RDP5
Instruction Manual

Contents

1 INTRODUCTION

RDP5 (Recombination Detection Program version 5) is a Windows VISTA/7/8/10 program for detecting and analysing recombination and/or genomic reassortment signals in a set of aligned DNA sequences. While a number of other programs have been written to carry out the same task (see Martin et al., 2011, and the web-site http://www.bioinf.manchester.ac.uk/recombination/programs.shtml), my motivation for writing RDP5 has been to produce an analysis tool that is both accessible to users who are uncomfortable with the use of UNIX/DOS command lines and permits a more interactive role in the analysis of recombination. I have particularly focused on making the program run with a minimum of fuss. This mean that it should be usable with most multiple nucleotide sequence alignments (unfortunately RDP5 cannot align your sequences for you, although the programs IMPALE, MUSCLE and CLUSTALW that are distributed with the RDP5 download can be used for this purpose) and should be able to give a detailed and reasonably accurate breakdown of the recombination events that have occurred during the evolutionary histories of the sequences being analysed.

The main strength of RDP5 is that it simultaneously uses a range of different recombination detection methods to both detect and characterise the recombination events that are evident within a sequence alignment without any prior user indication of a non-recombinant set of reference sequences. Besides the original RDP method, it includes the BOOTSCANning method (Salminien *et al*., 1995; Martin *et al*., 2005b), the GENECONV method (Padidam *et al*., 1999), the Maximum Chi Square method (MAXCHI; Maynard Smith, 1992; Posada and Crandall, 2001), the CHIMAERA method (Posada and Crandall, 2001), the Sister Scanning method (SISCAN; Gibbs *et al*., 2000), the 3SEQ method (Lam *et al*., 2018), the VisRD method (Lemey *et al*., 2009), the PHI test method (Bruen *et al*., 2006) and the BURT method.

If you are impatient and want to start analysing your sequences without reading the manual it is strongly recommended that you go straight to the step-by-step guide in section 10. This guide will help you use the program in the way it was intended to work. Also, if you want to run the program under Windows VISTA/8/10 you will need to give RDP5 administrator rights. Find out how to do this in section 12.5.

2 OPENING ALIGNMENTS AND OTHER FILES

A number of different alignment file formats are recognized by RDP5 including PHYLIP, GDE, FASTA, CLUSTAL, GCG, NEXUS, MEGA and DNAMAN. To open a file press the "Open" button (Fig 1 in the command button panel) and select the file to be opened. The directory from which files are loaded and the last files analysed are "remembered" by RDP5 when it is shut down. Once loaded the aligned sequences and their names are displayed in the "sequence display panel" (Fig 1). Also displayed are the degrees of nucleotide identity in different regions of the aligned sequences in an "identity display panel" (Fig 1). When analysing datasets where sequences have been obtained either from different genomic components (in the case of viruses) or different genomic loci (in the case of bacteria), and these sequences have been concatenated for analysis, RDP5 can be made aware of the concatenation points by denoting them in an alignment using "!" symbols inserted at appropriate points within the first sequence of the alignment. When inserting these symbols make sure not to knock the first sequence out of alignment.

Besides alignment files, RDP5 project files (with a ".rdp" or ".rdp5" extension) may also be loaded. In addition to aligned sequences these files also contain information on possible recombination events detected in previous analysis sessions.

When an alignment file is opened in RDP5 it will automatically screen the NCBI virus reference genome sequence database for a reference genome that can be used to identify the starting and ending positions of genes. If the sequences loaded are either novel virus species that have not yet been assigned a reference genome by the NCBI, or are not virus sequences, gene positions can be mapped by two alternative methods: (1) by opening a GenBank file that contains information on gene start and end positions, or (2) by opening a a flat text "ORFMap" file. ORFMap files can be manually made in a text editor such as wordpad. The first line of an ORFMap file should have

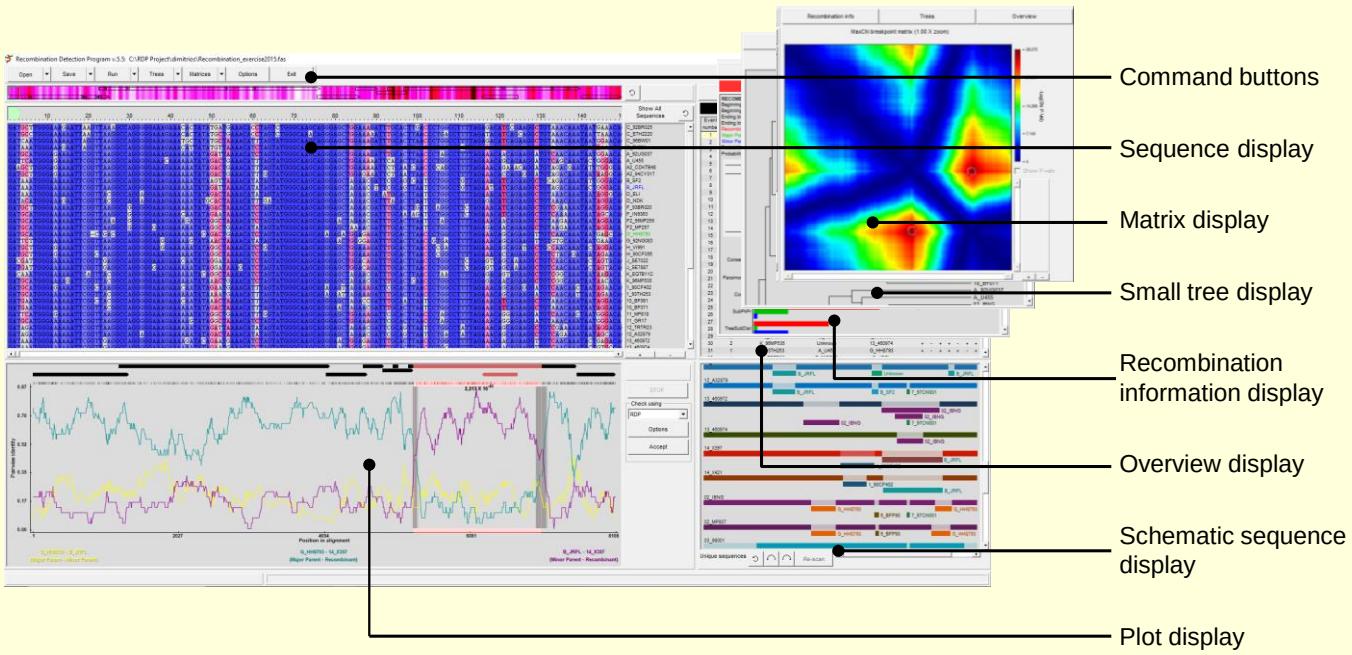

Figure 1. The main components of the RDP5 interface. Once sequence files (in any of a variety of formats) are loaded with the "Open" button in the command button panel, pressing the "Run" button will begin a automated exploratory scan for recombination with whatever analysis options are currently set (to see these press the "Options" button). Various phylogenetic trees and alignment analysis matrices can be drawn by pressing the arrows besides the "Trees" and "Matrices" buttons. Note that the small tree, matrix, recombination information and overview displays cannot be viewed simultaneously. It is possible to swap between these displays using the buttons above them.

the text "[ORF]" and each subsequent line should have three comma separated values in the following order: <genename>, <start nt coordinate>, <end nt coordinate>. To successfully load gene start and end positions GenBank and ORFMap files must be opened after an alignment file. In the case of GenBank files, one of the sequences within the multiple alignment must be the same as the sequence in the GenBank file. RDP5 will automatically scan the sequences in the alignment to check whether any match the sequence in the GenBank file. For ORFMap files the coordinates in the file must map either to the alignment or to one of the sequences within the alignment: the program will ask you how to interpret the coordinates and, if necessary, ask you to indicate the sequence to which the coordinates refer. If gene boundaries are available to RDP5 and breakpoint distribution analyses are performed, RDP5 will automatically test for variations in recombination breakpoint distributions relative to ORF boundaries as described in Lefuvre *et al.* (2009). If you are unable to load a particular GenBank or ORFMap file successfully, send me the file (at darrenpatrickmartin@gmail.com) together with your alignment and I'll fix the problem for you.

RDP5 can also read protein structure information from .pdb files. If the genome regions being analysed encode proteins with associated structures, any number of different .pdb files can be loaded. These .pdb files can include those containing multiple interacting proteins and RDP5 will automatically extract all information on the potential interactions of all amino acids encoded in the analysed alignments. Once .pdb files are loaded atomic coordinate positions can be used in protein SCHEMA analyses (See section 9.4; Voigt *et al*., 2002). Such analyses are described in Lefeuvre *et al.* (2007), and can be used to determine whether detectable recombination breakpoint distributions are influenced by natural selection acting against recombinants with disrupted intra- and/or inter-protein amino acid interactions (such as those that are respectively required for proper folding and optimal inter-protein binding).

3 SETTING ANALYSIS OPTIONS

Pressing the "Options" button in the command button panel will allow you to modify RDP5's settings. For casual users of RDP5, the program's default settings should work fine for most datasets. The only settings that you should ever need to change are *italicised in blue* below but should usually (unless you really know what you are doing) include only the (1) list of methods that should be used for automated recombination analyses (2) the window size settings of the various individual methods (3) the tree settings (where you can change substitution models and bootstrap replicates) and (4) the recombination rate settings. Unless you are particularly interested in exploring the influences of the various other settings it is OK to skip to section 4 of the manual.

3.1 General Settings

3.1.1 General recombination detection options. The various recombination detection methods can be set to perceive sequences as being either *linear or circular*. Note that even linear sequences can be analysed as though they are circular and this will in no way invalidate the analysis results unless an analysis of recombination breakpoint distributions is intended (see section 9.1). If linear sequences are analysed as though they are circular and some recombination is detected in an alignment, a strong recombination hotspot might be identified which spans the beginning and ending of the analysed sequences. While this will correctly indicate that the ends of recombinants tend to be inherited from different parental sequences, it should not be interpreted as the ends of the analysed sequences being genuine recombination hotspots. If recombination breakpoint distributions are of interest it would almost always be best to tell the program whether the sequences being analysed are linear or circular.

The *highest acceptable p-value* setting is the highest acceptable probability that observed patterns of nucleotide variation could have occurred in the absence of recombination/reassortment (the calculation of p-values differs for the different methods and will be discussed in section 8). The optimal highest p-value setting varies depending on the number of sequences in the alignment being analysed, the recombination methods being used to examine the alignment, the size of the sliding windows that are used (for RDP, Bootstcan, MAXCHI, CHIMAERA and SISCAN), and on whether the multiple comparison correction setting is on or off.

The default setting for *multiple comparison correction* is "on" as this makes the calculated p-values "experiment-wide" (or global) rather than "currently selected sequence triplet/pair wide" (or local) estimates of probability. Note that there are two multiple comparison correction "on" settings. The default is "Bonferroni correction" but a modification of this called "step-down correction" is also offered. These corrections

act as p-value modifiers that decrease the p-value cutoff according to the size of the dataset being examined. For a highest acceptable p-value setting of 0.05 with multiple comparison correction "off" you would expect that approximately 5% of p-values that are calculated would make the p-value cutoff by chance alone (i.e. without the need to invoke recombination). For a large dataset you would therefore expect many false positive results. For the same p-value cutoff but with multiple comparison correction set to "on" you would expect to only encounter one false positive in ~5% of the datasets that are examined. In most situations (<100 sequences with analysed sequences sharing >70% identity) a highest acceptable p-value setting of 0.05 when multiple comparison correction is on, or a p-value setting of 0.0001 when multiple comparison correction is off, should give few false positives but still enable the identification of most detectable recombination events. If the correction setting is off the p-value cut-off must be very carefully selected based on the number of false positives you are prepared to tolerate. When a large dataset containing sequences with low diversity (e.g. 100 sequences all sharing >95% identity) is analysed it may in fact be impossible to detect any of the recombination that is present if one of the multiple comparison correction settings are on. In these cases it may be best to analyse the dataset using the permutation tests offered (see section 3.1.2) with the multiple comparison correction setting off and a p-value cut-off of 0.001 – this will give you some idea of the expected false positive rate for each identified recombination signal. Be warned, however, that the permutation test should be used with extreme caution.

3.1.2 Permutation options: Unless you really know what you are doing leave the "number of permutations setting" at 0. In almost all cases the analysis results you will get without running permutations will be more credible than those that you obtain if you use this permutation test. If this setting is set to anything other than 0, RDP5 will run its automated recombination detection analyses in permutation mode. This involves generating a group of simulated recombination free datasets (the number that are simulated is specified by you in the space provided), which are then analysed by the program using the exact same settings that it uses to analyse a real dataset. There are several ways in which the results from such an analysis can be interpreted. Firstly, if RDP5 identifies more recombination events in the real dataset than it does in 95% or more of the simulated datasets then this is equivalent to a p-value <= 0.05 that there is no recombination evident in the dataset – i.e. you can be more than 95% sure that there is some evidence of recombination in the dataset. This result does not, however, tell you which of the detected recombination events are actual recombination events – the result simply tells you that some of them are probably real. Secondly if RDP detects a single recombination signal in the real dataset that has a better associated p-value than the best signals in 95% or more of the simulated datasets then this is the equivalent of saying that this signal has an associated p-value <= 0.05 – i.e. that you can be 95% confident that the recombination event associated with this recombination signal is a real event and not a false positive.

RDP5 can use two different approaches to simulate the sequences used in the permutation test. The simplest involves shuffling alignment columns to destroy most of the recombination signals evident in an alignment. While this has the pleasing effect of maintaining most of the properties of the sequences in the alignment (such as their phylogenetic relatedness and nucleotide composition), it does not maintain in the shuffled alignments the same spatial distribution of variable sites found in the original alignment. Maintaining the distribution of polymorphic sites in an alignment can, however, be important when evolutionary rates vary widely in different regions of the sequences being analysed. This is important for two reasons. The first is that it is generally easier to detect recombination in parts of an alignment where there are many polymorphic sites than it is in parts of an alignment with few polymorphic sites. If the distribution of detectable recombination breakpoints along an alignment is significant then so too will be maintenance of the spatial distribution of polymorphic sites in the simulated alignments. The second reason that spatial distribution of sites is important is that in very diverse parts of an alignment sequences are often poorly aligned. All recombination detection methods in RDP5 are particularly sensitive to sequence misalignment and whereas false positive signals due to misalignment of highly diverged sequence tracts in the real alignment will be detected as recombination events with significant p-values, these false positive signals will likely be undetectable in the shuffled alignments.

To solve this problem, the second (and default) method that RDP5 uses to simulate datasets employs the program SEQ-GEN to generate alignments with approximately the same spatial distribution of polymorphic sites as the real dataset (the "Use SEQGEN parametric simulations" setting). To obtain an appropriate spatial distribution of polymorphic sites in different parts of the alignment, different groups of columns in the alignment are separately simulated by SEQ-GEN where the input tree is scaled to reflect the degree of nucleotide diversity of the particular set of alignment columns being simulated.

Be very careful when using the permutation settings. Besides the program running very slowly, it may also crash unexpectedly. If you are sure that this kind of analysis is what you need and experience problems with it please e-mail me at darrenpatrickmartin@gmail.com and I'll do my best to help.

3.1.3 Data processing options: Once RDP5 has scanned an alignment and enumerated all detectable recombination signals, it begins the (often quite time consuming) task of trying to distil all the detectable recombination signals down to a minimal set of unique recombination events that could account for the signals. This process is necessary if you are hoping to make sense of the program's results because a single actual recombination event will almost always be detectable using multiple combinations of sequences in an alignment.

The "*require topological evidence*" setting allows you to specify whether you want the program to discard recombination signals that have no phylogenetic support. While this might seem an obvious thing to do, you should realise that many of the recombination detection methods implemented in RDP5 are fully capable of detecting real recombination events that do not result in any detectable change in phylogenetic tree topologies along an alignment. The default setting is that topological evidence is required but this is simply because most users (for good or bad reasons) would find this setting most desirable.

During automated scans the different detection methods will identify regions of sequence that are recombinationally derived. The boundaries of these regions, called breakpoints, will often be obviously suboptimal and selecting the "*polish breakpoints*" setting will prompt RDP5 to look, for better breakpoints using the BURT method (see section 8.13) in the immediate vicinity of those identified. Even if this setting is used you should realise that the program will still potentially identify the wrong breakpoint position – read section 10 on how to correct the obvious breakpoint detection errors that the program makes.

As mentioned earlier, misalignment of sequences is a major cause of false recombination signals. RDP5 is able to automatically assess whether the recombination signals it has detected are the product of misalignment. While it is possible to tell the program to not bother checking the consistency of alignments in the areas where it detects recombination signals (it makes the program a little faster), this is not advisable unless you are examining recombination in very good alignments with either no or very few inserted gap characters.

When it is trying to piece together a plausible set of recombination events that explain the recombination signals it has detected, RDP5 can be told to disallow the detection of recombination events in which one or both of the inferred parental sequences are themselves recombinant. This "disentangle recombination signals" setting should, however, only be used for datasets in which recombination is relatively rare (i.e. <10% of the analysed sequences are recombinant). If it is used for complex datasets where most of the sequences are recombinant, it can cause the program to get stuck in a never-ending analysis loop whenever it cannot find a viable set of recombination events that does not involve recombination between recombinant sequences. You should also be aware that there is no natural law that prevents recombinant sequences from recombining with one another (i.e. the actual parental sequences of some recombinants might in fact also be recombinants).

When RDP5 attempts to determine whether similar recombination signals that are detected in two or more different sequences might mean that these sequences all descended from the same recombinant ancestral sequence it is possible to make the rigor with which RDP5 does this more or less conservative with the "*group recombinants realistically/conservatively*" setting. The "realistic" version of this setting will ensure that groups of two or more sequences that are listed as having descended from the same recombinant ancestor could all plausibly cluster together within phylogenetic trees that are constructed from a portion of the analysed alignment that spans one or the other of the detected recombination breakpoints. The "conservative" version of this setting will identify sequences that have similar breakpoint patterns and similar degrees of genetic relatedness to the identified parental sequences, as having descended from the same recombinant ancestor even when there is no strong phylogenetic evidence that these sequences all share a more recent common ancestor with one another than they do with the remainder of sequences in the analysed dataset. The conservative setting is called

conservative because it will result in fewer unique recombination events being identified than the realistic setting.

When more than one recombination signal detection method is used to scan an alignment, the "*list events*" setting can be altered so that RDP5 will only display evidence detected by greater than a certain number of methods. If, for example, six methods are used during the primary screen for recombination (see below what the difference between a primary and a secondary screen is) and the "list events detected by >2 methods" setting is used RDP5 will only display recombination results that could be confirmed by between three and six different methods. If, after an analysis is completed, you would like to either relax this setting or make it stricter, you can do so and the list of detected events will then be instantly updated (i.e. unlike all the other settings described here, this setting can also be meaningfully changed even after the initial recombination screen is completed).

3.1.4 Analyse sequences using: RDP5 allows you to automatically analyse sequences for recombination using seven different recombination detection methods (see section 8 for a detailed description of the methods). These are the original RDP method, the BOOTSCAN/RECSCAN method (Salminen *et al*., 1995; Martin *et al*., 2005b), the method applied in the program GENECONV (Padidam *et al*., 1999; Sawyer, 1989), the MAXCHI method (Maynard Smith, 1992; Posada and Crendall, 2001), the CHIMAERA method (Posada and Crendall, 2001), the SISCAN method (Gibbs *et al*., 2000) and the 3SEQ method (Lam *et al*., 2018). It is possible to use the different methods either alone or in combination with one another. An indicator of the relative execution times of the different methods and an estimate of total execution time is given. Be warned that (1) estimates of relative and total execution times may be inaccurate and (2) the different methods may have vastly different speeds – take note when you are told that the analysis you are proposing will take a number of days or weeks. Also notice that BOOTSCAN and SISCAN have two associated selection boxes. If the left boxes are selected the methods will be used to explore for new recombination signals. If the right boxes are selected the methods will only be used to examine sequences in which recombination signals are detectable by other "primary scanning" methods that have been selected. This "secondary" scanning mode is also available for the LARD method. The reason these methods may be selected so that they will only run in this secondary scanning mode is that they are a lot slower than the other automated recombination signal detection methods implemented in RDP5. When analysing large datasets, therefore, it will often be desirable to explore for recombination signals using the fast methods and then use the slower methods to verify these results. Note that regardless of whether the 3SEQ, RDP, GENECONV, MAXCHI or CHIMAERA methods are selected for primary scans, these methods are so quick that they will always all be used in secondary scans of recombination signals detected by other methods.

3.2 RDP Method Settings

3.2.1 Reference sequence selection. Reference sequences used for identifying phylogenetically informative sites during analyses can be selected in five different ways. The default setting is to "use no reference" which means that all sites will be examined irrespective of whether they are phylogenetically informative or not. Whereas I have found that this setting provides the greatest power for recombination detection, it does tend to identify some false positive signals if very divergent sequences are being examined (i.e. if there are sequences sharing <60% identity in the alignment). This is not a problem if: (1) only recombination signals detected by multiple methods are to be accepted as genuine evidence of recombination, and (2) you heed the "this recombination signal may have been caused by a process other than recombination" warning in the recombination information and/or tree displays. You should take this warning seriously whenever you see it: it means that there is a good chance (maybe >50%) that the detected recombination signal is a false positive.

If the RDP method is to be used alone for an analysis of medium-large datasets (>30 sequences) containing both closely related and highly diverged sequences, I have found that the "using internal references only" setting provides the best unambiguous estimates of recombination breakpoints and the lowest frequencies of false positives. If small datasets are being examined (< 30 sequences) the "use internal and external references" setting would be better. For very small datasets (<5 sequences) the "use no reference sequences" setting is always recommended as long as all the sequences in the dataset are >70% identical. If you are examining a dataset containing a group of closely related sequences and you have access to a not too distantly related outlier sequence, then the outlier can be used as the "user defined reference sequence." This setting is, however, not recommended. Note that while the "use internal and external references" setting is meaningful for small datasets, as datasets become larger, the behaviour of an analysis with this setting will begin to approach that of the "use no references" setting. If accurate identification of breakpoints is desired it is not recommended that the "use external references" or "user defined reference" settings be used.

3.2.2 Recombination detection options. The *window size* used by the RDP method when scanning for evidence of recombination may be set. Note that the RDP method only examines polymorphic sites within triplets of sequences sampled from the alignment and the window size here refers to the number of these sites included in every window. While larger window sizes will lower signal:noise ratios but decrease the sensitivity of the analysis, smaller window sizes will increase the sensitivity but also increase the possibility of false positives.

Because some of the reference sequence settings can lead to a higher than desirable false positive rate when divergent sequences are being analysed, there is also a setting that will restrict RDP analysis to sequences that share identities that fall within a given range. This is also useful if, for example, within a genus an analysis of inter-species recombination is desired. If it has been determined that members of a virus species share greater than 90% identity whereas members of a genus share greater than 80% identity, only inter species recombination within a genus will be detected if the "only detect recombination" values are set to 80 and 90.

