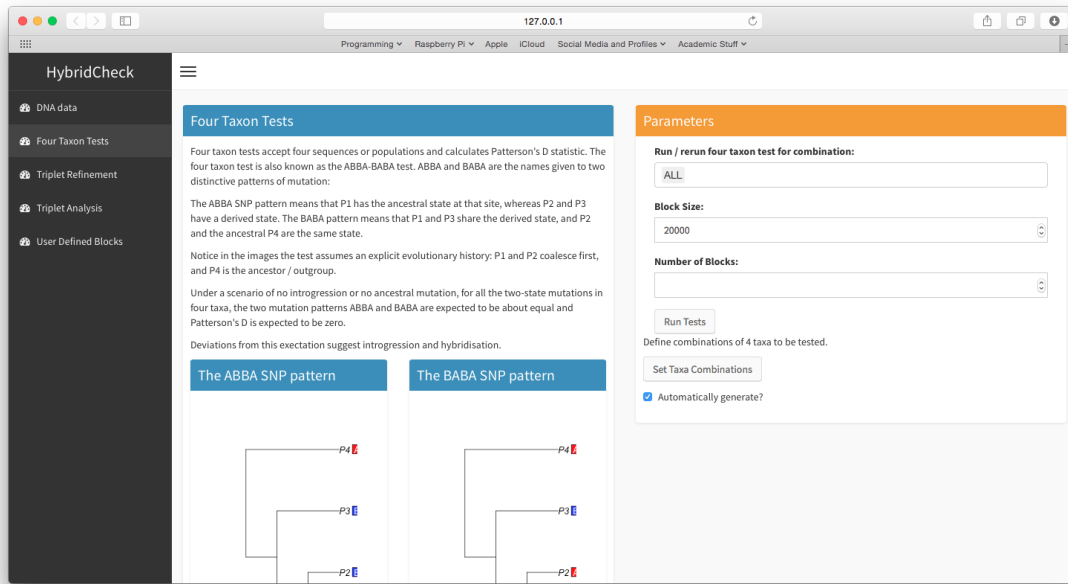

Figure 3. Four Taxon Test screen.

Figure 4. Running the four taxon tests. In this example, a jack-knife block size of 20,000bp is used.
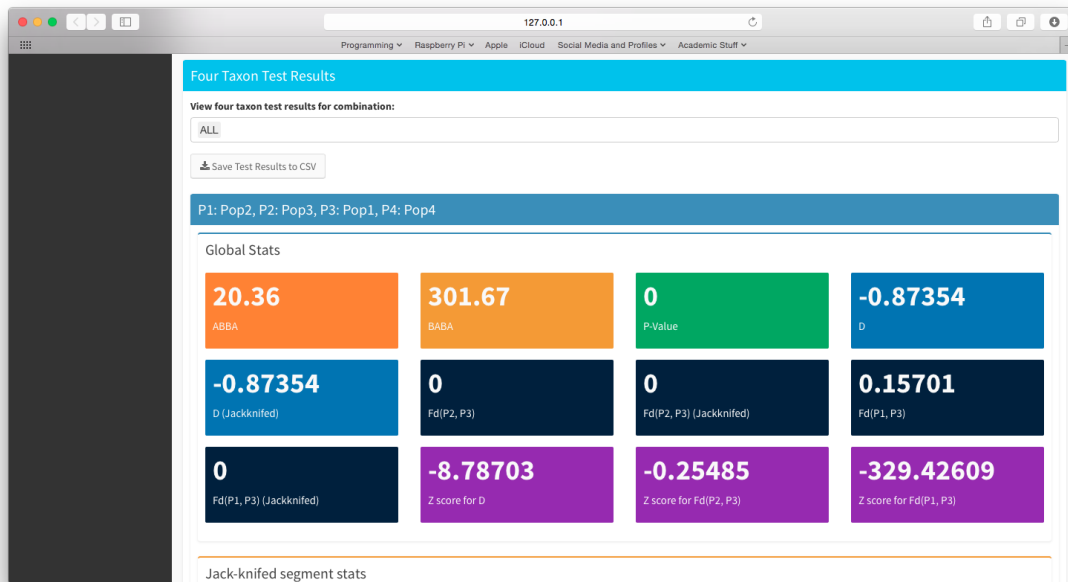

Figure 5. An example of the global statistics results printed for four taxon tests.

In Figure 5 the aforementioned global test statistics for a single test are shown. You can see from the image that the test was applied to the assumed history of (((Pop2, Pop3), Pop4), Pop1);. In the top panel are a series of stats computed for the whole length of the multiple sequence

alignment, including the ABBA and BABA scores, binomial P-value, D, two $\hat{f}_d$ statistics, and the jack-knife corrected values for D and $\hat{f}_d$.