3.3 GENECONV Settings

For additional information on GENECONV settings please consult the GENECONV manual. It can be obtained online from: http://www.math.wustl.edu/~sawyer/geneconv/

3.3.1 Sequence options. In RDP2 GENECONV could be set to screen sequences in an alignment in either pairs or triplets. In RDP5 only the triplet scan can be used for automated recombination signal detection with GENECONV and the "scan sequence pairs" setting can only be used during manual recombination detection. When the "scan sequence pairs" setting is used GENECONV will identify variable alignment positions as polymorphic sites and then check every possible sequence pair for evidence of recombination. If the "scan sequence triplets" setting is chosen the program will treat every possible sequence triplet in an alignment as independent alignments and screen them as it would if it were using the "scan sequence pairs" setting. Because there are many more possible sequence triplets in an alignment than there are sequence pairs, the triplet setting will have a more stringent multiple comparison correction than the pair setting. See section 8.2 for a detailed account of how screening triplets differs from screening pairs. I personally prefer the triplet setting as it yields results which are more consistent with the other automated recombination signal detection methods that are implemented in RDP5. This consistency greatly simplifies the task RDP5 faces when trying to reconcile all the recombination signals various methods have detected during its formulation of a feasible scenario of recombination events at the end of an automated analysis. Note, however, that the enforced triplet setting prevents the use of many standard GENECONV settings. The reason for this is that triplet scans are performed directly by RDP5, whereas RDP5 uses the GENECONV.exe to do pairwise scans.

The way in which gaps (or indels: "-" or "." Insertion symbols which are used to align sequences optimally) are handled can also be altered. A group of consecutive "-" insertions that correspond with nucleotides in another sequence can be treated as a single polymorphism, each individual insertion can be treated as an individual polymorphism, or gaps can simply be ignored. The best setting will depend on the alignment being analysed. If the sequences in the alignment have diverged somewhat and the alignment process has inserted a large number of gaps, it is probably best that each run of gaps be considered a single polymorphism. When gaps are ignored the program performs similarly to when runs of gaps are treated as a single polymorphism, except that occasionally the latter setting increases the number of polymorphisms. An increase in the number of polymorphisms may enable the identification of more difficult to detect recombinant regions. Stanley Sawyer (the author of GENECONV) recommends that the "treat each indel site as an individual polymorphism" setting never be used.

3.3.2 Fragment list options. The *G-scale* setting will influence how GENECONV handles nucleotide mismatches. Setting the G-scale to 0 will not allow mismatches within a fragment (See section 8.2 for

information on what a fragment is). Setting the G-scale to 0 is a special case that sets an infinitely high mismatch penalty. Setting G-scale to 1, however, sets the lowest possible mismatch penalty. Increasing the G-scale above 1 increases the mismatch penalty - at very high values the mismatch penalty will approach that used when the G-scale is set to 0. There is no optimal G-scale setting and it should be adjusted according to the dataset being examined – For detecting recent recombination events a G-scale of 0 or a G-scale with a high value (5+) would probably be best. For detecting older recombination events a G-scale value of 1 or 2 would probably be best. I personally only ever use a G-scale of 1 (the default). .

During its execution, GENECONV can be set to ignore potential recombinant regions that (1) have less than a certain length (the "Min. aligned fragment length" setting), (2) have fewer than a certain number of polymorphic sites (the "Min. polymorphisms" setting which is useful for differentiating between sequence conservation and recombination), and (3) have pair-wise scores that are below a particular cutoff (the "Min. pairwise frag score" setting). The program can also be set to ignore fragments with higher p-values that overlap with fragments that have lower p-values. By changing the "Max. overlapping frags" setting to >0 the program will report a specified number of potential recombinant regions that overlap with regions that have smaller p-value.

3.4 BOOTSCAN/RECSCAN Settings

3.4.1 Scan options. The *window and step sizes* used during BOOTSCANning should be carefully selected based on the length of the sequences being analysed, their relatedness and the sizes of recombinant regions that are anticipated. Note that the duration of a BOOTSCAN is effected far more by step size and number of bootstrap replicates than it is by the window size. The step size used must be smaller than the window size and should ideally be set to less than 50% of the window size. Window sizes should be selected so that, on average, there will always be more than ~10 variable nucleotide positions within every window examined. Whereas larger window sizes will increase signal:noise ratios, you should understand that obvious recombinant regions that are only slightly smaller than the window size may not be detected.

There are three different settings that determine how sequence relationships are measured during a BOOTSCAN. The "Use distances" setting will permit the quickest BOOTSCANs because, with it, pair-wise distance measurements without the construction of trees will be used to infer sequence relationships. The "Use UPGMA" and "Use NJ trees" settings determine relationships between sequences based on the positions of the sequences within trees. I would recommend that you use either the NJ tree or distance settings. Unless there are sequences in your alignment that are evolving at very different rates the distance method will give nearly identical results to the tree drawing methods and should always be tried first. Remember that the automated scan is just the first stage of the analysis and that once it is complete you will have the opportunity to scan any potential recombinants using more accurate (but slower) methods.

The *number of bootstrap replicates* that are used largely controls the significance of the recombination events that are detected using any particular percentage bootstrap cutoff (see below). It is strongly recommended that for any dataset containing more than ~20 2Kb+ sequences that the number of replicates be kept under 1000 and that the significance of results be controlled by increasing the percentage cutoff value. As a general rule 200 replicates with a 95% cutoff percentage seems to yield similar results to those obtained with the other methods when using a 0.05 p-value cutoff with multiple comparison correction on.

Using the same random number seed in two separate analyses will ensure that bootstrapped datasets remain the same for both analyses and that results are repeatable.

The cut-off percentage refers to the percentage bootstrap support that is required before any altered relationships between three sequences within an alignment are interpreted as evidence of potential recombination. Setting this value higher (it could be set as high as 100%) will increase the probability that any regions detected are recombinant. This value is only meaningful in the context of the number of bootstrap replicates selected. It should be noted that a value of 95% does not equate with a p-value cutoff of 5% (i.e. 0.05). The value (together with the number of bootstrap replicates) is simply proportional to the confidence that you have in the recombinant regions that the program detects – i.e. you could have more confidence in the recombinant nature of regions detected using 1000 replicates and a 100% cutoff percentage than regions detected with 50 replicates and a 70% cutoff percentage.

While it is possible to simply use bootstrap values as p-values during a scan (with any region exceeding the bootstrap cut-off being reported as possibly recombinant), it is strongly recommended that either the "calculate binomial p-value" or "calculate Chi Square p-value" settings be used. If either of these settings is selected a statistical test will be used to determine the probability that regions exceeding the bootstrap cut-off are recombinant. Using simulations I have found that the "calculate binomial p-value" sitting is by far the most powerful and this is the setting I strongly recommend you use.

3.4.2 Model options. Four different nucleotide substitution models may be used when calculating distance matrices from bootstrap replicated alignments. With all the models other than the Jukes Cantor, 1969 model it is possible to score transitions and transversions differently during pair-wise distance calculations. The Jukes-Cantor model is identical to the Kimura, 1980 model with a transition:transversion ratio set to 0.5. The Kimura model is in turn identical to the Felsenstein, 1984 model when equilibrium frequencies of all four bases are equal. The Felsenstein, 1984 model allows for differences in equilibrium base frequencies that may be either supplied by you or inferred from the alignment. The Jin-Nei, 1990 model is similar to the Kimura model except that it assumes that different rates of substitution occur at different sites. The Jin-Nei model determines site-specific substitution rates from a gamma distribution, the shape of which is determined by the coefficient of variation. Low values mean sites are expected to evolve at similar rates and high values mean rates are expected to vary more widely. RDP5 utilises code from the PHYLIP component DNADIST to calculate distances and additional information on this program can be obtained online from:
http://evolution.genetics.washington.edu/phylip/doc/dnadist.html

3.5 MAXCHI Settings

3.5.1 Scan options. Whereas in RDP2 it was possible to use the MAXCHI method to automatically screen an alignment either three sequences at a time or two sequences at a time, in RDP5 only triplet scans can be performed during automated recombination detection. Doublet scans are, however, still possible when using MAXCHI to manually screen sequences for evidence of recombination. The major difference between the triplet and doublet scans is that the doublet scans do not allow proper identification of parental and recombinant sequences.

As with other scanning window settings the optimal *window size* that should be selected for a MAXCHI analysis will depend on the sequences being analysed and the size of recombinant regions that must be detected. As is the case with the original RDP, CHIMAERA, GENECONV and 3SEQ methods, MAXCHI only examines variable nucleotide positions – i.e. the window size refers to the number of variable sites and not the number of nucleotide positions. The optimal window size for detecting recombinant regions with 20 variable nucleotide sites will be 40. The reason for this is that the MAXCHI scanning window is split into two with the halves being compared to one another (see section 8.4 for details on the MAXCHI method).

Because the χ^2 statistic is only calculated within individual windows a situation can arise where it is impossible to achieve a significant χ^2 p-value even with a fairly lax p-value cut-off. For example, with a window size of 20 it is impossible to achieve a p-value lower than $\sim 1X10^{-5}$. This isn't too much of a problem if the multiple comparison correction setting is set to off (a setting that is not recommended). However, with an alignment containing 20 sequences, multiple comparison correction on, a window size of 20 and a highest acceptable p-value cutoff of 0.01 it will be impossible to achieve a p-value below the cutoff (i.e. no recombination will be detected). Always remember this when selecting the window size.

Variable or set window sizes can also be used. Changing this setting to "variable" lets you specify which proportion of variable sites should be included in a window. If variable window sizes are used, windows will get larger for sequence triplets containing quite diverged sequences and smaller for triplets containing more closely related sequences. Note that if a sequence triplet has fewer variable sites than 1.5 times the specified window size, the window size will automatically be set to 0.75 times the number of variable sites. If the window size thus derived is smaller than 10, then the sequence triplet in question will not be examined.

It is always advisable to use the "strip gaps" setting for MAXCHI. If the "use gaps" setting is selected you should realise that each individual gap character ("-" or ".") will be treated as a fifth nucleotide. This may cause problems if, for example, one of the sequences in a triplet has a run of gaps in a particular region because the other two

sequences in the triplet will appear much more similar to one another in that region than they should and recombination will be inferred.

3.6 CHIMAERA Settings

3.6.1 Scan options. As with other scanning window settings the optimal *window size* that should be selected for a CHIMAERA analysis will depend on the sequences being analysed and the size of recombinant regions that must be detected. As is the case with the original RDP, GENECONV, 3SEQ and MAXCHI methods, CHIMAERA only examines variable nucleotide positions – i.e. the window size refers to the number of variable sites and not the number of nucleotide positions. The optimal window size for detecting recombinant regions with 20 variable nucleotide sites will be 40. The reason for this is that, like with the MAXCHI method, the CHIMAERA scanning window is split into two with the halves being compared to one another (see section 8.5 for details on the CHIMAERA method).

For information on setting *window sizes* refer to the previous section on appropriate window sizes for the MAXCHI method.

As with the MAXCHI method a variable window size setting may also be used with the CHIMAERA method, which allows you to specify the proportion of variable sites that should be included in a window. If variable window sizes are used, windows will get larger for sequence triplets containing quite diverged sequences and smaller for triplets containing more closely related sequences. Note that if a sequence triplet has fewer variable sites than 1.5 times the specified window size, the window size will automatically be set to 0.75 times the number of variable sites. If the window size thus derived is smaller than 10 the sequence triplet in question will not be examined.

3.7 SISCAN Settings

3.7.1 Scan options. The *window* and *step sizes* used during a SISCAN should be carefully selected based on the length of the sequences being analysed, their relatedness and the sizes of recombinant regions that are anticipated. The step size used must be smaller than the window size and should ideally be set to less than 50% of the window size. Window sizes should be selected so that, on average, there are more than ~10 variable nucleotide positions within every window examined. Whereas larger window sizes will increase signal:noise ratios, you should understand that obvious recombinant regions that are only slightly smaller than the window size may not be detected.

It is strongly recommended that the "strip gaps" setting be used. If gaps are used, each individual gap character ("-" or ".") will be treated as a fifth nucleotide.

It is also strongly recommended that the "use 1/2/3 variable positions" setting be used. This setting will focus the analysis on sites that differ between the sequences in a triplet. Whereas the "use 1/2/3/4 variable positions" setting will focus the analysis on sites that vary between the sequences in a triplet and/or the sequences in a triplet and an outlyer sequence (see 3.7.2 for information on outlyer sequences), the "use all positions" setting will examine all sites both variable and constant. The "use 1/2/3 variable positions" setting is recommended because the other settings tend to "dilute" recombination signals by including a lot of irrelevant sites in the analysis.

3.7.2 Fourth sequence selection. During a "SISCAN" sequence triplets are examined together with a fourth outlyer sequence (See section 8.6 for details of the SISCAN method). The outlyer can either be another sequence in the alignment or a randomised sequence constructed from the sequences in the triplet. With the "use nearest outlyer" setting, for every sequence triplet examined, RDP5 will scan an alignment for an outlyer sequence that most closely resembles the three sequences in the triplet. With the "use most divergent sequence" setting, RDP5 will always use the most divergent sequence in the alignment as an outlyer. The "use radomised sequence" setting will, for every window analysed in every sequence triplet, require construction of a new randomised sequence. It is recommended that the "use nearest outlyer" setting be used because this is both the quickest setting and, unlike the other settings, it yields results that are usually well supported by other recombination signal detection methods.

3.7.3 Permutation options. When determining the significance of potential recombination signals SISCAN uses a permutation test (for details of the calculation of p-values see section 8.6). Because the test can be quite time consuming RDP5 can be set to use fewer permutations during an exploratory scanning phase (the scan permutation number) and, when a possible recombination signal is detected, use more permutations to accurately determine p-values for likely recombinant regions (the p-value permutation number).

Because SISCANning involves the generation of randomised sequences (see section 8.6 for details) there is the option to provide a random number seed. Using the same random number seed in repeated analyses will ensure that SISCAN results are reproducible.

If the "do fast scan" setting is used RDP5 will only use permutation tests to analyse windows in which the pair-wise relationships between the sequences in a triplet differ relative to the relationships of the sequences over their entire lengths (these are the only windows within which a recombination signal is likely to be found). The "do exhaustive scan" setting will perform permutation tests on every window – regardless of how unlikely it is that a recombination signal will be detected in windows where sequence relationships are the same as they are over the entire length of the sequences.

3.8 LARD Settings

For additional information on LARD settings please consult the LARD manual. It can be downloaded from:
http://evolve.zoo.ox.ac.uk/software/Lard/main.html

3.8.1 Model options. LARD offers the option of using three different nucleotide substitution models for the maximum likelihood reconstruction of three sequence phylogenies. (1) The Hasegawa Kishino and Yano, 1985 (HKY) model allows different transition and transversion rates and unequal nucleotide frequencies. The Kimura, 1980 and Jukes-Cantor, 1969 models are specific cases of this model. (2) The Felsenstein, 1984 model is similar to the HKY model but allows nucleotide frequencies to be estimated from the alignment and handles transition/transversion rates differently. (3) The reversible process model allows different rates for all six different types of substitution and assumes, for example, that the frequency of T to C substitutions will be the same as the frequency of C to T substitutions.

Besides the different nucleotide substitution models, LARD also offers the option of using two different models that allow for site-specific variation in substitution rates. (1) A codon-based model allows different substitution rates at each codon position (this is obviously only applicable to coding regions). In general the last codon position should have the highest substitution rate, the middle position the lowest rate and the first position an intermediate rate; (2) A model that assigns different substitution rates to sites based on a gamma distribution. Whereas the gamma distribution is scaled so that the average rate is equal to 1, it is possible to specify the shape of the distribution using the "gamma shape for site rate heterogeneity" setting. A low value (<1) will mean that sites vary greatly in their evolution rate whereas higher numbers for this setting will specify that sites evolve at more similar rates. Setting "# categs for gamma rate heterogeneity" to 0 will give all sites the same substitution rate. Setting this number to a positive integer (N) will assign each site with a different probability to each of the N substitution rate categories specified

3.8.2 Scan options. LARD examines three aligned sequences at a time. It can be set to scan sequences in three different ways. The first and quickest way involves moving a partition along the alignment and determining the likelihood that trees constructed from sequences on either side of the partition have the same branch lengths (the "test one breakpoint" setting; for a detailed description of what LARD does see section 8.7). The second way is to move a window along the alignment with a partition in the centre (this is similar to that used for the MAXCHI and CHIMAERA methods; the "sliding windows scan"). The third, and by far the slowest, way to scan the alignment is to search for two optimal breakpoint partitions (the "test two breakpoints" setting). This could involve evaluating every possible pair of partitions of the alignment.

The "*step size*" setting will specify how many nucleotides along the alignment the partition(s)/window will move at each step of the analysis. While setting the step size to 1 will ensure the highest possible scan resolution, the scan will most likely be quite slow. Increasing the step size will speed up the analysis but decreases the scan resolution. A step size of 10 nucleotides should be a good compromise.

If a sliding window scan is chosen, you can specify the window size that is used – remember though that the window has a partition in the centre so that a window size of 400 indicates that the 200 nucleotides on the left of the window get compared with the 200 on the right. The LARD method examines both conserved and variable alignment positions and the window size setting should be large

enough that every window examined contains at least 20 variable nucleotide sites

3.9 PHYLPRO Settings

For additional information on PHYLPRO settings and how PHYLPRO works please consult either section 8.8 or the PHYLPRO manual. It can be downloaded from:
http://www.rsbs.anu.edu.au/ResearchGroups/GIG/Products/phylpro/

3.9.1 Scan options. PHYLPRO is another recombination detection method (like the LARD, BOOTSCAN and SISCAN methods) that examines both variable and conserved alignment positions. The *window size* setting should be large enough that all examined windows contain 20 or more variable alignment columns. Like with the LARD method this number is twice that recommended for the BOOTSCAN and SISCAN methods because the PHYLPRO method involves moving a window with a partition in its centre along the length of an alignment with each half of the window being compared to the other. See section 3.4.1 of this manual for advice on selecting window sizes.

During pair-wise distance calculations (see section 8.8) the PHYLPRO method can be set to handle gaps in two different ways: Alignment positions with any gap characters can be either completely ignored (the "strip gaps" setting) or these positions can be considered as long as both of the sequences compared have a nucleotide in the relevant position (the "ignore gaps setting).

When calculating correlation coefficients for sets of pair-wise distances on either side of the moving window (see section 8.8) the PHYLPRO method can be set to either use or not use the zero distance values obtained when sequences are compared with themselves. The permutation test is not currently implemented and the permutation options will have no influence on the analysis results.

3.10 DNA Distance Plot Settings

3.10.1 Scan options. The *window* and *step sizes* used during the construction of distance plots may be set. You should set window sizes based on the relatedness of parents that are being examined. Ideally each window in the scan should contain at least 5 variable positions. The optimal step size is also dependent on the relatedness of the sequences being examined and should be smaller than 20% of the window size.

3.10.2 Model options. RDP5 uses code from the PHYLIP component, DNADIST, to construct distance plots and the model options on offer are those available in that program. For additional information on the DNA distance models used by DNADIST please consult the online manual at:
http://evolution.genetics.washington.edu/phylip/doc/dnadist.html
Consult section 3.4.3 of this manual for a brief description of the model options.

3.11 TOPAL Settings

For additional information on TOPAL settings please consult the TOPAL manual. It can be obtained online from:
http://www.bioss.sari.ac.uk/~frank/Genetics/manual.html

3.11.1 Scan Options. As with the PHYLPRO, BOOTSCAN and SISCAN methods (see sections 3.9, 3.4 and 3.7 respectively) the optimal *window* and *step* sizes used during a TOPAL scan are dependent on the relatedness of the sequences being examined. Note, however, that the TOPAL method is similar to the PHYLPRO method in that the windows examined are split in two and have an optimal size that is twice that of the BOOTSCAN and SISCAN methods. You should attempt to set the window size so that each window will cover more than ~20 variable nucleotide positions. See section 3.4.1 of this manual for advice on selecting window sizes.

When drawing a difference in sum of squares (DSS) plot you can opt to smooth it by averaging DSS values over a "smoothing window" that is moved across the plot one DSS value at a time.

3.11.2 Tree options. During a TOPAL scan RDP5 uses the PHYLIP components NEIGHBOR and FITCH to calculate neighbour joining (NJ) and least squares (LS) trees, respectively. Although the "calculate NJ and LS trees" setting is substantially faster than the "use only LS trees" setting, according to the people who developed the method, it should only be used during manual TOPAL analyses of >20 sequences. I'm not sure if I agree with this though as both settings seem pretty similar in practice – except of course that the one is much quicker than the other.

The "Power" setting will influence the magnitude of the DSS values that are calculated – if DSS values are very small (e.g. 0.002) increasing the Power setting will increase them to values that are easier to compare.

A random number seed used during generation of simulated sequences, and randomising the input order of sequences in FITCH and NEIGHBOR can be provided. Using the same seed will result in identical DSS plots in repeated analyses.

3.11.3 Parametric bootstrapping options. If the *number of permutations* is set to a number greater than 10, RDP5 will perform a permutation test called a parametric bootstrap to determine the significance of any detected DSS peaks. The parametric bootstrap alignments are simulated using SEQ-GEN (Rambaut and Greaaly, 1997) and the DSS plots generated from these alignments are presented together with plots from the real data for comparison purposes.

3.11.4 Model options. RDP5 uses the PHYLIP component DNADIST to construct distance matrices and the model options on offer are those available in that program. For additional information on the DNA distance models used by DNADIST please consult the online manual at: http://evolution.genetics.washington.edu/phylip/doc/dnadist.html
Consult section 3.4.3 of this manual for a brief description of the model options.

3.12 VisRD Setting

VisRD, like the PHYLPRO, LARD, BOOTSCAN and SISCAN methods), is a recombination analysis method that examines both variable and conserved alignment positions. The scanning *window size* is the only setting that can be changed and should be made large enough that all examined windows contain 10 or more variable alignment columns. See section 3.4.1 of this manual for advice on selecting window sizes.

3.13 Breakpoint Distribution Plot Settings

Breakpoint distribution plots are a useful way of analysing alignments for evidence of recombination hot and cold spots (see section 9.1; Heath *et al*., 2006). The test used to detect breakpoint hot and cold spots is based on permutations. The number of *permutations* used in this test can be specified. The number should be 100 or greater. The size of breakpoint clusters that you wish to examine can be specified with the "*window size*" setting. Note that small window sizes (<=50nts) are useful for detecting unusually tight clusters of breakpoints (i.e. highly focused recombination hotspots) but are not very good for detecting either recombination cold spots or dispersed recombination-hotspots. Window sizes between 100 and 200 nt are generally a good compromise between detecting hot and cold spots but might miss evidence of unusually tight clusters of breakpoints within regions smaller than the specified window size. It is therefore advisable to try a range of window size settings

3.14 Recombination Rate Settings

RDP5 uses the programs CONVERT and INTERVAL from the LDHAT package (McVean *et al*., 2002; McVean *et al*., 2004) to construct plots of varying recombination rates across sequences. For additional information on the settings used by these programs consult the LDHAT manual at:
http://www.stats.ox.ac.uk/~mcvean/LDhat/instructions.html

The INTERVAL program that RDP5 uses to draw recombination rate plots, estimates variations in recombination rates along an alignment using a penalised approximate likelihood approach within a Bayesian reversible-jump Markov chain Monte Carlo (RJMCMC) scheme. INTERVAL requires an initial estimate of the alignment-wide population scaled recombination rate (rho) as a starting point. The "*starting rho*" value should be a number between 0 and 100 that should ideally be an actual estimate of the alignment-wide population scaled recombination rate. An estimate of this can be obtained by firstly drawing a plot with an arbitrary starting rho value (say 10) which, apart from giving you a plot of recombination rates along your alignment, will also give you an estimate of the alignment-wide population scaled recombination rate. This value, displayed in the

recombination information panel, can then be used as a better starting value when you redo the plot.

INTERVAL allows you to specify a "*block penalty*" to prevent the RJMCMC invoking the existence of too many changes in recombination rate across a region of sequence – i.e. you can set the block penalty to prevent INTERVAL from over-fitting a complex variable recombination rate model to the data. I cannot give any really good advice on what constitutes an appropriate penalty other than that you should try constructing plots with a range of penalties between 0 and 50. Lower penalties will enable the analysis to detect smaller, more subtle variations in recombination rates but could also result in over-fitting of the inferred changes to the data. Conversely, higher block penalties will sacrifice sensitivity in return for greater confidence in the recombination rate changes that are detected. Gill McVean advises the use of simulations with sequences resembling those you are analysing to determine the most appropriate block penalty. As this approach will probably be beyond most RDP5 users, I'd recommend that you settle on a penalty somewhere in the range 5-30 and don't over-interpret the peaks and valleys in the plots that you get.

The "*minor allele frequency cutoff*" setting determines which variable alignment columns INTERVAL will examine. Having a cutoff that excludes rare polymorphisms focuses the analysis on the most reliable and least noisy evidence of recombination – i.e. that which have left a mark on the distributions of the older, most phylogenetically informative nucleotide polymorphisms in the dataset. It is strongly advisable that a cutoff is chosen which excludes alignment columns that contain a single sequence with a site that is different from all the rest in the alignment. I recommend that the cut-off is chosen so that only sites carried by three or more sequences are included in the analysis. The value of this setting will therefore need to be changed with every analysis you do. For example, with an alignment containing 100 sequences, a minor allele frequency cut-off of 0.05 will exclude all variable alignment positions where fewer than six sequences share one of the two alternative nucleotides at that position.

The "gap frequency cutoff" can be used to exclude from an analysis any alignment columns with more than a certain amount of missing data.

The number of MCMC updates performed during the analysis can be set. The first 10% of updates will always be discarded as burn-in and the number of updates must always be greater than 10^5. It is strongly recommended that you never use less than 10^6 updates.

3.15 Matrix Settings

RDP5 can make several different types of matrix plots. Many of the different matrix plots share settings such as their colour scales, permutation numbers and window sizes. Although it is not a matrix, various matrix settings (window size, permutation number and type species) are shared with the **recombination breakpoint plot** (see section 3.13).

Note that the **Rmin(HK)**, **Rmin(HK)/D** and **LD** matrices that RDP5 presents are constructed by the program PAIRWISE (a component of the LDHAT package) using minor allele frequency, gap frequency, gene conversion model and average tract length settings that are specified in the recombination rate options section (see 3.14). See sections 9.3.6 – 9.3.8 for what is being plotted in these matrices.

3.15.1 Ingrid Jakobsen (IJ) compatibility matrix. The IJ compatibility matrix in RDP5 is only a partial implementation of that implemented in the program Reticulate (Jakobsen and Easteal, 1996; Jakobsen et al., 1997) in that a statistical test using Ingrid Jacobsen's neighbour similarity scores is not available in RDP5 (It is, however, implemented in RDP2 which is available from the RDP web-page). See section 9.3.1 for details of what is being plotted in an IJ compatibility matrix. For additional information on compatibility matrices and the program reticulate please consult the manual: http://jcsmr.anu.edu.au/dmm/humgen/ingrid/ftp/reticulate/instructions

3.15.2 Tevor Bruen (TB) compatibility matrix. The TB compatibility matrix is similar to a non-binary version of a IJ compatibility matrix in that whereas an IJ compatibility matrix is comprised exclusively of white and black cells respectively representing compatible and incompatible site pairs, the TB compatibility matrix is comprised of cells representing varying degrees of phylogenetic compatibility (indicated an incompatibility score). See section 9.3.2 for details of what is being plotted in a TB compatibility matrix. For additional information on TB compatibility matrices and the program PhiPack which Trevor Bruen has made to construct these, please consult the PhiPack manual: https://www.maths.otago.ac.nz/~dbryant/software/phimanual.pdf

3.15.3 Robinson-Foulds (RF) compatibility matrix. The "*window size*" setting refers to the number of nucleotides that are used to construct the various phylogenetic trees that are to be compared with one another and the "*step size*" refers to the number of nucleotides that are skipped between consecutive windows. As with the SH compatibility matrix, if the step size is set to larger than half the window size, the window size will be automatically adjusted so that it is twice the step size. While decreasing the step size will increase the resolution of RF matrices, it will also exponentially increase the amount of time it takes to construct the matrix (i.e. it can take a very long time to construct SH matrices if the step size is small). If the step size to smaller than 1/2000 the length of the analysed sequences it will be increased so that it is 1/2000 the length of the analysed sequences. See section 9.3.2 for details of what is being plotted in a RF compatibility matrix.

3.15.4 Shimodaira-Hasegawa (SH) compatibility matrix. The "*window size*" setting refers to the number of nucleotides that are used to construct the various phylogenetic trees that are to be compared with one another and the "*step size*" refers to the number of nucleotides that are skipped between consecutive windows. As with the RF compatibility matrix, if the step size is set to larger than half the window size, the window size will be automatically adjusted so that it is twice the step size. While decreasing the step size will increase the resolution of SH matrices, it will also exponentially increase the amount of time it takes to construct the matrix (i.e. it can take a very long time to construct SH matrices if the step size is small). If the step size to smaller than 1/2000 the length of the analysed sequences it will be increased so that it is 1/2000 the length of the analysed sequences. See section 9.3.3 for details of what is being plotted in a RF compatibility matrix.

3.15.5 Recombination matrix. The "*type sequence*" setting can be used to specify the sequence in an alignment that will be used as a reference when numbering the nucleotide coordinates that are plotted. See section 9.3.4 for details of what is being plotted in a recombination matrix.

3.15.6 Modularity matrix. See 3.15.4 for what the "*type sequence*" setting means The "*window size*" setting refers to the number of nucleotides that are examined when comparing how closely the parental sequences of detected recombinants resemble one another. See section 9.3.5 for details of what is being plotted in a modularity matrix.

3.15.7 Recombination region count matrix. See 3.15.4 for what the "*type sequence*" setting means. The "*window size*" setting here refers to the diameter of the circle drawn around every recombination breakpoint pair plotted on a breakpoint pair matrix. See section 9.3.6 for details of what is being plotted on a recombination region count matrix.

3.15.8 Breakpoint distribution plot. See 3.15.4 for what the "*type sequence*" setting means. See 3.13 for what the other settings mean and section 9.3.7 for details on what is plotted.

3.16 Tree Settings

RDP5 can construct UPGMA, neighbor joining (NJ), Fast neighbour joining (FatNJ or approximate least squares; LS), maximum likelihood (ML) or Bayesian trees. To set tree options for a specific tree construction method you must first select the type of tree you'd like to set options for. Note, however, that there are no uder-defineable settings for how RDP5 makes UPGMA and FastNJ trees.

3.16.1 Neighbor joining trees

3.16.1.1 Tree drawing options. RDP5 utilises the PHYLIP component NEIGHBOR to construct NJ trees and additional information on this program and its settings can be obtained online from: http://evolution.genetics.washington.edu/phylip/doc/

It is possible to specify whether or not negative branch lengths are to be permitted in the finished tree. Negative branch lengths are possible when constructing trees with the NJ method. By not allowing negative branch lengths you will force RDP5 to report negative branch lengths as having zero length.

Randomising (or jumbling) the order in which sequences are added to NJ trees will influence the way NEIGHBOR produces the final tree (if ties are obtained in any of the iterative rounds of branch addition the first sequence in the order will win the tie with possible consequences for the topology of the finished tree). To test the influence of sequence input order on the topology of a tree, use the

"randomise input order" setting, set the bootstrap number to 0 and then construct trees with a range of different random number seeds. If the tree topology changes with different random number seeds then the input order has had an influence on the tree's topology.

3.16.1.2 Model options. RDP5 uses the PHYLIP component DNADIST to calculate distance matrices for NJ tree construction. The model options on offer are those available in DNADIST. For additional information on the DNA distance models used by DNADIST please consult the online manual at:
http://evolution.genetics.washington.edu/phylip/doc/dnadist.html
Consult section 3.4.3 of this manual for a brief description of the model options.

3.16.1.3 Branch support tests. The number of *bootstrap replicates* used during the construction of NJ trees can be set. A random number seed used during generation of bootstrapped alignments can be provided. Using the same seed will result in identical bootstrapped alignments in repeated analyses.

3.16.3 Maximum likelihood trees

3.16.3.1 Model options. RDP5 can use the programs PHYML (versions 1 and 3; Guindon and Gascuel, 2003; Guindon *et al.*, 2010), RAxML (version 8; Stamatakis, 2014) and FastTree (version 2; Price *et al.*, 2010) to construct maximum likelihood (ML) trees. Model options can, however, only be set for PHYML. For additional information on the models that are applied by these programs please consult their online manuals at:
http://bioweb2.pasteur.fr/docs/modules/phyml/3.0.1/phyml_manual_2008.pdf; (PhymL) http://sco.h-its.org/exelixis/resource/download/NewManual.pdf (RaxML); http://meta.microbesonline.org/fasttree/ (FastTree).

Eight different *nucleotide substitution models* are available for PHYML. These include the Jukes Cantor-1969 (JC69), Kimura-1980 (K80), Felsenstein-1981 (F81), Felsenstein-1984(F84), Tamura and Nei- 1993 (TN93), General time reversible (GTR; Lanave *et al.* 1984, Tavaré 1986, Rodriguez et al. 1990) and Hasagawa, Kishino and Yano -1985 (HKY85) models. While PHYML allows users to specify their own GTR rate matrix this option is not implemented in RDP5. RDP5 will also automatically select a best fit model using an Aikaike information criterion (AIC) test such as that described in Posada and Crandall (1998). This test compares the likelihoods of trees constructed with various standard nucleotide substitution models (including or excluding extra parameters permitting site-to-site variations in substitution rates) and, accounting for the number of parameters the different models contain, selects the model that fits the data best.

Depending on the model selected you may be able to specify the transition:transversion rate ratio (note that to keep things consistent with PHYLIP components used elsewhere this is the "rate ratio" and not the "ratio" normally used in PHYML – the number that will be passed to PHYML for phylogeny construction will be twice the number specified here). If a value of 0 is specified PHYML will determine the maximum likelihood value of this parameter during tree construction (doing this will make tree construction slower).

The proportion of invariable sites can be set to any number between 0 and 1 inclusive. Setting this value to 1 will prompt PHYML to find the maximum likelihood value of this parameter during tree construction.

Depending on the model selected, equilibrium base frequencies may be estimated either empirically from the data, or by maximum likelihood during tree reconstruction (with the later making tree construction slower).

PHYML allows specification of multiple substitution rate categories – i.e. it can take into account that different sites along an alignment may evolve at different rates. The value of each substitution rate category is drawn from a discrete gamma distribution of possible categories. The greater the number of categories specified, the more accurate will be the fit of actual substitution rates to the rate categories chosen. However, the program should take four times longer to construct a tree using four rate categories than it will take to construct a tree using one. Whereas allowing fewer than four rate categories can be unrealistic, allowing more than eight does not really improve the accuracy of tree construction but seriously slows the tree construction process down.

If trees are to be constructed using more than one substitution rate category, the exact shape of the gamma distribution from which the categories are drawn can be changed using the gamma distribution parameter. Values of this parameter below 0.7 correspond with high variations between the evolution rates of sites in the sequences being examined. Values between 0.7 and 1.5 correspond with moderate variation and values larger than 1.5 correspond with low variation. If a value of 0 is specified the shape parameter will be inferred by maximum likelihood during tree construction (again, this will increase the tree construction time).

3.16.3.2 Branch support tests. For small datasets PHYML is fast enough to perform standard bootstrap tests of branch support. The number of bootstrap replicates used during the construction of ML trees can be set. Unlike with the NJ and LS trees, however, the random number seed will automatically change for each tree constructed.

3.16.3.3 Tree search strategy. Various different compute-program dependent strategies can be used to search for the ML tree. In order of fastest to slowest these are: fastest FastTree (the default), faster RAxML, fast PHYML1 tree search, PHYML3 tree search with NNI, PHYML3 tree search with SPR, and PHYML3 tree search with NNI +SPR. The relative accuracies of these different tree searching methods is disputed. FastTree seems to excellently balance speed and relatively high accuracy, but over-all RAxML or PHYML3 may be slightly more accurate. RaXML is, however, definitely more accurate than both PHYML and FastTree when it comes to analysing alignments with large amounts of missing data.

3.16.4 Bayesian trees

RDP5 uses the program MrBayes 3.2 (Ronquist *et al*, 2012) to make Bayesian trees. The options on offer in RDP5 are only a very small subset of those available in MrBayes. For additional information on these options please consult the MrBayes online manual at:
http://mrbayes.csit.fsu.edu/wiki/index.php/Manual

3.16.4.1 Model options. Three different nucleotide substitution models are available. You will notice that the model names do not correspond to those of any of the other three drawing methods in RDP5. However, MrBayes run with the "all 6 substitution types are equally likely" and "no rate variation across sites" settings corresponds with the Jukes cantor, 1969 model. Similarly, MrBayes run with the "all 6 substitution types are equally likely" and "gamma distributed rate variation" corresponds with the Felsenstein, 1981 model. You should be able to find a suitable mixture of the three model settings to recreate most of the common nucleotide substitution models. The "transitions and transversions can be unequally likely" setting will result in the Transition:transversion rate ratio being approximated along with the phylogeny. The "all six substitution types can be unequally likely" setting can be used to approximate the GTR model with Bayesian probabilities of the six different substitution types being inferred during tree construction.

You may also specify whether trees are to be inferred assuming gamma distributed rate variation across sites. Only three of the five types of rate variation (including no variation) on offer in MrBayes are offered in RDP5 (the options with invariable sites are not included). See section 3.16.3.1 for details on what gamma distributed rate variation means. The "auto-correlated" rate distribution setting will allow you to specify that the rates of adjacent sites are not chosen independently of one another. Although tree construction with the auto-correlated gamma distribution setting is always slower than that with the plain gamma distribution setting, the difference in construction times decreases with increasing dataset size.

See section 3.16.3.1 for advice on selecting the number of rate categories that are to be used during tree construction.

3.16.4.2 MCMC options. Use the "*number of generations*" setting to indicate the maximum number of MCMC generations that should occur during tree construction. RDP5 is incapable of providing you access to an interactive use of MrBayes which means that you will not have the MrBayes option of simply continuing with the tree construction process until enough convergence is reached. Therefore, RDP5 uses the "average standard deviation of split frequencies" convergence diagnostic to tell MrBayes when it should stop trying to find better trees. It will stop MrBayes when the average standard deviation of split frequencies is smaller than or equal to 0.1. If this degree of convergence is never reached then the trees should either be examined keeping this in mind, or another run with more generations should be started from scratch. Note that with MrBayes you could simply continue a run which means it will sometimes be a better idea for you to simply construct these trees using MrBayes directly.

Anyway, the number of MCMC generations should probably never be set below 10^6. If convergence doesn't happen in this number of generations, the generation number could be set as high as 10^{10}. Remember that the "stop rule" is in place so that as soon as the stop condition is reached (even if it is reached after only 10^5 generations) the run will terminate and your tree will be displayed.

The *sampling frequency* setting should be used to specify how many generations should pass between samples drawn from the Markov chain. The number should never be less than 10 or greater than 100th of the expected MCMC generations before convergence. 100 is a safe number to choose for this setting.

If the *number of chains* is set higher than 1 MrBayes will run multiple MCMC chains in parallel which it uses for something called "Metropolis coupling" to improve its sampling of potentially good trees. It will always run one "cold" chain and any extra chains specified will be "heated". Running heated chains in parallel to the cold chain may be absolutely essential to achieve a good tree for alignments containing more than ~50 sequences. Basically, the more chains you specify the better will be your chances of obtaining a good tree. However, the time taken for the program to create and examine a specified number of MCMC generations will increase in proportion with the number of chains specified. Also, if your computer does not have enough RAM for MrBayes to store all the chains you ask it to analyse, the program can start running really slowly.

Another parameter influencing the Metropolis coupling behaviour of MrBayes is the *temperature*. The temperature parameter controls the rate at which the heated chains get hotter. The whole rationale behind heating of the chains is to reduce the penalisation of potential trees that are relatively less probable than the best trees sampled at any given point in the program's execution. These less probable trees might more closely resemble, and therefore provide access to, some really good trees that the MCMC sampler would never otherwise find without the heating process. Low temperature values will heat the heated chains more slowly than high values. I'm not sure how high the temperature setting might be set without there being a complete collapse in the sampling scheme but the default value in Mr Bayes is 0.2 (corresponding with a 20% increase in temperature) at every heating step.

The *swap frequency* and *swap number* determine the rate at which states (from hot to cold and *vice versa*) are swapped between the chains being analysed. The swap frequency setting specifies the number of generations that pass between attempted exchanges of states between a randomly picked hot chain and the cold chain. The swap number determines how many swaps are attempted between different hot chains and the cold chain at every "swapping generation"

3.17 SCHEMA Settings

SCHEMA (see section 9.4; Voigt *et al*., 2002; Lefeuvre *et al*., 2007) is a method that takes protein atomic coordinates and estimates degrees of protein or single stranded nucleic acid fold disruption expected in recombinant proteins or single stranded DNA/RNA molecules. RDP5 uses a permutation test to determine whether natural recombinants are significantly less disruptive of protein/nucleic acid folding than randomly generated recombinants. The number of permutations used in this test can be specified with the *permutation number* setting

3.17.1 Protein folding disruption. The SCHEMA method finds all amino acid pairs that are within a user defined distance of one another (which is usually between 2 and 20 angstroms) and identifies these as being potentially interacting within the folded protein. This distance can be defined with the *interaction distance* setting.

3.17.2 Nucleic acid folding disruption. RDP5 uses the program hybrid-ss-min from the UNAFOLD package (Markham and Zuker, 2008) to infer the secondary structures of DNA and RNA molecules. The *temperature* at which this inference is carried out at is important and should be set to the approximate physiological temperature at which the DNA/RNA being analysed occurs (e.g. 37°C for human viruses and 20°C for plant viruses). For accurate secondary structure inference it is also necessary to indicate whether the *sequences being analysed are RNA or DNA*.

4 FINDING EVIDENCE OF RECOMBINATION

4.1 Automated Exploratory Recombination Analysis

4.1.1 Masking and disabling sequences. When large numbers of sequences are to be analysed, certain sequences in an alignment can be either "masked" or completely removed from the analysis ("disabled") by clicking (with the left mouse button) on the name of the sequence either in the sequence display panel or in the small tree display panel(Fig 1). Masking does not stop the sequence being used in either tree construction, BOOTSCANning or as a reference sequence in determining informative sites (for the original RDP method, SISCAN or VisRD). Masking of sequences is useful for both focusing the analysis on groups of sequences within an alignment and, because fewer sequence comparisons are made when some sequences are masked, increasing the power of recombination detection amongst a smaller subset of sequences within an alignment. Disabling sequences is useful for temporarily discarding sequences from an alignment.

RDP5 will, by default, automatically mask sequences to ensure optimal recombination detection. Auto masking will minimise the number of comparisons the program makes during an exploratory recombination screen. This will ensure that the multiple testing correction needed for p-values will be kept to a minimum and will therefore guarantee that at least as many (but probably more) recombination events will be detected as would have been detected if no sequences were masked.

4.1.2 Grouping sequences. Grouping of sequences provides and additional means of focusing analyses onto a specific group of sequences. To make a group right click on the sequence names in the sequence display panel or the small tree display panel and select the "group" option that is offered. Then simply click on the names of the sequences (in either the sequence display or small tree display) that you wish to form part of the group. When a group of sequences is selected and an automated exploratory scan for recombination is subsequently carried out, the only sequence triplets that will be examined will be those for which two or more of the sequences are within the selected group. As with masking, this minimises the numbers of tests that are performed and increases the program's power to detect recombination events within the specified group of sequences.

4.1.3 Running an automated exploratory analysis. Once the appropriate settings have been selected, pressing the "Run" button in the command button panel (Fig 1) starts the analysis. A progress bar, the time taken, the number of unique events and the number of recombination signals detected are displayed for each of the different methods selected for the primary exploratory scan for recombination. It is recommended that the "Do not show plots" or "show overview during scan" option be selected in the "General Options" (see section 3.1). If the "show plots" setting is selected the program will display plots of raw data which could more than double the analysis time.

If the "show overview during scan" setting is selected the program will display plots during a scan indicating in real time the positions in the alignment where recombination is being detected. Displayed during the primary scanning phase of the analysis are plots indicating the genetic distances between parental sequences involved in generating the detected recombination signals (PDist), the minimum probability values associated with detected events (P-Val), and the number of events detected in particular regions of the alignment (#Hits). Different versions of these plots are displayed during the secondary phase of the analysis during which all the detected recombination signals are reconciled to determine the actual numbers of recombination events that yielded the signals. During this secondary phase only information on actual "unique" recombination events (or rather RDP5's interpretation of what these are) is displayed in these plots.

4.1.4 Identification of unique recombination events. RDP5 will sequentially scan the input alignment with each of the recombination detection methods that are selected as "primary scan" methods (see section 3.14). The number of detected recombination signals will be displayed as the primary scanning phase progresses. The recombination detection methods implemented in RDP5 examine every possible triplet of sequences within an alignment for patterns of nucleotide variation indicative of recombination. Once identified, the characteristics of each "recombination signal" (sequences in the triplet, the approximate breakpoint positions, approximate probability of recombination and the method used to detect the recombination event) is stored until every recombination signal in every sequence triplet has been identified. It is important to note that not every recombination signal is indicative of a single unique recombination event. A recombination event between two nucleotide sequences produces a recombinant molecule that has two pieces each of which is most closely related to one or the other of the two recombining sequences (also called the parental sequences). It is important to note that these "parental" sequences are not the actual parents of the recombinant

sequence – they are instead sequences within the analysed dataset that were used to infer the existence of the actual parents).

When detecting recombination amongst a sample of aligned sequences, the recombination signal detection methods in RDP5 will be able to detect a recombination event if:

1. One or more descendents of the recombinant have been sampled.
2. One or more reasonably close relatives of at least one of the parental sequences have been sampled

Once a preliminary account has been made of all the recombination signals detected by all the selected primary recombination signal detection methods, RDP5 will begin trying to determine how many unique recombination events are responsible for the recombination signals detected. If more than one descendent of a recombinant is sampled, or more than one close relative of either of the parental sequences has been sampled, then the recombination event will be detectable with more than just one combination of three sequences within the total sequence dataset being analysed. These multiple detections of the same event must be taken into consideration when RDP5 attempts to identify the set of unique recombination events responsible for the recombination signals in the alignment.

RDP5 handles multiple detections of the same events using repeated cycles of recombination signal detection. All detectable recombination signals in an alignment are identified, the strongest signal is chosen and a piece of sequence between the detected recombination breakpoints – i.e. the piece of sequence that is responsible for the recombination signal – is removed. See section 4.14 to find out how RDP5 identifies the sequence in a triplet that is the recombinant. The alignment is then re-analysed and the process repeated until there are no longer any recombination signals detectable.

During this second phase of an exploratory recombination analysis a second set of graphs may be displayed (if the "show overview" setting is selected). These graphs indicate the same stats as those displayed during the first phase except that (1) the PDist plot is replaced by a plot of recombination breakpoint numbers (BPNum) and (2) the data plotted is only that from unique recombination events (previously the data plotted was a composite of all detected recombination signals).