Finally there are three Z scores for D and $\hat{f}_d$ based on the jackknife procedure.

In Figure 6 the table of statistics produced for each jack-knife segment is displayed, each row represents one non-overlapping jack-knife block.

The first nine columns are the statistics of D, $\hat{f}_d$ and P, computed for the local region of the sequence within the jack-knife segment. The column labeled "Block Fraction" quantifies how much of the total ABBA and BABA sites across the sequence are contained within the block, expressed as a proportion. This value quantifies how influential the block is on the global statistics – if a jack-knife block has extreme D or $\hat{f}_d$ values, but a low Block Fraction value, it may have an 'unfair' influence on the observed global statistics, and it's influence will be reduced when computing the jack-knife corrected global statistics. In the example in Figure 6, we see one jack-knife block contains almost all the two-state (ABBA and BABA) sites, so any other block with high or low D or $\hat{f}_d$ statistics has a larger influence on the observed global statistics than is warranted by the amount of all the two state sites it represents.

The remaining columns after the tenth "Block Fraction" column are the scaled pseudo estimates of the global statistics computed by leaving out the given jack-knife block.

Figure 6 Table of statistics computed for each jack-knife segment.

Specify Triplets to Analyse

In this section you determine which triplets to analyse. By default, HybridCheck will define which triplets to analyse based on the results of the four-taxon tests performed. If there are no tests performed then every possible triplet will be analysed. In the example figure, for 10 sequences there are 120 possible triplets (Figure 7), but due to the results of the four-taxon tests, HybridCheck has narrowed down the choice to 60 triplets to analyse. If you have no four-taxon test results, but still want to ignore triplets that are comprised of sequences from the same population or species, whilst keeping triplets comprised of sequences from separate populations or species, this option can be specified instead. To do this, simply switch the triplet generation method to "Generate triplets to scan for recombination between populations". If you select that option, you will be able to choose how many sequences from the same population are allowed in a triplet; one or two.

In addition, you can also select the checkbox marked "Raw p-distance based". This option allows you to exclude triplets containing sequences that are too closely related. This can be done by setting a manual

distance threshold, or by allowing HybridCheck to decide on the distance threshold used through analysing the distribution of the pairwise distances of all sequences (Figure 8).
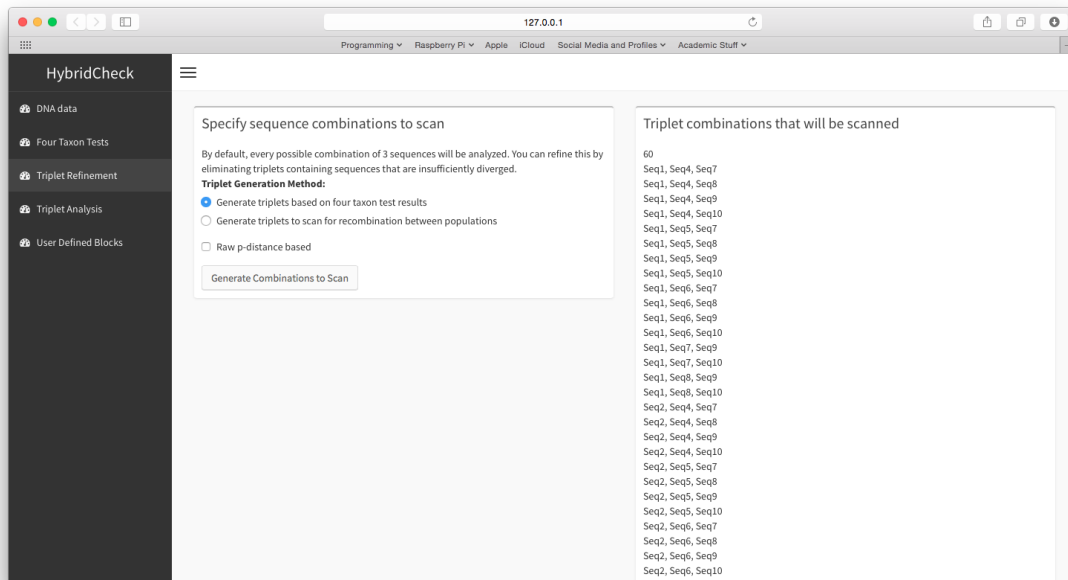