The procedure used for detecting unique recombination events can become a little complicated when there are multiple descendants of a single recombinant in a sample of analysed sequences. It is important not to count each of the descendents as though they possess a unique recombination event. Therefore, when a recombination signal is detected, RDP5 uses a mixture of statistical and phylogenetic methods to identify multiple descendents of ancient recombinants. Note that whenever a sequence is referred to as the "presumed recombinant" in the following sections it does not mean it is the sequence that will ultimately be identified as the recombinant. In fact all three sequences used to detect the recombination signal are in turn analysed as if they are the recombinant and the other two sequences are parental. These various methods involve:

1. Making six "sub-alignments" of the alignment being analysed. Two sub-alignments are taken from the regions 3' and 5' of each identified recombination breakpoint (i.e. four alignments in total) with the length of each sub-alignment corresponding to 20 variable nucleotide positions between the presumed recombinant and either of its presumed parental sequences. If there is only one breakpoint in a linear sequence the sequences are treated as if they are circular and the join between the two ends are treated as a second breakpoint. The final two sub-alignments are the bits of sequence bounded by the recombination breakpoints. Again, if there is only one breakpoint in a linear sequence then the sequence is treated as circular and the region 5' of the 5' breakpoint is "ligated" to the region 3' of the 3' breakpoint.
2. A Jukes Cantor distance matrix and a bootstrapped neighbor joining tree (which branches being collapsed if they have <50% support) is constructed for each of the six sub-alignments. The six distance matrices and six trees are divided into three pairs – one for the sub-alignments bounding the 3' breakpoint, one for the sub-alignments bounding the 5' breakpoint and one for sub-alignments obtained by partitioning the entire alignment into two pieces.
3. A "presumed recombinant" is selected from the three sequences used to detect the current event.
4. The trees are used to identify sequences that are "phylogenetically correlated" with the presumed recombinant – i.e. sequences that tend to move around in trees with the presumed recombinant. A set of sequences are identified that "move" with the presumed recombinant relative to the parental sequences between the trees. All of the sequences thus identified are included in a ***phylogenetic correlation set***. Due to the lack of either a known statistical test for tree robustness, or multiple testing correction, the statistical meaning of grouping sequences into such sets is obscure. However, due to the multiple testing carried out, the groupings are expected to be reasonably unconservative and although a large number of false positives are expected, the number of false negatives will be correspondingly low.
5. Each sequence in the alignment is then compared with the presumed recombinant by correlating distances between each sequence and the parentals with those of the presumed recombinant and the parentals in the paired matrices – ie the distance between sequence X and parental 1 in matrix 1 is regressed against that of the presumed recombinant and parental 1 in matrix 1. Altogether the regression analysis of each sequence using each matrix pair involves the correlation of six distance measures (those of the selected sequence/presumed recombinant against both the parentals in both matrices and the distances between the parentals in both matrices). Significant correlation (Pearson's correlation using a t-test and $P < 0.05$ cutoff) between the distances of a selected sequence to the parental sequences with those of a presumed recombinant sequence to the same parental sequences using any of the three matrix pairs, is used to identify the sequences that have potentially descended from the same ancestral recombinant as the presumed recombinant. This mechanism of grouping sequences into a ***distance correlation set*** is also extremely unconservative because p-values are not Bonferroni corrected and again one would therefore expect a large number of false positives and few if any false negatives.
6. The total pool of identified recombination signals in the entire alignment is then scanned for potential matches to the current recombination event under consideration. Potential matches are recombination signals (a) that were detected with two of the sequences in the triplet used to detect the event under consideration, and (b) where the amount of sequence bounded by the approximated recombination breakpoints overlaps that bounded by the breakpoints estimated for the current event by greater than 30%. Sequences identified in this way are placed into a ***detectable signal set***.
7. Sequences occurring in at least two of the phylogenetic correlation, distance correlation and detectable signal sets are presumed to have descended from the same original recombinant sequence as the presumed recombinant currently under consideration. These sequences are grouped into a ***co-recombinant*** set.
8. Another, different, presumed recombinant is selected from the three sequences used to detect the current event and the process from (4) through (8) is repeated until all three sequences have been considered as the presumed recombinant.

For every detectable recombination event this process conservatively identifies the sequences potentially carrying trace evidence of the same original recombination event.

4.1.5 Identification of recombinant sequences. Identification of the recombinant sequence in a sequence triplet used to detect a recombination signal is achieved using the consensus of various statistical and phylogenetic methods. These include:

1. **PhPr**: The phylogenetic profile or PHYLPRO method of Weiller (1998). Pair-wise Jukes Cantor distances between a query sequence and all the other sequences sampled are calculated using two portions of the multiple sequence alignment bounded by the approximate recombination breakpoints. Pearson's correlation coefficient (R) is then determined for these two lists of pairwise distances. The recombinant sequence is likely to be the sequence with the lowest R score of all three sequences in the triplet. However, it is possible that if a substantial proportion of the sequences in a sample are descended from the same recombinant, correlation of distances between the recombinant and the other sequences in the alignment (many of which share the same recombinant sequence mosaic as the recombinant) will be high and the PHYLPRO method may fail to identify the correct recombinant.
2. **TreePhPr**: A variation of the PHYLPRO method in which branch lengths sparating sequences within neighbor joining trees (constructed from the same distance matrices used for the PHYLPRO method) rather than genetic distances are used.

3. **SubPhPr & TreeSubPhPr**: Other variations of the PHYLPRO method in which the sum of squares of differences in the distances between sequences in a triplet and the remainder of sequences in the two alignments is calculated. The difference in distances between the recombinant and the remainder of sequences in the alignment is expected to be greater than that of the parental sequences. The first variant (SubPhPr) uses genetic distances (as with PhPr) and the second (TreeSubPhPr) uses tree branch length distances (as with TreePhPr).
4. **SubDist & TreeSubDist**: Yet more variations of the PHYLPRO method in which the average phylogenetic correlation between the two alignments is measured when each sequence in the triplet is in turn removed from the alignment. It is expected that removal of the recombinant sequence will result in the greatest increase in average phylogenetic correlation between the alignments. Whereas SubDist uses distance matrices (as with PhPr), TreeSubDist uses tree branch length distances (as with TreePhPr).
5. **ParsimonyO & ParsiomonyI:** Modifications of the subtree pruning and re-grafting (SPR) methods of McLeod *et al*. (2005) and Beiko and Hamilton (2006). These methods involve using neighbor joining trees constructed from portions of the alignment bounded by the recombination breakpoints (as opposed to trees constructed using different genes as in McLeod *et al*., 2005 and Beiko and Hamilton, 2006), and determining the minimum number of SPR operations required to convert one tree into the other. Other modifications of the McLeod et al./Beiko and Hamilton method are that, for each potentially recombinant sequence under consideration, (a) only the subtree containing the sequences in the co-recombinant set for that sequence is considered, (b) it is assumed that the co-recombinant set is monophyletc and (c) rather than comparing the two trees to one another, the number of SPR operations required to reconstitute the monphyletic co-recombinant subtree is determined separately for both trees and averaged. These modifications take into consideration the fact that in taxa where recombination is very frequent there will be many conflicting phylogenetic signals within and between both trees that have nothing to do with the recombination event currently under consideration.
6. **O:E & O:EDist**: Methods that compare observed recombination signals with those that would be expected if each of the sequences in the triplet were recombinant. As mentioned previously, whenever a recombination event occurs it will potentially be possible to detect it if there is at least (a) one close relative of at least one of the parental sequences and (b) one descendent of the recombinant in the alignment. Whenever a sample contains more than one descendent of the recombinant or more than one close relative of one of the parental sequences, the recombination event will be detectable with more than one combination of sequences. Therefore, recombination signals (a) detected with close relatives of each of the sequences in the triplet used to identify the current event and (b) involving at least 30% sequence overlap between approximated breakpoints are identified and used to infer which of the sequences in the triplet is recombinant. This can be achieved because, depending on which sequence is recombinant, it would be expected that the recombination event should be detectable with different sets of sequence triplets. The sequence with the corresponding set of expected sequence triplets that has the greatest overlap with the set of observed triplets is most likely to be the recombinant.
7. **dMax(VisRD)**: The recombinant identification statistic described by Lemey *et al* (2009). dMax is a quartet mapping statistic that is calculated by constructing large numbers of four taxon maximum parsimony trees containing, in turn, each of the three sequences in the triplet used to detect recombination signals. Quartet map locations are determined using the fragment of the alignment between the recombination breakpoints and the remainder of the alignment. The difference between these map locations, d, is recorded for large numbers of quartets containing each of the sequences in the triplet used to detect the recombination signal. The triplet sequence that yields the greatest d across all examined quartets (i.e. dMax) is assumed to be the recombinant.
8. **Conflict**: Indicates the degree to which distances are smaller between the members of potential "co-recombinant" sets (see 4.13 above) than they are with other sequences in the alignment. Whereas it is expected that the potential co-recombinant sets of the real recombinant sequence should all be more similar to one another than any is to any other sequence in the alignment (i.e. recombinants descended from the same recombinant ancestor should be monophyletic), this is not expected to be the case for the potential co-recombinant sets of the parental sequences.
9. **OuCheck**: Indicates the degree to which phylogenetic relationships between the triplet sequences and other individual sequences in the alignment are disturbed by recombination (similar to a doublet scanning version of the dMax statistic above). It is calculated by considering the topologies of rooted NJ trees constructed from the region of the alignment between the recombination breakpoints and the remainder of the alignment. For each of the triplet sequences, the number of times relationships are maintained between the individual triplet sequences and each other sequence in the alignment across both trees is counted. The recombinant can be identified as the sequence that maintains the fewest unchanged relationships relative to the other triplet sequences.
10. **TrpScore**: Measures the change in rooted NJ tree positions (without taking actual distances into account) for each sequence in the triplet between a phylogenetic tree constructed from the fragment of the alignment between the recombination breakpoints and the tree constructed from the remainder of the alignment (similar to a triplet scanning version of the dMax statistic above). Differences in tree positions between each triplet sequence relative to every other pair of sequences in the alignment are calculated. Using averaging over branches to account for sampling biases, the enumerated topology changes are expected to be highest for the recombinant sequences
11. **SetDistT & SetDistP**: Focus on the triplet sequences and compares the numbers of polymorphic sites found between the recombination breakpoints in these three sequences with those found in the remainder of the alignment. It is expected that if the polymorphic sites are evenly distributed between the two regions, the recombinant sequence will be the one that is alternatively most closely related to the major and minor parents. If the polymorphic sites are sparser between the breakpoints then this implies both that there is an un-sampled major parental sequence and that it is the sequence that is most distantly related to the other two in the remainder of the alignment that is the recombinant. Conversely, If the polymorphic sites are more dense between the breakpoints then this implies both that there is an un-sampled minor parental sequence and that it is the recombinant sequence is most distantly related to the other two sequences in the triplet across the alignment region between the breakpoints.

A weighted consensus of these methods is used to identify the recombinant from amongst the sequences in a triplet.

It is important to note that although all of these methods work very well in sequences where recombination has been relatively rare, they all suffer from an elevated failure rate when recombination is frequent. The main reason for this is that when recombination is frequent many of the clearest recombination signals will be achieved when either two or all three of the sequences in a triplet are recombinant. Another reason is that the accuracy of trees and distance measures used to infer which sequences are recombinant, decrease as the number of detectable recombination events in an alignment increases.

In analyses where large numbers of independent recombination events are detectable it can be very difficult, if not impossible, to properly resolve the origins of sequence fragments within the recombinant sequences. However, for purposes of identifying the number of unique recombination signals in an alignment neither incorrect identification of recombinants, nor multiple overlapping recombination signals, is a fatal problem. This is because when a recombination signal is detected, a recombinant sequence is chosen and the pieces of sequence between the estimated breakpoints in all the assumed descendants of the inferred ancestral recombinant are deleted. The signal originating from that event disappears and it is not counted again during the next round of analysis. This will be true even if the incorrect sequence is chosen as the recombinant.

4.1.6 Cyclical detection and erasing of recombination signals. The systematic detection and erasing of recombination signals from an alignment is specifically carried out in the following manner:

1. An alignment is screened for recombination signals using one or more of the exploratory recombination signal detection methods that have been selected (see section 8).
2. The total pool of detectable recombination signals is examined and the signal with the best approximated probability of being a real recombination event is selected.
3. All sequences in the alignment are compared with each sequence in the triplet used to detect the selected recombination event as described in section 4.1.3. Three groups of sequences, called co-

recombinant sets, are identified as possibly having the same recombinant origin as each of the three sequences in the triplet.
4. One of the sequences in the triplet is identified as the most likely recombinant as outlined in section 4.1.4.
5. The tracts of sequence responsible for the recombination signals in the identified recombinants and all the sequences in the corresponding co-recombinant set are erased. This simply involves replacing the nucleotide characters (i.e. A, C, G and T) with gap characters (i.e. -) in the region bounded by the approximated recombination breakpoints in each of the sequences in the co recombinant set. For every tract of sequence erased a new sequence is added to the alignment. Each new sequence contains a copy of the erased sequence tract and gap characters at all other un-copied sequence positions. What this in effect achieves is to uncouple from one another the two bits of sequence that have different evolutionary histories.
6. The cycle then resumes from step (1) and continues until no further recombination signals are detectable.

It is important to note that once sequences have been erased from the alignment and the alignment is re-screened, the part of the detection procedure dealing with the identification of recombination breakpoint positions is altered slightly. When recombination events are determined to involve breakpoints that either bracket, or are predicted to be close to a portion of deleted sequence, then one or both of the breakpoint positions are marked as being "uncertain." The number of variable nucleotide positions in the sequence triplet being examined that fall between the deleted region and the position identified as the likely breakpoint, and the recombination signal detection method estimating the breakpoint position, determine when a breakpoint position is identified as uncertain. For example, for the RDP method any breakpoint within one window length (i.e. in variable nucleotides) of a deleted region is labelled as "uncertain." In cases where breakpoints bracket one or more deleted regions, detected signals are broken into two or more pieces, each corresponding to the portions of continuously uninterrupted sequence between the identified breakpoints. The recombination signals within these regions are reanalysed independently and breakpoints adjacent to deleted tracts of sequence are labelled as being uncertain.

Identifying breakpoints that are uncertain (due mostly to overlapping recombination events within a sequence triplet used to identify a recombination event) is vital for the accurate determination of detectable breakpoint distributions within a set of aligned sequences.

See section 10 of this manual for a step-by-step guide on how features in RDP5 should be used to formulate a recombination hypothesis and section 9.1 on how approximated breakpoint positions for unique events can be used to detect recombination hotspots.

4.2 Automated Query vs Reference Analyses

Besides automated fully exploratory scans it is also possible to scan for recombination using a pre-defined set of known (or strongly suspected) set of non-recombinant sequences as references. With this analysis scheme RDP5 will treat all non-reference sequences, called query sequences, as though they were potential recombinants. It will then screen each of these query sequences either against every pair of reference sequences, or, if references are grouped, against pairs of sequences drawn form different reference sequence groups (see below).

Prior to running such an analysis it is necessary to tell RDP5 which of the sequences are references. This can be done in two different ways: (1) by right clicking on the sequence names in the sequence dsplay and selecting the "*select references (for query vs reference scan)"* option; or (2) by prefixing the names of the reference sequences in the input alignment with the letters "REF" followed by a consistent reference group name identifier (if you want to use reference sequence groups). If, for example, you are analysing a virus species with known strain groupings, say strains A through D, you may want to treat reference sequences within each strain as a group. To do so you could prefix the names of the sequences in the input alignment that are is strain group A with the text "REF-A" those in strain group B with "REF-B" etc to obtain an input file in fasta format that looks something like this:

>REF-A<sequence1 name>
AAAAGCATTT
>REF-A<sequence2 name>
AAAAGCATTT
>REF-B<sequence3 name>
AAAAGCATTT
>REF-C<sequence4 name>
AAAAGCATTT
>REF-D<sequence5 name>
AAAAGCATTT
> <sequence6 name>
AAAAGCATTT

Immediately upon opening a fasta file like this in RDP5 you will be asked whether you would like RDP to automatically select reference sequence groups. If you say "yes" to this request RDP will treat sequence 6 as a query sequence, group sequence 1 and sequence 2 into reference group A and it will respectively place sequence 3, sequence 4 and sequence 5 into groups B, C, and D.

To run a query vs reference analysis you will need to press the arrow beside the "Run" button and select the "*do a full query vs reference recombination scan…..*" option. The analysis process will then proceed exactly as for an automated exploratory scan (see section 4.1 above) except that every triplet of sequences examined will contain one query sequence and two reference sequences each of which will be drawn from two different reference sequence groups. Given this constraint on the composition of scanned sequence triplets query vs refence scans will always involve fewer sequence comparisons (and therefore less severe multiple testing corrections of p-values), than if the sequences were scanned in the fully exploratory mode.

Note that query vs reference scans will not impose the constraint that reference sequences cannot be detected as recombinants: i.e. in cases where recombinant sequences have been used as references (as is common for example in HIV query vs reference recombination analyses where circulating recombinants are commonly among the reference sequences used), these will in some cases be detected as recombinants. This will occur, for example, if some of the query sequences being scanned are closely related to the parental sequence(s) of the recombinant reference sequence. In cases where a reference sequence is detected by RDP5 as being recombinant, the associated recombination event in the overview display (Fig 1) will be displayed in orange.

4.3 Manual Query vs Reference Analyses

It is possible to use RDP5 detect recombinant sequences in an alignment using a manual query vs reference sequence approach such as that used in programs like SIMPLOT (Lole *et al*., 1999) or cBrothers (Fang *et al*., 2007). Pressing the arrow button beside the "Run" button in the command button panel (Fig 1) will display a menu from which you can select any of seven manual recombination detection methods (GENECONV, BOOTSCAN, MAXCHI, SISCAN, LARD, 3SEQ, Distance Plot or TOPAL). You may be prompted to:

1. **Select a potential recombinant sequence** (GENECONV, BOOTSCAN, MAXCHI, and Distance Plot). You should choose the potential recombinant sequence against which you would like to scan potential parental sequences.
2. **Select an Outlyer Sequence** (SISCAN): Select a sequence that is more distantly related to the potential recombinant sequence than either of its parents.
3. **Select parental and/or outlyer sequences** (GENECONV, BOOTSCAN, MAXCHI and Distance Plot): Select the sequences against which you would like to screen the potential recombinant sequence by clicking on the name of sequences in the left panel. You can unselect sequences in the right panel by clicking on them. For Distance Plots you need only select one sequence, for MAXCHI and GENECONV scans you need to select at least two sequences and for BOOTSCANs you must select at least three (two potential parental sequences and an outlyer). If you are attempting to determine the origin of sequences in a recombinant you should always try to select the likely parents of the recombinant and a sequence that is more distantly related to the parental sequences than they are to one another. Note, however, that for manual MAXCHI and GENECONV scans a very divergent outlyer may decrease the power of the scan – You should try select a outlyer that is as closely related to the parental sequences as possible. Also note that when selecting parental sequences for a manual BOOTSCAN you should avoid selecting potential parental sequences that are more closely related to one another than they are to the recombinant. If you are unable to avoid selecting parental sequences that are more closely related to one another than they are to the recombinant you should use the "closest relative scan" option (see below).

4. **Select parental and recombinant sequences** (SISCAN, LARD, 3SEQ). Select three sequences by clicking on sequence names in the left panel. Try to select one recombinant sequence and its two parental sequences. If one of the parental sequences is absent from the alignment recombination could still be detectable using these methods if you select a "parental" sequence that is more distantly related to both the recombinant and the parental sequence that is in the alignment than these two sequences are to one another. This "parental" sequence should, however, still be more closely related to both the recombinant and the parent than either of these sequences are to the actual parent that has gonr unsampled.
5. **Select Sequences** (TOPAL). Select four or more sequences by clicking on their names in the left panel. The sequences chosen should include a recombinant sequence, at least one parental sequence, and an outlyer sequence that is more distantly related to the parental and recombinant sequences than they are to one another.
6. **Closest relative scan option** (BOOTSCAN). If any of the parental and/or outlyer sequences used in a scan are more closely related to one another than they are to the potential recombinant, you should select this option. If you scan without this option, parts of the scan over which parental sequences are more closely related to one another than they are to the recombinant will contain no information on which of the parental sequences the recombinant most resembles.

Once you have selected enough sequences, pressing the "OK" button will perform the analysis. Results of the manual scan will be graphed in the plot display (Fig 1). A key indicating the meaning of the different plotted lines will be given in the recombination information display (Fig 1). Clicking on the names or coloured boxes in this display will highlight the corresponding plot in the plot display.

5 EXAMINING AUTOMATED ANALYSIS RESULTS

The basic RDP5 interface is broken up into six separate panels, four of which are displayed at any one time (see Fig 1). From top left, moving clockwise these are (1) the sequence display, (2) the overview display, (3) the recombination information display, (4) the dendrogram display (5) the matrix display (you can toggle between (2), (3) (4) and (5) but they are not all displayed together), (6) the schematic sequence display, and (7) the plot display. Each display has a range of associated features many of which are accessible through a series of display-specific menus which are accessible by pressing the right mouse button when the mouse pointer is over the different displays. Whenever specific menu items are discussed below, they will be identified with blue text. Because the examination of results proceeds via either the overview or schematic sequence displays, it is these displays that will be described first.

5.1 The Overview & Schematic Sequence Displays

Once an automated analysis has concluded, a summary of the detected recombination events is tabulated in the overview display and schematic representations of the aligned sequences indicating positions of potential recombination events are presented in the schematic sequence display (Fig 2). These two displays give overviews (tabular and graphical) of the over-all recombination hypothesis that RDP5 has come up with. It is very important that you realise that the program is fallible and that it is very likely that its hypothesis can be improved with your guidance.

The program displays only the best evidence (i.e. the evidence with the best associated p-value) of recombination that it has detected. The unique recombination events that have been detected are presented in each row of the overview display table, and as coloured rectangles in the schematic sequence display. Each of the colored rectangles in the schematic sequence display represents a recombination signal. The left and right bounds of each rectangle mark the inferred breakpoints flanking a fragment of sequence transferred by recombination. Each rectangle is also labelled with the name of a sequence in the alignment that most closely resembles the presumed donor (or minor parent) of the depicted piece of sequence.

These representations of potential recombination events can be colour coded according to:

1. Their most likely parental origins (unique colours are given to every potential donor sequence in the alignment).
2. The recombination signal detection methods that identified them.
3. Their associated p-value's.
4. The relatedness of their inferred parental sequences.

The colour coding can be changed by pressing the "cycle through display options" button (Fig 2) on the bottom of the schematic sequence display. A key to the currently selected colour coding can be viewed by clicking on the left mouse button when the mouse pointer is over any grey area of the schematic sequence display (note that a key is not available for the "unique sequences" display).

Menus that provide various analysis and data management options can be accessed by right clicking in either the overview or schematic sequence displays. If the mouse pointer is over either particular row of the overview display table or a rectangle representing a specific recombination event, a menu will appear with options that relate to that event. Right clicking on any other part of the schematic sequence display provides a menu with options relating to the recombination display as a whole.

5.1.1 Using the overview and schematic sequence displays to select recombination events. The rectangles representing recombination events in the schematic sequence display are sensitive to the mouse pointer and, when the pointer is moved over one of these rectangles: (1) the row of the overview display table relating to that recombination event will be highlighted, and (2) detailed information relating to the event will be presented in the "recombination information display" (see section 5.2 and Fig 3). Either clicking on the left mouse button when the mouse pointer is over the rectangle or clicking on the left mouse button when the mouse pointer is over a row of the overview display table will select that recombination event for more in-depth analysis. Immediately after right clicking on the rectangle representing a recombination event, a plot of the raw data that was used to identify the event will be presented in the plot display (section 5.3 and Fig 4), the nucleotide sites used during the analysis will be highlighted in the sequence display (section 5.4 and Fig 5) and UPGMA or FastNJ trees (useful for visually checking the RDP5's identification of parental and recombinant sequences) will be presented in the tree display (section 5.5 and Fig 6).

5.1.2 Saving a graphic of the schematic sequence display. An enhanced metafile (.emf) graphic of this display can be saved to disk by clicking on the right mouse button when the mouse pointer is over any grey area of the schematic sequence display and then selecting the "Save to .emf file" menu option that is offered. Alternatively if you select the "Copy" menu option then the graphic will be copied to the clipboard and can be pasted into other programs that accept the .emf graphic format (e.g. Word and Powerpoint). Note that to edit copy and pasted images of the schematic sequence display in a program like powerpoint you will need to "ungroup" the image and convert it into a powerpoint object. After converting it to a powerpoint object you may need to ungroup it a second time.

5.1.3 Navigating through data presented in the overview and schematic sequence displays. Evidence of recombination can be presented within the schematic sequence display in various different ways. Apart from changing the way different kinds of events are colour coded (see the beginning of section 5.1), you can change the types of event that are displayed. Click on the right mouse button when the mouse pointer is over any grey area of the schematic sequence display and a menu will be displayed with the following three options: (1) "Show all events for sequence X" (sequence X is the specific sequence who's "space" the mouse pointer is closest to), (2) "Show only best events for all sequences," and (3) "Show all events for all sequences." If you choose to show all events RDP5 will display, stacked one on top of the other, representations of all the "best" recombination signals associated with specific recombination events that have been detected by different recombination analysis methods. Whereas obvious recombination signals might be detectable with all seven or eight of the methods that RDP5 uses to automatically check signals, less obvious signals might only be detectable with one or two different methods. If you choose to show only the best events (the default) the stacked representations of recombination signals will be collapsed and only the "best" signals (i.e. those associated with the lowest p-values) will be displayed.

Although it is possible to query the evidence for any particular recombination signal represented in the schematic sequence display it is strongly recommended that you use the tools RDP5 provides to navigate through the data in a structured way. If you select the "Go to event" menu option you will see that various alternatives are offered. You can opt to go to the "best unaccepted event," the "previous event" or the "next event." You can also select whether you wish to skip

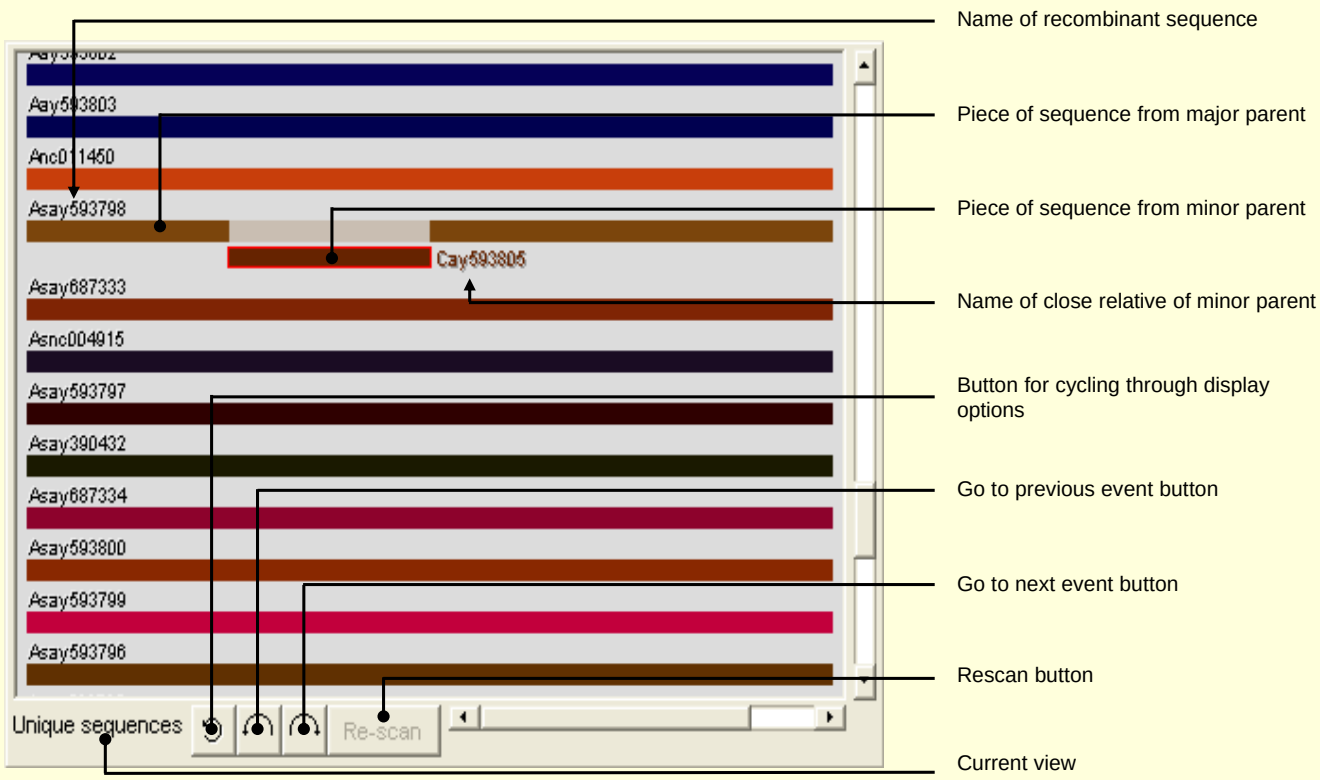

Figure 2. The schematic sequence display. This is where the results of automated recombination scans are presented and it is the part of the program that is used to drive the manual checking of automated analysis results. The coloured rectangles represent sequence fragments. The sequence names in black on the left refer to the rectangles beneath them (labelled "Piece of sequence from major parent" in the figure). The rectangle labelled "Piece of sequence from minor parent" is a graphical representation of a sequence fragment that has potentially been derived through recombination from a sequence resembling the one named to the right of the rectangle. These rectangles represent recombination events. If the mouse pointer is moved over such a rectangle (it will become highlighted) and the left mouse button is clicked the recombination event represented by the rectangle is "selected" for more detailed analysis (The rectangle will begin to flash, information will fill the recombination information display (see Figures 1 and 3 or Section 5.2), a plot indicating the exact recombination signal used to detect the recombination event will be drawn in the plot display (see Figures 1 and 4 or Section 5.3), and trees describing the phylogenetic consequences of the recombination event will be drawn in the tree display(s) (see Figures 1 and 6 or Section 5.5). Right clicking on either coloured rectangles or the grey areas around rectangles will bring up two different command option menus. The "cycle through display options" button will change the colour scheme to highlight different aspects of the recombination events being displayed (such as the methods used to detect the depicted recombination events, the p-values of the recombination signals and degrees of parental sequence relatedness). Use the "Go to previous event" and "Go to next event" buttons to navigate through the results in an ordered way (preferably in the same order as the recombination events are numbered in the recombination information display – see Figure 3). These buttons will help you find the best evidence of particular recombination events (initially "event 1"). The "Rescan" button will start flashing whenever automated analysis results are manually modified in a way that could have an influence on the interpretation of other detected recombination events.

"accepted events" and "rejected events" – these will be explained later in section 5.14.

During its automated recombination detection scanning phase of an analysis, RDP5 attempts to formulate a consistent recombination hypothesis to explain the detected recombination signals in an alignment (see section 4.2 for some details of what the program does to formulate this hypothesis). The hypothesis is formulated in a step-wise fashion with the most obvious recombination signals being accounted for first and the least obvious last. Unfortunately, the program is fallible and will probably make mistakes at some stages of this process. When it makes a mistake at a particular step it will be more likely to make a mistake in all subsequent steps and it is therefore advisable that you analyse the recombination signals in the same order that RDP5 dealt with them. This way when you see the program has made a mistake you can tell it to only re-evaluate the recombination signals that it dealt with after the mistake was made.

You can navigate through the events in the same order as RDP5 dealt with them by starting at the first event listed in the overviewdsiplay table. At the end of an automated scan if you select the "Go to next event" menu option you will be taken to event number 1. Alternatively you can press the left mouse button on a grey background section of the schematic sequence display and then press the "Pg Dn" button on the keyboard and you will also be taken to event 1. Alternatively, the event navigation buttons at the bottom of the schematic sequence display (Fig 2) can be used to navigate through the events in a structured way. Collectivelty, you can navigate backwards and forwards through the events by clicking on them in the overview table, using the schematic sequence display menu options, the "Pg Up" and "Pg Dn" buttons or the navigation buttons.

5.1.4 Managing data presented in the schematic sequence display. Pressing the right mouse button when the mouse pointer is over a recombinant region will display an "editing" menu that will allow you to accept and reject evidence of recombination, and "correct" any mistakes that the program has made in its parental/recombinant designations. You should take care when using the parent/recombinant swapping options because: (1) correctly identifying parents and recombinants is often very difficult; and (2) the program is not infallible when identifying recombinant/parents but it is objective whereas you may not be. Make sure that you do not put too much faith in the identified (either by you or the computer) polarity of recombination events.

It is very important that you use the "Accept" or "Reject" evidence of recombination options via either the "accept" button on the plot display (see section 5.3 and Fig 4), or the overview and schematic sequence display menus as you go along. This both helps you keep track of where you are when going through the results of an analysis, and tells RDP5 which events it should not reconsider when you tell it to reformulate an improved recombination hypothesis. As you move sequentially through the proposed recombination events you should

specifically "accept" evidence for which RDP5 has (1) correctly identified the recombinant sequence, (2) correctly identified the recombination breakpoints, and (3) has neither over- nor under-grouped sequences that have similar evidence of recombination that may/may not indicate they are descendents of a common recombinant ancestor (for help making these decisions see section 10.4 of the step-by-step guide). If for a particular recombination event RDP5 incorrectly identifies the recombinant sequence(s), incorrectly identifies recombination breakpoints, or incorrectly groups sequences as having descended from an ancestral recombinant, it will have been more error prone when analysing all subsequent events. You must therefore try to correct the most glaring of these errors (see section 5.15) when you find them, "Accept" your corrections and then tell the program to "Re-Identify recombinant sequences for all unaccepted events" – this is one of the menu options that appear whenever you press the right mouse button anywhere in the schematic sequence display. You can also do this by pressing the flashing red "Re-scan" button beneath the schematic sequence display (Fig 2).

When an event is "accepted" RDP5 draws a red rectangle around its representative row on the overview display table and around its representative coloured block(s) in the schematic sequence display. The "Accept this event in all [number of sequences] sequences where it is found" option should be used when you are happy with the way that RDP5 has grouped both the recombination signals it has (1) detected in different sequences descended from an ancestral recombinant, and (2) detected by different recombination detection methods within individual sequences. If you are not happy with how RDP5 has grouped the sequences you can opt to individually accept a given recombinaton event in specific sequences using the "Accept this event only in this sequence" option. When an event is accepted in a particular sequence RDP5 will not re-evaluate the event when you tell it to make an improved recombination hypothesis using either the Re-scan button or the "Re-Identify recombinant sequences for all unaccepted events" menu option.

5.1.5 Correcting RDP5 via the schematic sequence display. Two of the three main errors that RDP5 will make can be corrected via the menu options provided in the schematic sequence display.

Whereas the schematic sequence display can be used to identify possible inaccuracies in recombination breakpoint prediction, these must be corrected using either the plot display (see section 5.3) or the sequence display (see section 5.4). When you select the "show all evidence" menu option and representations of the signals detected by different methods are all displayed together, you can quickly assess whether there are differences in the breakpoint positions identified by different methods. If there are differences it will often be worthwhile to carefully check the identified breakpoint positions - even if this involves looking at the sequences by eye.

Conversely, inaccurate identification of recombinant sequences (i.e. when a sequence identified as parental is in fact the recombinant) cannot be determined using the schematic sequence display (see section 10.4 in the step-by-step guide on how such errors are identified) but it can be fixed using the schematic sequence display menus. If you right click on the representation of a recombination event the last three menu items displayed give you the option of "swapping" the recombinant and parental sequences. For example, if the sequence identified as the "minor parent" is the sequence you think should have been identified as the recombinant select the "Swap recombinant and the minor parent" option.

Remember to "Accept all similar" if you are satisfied that all sequences in the alignment that carry traces of the current recombination event (i.e. all those sequences that are descended from the ancestral sequence in which the recombination event occurred) have been identified. If only some of the recombination signals have been correctly identified, then individually "Accept" only the specific signals that you believe represent evidence of the recombination event. If you choose to discount some signals in this way (there is another way of doing this via the phylogenetic trees – see Section 5.5) make sure that you individually accept all of the appropriate signals – If, for example, you only select the best signal (the one that that is always displayed) for a particular sequence, RDP5 will assume that all the other unselected signals are incorrect and should be discarded. If you leave some signals unaccepted but RDP5 has identified them as being evidence of the same event you are analysing, you will in effect be telling RDP5 that you think it has over-grouped the detected recombination signals (i.e. RDP5 will assume you are telling it that the current recombination event is in fact two or more distinct recombination events). If the unaccepted recombination signals are re-detected, RDP5 will interpret these as being evidence of a different recombination event.

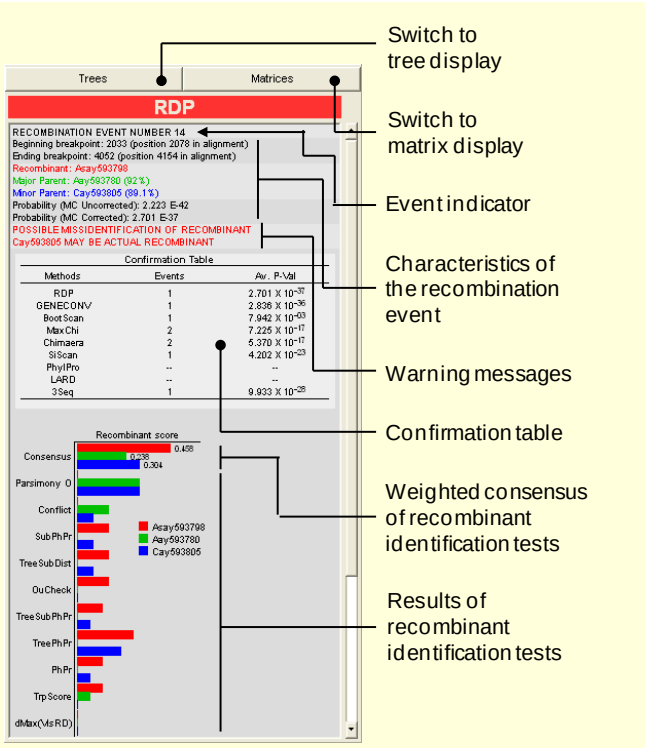

Figure 3. The recombination information display. Each apparently unique recombination event is numbered according to the order in which RDP4 characterised the event. You should start checking results from event 1 and move through the potential recombination events in the same order that RDP4 characterised them. Breakpoint positions are specific for the recombinant (or potentially recombinant) sequence that has been selected (the one with the flashing yellow rectangle in the schematic sequence display – see Figure 2 and Section 5.1) – the breakpoint positions in the alignment are given in parentheses. The "major parent" is usually (but not always) a sequence closely related to that from which the greater part of the recombinant's sequence may have been derived. The "minor parent" is usually a sequence closely related to that from which sequences in the proposed recombinant region may have been derived. p-values that are displayed are either multiple comparison (MC) corrected or uncorrected. Also displayed in **BOLD RED CAPITALS** are various warnings. The confirmation table gives an overview of (1) how many methods have detected the current recombination signal, (2) in how many different sequences and (3) with what degree of certainty. The bar graphs beneath the table indicate relative degrees of support for different sequences being identified as the recombinant based on various tests (see Section 4.1.4 on what these tests are showing). The colour scheme is: Red = the probable recombinant; blue = a close relative of the presumed minor parent; green = a close relative of the presumed major parent. The consensus scores indicate the relative degrees of support for each of the sequences being identified as the recombinant.

Besides using different combinations of "accepts" and "rejects" to correct mistakes the program makes in over-grouping sequences, the menus of the overview and schematic sequence display can also be used to correct under-grouping of events – i.e. when RDP5 has identified sequences descended from the same ancestral recombinant as carrying evidence of two different unique recombination events. The "Merge events" menu option gives you the opportunity to group signals from any two identified events as having originated from the same original recombination event. Grouping and ungrouping events can also be achieved using the tree displays (Section 5.5).

If you modify breakpoint positions, recombinant designations or groupings of detectable recombination signals, you must first accept your modifications and then select either the "Re-identify recombinant

sequences for all unaccepted events" menu option or press the flashing red "Re-scan" button. If you evaluate recombination events in the same order that RDP5 identified them and accurately correct mistakes that the program has made, then each new recombination hypothesis RDP5 formulates when you select this option will be an improvement on the last and eventually a consistent story should emerge from the data.

5.2 The Recombination Information Display

When the mouse pointer is moved over a coloured rectangle representing a potential recombination signal in the Schematic Sequence Display (Fig 2), information on that region is printed in the Recombination Information Display (Fig 3). This information includes the event number, the method(s) used to detect the recombination signal, the names of sequences that are closely related to likely parental sequences (major and minor parents) and the approximate probability that the recombinant sequence could have been more closely related to the "minor parent" than the "major parent" in the specified region because of chance deviations of mutational patterns from what would be expected in the absence of recombination. For any particular recombination signal the meaning of the p-values that is displayed here will vary slightly according to the recombination detection method used to detect the signal. The p-values displayed for the different methods are described in Section 8.

The names of the recombinant, major parent and minor parent are sensitive to the mouse pointer and left clicking on these names will result in the schematic representation of these sequences being displayed in the schematic sequence display.

Also displayed are warnings if:

1. There is only a single suitable parent-like sequence in the set of aligned sequences.
2. There is a fair likelihood (an approximately 30% or greater chance) that the program has misidentified the recombinant sequence (i.e. the actual recombinant is one of the sequences identified as a parental sequence). If one or both of the identified parental sequences is almost as likely to be the recombinant then the name(s) of the sequences are given.
3. One or both breakpoints could not be identified.
4. One or both breakpoints may have been misplaced.
5. The signal represents only trace evidence (i.e. it is not statistically significant) of a recombination event detectable in one or more other sequences (i.e. it has an associated p-value > than the cut-off)
6. If the recombination signal is a possible/probable misalignment artefact.

These warnings are meant as a prompt for you to carefully examine the presented data and make a judgment on whether the program's interpretations are correct or not. Even when no warning is given it is always advisable to properly examine results. There is always a fair chance that the methods implemented in RDP5 will inaccurately determine breakpoints, incorrectly identify parental and recombinant sequences and over- or under- group sequences believed to be descended from ancestral recombinants. For example, the original RDP method will misidentify recombinant sequences without giving a warning when a substantial proportion of the reference sequences being used are themselves recombinant. You should carefully examine all potential recombination events using the supplementary analyses that are offered by RDP5 (see the step-by-step guide in Section 10).

The "confirmation table" part of the recombination information display gives some indication of (1) the number of sequences in the alignment that the currently selected recombination event has been detected in and (2) the degree of agreement between different detection methods regarding the currently selected recombination event.

The histogram beneath the confirmation table summarises the results of various assays that the program uses to infer which of the sequences used to detect a recombination signal, is the recombinant. The assays are briefly outlined in Section 4.1.4. The only really relevant bit of this plot to 99% of users will be the top three bars representing the "consensus" scores of the three sequences indicated. The numbers next to these bars are the "consensus scores" of the three sequences. These scores have no real meaning other than that the higher the score the more confident you should be in the program's assessment of which sequence is recombinant. A score >60 indicates that the identified sequence is almost certainly the recombinant. A score <60 but >40 means that the program may have made a mistake (but probably didn't). Anything lower than this indicates that the program is VERY unsure about which sequence is the recombinant. It is under these circumstances where your input can be most useful. You should realise though that your opinion may not be very valuable if, for example, you are not very good at interpreting phylogenetic trees.

The Information display can also be used to modify how RDP5 interprets breakpoints. You will notice if you left click on the "Beginning breakpoint" or "Ending breakpoint" fields within this display, that the breakpoints will be given an "Undetermined" label. This label is important because undetermined breakpoints will be ignored when RDP5 tests breakpoint distributions for evidence of recombination hot- and cold-spots.

5.3 The Plot Display

Left clicking on the coloured rectangles that represent rescombination signals within the schematic sequence display (Fig 2) will produce a graphical plot of the actual signal (Fig 4). The whole plot is sensitive to the mouse pointer and:

1. Double clicking anywhere in this panel will take you to the corresponding region in the sequence display panel (Fig 5).
2. Moving the pointer around the plot will display a cross hair for which X and Y coordinate values are displayed (Fig 4).
3. When a SISCAN plot is being displayed left clicking will produce a key that describes the meaning of the various plotted lines. Clicking on any of the plots indicated in the key will highlight that plot in the Plot Display. For a key of what the different plots represent see Gibbs *et al.* (2000)

At the top of the plots is a graphical representation of the distribution of polymorphic/analytically relevant sites that were used to detect the recombination signal. In the MAXCHI, CHIMAERA, SISCAN and GENECONV plots, broken lines indicate the p-value cutoffs that were used to determine the significance of breakpoints (MAXCHI, CHIMAERA) or potentially recombinant fragments (GENECONV, SISCAN). See section 8 for specific descriptions of what is being plotted for the various methods.

When you right click on the plot display you will be given the option to (1) save a graphic of the plots (in either .emf or .bmp format) (2) save the actual raw data used to construct the plots (in comma separated value or .csv format) or (3) copy an image of the plots to the clipboard (so that the plots can be pasted into Word, Powerpoint or any other .emf viewer).

Beside the plot display is a panel with the caption "Check using." In this panel are two buttons with the words "Options" "and STOP" on them. There is also a "combo" box that should have the name of a recombination detection method displayed. This combo box can be used to test whether various other recombination analysis methods are also capable of detecting the current recombination signal. The Options button can be used to adjust parameter settings for the method currently selected in the combo box. The "STOP" button can be used to terminate a scan that is taking too long (as sometimes happens with the LARD or TOPAL methods).

Besides being used to cross-check different recombination detection methods, graphical overviews of the detected recombination events can also be accessed via this combo box. These include

1. **Overview**: These plots are similar to those displayed during the automated recombination screening scan. The main additional feature in the overview plots is that the recombination signals being represented are broken down according to the methods used to detect the signals. You can see a colour key indicating the methods that detected the various signals by left clicking on the plot. The vertical lines in these plots indicate the estimated positions of breakpoints and the upper horizontal lines indicate either the genetic distance between parental sequences (PDist), the p-values associated with the detected recombination signals (PVal) or the number of times individual regions of the aligned sequences were inferred to have been transferred by recombination (#Hits).

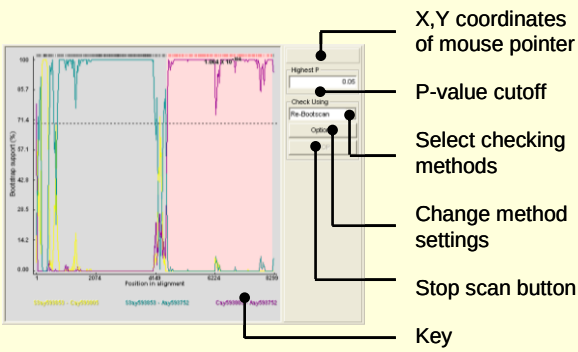

Figure 4. The plot display. Interpretation of plots varies between different checking methods (see section 8 for details on what is being plotted). Different coloured lines usually indicate different sequence pairs (the names are given in the key). Vertical lines above the plot indicate positions of the variable nucleotide sites that have yielded the signals being plotted (these sites can be individually colour coded in the sequence display – see Figures 1 and 5 and section 5.4). The left and right boundaries of the pink area indicate approximated recombination breakpoint positions. The X,Y coordinates of the mouse pointer are displayed whenever the pointer is over the plot. Analysis settings used to generate the plots can be changed by pressing the "Options" button beside the plot and the recombination signal depicted by the plot can be examined with various other recombination detection/analysis methods using the "Select checking methods" box.