Figure 7. The triplet generation screen.
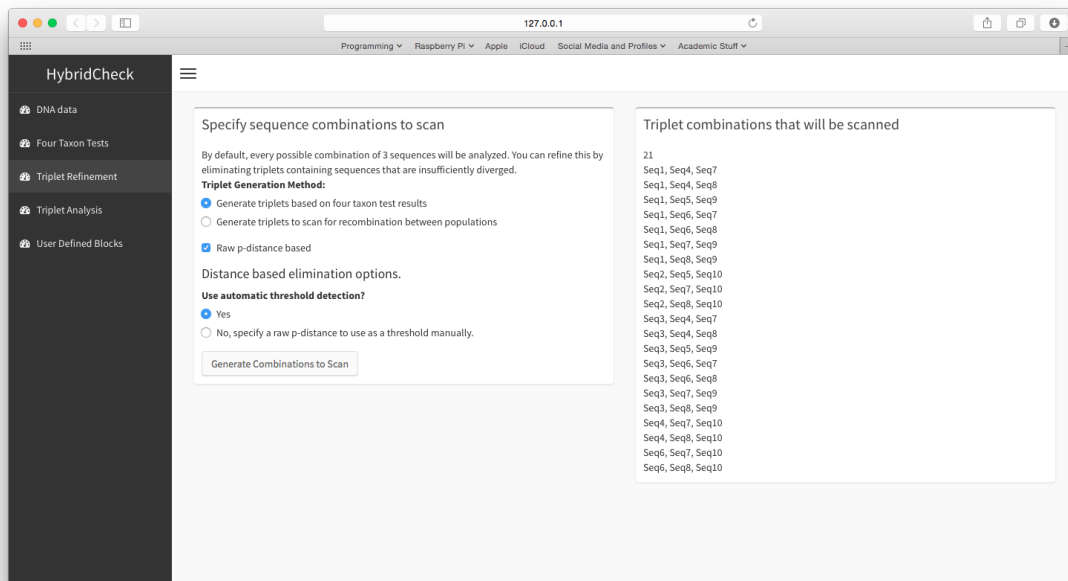

Figure 8. Triplets to scan have been refined according to the results of four-taxon tests, and according to the distances between sequences.

With the triplet selection set, the next section to complete is the *Triplet Analysis* stage.

Analyse Triplets

The *Triplets Analysis* section is where you set options for, and see the results of analyses of the triplets.

The page is split into two sections; the *Parameters* section and the *View Results* section (Figure 9). The Settings section is where you set the parameters for A) the sliding window scan, B) the block detection, and C) block testing and dating stages. It is also the section where you specify which triplets (generated in the previous section) will be analysed with those settings. By default every triplet generated in the previous stage will be analysed, as indicated by the *ALL* displayed in the *Run / rerun analysis for triplets* box (Figure 9).
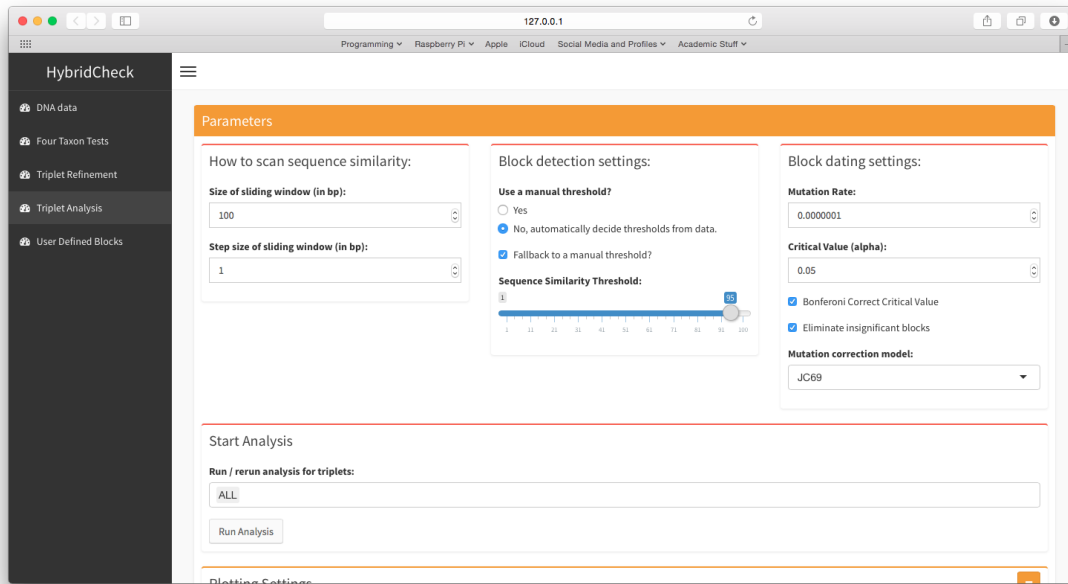
Figure 9. Analyse and explore sequence triplets section.