2. **Recombination event map**: This plot is similar to the p-value portion of the overview plots described above, except that the colours that are displayed represent degrees of parental sequence relatedness. Whereas cooler colours indicate that parental sequences were more distantly related, warmer colours indicate that they were more closely related.
3. **Breakpoint density**: This is a sliding window plot indicating the clustering of detectable recombination breakpoints along the alignment and can be directly used to infer the existence of statistically supported recombination hot- and cold-spots. See Section 9.1 for a description of how this plot is produced and the underlying tests performed. Whereas the plotted line represents the number of breakpoints detectable within a moving window of user specified size (press the "options" button to change the window size), the grey and white areas around the line respectively indicate the 95% and 99% confidence intervals for the expected degrees of breakpoint clustering in the absence of recombination hot- and cold-spots. Whereas if the black line emerges above these shaded areas it indicates the existence of a recombination hot-spot, if it drops below the shaded areas, it indicates the existence of a recombination cold-spot. The upper and lower dotted lines respectively indicate "global" 99% and 95% confidence intervals of there being recombination hot-spots. Note that this test is extremely conservative. See Section 9.1 for a description of what the global confidence intervals mean.
4. **Breakpoint P-density**: This plot is a version of the breakpoint density plot described above in which the plotted values correspond to probabilities (rather than absolute breakpoint numbers) that breakpoints are not significantly clustered. It is essentially a transformed version of the breakpoint density plot in which the dimensions of the shaded bits are held constant and the black line is plotted relative to these.

5.4 The Sequence Display

The sequence display (Fig 5) can be cycled to show (1) the entire sequence alignment, (2) only the sequences involved in identifying the currently selected recombination signal, or (3) only the informative sites within the sequences involved in identifying the currently selected recombination signal. Left clicking in the sequence display will produce a key that describes the colour coding of the nucleotides in the display.

Holding the mouse pointer over any nucleotide in the sequence display will indicate the position of that nucleotide in its unaligned sequence.

You can also save alignments in various formats and with various pieces of sequence/whole sequences omitted using the menu that is accessed when you right click anywhere in the sequence display. The alignment saving options include:

1. Save entire alignment: Will save the full alignment in whatever format you specify.
2. Save alignment with recombinant sequences removed: Will save an alignment minus any of the recombinant sequences identified during an automated recombination scan. To tell which sequences will be included in the alignment look, at the schematic sequence display. Any sequence that is represented by an unbroken line will be included
3. Save alignment with recombinant columns removed: All alignment positions that fall between pairs of identified recombintion breakpoints in ANY sequence in the alignment will be removed for all sequences. If many recombinant regions have been detected with an alignment, this option could very easily yield an empty or nearly empty alignment.
4. Save alignment with recombinant regions removed: All nucleotide positions in any sequences that are between any identified recombinantion breakpoint pair will be removed and replaced with gap ("-" or ".") characters.
5. Save alignmnet with recombinant regions seperated: Recombinant sequences within the alignment will be split into two or more sequences. For every detected recombination event the sequence(s) carrying evidence of the event will be split into two parts – one part between the identified recombination breakpoints, and the other from the remainder of the sequence. Gap characters will be inserted into the two sequences to properly maintain their alignment positions. The resulting alignment should be free of detectable recombination events.
6. Split alignment into common mosaics: All sequences in the alignment that have either identical recombination mosaics (i.e. the same pattern of recombination detected events) or are non-recombinant will be split up into separate alignments.
7. Split alignment into recombination free sub-alignments: The alignment will be split into multiple sub-alignments each containing no detectable recombination signals.
8. Save only enabled sequences: Only sequences that are "enabled" (see section 4.1.1) will be saved. This is useful for manually splitting the sequences in the alignment up into related groups.
9. Save only disabled sequences: Only sequences that are either disable or masked (see section 4.1.1) will be saved.

When you are saving modified alignments you will often be asked whether to consider all of the detected recombination signals or only those that you have accepted (see Section 5.1.4).

Left clicking on the names of sequences to the right of the sequence display will cyclically mask, disable and enable the sequences in the alignment. See section 10.1 for reasons why you should sometimes mask or disable sequences. Masking or disabling some sequences in an alignment will reduce the number of recombination signal detection scans and thereby both speed up an analysis and reduce the severity of multiple testing correction needed during p-value calculation. Whereas masking a sequence will mean that RDP5 will avoid looking at the sequence during a primary automated recombination screen, the sequence will still be looked at during secondary screens and will also be used within the context of phylogenetic trees to determine which sequences are recombinants. Disabled sequences will not be examined at all for evidence of recombination (even during the secondary scanning phase) but will still be included within phylogenetic trees.

Right clicking over the sequence names will display a menu of options. You can "Mask all", "Enable all", "Disable all" or "Invert masking." The most useful option for general recombination analysis is "Auto mask for optimal recombination detection." This setting will focus the analysis on sequences where it is possible to detect recombination while ignoring efforts to detect recombination between sequences that are too similar. This can substantially increase the power of RDP5 to detect recombination, particularly in large alignments containing mixtures of very similar sequences (sharing <99% identity) and more diverged sequences (<90% identical).

If you are interested in looking for recombinants in a specific group of sequences but would like RDP5 to check a larger set of sequences in case some of these are good candidate parents, you can designate a group using the "Select group" menu option. To select a

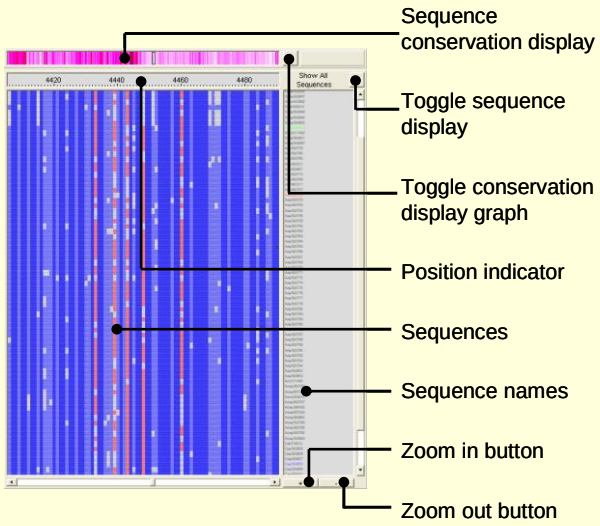

Figure 5. The sequence display. The sequence conservation display is a graphical overview of the sequence alignment that also indicates the portion of the alignment that is currently presented. Within the sequence part of the display, individual nucleotides are colour coded according to their degree of conservation. When a recombination event is selected (see Figures 1 and 2 or Section 5.1), the "toggle sequence display" button can be used to highlight nucleotide polymorphisms that contribute to the recombination signals depicted in the plot display (see Figures 1 and 4 or Section 5.3). Red green and blue highlighted sequence names indicate recombinant, major parent and minor parent sequences, respectively. Use the "Zoom in " and "Zoom out" buttons to either reduce or enlarge the portion of the alignment that is shown.

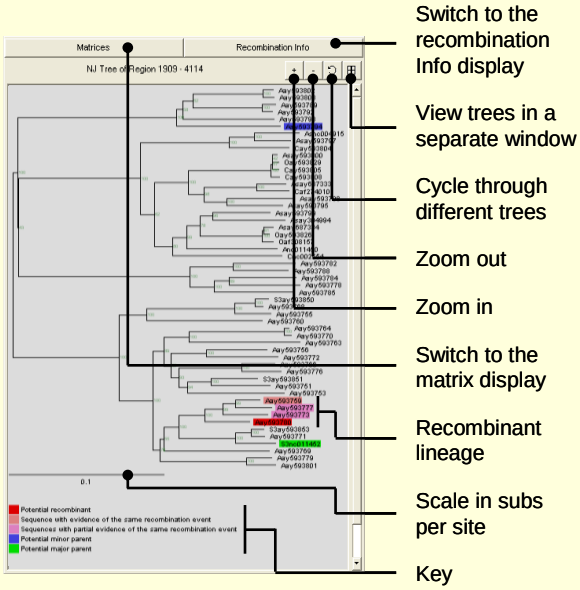

Figure 6. The tree display. Green and blue highlighted sequences indicate reasonably close relatives of major and minor parents. The red highlighted sequence is the currently selected recombinant sequence (i.e. the sequence with the flashing yellow rectangle beneath its name in the schematic sequence display – see Figures 1 and 2 or Section 5.1). Pink and purple sequences are sequences with similar (Pink) or somewhat similar but notably different (Purple) recombination signals to that observed in the sequence highlighted in red. These Red, pink and purple sequences possibly evolved from a common ancestral recombinant sequence. Enlarge or reduce the tree using the "Zoom in" and "Zoom out" buttons. When using trees to test whether RDP has correctly identified recombinant sequences it will usually be best to look at the side-by-side tree display in a separate window – Press the "View trees in a separate window" button to do this. The "cycle through different trees" button will change the fragment of the alignment that is used to construct the tree that is displayed – Note that only UPGMA trees are usually displayed here. To see trees drawn with other methods press the "View trees in a separate window" button, right click on the tree displays that are shown and select the "Change tree type" option that is displayed – this will allow you to draw neighbour joining (recommended for all datasets), maximum likelihood and Bayesian Trees (only recommended for datasets with <50 sequences).

group choose this option and then click on the names of sequences you would like to include as candidate recombinants. When you click on the sequence names they will turn blue. If you click on a blue name it will turn black again. Whereas names in blue denote candidate recombinants, those in black denote sequences against which these recombinants will be screened.

If you would like RDP5 to adjust the schematic sequence display to show a particular sequence in the sequence display, move the mouse pointer over the name of the sequence, right click and select the "Go to" option. The representation of the sequence that the mouse pointer is over will be indicated in the schematic sequence display (Fig 2).

5.5 The Tree Displays

If you press the "Trees" buttons (Figs 3 and 7) a number of different trees expressing the relationships between the identified recombinant and other sequences in the alignment will be displayed in phylogenetic trees constructed using various different parts of the alignment. If the "Trees" button at the top of the screen in the command button panel (Fig 1) is pressed, two trees will be displayed side-by side. Alternatively if you press the "Trees" button above the recombination information display (Fig 3), a tree (Fig 6) will be displayed in the same space as the recombination information display. Different trees constructed using different bits of the alignment can be viewed by pressing the "cycle through trees" button (Fig 6). These trees include those constructed using (1) all regions of recombinant sequences examined separately, (2) only the identified recombinant region (the region related to the "minor" parent in the selected sequence), (3) only the identified "non-recombinant" region (the region related to the "major" parent in the selected sequence) or (4) all regions ignoring recombination.

When the side-by-side trees are displayed in the separate window (i.e. when you press the "Trees" button in the command button panel indicated in Fig 1) it is possible to mark sequences in one tree and have the corresponding sequences in all other trees marked at the same time. This feature is very useful for tracking the "movement" of recombinant sequences around trees constructed from different parts of an alignment. Sequences can be marked/unmarked by left clicking on their names in the trees.

Right clicking in the side-by-side tree display gives you a number of options. Selecting the "Find sequence" option will allow you to search the tree for a specific sequence (which, if found, will be highlighted in the tree with a white background). The "Clear colour" option will remove all markings from the trees, the "Auto colour" option will colour all sequence names in the tree the same colours as sequences presented in the schematic sequence display, and the "Select colour" option will allow you to select a colour with which to mark sequences.

When the mouse pointer is moved over nodes within the displayed trees a blue spot appears. If the left mouse button is pressed then all the sequences represented on the right of the node will be marked with whatever the currently selected colour is. If the right mouse button is pressed a menu is displayed. Options on this menu include: "Mark/Unmark sequences above this node as having evidence of this recombination event" which can be used to correct mistakes that RDP5 has made in over- or under-grouping sequences it thinks have descended from a common ancestor; "Find best major/minor parent above this node" which can be used to identify the sequence above this node that, if swapped for the currently indicated major/minor parent would yield the strongest signal of recombination; "Accept/Reject all recombination events above this node" which can be

used to inform the program that you are happy/unhappy with the characterised recombination signals detectable in whole groups of sequences; and "Colour/Uncolour all sequences above this node" which can be used to simultaneously colour/uncolour large groups of sequence names within the tree. The last menu option, "Determine ancestral sequence at this node," will prompt RDP5 to attempt the determination of the ancestral sequence at this node using the maximum parsimony (with the DNAPARS component of PHYLIP; Felsenstein, 1989), maximum likelihood (with RAxML; Stamatakis, 2006) and/or Bayesian (with MRBAYES 3.2; Ronquist *et al*., 2012) approaches. Note that estimations of ancestral sequences using a Baysian approach can take a very long time. When an ancestral sequence has been inferred it can be saved to a .csv file by right clicking on the ancestral sequence that is displayed.

Other options on offer in the standard tree menu (the menu that is shown when you press the mouse button while the pointer is over an empty grey area of the tree display) relate to saving either the tree image (the "Copy", "Save to .emf file" options), or the Newick format encoding (the "Newick format" option) that will allow you to reload the tree in programs like Mega (Kumar *et al*., 2008), FigTree (an excellent tree viewer and annotation program by Andrew Rambaut that is available for free from http://tree.bio.ed.ac.uk/software/figtree) and TreeView (Page, 1996). Unlike with the tree display in the main RDP5 window, in the side-by-side tree display you are also given the option of changing the default trees that are constructed every time you select a new recombination event from UPGMA trees to FastNJ trees (with the "Make FastNJ the default tree" option). Individual UPGMA/FastNJtree can be redrawn as neighbour joining, maximum likelihood, or Bayesian trees by selecting the "Change tree type" option. Be very careful when selecting the latter two tree types – they might take much longer to construct than you will be prepared to wait.

The side-by-side tree display also has nine additional menu options that are only accessible if the mouse pointer is over one of the sequences in the tree when the right mouse button is pressed. The "Mark [sequence name] as also having evidence of this event" option alternates with the "Mark [sequence name] as not having evidence of this event." These menu options can be used to correct mistakes that RDP5 has made in over- or under-grouping sequences it thinks have descended from a common ancestor.

The "Accept this event only in this sequence", "Accept this event in all [number of sequences] sequences where it is found", "Reject this event only in this sequence", and "Reject this event in all [number of sequences] sequences where it is found" options are the same as those found in the schematic sequence display menu (see section 5.1.4). These should be used to inform RDP5 that you are satisfied with the description of particular recombination events within specific sequences or groups of sequences so that it does not re-evaluate these during subsequent rescans (See section 10.4 for how and why accepting and rejecting sequences is done).

The "Make [sequence name] the [major/minor] parent" options let you manually assign major or minor parental sequences. Use them if you feel you are able to identify better candidate parental sequences than those which were automatically identified by RDP5. You should, however, be very careful when manually choosing "better" parental sequences. In some cases, such as when recombination events are very old or have occurred between very closely related sequences, a recombination signal can completely disappear because the sequences you assume are parental are in fact not the best pair of sequences for identifying the recombination event. This could be due to many different factors but most commonly can be attributed to misleading inaccuracies in the trees used by you to identify the parental sequences.

Before you go ahead and select an alternative parental sequence or group/ungroup recombinant sequences, the "Recheck plot with [sequence name] as recombinant/minor parent/major parent" option can be used to test what a recombination signal would look like if one of the sequences in the currently selected sequence triplet (i.e. either the red, green or blue highlighted sequences in the tree) were replaced with the sequence the mouse pointer is over. These options can also be particularly useful for determining whether RDP5 has over- or under-grouped sequences it thinks have descended from a common recombinant ancestor (See step 10 in section 10.4 of the step-by-step guide).

The "Go to [sequence name]" option will centre the graphical representation of the sequence that the mouse pointer is over in the schematic sequence display (Fig 2).

At the bottom of the side-by side tree display is a button labelled "Run tests". Pressing this button will run Shimodaira-Hasegawa and approximately unbiased tests that compare the topologies of the trees

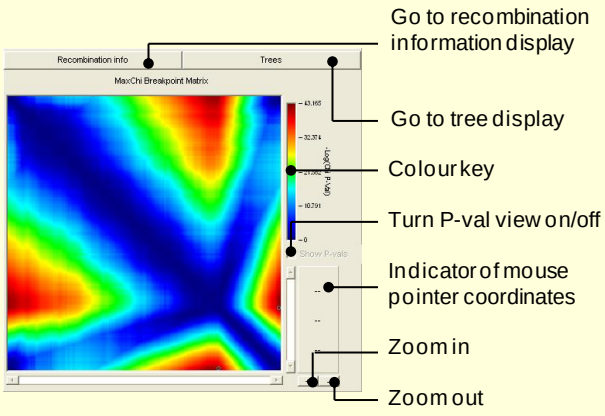

Figure 7. The matrix display. Although many different matrices can be constructed with RDP4, most of the matrix types can only be accessed once an automated recombination analysis has been completed. Moving the mouse pointer over the matrix window and right clicking will provide a range of addition options, including those to change the matrix type, change its colour scheme and save the matrix to a graphics file. For large alignments it might be necessary to enlarge the matrix with the "Zoom in" button to see sufficient detail. The X and Y coordinates of the mouse pointer and the value depicted in the matrix beneath the mouse pointer are given in the panel beside the matrix.

on the left and the right of the side-by side tree display. p-values <0.05 for both of these tests should be interpreted to mean that the topologies of the trees are probably significantly different from one another. Note, however, that the trees in the different panels of the tree display are expected to almost always have significantly different topologies. Further, absence of evidence for significantly different tree topologies is not evidence that the tree topologies are the same –i.e. it is not evidence that recombination has not occurred. It simply means that there is an absence of phylogenetic support for a particular recombination event having occurred.

5.6 The Matrix Display

Pressing the "Matrix" button either above the recombination information display or in the command button panel at the top of the screen (Figs 1, 3 and 6) will result in the recombination information display being replaced by the matrix display.

A number of different matrix types can be drawn in this display. You may select the matrix type that you would like to view by either right clicking in the matrix display and selecting the "Change matrix type" option or by clicking on the small arrow beside the matrix button in the command button panel (Fig. 1). For a brief description of all the different matrix types see section 9.3.

Other options that are available on the menu are to "Copy" the matrix to the clipboard, "Save to .bmp file" and "Save to .csv file." The latter option will save information on each cell within the matrix to a spreadsheet that can be opened in programs like Excel or Open Office. The "Change colour scheme" option allows you to change the scheme used to express the range of cell values presented in the matrix.

If a MAXCHI or LARD matrix is being displayed, two additional menu options, "Place breakpoint here" and "Place ancestral breakpoint here," are offered whenever the right mouse button is pressed. If the former option is selected then the breakpoint positions of the recombinant being analysed will be changed to the X,Y coordinate positions at the tip of the mouse pointer – these coordinates are displayed to the right of the matrix display. If the latter option is selected then the breakpoint positions of every sequence carrying evidence of the same recombination event will be changed along with the currently selected recombinant (see points 1-4 in section 10.4 of the step-by-step guide to using RDP5 for information on when/why breakpoints should sometimes be adjusted).

Table 1. The different recombination detection and analysis methods available in RDP5

Method	Implementation	Identifies Recombinants	Estimates Breakpoints	Estimates Regions	p-Value Calculation	References
Original RDP method	RDP5	+	+	+	Binomial distribution	Martin and Rybicki, 2000
GENECONV	RDP5 & GENECONV	+	+	+	Blast-Like Karlin-Altschul & Permutation	Padidam *et al*., 1999
BOOTSCAN	RDP5 & PHYLIP	+	+	+	Bootstrapping & binomial distribution & χ^2	Salminen *et al*., 1995
Maximum χ^2	RDP5	+	+	+/-	χ^2 & Permutation	Maynard Smith, 1992
CHIMAERA	RDP5	+	+	+/-	χ^2 & Permutation	Posada and Crandall, 2001
Sister Scan	RDP5	+	+	+	Permutation and Z-Test	Gibbs *et al*., 2000
3SEQ	RDP5	+	+	+	Exact test	Lam et al., 2018
LARD	LARD	-	+	-	Likelihood ratio	Holmes *et al*., 1999
Distance Plots	RDP5 & PHYLIP	-	+	+	-	-
PhylPro	RDP5	+	+	-	-	Weiller, 1998
DSS/TOPAL	RDP5, PHYLIP & SEQ-GEN	-	+	-	Parametric bootstrap	McGuire and Wright, 2000
VisRD	RDP5	+	+	+	-	Lemey *et al*., 2009
BURT	RDP5	-	+	+	-	-

6 SAVING RESULTS AND RECOMBINATION FREE DATASETS

Besides the various save options that are provided when the right mouse button is clicked while the pointer is over particular display panels (which enables images of trees, matrices, plots and other graphics to be either saved in various formats or copied and pasted into other programs), RDP5 has two different classes of analysis outputs that can also be saved following a successfully completed automated scan for recombination:

(1) For people who are interested in recombination, analysis results depicting the recombination events that are evident within a dataset can be saved in one of two different formats by pressing the "Save" button at the top of the program screen. Results saved in an RDP5 project file (a file with a ".rdp5" extension) can be reloaded at a later date for further study using RDP5. Saving results to a .csv file (a text file that can be read with a spreadsheet program like Excel) will give you a tabulated summary of all of the unique recombination events that the program has detected. In order for different fields of the text file to be read correctly by a spreadsheet program (such as Excel) you may need to specify when loading the file that columns are delimited by commas. Note that for versions of RDP before 2.0 columns were delimited by TABS and for versions before 1.07 the columns were delimited by spaces.

(2) For people who are mostly interested in removing evidence of recombination from their analysed datasets, recombination-free alignments can be saved by right clicking on the sequences in the schematic sequence display (Fig 1). Alignments can be saved in a variety of different formats with recombinant sequences completely removed, with the bits of recombinationally derived sequence removed (the recombinationally derived bits are replaced by the "gap" character, "-"), or with the recombinant sequences split into their constituent parts (the distributed alignment option). For this latter option each recombinant sequence is "decomposed" into two or more different sequences (a sequence with one detected event will be split into two sequences, one with three detected events into three sequences and so-on) each with gap characters added to ensure that the nucleotides they retain remain aligned.

7 SUPPLEMENTARY ANALYSES

RDP5 allows you to "check" results obtained with any particular method using the original RDP method, GENECONV, BOOTSCAN/Recsan, MAXCHI, CHIMAERA, SISCAN, LARD, 3SEQ, BURT, distance plots VisRD and TOPAL/DSS. To select a method for checking results press the button in the "Check using" section of the plot display (Fig 4). The list of methods that can be used to check a result will be displayed and you can select whichever one you want.

It is recommended that once a recombinant region has been identified and appears to represent evidence of a genuine recombination event (i.e. there is evidence from at least two different analysis methods that a particular sequence has a recombinant origin), you should both carefully examine whether RDP5 has correctly identified breakpoint positions in the recombinant sequence(s) and check whether it has not over- or under- grouped recombination signals when it has tried to work out how many unique events account for the recombination signals in the alignment. See section 10.4 for a detailed walk-through of how various supplementary analyses can be used to check the accuracy of automated RDP5 results.

Other supplementary analyses that you can do in RDP5 following an automated exploratory scan for recombination are the construction of **recombination breakpoint distribution plots** (these are useful for identifying recombination breakpoint hotspots; see section 9.1), **recombination rate plots** (parametric approximation of variations in recombination rates across an alignment that can also be used to identify recombination hotspots; see section 9.3), **recombination event maps** (a simple graphical over-view of all the unique recombination events detected; see section 5.3), **tests of recombination induced protein/nucleic acid folding disruption** (see sections 9.5 and 9.6), **recombination region count matrices** (a more complex overview of the unique events detected indicating how often different parts of the analysed sequences are separated from one another by recombination; see section 9.4.4), **recombination breakpoint matrices** (useful for telling whether specific breakpoint pairs tend to occur together; see section 9.4.5), **recombination matrices** (an overview of recombination expressing the bits of sequence exchanged in terms of the relatedness of parental sequences; see section 9.3.2), and **modularity matrices** (useful for identifying bits of sequence that always tend to be co-inherited from the same parental sequence; see section 9.4.3).