Settings to scan sequence similarity:

Set the sliding window size and step size for HybridCheck to use when it scans the pairwise sequence similarity across the informative sites of a sequence triplet. These two settings are located in the box titled *How to scan sequence similarity* (Figure 9). By default these values are 100 and 1, respectively.

Smaller windows for scans will pick up smaller regions of elevated sequence similarity that would be less visible in larger windows, but data may be prone to noise. Smaller windows and step sizes will also require more memory and more computation time.

Block detection settings:

Settings for block detection are located in the box titled *Block detection settings* (Figure 9).

HybridCheck can determine a suitable threshold for identifying significantly elevated regions of sequence similarity if you check the *No, automatically detect thresholds from data* checkbox for the *Use a manual threshold?* option. Alternatively, tick *Yes* for this option and then use the slider beneath to set a manual threshold for HybridCheck to use. It is recommended that the box labeled *Fallback to a manual threshold* remain checked.

14

Block dating settings:

The settings for block testing and dating are located in the box titled *Block dating settings* (Figure 9).

Set whether you want to *eliminate insignificant blocks* that fail an exact probability test. The formula for this test is given as:

$$Pr(X \leq k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$$

Where *p* is the proportion of mutations between the two aligned sequences to which the recombination block applies, and *k* is the observed number of mutations between the two aligned sequences, within the bounds of the potential recombination region. The default critical alpha for this test is 0.05, and values less than 0.05 pass the test. The alpha can be set by entering a new alpha in the box labeled *Critical Value (alpha)* or by checking the box labeled *Bonferroni correct critical value?*

Set an assumed substitution rate and select a choice of mutation model from the selection of *JC69, K80, F81, K81, F84, BH87, T92*, and *TN93*. These are used in estimating the divergence times of recombinant regions by solving the function for $2\mu t$:

$$f(n, k, 2\mu t, Pr(X \leq k)) = \left(\sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} 2\mu t^i (1-2\mu t)^{n-i} \right) - Pr(X \leq k)$$

Where $2\mu t$ is the divergence between the two sequences, *k* is the observed number of mutations in the block, and *n* is the size of the block. The root is then solved for *t*, using the assumed mutation rate provided by the user in the settings.

Once the settings in these three boxes are set as desired, and the triplets to be analysed or re-analysed with those settings are set as desired, hit the *Run Analysis* button, and wait whilst HybridCheck performs the sequence similarity scans for each triplet, and then identifies blocks in the scan data for each triplet, and then finally tests and dates blocks for each triplet.

Once you have run an analysis, you can view the results of a given triplet in the *View Results* panel. The top of this panel features a dropdown panel called *View Triplet,* which selects a triplet, the results of which shall be displayed to screen. The panel also features two buttons; one for saving the table of detected and dated recombinant

blocks for the selected triplet, and one for saving the generated plots to image file.

Below this pane, two panes showing plots of the recombination signal can be viewed (Figures 10 & 11). Figure 11 also shows the view of the data-table of detected recombinant blocks. This table prints one line for every such block detected and displays the two sequences that share the block, the location in BP, the number of SNPs in the block, the 95% CI of the age estimate, its assigned P-value, and the P-value threshold it had to beat to be counted as significant.

Figure 10. Viewing the results of an analysed triplet: A panel showing coloured heat-plots of recombination signal.

To save the plots, you can use the *Save Plots* button, visible in the top panel of the *View Results* section (Figur 10) and set a download destination and name for the plot in the resulting dialog window. Alternatively, right click on them and select "Save Image As...", exactly as for any image in a web browser.

Use the *Save Table* button visible in the top panel of the *View Results* section (Figure 10), and set a filename and destination folder for the comma-separated-values (CSV) formatted file. This file can be further opened and used in other programs such as spreadsheets software.

Re-running an analysis or editing the output:

If you have done an analysis but want to redo it with different analysis settings, simply change the options again using the interface and press the *Run Analysis* button (Figure 9) again. You can specify to re-analyse and view results of only a few interesting triplets out of the total set of triplets you first analysed using the *Run / rerun analysis for triplets* and *View Triplet* boxes, which are visible in Figure 9.

Figure 11. Viewing the results of an analysed triplet: A panel showing line-plot of recombination signal.

If you just want to change the aesthetic values of the plots however, you do not need to re-run the analysis, the plots are "reactive". This means that they are automatically redrawn for the user, whenever one of the settings that affect how the plot is drawn is changed. The settings for editing how the plots are drawn are in the collapsed panel labeled *Plotting Settings* (Figure 9). To view the settings for drawing the plots, click the collapsed pane to expand it and view its contents.

These settings allow you to make simple changes such as altering the colour, face, and size of axis labels, and include or exclude legends.

The settings provide a reasonable amount of flexibility, but this is very little compared to the flexibility gained by programming in R with the HybridCheck, grid and ggplot2 packages. However, whilst that provides maximum flexibility in the appearance of the plots, it requires the user to be comfortable at programming with R.

Analyse user defined blocks

There are a wide variety of recombination breakpoint detection algorithms available that vary in sophistication and ease of use, and HybridCheck provides a method of detecting blocks as described in previous sections of this manual. However, you may already know of some recombinant regions in their sequence data, either through searching the literature, or by using one of the other recombination breakpoint detection programs available. In such a case, you may not want to use the scanning and detection functionality of HybridCheck, but may want to use the significance testing and divergence time estimation functionality HybridCheck provides.

This is where the *User Defined Block* screen is useful (Figure 12):

Figure 12. Known recombination regions can be tested and dated with HybridCheck.

After loading DNA sequence file as described previously in this manual, you can add known recombinant regions between two sequences by filling the fields of the *Add or clear user blocks* panel, and then clicking the *Add user defined block between sequences* button. The added block is visible in a table below the panel (Figure 13).

After adding all the regions you want to test, check the settings in the fields of the *Block dating settings* panel, and click the *Test and date user blocks between sequences* button, the data-table will be updated with the results of the test and the estimated divergence times of blocks (Figure 13).
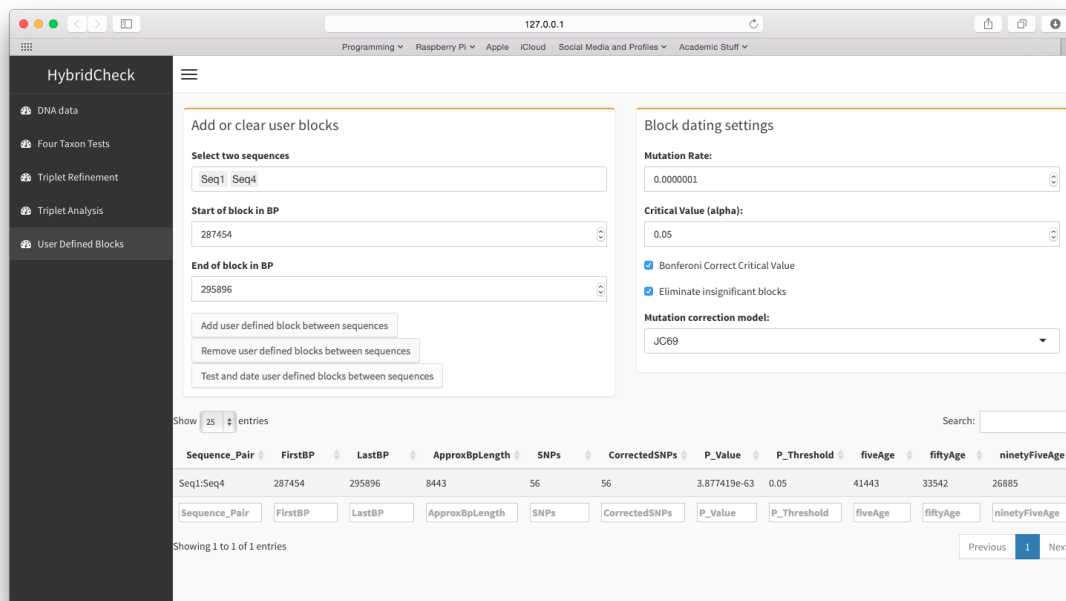
19

Figure 13. The user-defined block has a p-value, and a 95% confidence range for divergence times calculated for it.

Exiting the Program

Whilst you are using the HybridCheck graphical interface, an R session is running in the background and you can see its progress and printed messages in this console. Closing this background session will stop HybridCheck and the graphical interface will go grey and inactive, and you can simply close it.

Contacting us if you have problems

HybridCheck is an open source and continually improving set of software. If you get stuck or something goes wrong, you can contact Ben J. Ward at b.ward@uea.ac.uk. Alternatively, the GitHub repository for HybridCheck allows you to file a bug report, question, or feature request. It also has links to the website and the Gitter chat room for this software project. We will do our best to assist you; a sample of your data or code that can reproduce the error may be required.