8 RECOMBINATION SIGNAL DETECTION METHODS

RDP5 provides access to seven primary exploratory recombination signal detection methods (Table 1). These (named after the programs first implementing them) are the original RDP method, GENECONV, BOOTSCAN/RECSCAN, MAXCHI, Cimaera, 3SEQ and SISCAN. An additional five, supplementary/secondary methods can be used to check how accurately recombinant regions or breakpoints have been detected by the primary exploratory methods. These methods (also named after the programs originally implementing them) are LARD, PHYLPRO, VisRD, distance plots and TOPAL/DSS. The exploratory methods can be used to scan alignments for recombination signals and/or, following the completion of a scan, to check the validity of the results produced by other detection methods. In this section a brief description will be given of the twelve methods (for additional information please consult the supplementary material indicated).

8.1 The RDP Method

8.1.1 The method. The original RDP method (Martin and Rybicki, 2000) screens multiple sequence alignments for evidence of recombination by examining every possible sequence triplet using a three-step procedure (Fig 8 A) as follows:

1. Within each unique set of three sequences (or triplet) sampled from an alignment all phylogenetically uninformative sites are discarded. Given an UPGMA dendrogram constructed from the full alignment, in any particular triplet there will be two sequences, A and B that are more closely related to one another than to a third sequence, C. Non-informative sites are those that are identical in all three sequences, different in all three sequences, or (if reference sequence settings are used) are not present in any member of a group of reference sequences.
2. A window is moved along the alignment of informative sites one nucleotide at a time, and an average percentage identity for each of the three possible pairs is calculated at each position (Fig 8 B). Potential recombinant regions are identified as regions where the percentage identity of A-C or B-C is higher than that of A-B.
3. In a potential recombinant region the probability that a particular run of nucleotide identities occurred by chance is approximated using the binomial distribution. A p-value is calculated from this probability by multiplying it by the number of unique windows examined. A multiple comparison corrected (or Bonferroni corrected) p-value is calculated from this p-value by multiplying it by the total number of triplets examined within the alignment.

Once a potentially recombinant region has been detected it remains to be determined which of the three sequences is recombinant and which are "parentals." This is achieved using the approach outlined in section 4.1.4.

8.1.2 Potential problems. Depending on the method of reference sequence selection that is used, the RDP method may not be able to analyse certain sequence triplets in an alignment for recombination. If, for example the, "use only internal references" setting is used, the RDP method will not analyse triplets that are one another's nearest relatives. Also, given an alignment of 4 sequences with this setting, the RDP method will not be able to examine any of the three possible sets of triplets unless the UPGMA for the alignment has the appropriate branching pattern. For small alignments, you should therefore always use either the "internal and external references" or the "no references" settings.

You should note that the original RDP algorithm has no way of explicitly handling rate variation across lineages (leading to non-ultrametric/non-clock-like trees – ie. Trees where different sequences in the alignment appear to be evolving at vastly different rates). There is a real possibility that if either sequences are evolving at different rates or have been sampled at different times), the part of the method that relies on UPGMAs (selection of reference sequences) will not function the way it was intended to. For such datasets you should use either the "use internal and external references" or "use no references" settings (the latter is the default).

Because the method only uses informative sites it should be fairly insensitive to differences in the rate at which different regions of a sequence are evolving. It does, however, have no way of explicitly handling unusual nucleotide compositions or extreme variations in different types of nucleotide substitutions – ie the RDP method does not apply any substitution models during the calculation of distances. This is not a problem except that: (1) Extreme differences in certain types of substitution may obscure the evidence of recombination that

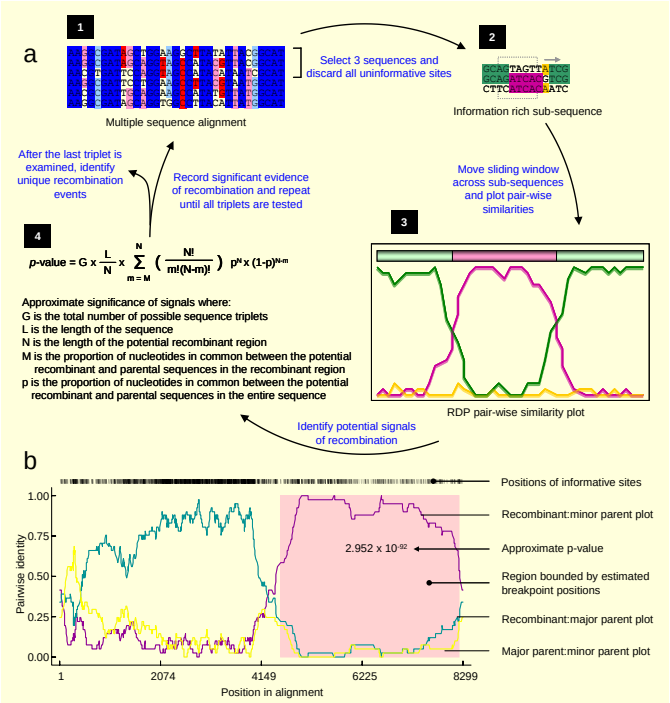

Figure 8. The original RDP method. (a) The automated analysis procedure. (b) An example RDP pairwise identity plot. This is the same example recombination signal that is depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

the RDP method requires and (2) extreme nucleotide compositions may compromise the (already imperfect) validity of the p-values that the RDP method calculates.

Finally, the different reference sequence settings have vastly different powers to detect recombination and can have quite different false positive rates. The most powerful option is to use no reference sequences but this setting is also more prone to false positive inference of recombination than the other settings. While the false positive rate with this setting is low enough that it should never be a problem for datasets with moderate to low diversity (ie such as when all sequences in an alignment share >70% identity), it is not advisable to rely exclusively on this method for the detection of recombination in alignments with highly diverged sequences.

8.2 GENECONV

8.2.1 The method. GENECONV (Padidam *et al*., 1999; Sawyer, 1989) looks for regions within a sequence alignment in which sequence pairs are sufficiently similar to suspect that they may have arisen through recombination (Fig 9 A). Note that the method used for triplet scanning (used in exploratory analyses) is identical to that used for pair scanning (used in manual analyses) except that instead of analyzing the entire alignment the triplet scan splits the alignment up into every possible alignment of three sequences and analyses each of these alignments separately. The basic procedure is as follows:

1. Monomorphic sites are excluded from the alignment as a control for constant or highly selected sites. What remains is an alignment of polymorphic sites.
2. For every possible sequence pair in the alignment, regions are found that are either (a) identical and unusually long for that pair of sequences or (b) have an unusually high degree of similarity. Similarity is scored based on a scheme where (a) matches (or concordant sites) count as +1 and (b) there is a penalty for mismatches (or discordant sites). The mismatch penalty depends on the density of polymorphic sites between the two sequences and on a user-specified mismatch intensity parameter or G-scale (Fig 9 A).
3. p-values are assigned to high scoring regions (Fig 9 B and C; also called fragments, high scoring aligned pairs or HSAPs). The p-values assigned to these regions are derived through (a) permutations (slow but accurate) and/or (b) a BLAST derived Karlin and Altschul (KA, 1990) method (approximate but fast). Although approximate, multiple comparison corrected (also called Bonferroni corrected or global) KA p-values are generally far more conservative than permutation p-values. Multiple comparison correction simply involves multiplication of pair-wise KA p-values by the number of pair-wise comparisons made during an analysis.

For additional information on the GENECONV algorithm please consult the GENECONV manual. It can be obtained online from: http://www.math.wustl.edu/~sawyer/geneconv/

As with the RDP method, parental and recombinant sequences are identified using the approach outlined in section 4.1.4.

8.2.2 Potential problems. GENECONV works with polymorphic sites that are determined based on the entire alignment. When doing pair-wise or triplet scans (but particularly with pair-wise scans) A single highly diverged sequence in an alignment will therefore introduce many polymorphic sites that are irrelevant for detection of recombination amongst sequences in the alignment that are more similar to one another. These irrelevant sites can have two effects on an analysis: (1) They could result in apparently significant runs of concordant sites when GENECONV is examining closely related sequences (these runs will be interpreted by GENECONV to be recombinant regions). (2) By needlessly increasing the number of polymorphic sites they will decrease the apparent significance of p-values and could result in small (but genuine) recombinant regions being missed. Before doing an analysis with GENECONV care should therefore be taken either to carefully select sequences at the alignment stage or to disable potentially problematic sequences in RDP5. If you notice that results obtained with other methods can be confirmed when you do a GENECONV check but that GENECONV did not detect these results during the automated pair-wise analysis it is very likely that GENECONV had a problem with the structure of the dataset. Doing a triplet scan instead of pair-wise scan may solve this problem.

When analysing GENECONV derived results you should also always be very cautious when accepting evidence that recombination has occurred between two sequences that are one another's nearest relatives. It is always possible that a run of conserved sites between the sequences has been misinterpreted as being evidence of recombination.

The final problem with GENECOV is that simulations have revealed that it has the lowest recombination breakpoint detection accuracy of the seven methods that can be sued to automatically screen for recombination in RDP5. Always recheck the positions of recombination breakpoints detected with GENECONV with those detected by the MAXCHI and CHIMAERA methods (the most accurate breakpoint detection methods implemented in RDP5)

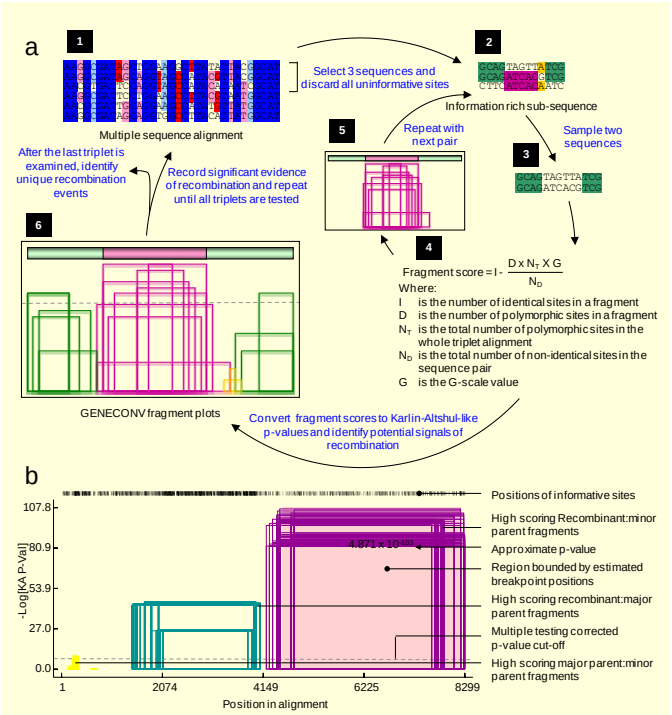

Figure 9. The GENECONV method. (a) The automated analysis procedure. (b) An example plot of high scoring fragments. This is the same recombination signal depicted in Figures 8 to 19. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.3 BOOTSCAN/RECSCAN

8.3.1 The method. BOOTSCAN is a sliding window method that was developed to identify the parental origins of sequence blocs within know or suspected recombinant sequences (Salminen *et al*., 1995). In its original implementation BOOTSCANning involved: (1) Construction of an alignment containing a potentially recombinant sequence and a set of (non-recombinant) reference sequences. (2) Moving a window of set length along the alignment a set number of nucleotides at a time and calculating a bootstrapped neighbour joining tree for each window. (3) Plotting the relative bootstrap support for nearest neighbour groupings of the potentially recombinant sequence with each of the reference sequences at each window position. Whereas non-recombinant sequences should group (with an excess of ~70% support) with a single reference sequence across its entire length, recombinant sequences should group alternatively (with an excess of ~70% support) with two or more different reference sequences. With recombinant sequences the midpoint between the transition of high bootstrap values grouping it with one reference sequence to high values grouping it with another, should approximate the recombination breakpoint. The reference sequences with which the recombinant is alternatively grouped are assumed to be the parental sequences.

Although RDP5 implements this type query vs reference scan with its "Manual BOOTSCAN" method, the exploratory BOOTSCAN RDP5 uses to automatically search for recombination signals (called RECSCAN in Martin *et al*., 2005b) (Fig 10 A) differs from that described above in that it requires no prior identification of recombinant and non-recombinant sequences. The RDP5 BOOTSCANning procedure involves the following steps:

1. A window of set size is moved along the alignment a specified number of nucleotides at a time
2. Bootstrap replicates of each window are constructed and pair-wise distances are calculated that can either themselves be used for a pair-wise distance BOOTSCAN or they can be used in a UPGMA or neighbour joining tree BOOTSCAN.
3. At each window position the relative grouping (based either on pair-wise distances or tree positions) of every possible sequence triplet in the alignment is determined over all bootstrap replicates. Nucleotide sequence distances, and trees are all produced using recoded versions (in dna.dll) of the PHYLIP components DNADIST, and NEIGHBOR (Felsenstein, 1989).
4. Following completion of the last window in the scan, stored bootstrap data on pair-wise sequence relationships in every possible sequence triplet over all windows, is scanned for alterations in relative bootsrap support for sequence pairs. High degrees of bootstrap support alternating between two different sequence pairs (Fig 10 B) are indicative of potential recombination events.
5. Either binomial (see 8.1) or χ^2 p-values (see 8.4) can be calculated for identified regions.

As with the RDP5 method, parental and recombinant sequences are identified using the approach outlined in section 4.1.4.

8.3.2 Potential problems. A major problem with this and other implementations of BOOTSCAN is that there is no defined "appropriate" level of bootstrap support above which one should have a high degree of confidence that detected regions are recombinant. It i s, for example, widely accepted that 95% support for sequences A and B being more closely related in region 1 and 95% support for sequences B and C being more closely related in region 2 does not equate with 95% confidence that a recombination event has occurred. Binomial and χ^2 p-values can be used to identify which identified regions are significant. Also, although bootstrap values are generally conservative indicators of significance (and there is therefore a good chance that many real recombinant regions will be missed with a bootstrap cutoff of, for example, 95%) there is no obvious way of correcting bootstrap values for multiple testing. This means that relying entirely on bootstrap scores in analyses of large datasets can potentially yield a lot of false positives.

Another problem with all implementations of BOOTSCANning is that they require fixed window sizes. This is a problem for two reasons: (1) In situations where nucleotide substitution rates vary widely along the length of sequences, the information content of different windows will vary greatly. This may, for example result in a 95% bootstrap cutoff being far more meaningful in parts of the alignment with a lot of sequence variation than in parts of the alignment where there are only a few variable nucleotides per window.

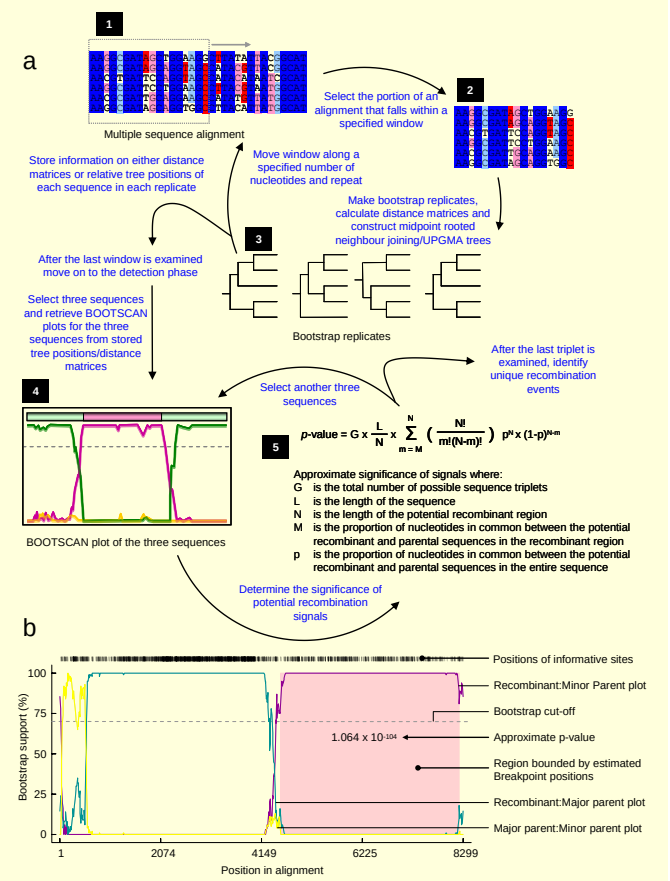

Figure 10. The BOOTSCAN method. (a) The automated analysis procedure. (b) An example BOOTSCAN plot. This is the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

(2) In regions of an alignment with a lot of variability, small recombinant regions are a lot more easily detectable than in portions of an alignment with low variability. Setting the window to accommodate areas of an alignment with low variability (ie making it larger) will often result in smaller, otherwise easily detectable recombinant regions in areas of high variability being missed. The obvious solution to this problem is that care should be taken in the construction of alignments that are to be analysed by BOOTSCANning. Areas of a sequence that differ greatly in their variability should simply be analysed separately.

8.4 MAXCHI

8.4.1 The method. Maynard Smith (1992) proposed a method (called the maximum χ^2 method) for identifying recombination breakpoints that has since been implemented by David Posada in the program MAXCHI (Posada and Crandall, 2001). Given an alignment MAXCHI examines sequence pairs and seeks to identify recombination breakpoints by looking for significant differences in the proportions of variable and non-variable polymorphic alignment positions in adjacent regions of sequence (Fig 11 A). The method involves the following steps:

1. All monomorphic sites in an alignment are discarded. What remains is an alignment of polymorphic sites. There is also an option to discard sites that contain gaps.
2. For every possible sequence pair in the alignment, a window of set length with a partition at its center is moved along the sequences one nucleotide at a time.
3. At each window position a 2 x 2 χ^2 value is calculated as an expression of the difference in the number of variable sites between the pair of sequences on either side of the central partition. When plotted along the length of the alignment, peaks in these χ^2 values (Fig 11 B) indicate potential recombination breakpoints.

Although the maximum χ^2 method performs best when only two parental sequences and a recombinant sequence are compared, it is possible to use the method to examine alignments of more than 3 sequences. RDP5 can be set to either examine three sequences at a time (the "scan triplets" setting) or examine pairs of sequences with variable site positions being determined from the entire alignment (implemented in the "manual" MAXCHI scan).

MAXCHI provides information on the positions of potential breakpoints but does not give information on the extent of recombinant regions. When the "scan triplets" setting is used RDP5 will make (a rather crude) attempt to match potential breakpoints and will assume that sequences between matched breakpoints are within a single recombinant region. To match breakpoints RDP5 uses the following procedure:

1. All χ^2 peaks in all 3 pair-wise χ^2 plots along the length of the alignment are identified.
2. Centering a window partition on the highest χ^2 peak RDP5 incrementally increases the window size by one nucleotide on either side of the partition until χ^2 values begin to drop.
3. The window size yielding the maximum χ^2 should encompass the entire recombinant region. To determine whether the left or right window encompasses the recombinant region the χ^2 values at the extreme ends of the windows (determined during the first scan) are compared and the side corresponding with the highest peak is assumed to be the recombinant region.
4. The process is repeated using the next largest χ^2 peak until no significant χ^2 peaks remain.

Parental and recombinant sequences are identified using the approach outlined in section 4.1.4.

Along with CHIMAERA (section 8.5), MAXCHI is one of the most accurate breakpoint detection methods implemented in RDP5.

8.4.2 Potential problems. When scanning triplets the most serious problem with MAXCHI is that it will include sites that vary between all three sequences in the analysis. In my hands this has only been a problem when examining very diverged sequence triplets. In these situations MAXCHI will yield what appear to be some quite convincing recombination signals that turn out to be false positives – i.e. results with high p-values that cannot be confirmed with any other recombination detection methods.

As with GENECONV, pair-wise scans that are perfored duing manual MAXCHI analyses use the entire alignment to identifying variable sites. If alignments contain sequences that are both highly diverged and very similar, MAXCHI will occasionally give false negative results for otherwise easily detectable events between closely related sequences. In these situations the datasets should be split (either at the alignment stage or in RDP5 by disabling sequences) into, for example, groups containing only sequences from the same species, a group containing one representative sequence from different species, a group containing one representative sequence from different genera etc.

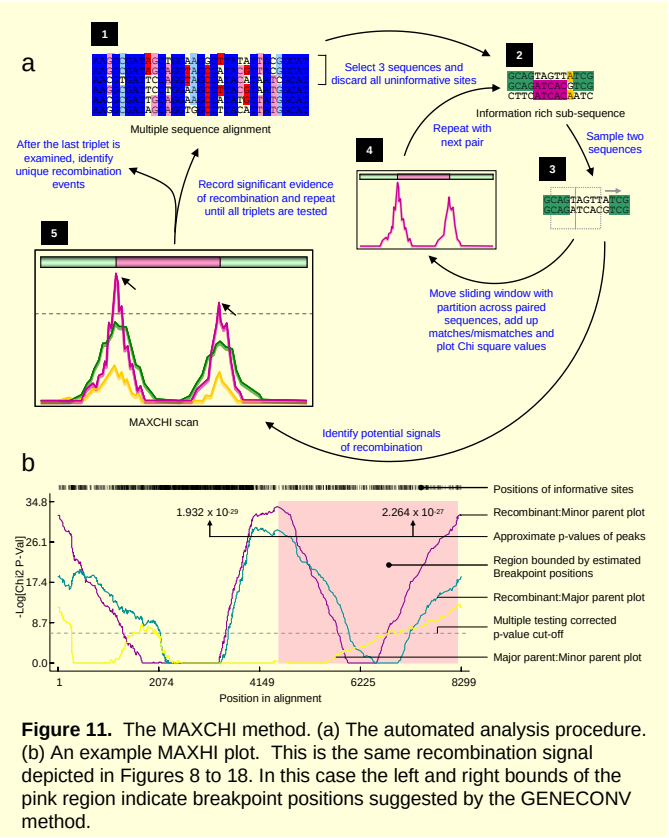

Figure 11. The MAXCHI method. (a) The automated analysis procedure. (b) An example MAXHI plot. This is the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.5 CHIMAERA

8.5.1 The method. CHIMAERA is David Posada's modification of Maynard Smith's maximum χ^2 method (see section 8.4). The differences between CHIMAERA and MAXCHI are (1) the way in which polymorphic sites are chosen and (2) CHIMAERA can only be used to screen triplets. Every possible sequence triplet in an alignment is screened. Each sequence in a triplet is in turn examined to determine if it could potentially be the recombinant of the other two sequences in the triplet using the following steps:

1. All monomorphic sites and sites at which neither of the two "parental" sequences matches the selected "recombinant" sequence are discarded. The three sequences are compressed into a linear string of 1's and 0's with 1 representing a match of the recombinant with one parent and 0 representing a match with the other.
2. A window of set length with a partition at its center is moved along the string of 1's and 0's one position at a time.
3. At each window position a 2 x 2 χ^2 value is calculated as an expression of the difference in the proportion of 1's and 0's on either side of the central partition. When plotted along the length of the alignment, peaks in these χ^2 values (Fig 12 B) indicate potential recombination breakpoints.

As with MAXCHI, CHIMAERA provides information on the positions of potential breakpoints but does not give information on the extent of recombinant regions. RDP5 determines recombinant regions from χ^2 peaks in exactly the same way as it does for MAXCHI (See section 8.4.1). Along with MAXCHI (section 8.4), CHIMAERA is one of the most accurate breakpoint detection methods implemented in RDP5.

Parental and recombinant sequences are identified using the approach outlined in section 4.1.4.

8.5.2 Potential problems. CHIMAERA does not suffer from the same problems as MAXCHI when examining very diverged sequences but, because it relies on matches between parental and recombinant sequences, may have trouble identifying recombination when only one parental sequence is present in an alignment.

Because of the similarities between MAXCH and CHIMAERA it is probably not a good idea to confirm results obtained with the one method with the other – i.e. a recombination signal that was only detectable with the MAXCHI and BOOTSCAN methods would be better evidence of recombination than a recombination signal that was only detectable by the MAXCHI and CHIMAERA methods.

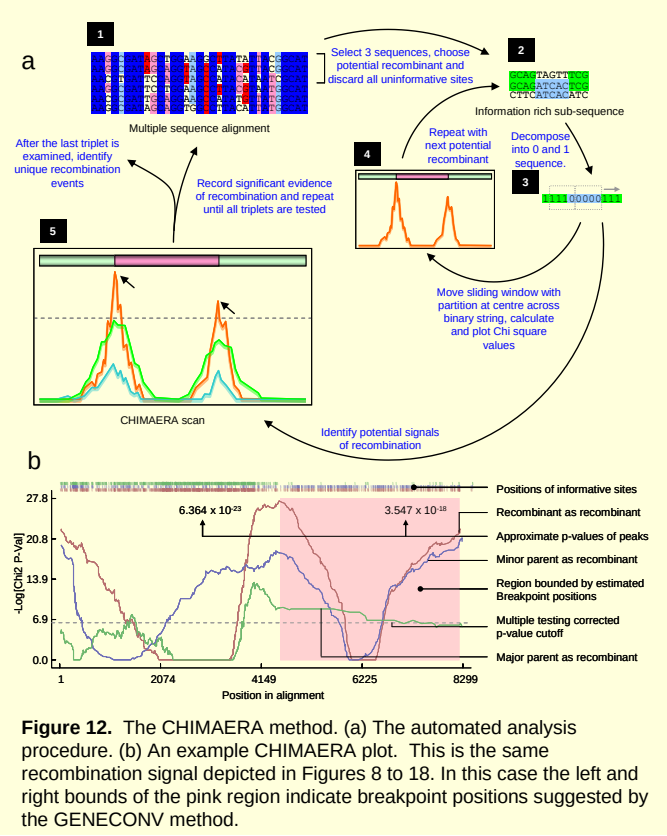

Figure 12. The CHIMAERA method. (a) The automated analysis procedure. (b) An example CHIMAERA plot. This is the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.6 SISCAN

8.6.1 The method. The Sister scanning method was developed by Adrian Gibbs, Mark Gibbs and John Armstrong (2000) as a means of analysing recombination signals in nucleotide sequence data. Every possible triplet in an alignment is examined for evidence of recombination using the following steps.

1. A fourth sequence is either constructed by "horizontal" randomisation of one of the sequences in a triplet or drawn from the alignment (either the most diverged sequence in the alignment or the sequence in the alignment that is most closely related to the three sequences in a triplet but is more distantly related to the three sequences than they are to one another - i.e. is the nearest outlyer).
2. A window of set length is moved along the four sequence alignment a set number of nucleotides at a time. If a randomized sequence is being used, a new randomized sequence (constructed by a process called horizontal randomization which maintains nucleotide content) is produced for every window (Fig 13).
3. Each column of the alignment is sorted into one of fifteen different categories.
4. The nucleotides in each column of the alignment are then randomised (in a process called "vertical" randomisation) to produce a user defined number of permuted alignments. The number of columns falling into the fifteen different categories is determined for each of the permuted alignments.
5. At every window position a Z-test is used to determine whether the number of columns in that window corresponding to any of the 15 site categories differed significantly from those determined for the vertically randomised alignments.

For more information on sister scanning and details on site categories consult Gibbs *et al*., 2000).

Parental and recombinant sequences are identified using the approach outlined in section 4.1.4.

8.6.2 Potential problems. The main problem with SISCAN is that it, like BOOTSCAN (see section 8.2.2), examines all sites rather than just variable sites. Although the method can be set to "ignore" invariant sites, throughout an analysis the SISCAN window size remains constant with respect to the underlying alignment. What this means is that in less variable parts of an alignment (or when less divergent sequences are examined) there may be too few sites per window for the analysis to be effective. Increasing the window size to accommodate less variable regions may solve part of this problem but with larger window sizes recombination signals from small recombinant regions (less than half the window size) will be more difficult to detect. You should try to set window sizes so that each window will, on average, contain between 10 and 20 variable sites.

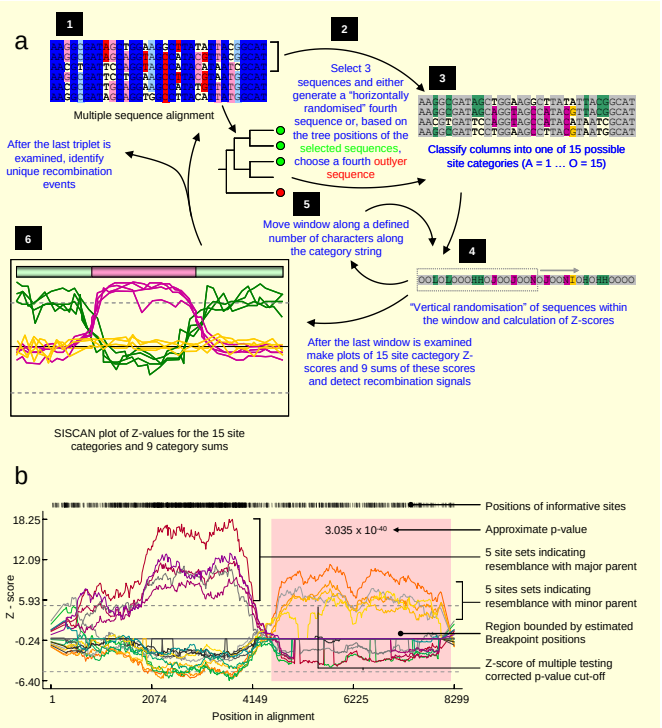

Figure 13. The SISCAN method. (a) The automated analysis procedure. (b) An example SISCAN plot. This is the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.7 3SEQ

8.7.1 The method. 3SEQ is a triplet scanning method (like BOOTSCAN/RECSCAN, RDP, MAXCHI, CHIMAERA and GENECOV) developed by Maciej Boni (Lam et al., 2018). For more information on 3SEQ please consult the 3SEQ user guide available from http://www.cggh.ox.ac.uk/3seq-source/3seq_manual.pdf. As with the MAXCHI, CHIMAERA, GENECONV and RDP methods, 3SEQ focuses only on polymorphic sites within sequence triplets drawn from a larger alignment. The sites that are examined by 3SEQ are, in fact, exactly the same as those examined by CHIMAERA. Each sequence in a triplet is in turn queried to determine if it could potentially be the recombinant of the other two sequences in the triplet using the following steps:

1. All monomorphic sites and sites at which neither of the two "parental" sequences matches the selected "recombinant" sequence are discarded. The three sequences are compressed into a linear string of +1's and -1's with +1 representing a match of the recombinant with one parent and -1 representing a match with the other (Fig 14).
2. Starting at each end of the -1 & +1 sequence a running total of the sum of -1's and +1's is recorded at each new position.
3. The maximum difference in the running total across any two sites in the sequence is then noted together with the distance between the sites.
4. Whereas the sites bounding the maximum change in the running total indicate the most probable positions of potential recombination breakpoints, the difference between the running totals recorded at the sites and the number of nucleotides separating them can be used to either calculate a p-value, or, in the case of the "Big" RDP5 download, read a p-value from a pre-computed p-value table.

The really great thing about the 3SEQ method is that it does not require that a user provide any analysis settings.

8.7.2 Potential problems. As with the CHIMAERA method, 3SEQ relies on matches between parental and recombinant sequences and may have trouble identifying recombination when only one parental sequence is present in an alignment.

Because 3SEQ and CHIMAERA query exactly the same combinations of nucleotide sites when looking for recombination it is probably not a good idea to confirm results obtained with the one method with the other – i.e. a recombination signal that was only detectable with the 3SEQ and BOOTSCAN methods would be better evidence of recombination than a recombination signal that was only detectable by the 3SEQ and CHIMAERA methods.

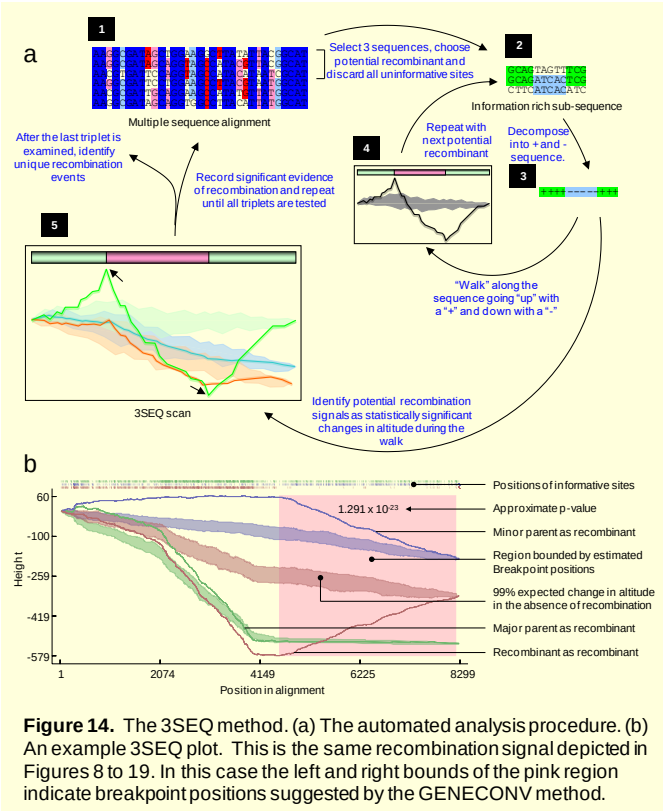

Figure 14. The 3SEQ method. (a) The automated analysis procedure. (b) An example 3SEQ plot. This is the same recombination signal depicted in Figures 8 to 19. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.8 PHYLPRO

8.8.1 The method. PHYLPRO (Weiller, 1998) is one of the few recombination detection methods that directly identify recombinant sequences and it is therefore the basis of a series of tests used by RDP5 for this purpose (see section 4.1.4). Besides forming a core part of all automated recombination scans carried out with RDP5, the PHYLPRO method can also be used to test how accurately other methods have identified breakpoint positions. Because it lacks a computationally simple method for quantifying the significance of potential recombination signals, it cannot, unfortunately, be used for automated exploratory scans of recombination. The method works as follows:

1. As with the MAXCHI and CHIMAERA methods, a window of user-defined width and with a partition at its centre is moved one nucleotide at a time along the length of the alignment.
2. At each position the Hamming or p-distance of every sequence to every other sequence is estimated for each half of the window.
3. For each sequence the distance measures of that sequence to all others in the left hand window are regressed against the corresponding distance measurements from the right hand window and Pearson's regression coefficient (R) is calculated an d recorded.
4. Besides the lowest values of R potentially corresponding with recombination breakpoint positions, the sequence(s) with the lowest value(s) of R at recombination breakpoints are likely to be the recombinants (Fig 15).

8.8.2 Potential problems. The main shortcoming of the PHYLPRO method is that there is no computationally quick way to test the statistical significance of the potential recombination signals that are detectable with the method. Also, as with Bootscan/RECSCAN and SISCAN methods the PHYLPRO method queries all alignment sites rather than just the polymorphic ones. See section 8.6.2 for why this is a problem. I have also not yet determined whether PHYLPRO is more or less accurate at identifying breakpoint positions than relatively accurate methods like MAXCHI and CHIMAERA. Therefore, for checking and adjusting breakpoint positions it is recommended that you should rather use the CHIIMAERA and MAXCHI methods.

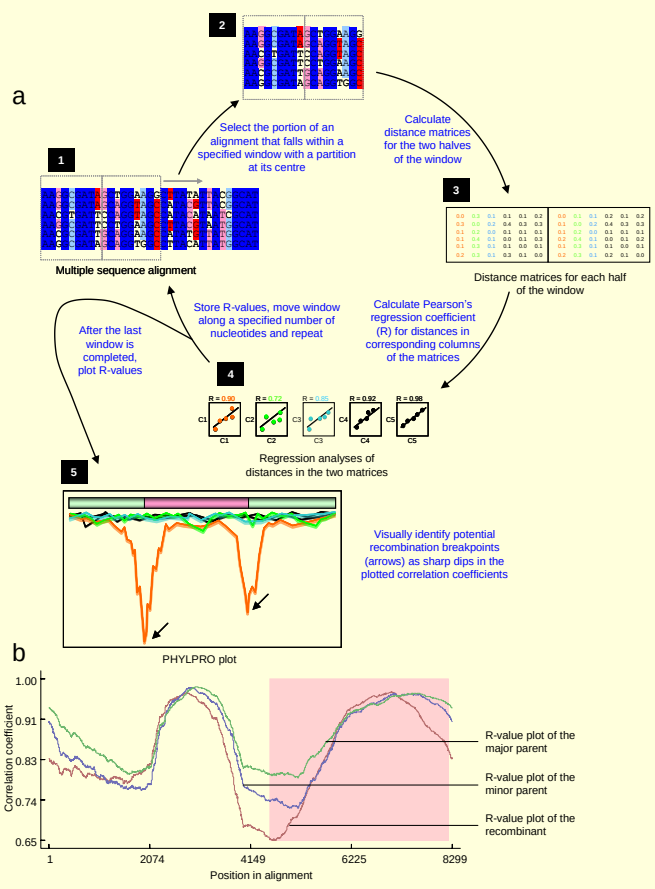

Figure 15. The PHYLPRO method. (a) The analysis procedure. (b) An example PHYLPRO plot. This is the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.9 VisRD

8.9.1 The method. VisRD (Strimmer et al., 2003; Lemey et al., 2009) is one of the few recombination analysis methods (PHYLPRO is another) that directly identify recombinant sequences. As with the PHYLPRO method, VisRD is automatically used behind the scenes for recombinant identification in conjunction with all the automated exploratory recombination signal detection methods. Within RDP5 it is also possible to construct VisRD "Highway" and "Occupancy" plots as checks of the recombination signals detected by other methods. Note that the RDP5 implementation of VisRD is not complete and there are many features available in the program VisRD3.0 (available at http://www.uea.ac.uk/cmp/research/cmpbio/Phyologenetics+Software+-+VisRD) that are not available in RDP5. For more information on the VisRD method see the manual at http://www2.cmp.uea.ac.uk/~vlm/visrd/manual.pdf. The method works as follows:

1. Four sequences (called a quartet) are drawn from a larger alignment and a window of fixed width is moved along these sequences one nucleotide at a time (Fig 16).
2. In every window a four taxon tree is constructed and the topology is plotted on a quartet mapping triangle (something that geometrically expresses in a 2 dimensional space the relative degrees of support for all three of the fully resolved, and all the various partially/unresolved tree topologies that could explain the phylogenetic relationships between the sequences in a quartet) .
3. Changes in the coordinates of points mapped for quartet trees constructed from sequences on either side of recombination breakpoints can be used to indicate which sequence(s) are recombinant. The window positions where support shifts from one fully resolved tree to another can indicate the locations of recombination breakpoints (Fig 16).

8.9.2 Potential problems. The main shortcoming of the original VisRD method (Strimmer, 2003) is that there is no simple way to test the statistical significance of potential recombination signals. Although this has subsequently been addressed in an updated version of the method (Lemey et al, 2009), is still a shortcoming of the VisRD implementation in RDP5. With the new statistical tests introduced by Lemey et al. (2009), VisRD should be implemented in RDP5 in the future as an exploratory recombination screening method.

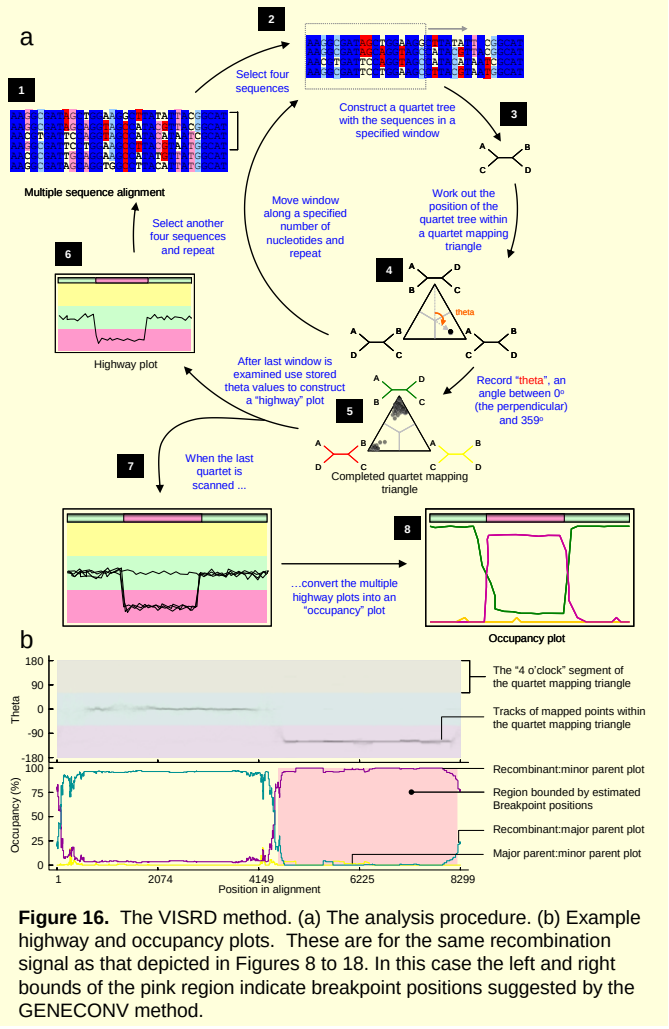

Figure 16. The VISRD method. (a) The analysis procedure. (b) Example highway and occupancy plots. These are for the same recombination signal as that depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.10 LARD

8.10.1 The method. LARD (Holmes *et al*., 1999) detects recombination breakpoints using a method similar to that used by MAXCHI. The method scans an alignment of three sequences (a recombinant and two parental sequences) for the point in the alignment that optimally separates regions of conflicting phylogenetic signal. The method is as follows:

1. A three sequence alignment is partitioned into two pieces and an unrooted maximum likelihood tree is constructed with branch lengths being permitted to vary on either side of the partition.
2. The improvement in likelihood obtained by permitting branch lengths on either side of a partition to vary different (i.e. that they have different branch lengths due to recombination) is assessed with a likelihood ratio test that compares the likelihood of the six parameter partitioned tree with that of a three parameter non-partitioned tree constructed from the same sequences.
3. Every possible partition of the alignment is examined as above and the partition(s) yielding the greatest improvement in likelihood over that of the unpartitioned tree is taken to be the most likely recombination breakpoint(s) (Fig 17).

Unfortunately this method is computationally too slow to automatically screen datasets three sequences at a time for recombination. It is therefore included as a means of checking the parental and recombinant sequence triplets identified by the other methods.

For additional information on LARD please consult Holmes *et al*. (1999).

8.10.2 Potential problems. Although LARD accounts for rate heterogeneity among sites it is unable to distinguish recombination from cases in which a portion of one of the sequences in a triplet is evolving at a greater or reduced rate relative to the corresponding regions in the two other sequences. Note that this is almost certainly a problem with all other recombination detection methods too (it is mentioned here merely because it is the only problem with LARD mentioned by Holmes *et al*., 1999).

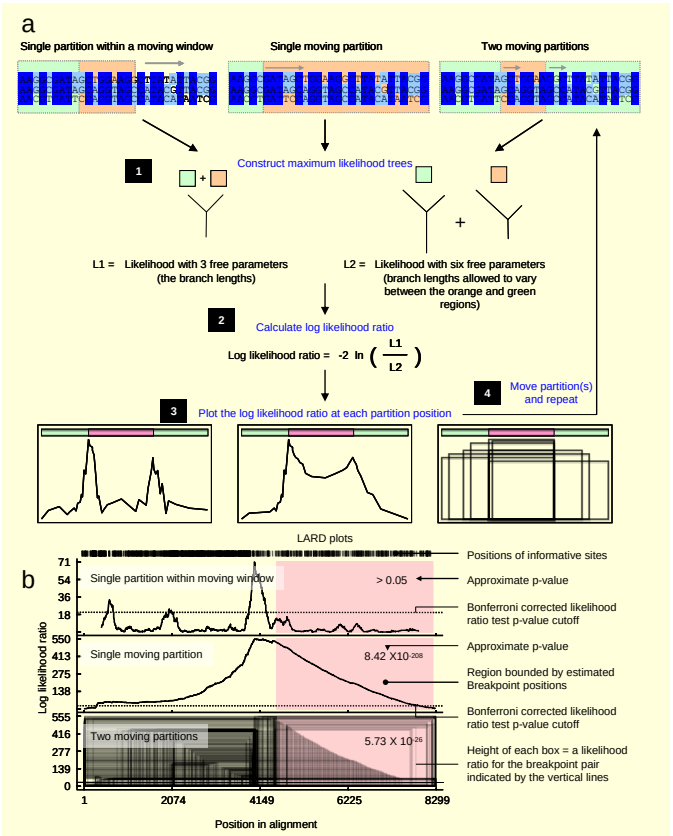

Figure 17. Three variations of the LARD method. (a) The analysis procedure. (b) Example LARD plots for the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.11 DNA Distance Plots

8.11.1 The method. DNA distance plots can be used to provide a graphical description of the relationships between potentially recombinant sequences and their proposed parental sequences. Plots are constructed in the following way:

1. A window of set length is moved a set number of nucleotides at a time along an alignment of the proposed parental and recombinant sequences.
2. Pair-wise distances are calculated for each window using DNADIST (a component of the PHYLIP package) and are plotted against the position in the alignment of the window's centre.

Note that for all distance models other than the "similarity" one, distances are measured in "evolutionary units" which are proportional but not equal to the number of nucleotide substitutions that have occurred between the sequences. I only mention this because at least one user has mistakenly taken DNA distance plot data, subtracted each distance measurement from 1, multiplied by 100 and referred to the resulting plot of "percentage identities" as a similarity plot (as is drawn by the program SimPlot). For additional information on how distances are measured by DNADIST consult the DNADIST documentation from the PHYLIP manual:
http://evolution.genetics.washington.edu/phylip/doc/dnadist.html

8.12 TOPAL/DSS

8.12.1 The method. Given a sequence alignment TOPAL attempts to identify recombination breakpoints by looking for differences in phylogenetic trees constructed from adjacent regions of sequence. It is somewhat of a hybrid between the LARD, BOOTSCAN and MAXCHI methods. TOPAL employs the following approach:

1. A sliding window of set length with a partition at its center is moved along an alignment a set number of nucleotides at a time.
2. At each window position a distance matrix (normalized to that of the entire alignment) is calculated and either a neighbour joining or least squares tree is constructed for the sequences on either side of the partition. Optimal branch lengths are determined by unweighted least squares and the corresponding sum of squares and tree topologies on either side of the partition are recorded.
3. The topologies on either side of the partition are swapped and optimal branch lengths with the forced topology are determined by the unweighted least squares method and sum of squares are recorded.
4. The difference between the sum of squares (DSS) of the forced and unforced tree topologies is recorded for each partition. The higher of the DSS scores for each window is recorded. DSS peaks along the length of the alignment are indicative of potential recombination breakpoints (Fig 18).
5. The significance of DSS peaks can be determined by parametric bootstrapping.

For additional information on the TOPAL algorithm please consult either McGuire and Wright (1998,2000) or the TOPAL manual which can be obtained online from:
http://www.bioss.sari.ac.uk/~frank/Genetics/manual.html

RDP5's implementation of TOPAL that is used for checking the results of automated triplet scanning methods (the original RDP, MAXCHI, GENECONV, BOOTSCAN, CHIMAERA, SISCAN and 3SEQ methods), differs slightly from that described above. In its checking role in RDP5 it is only used with alignments containing three sequences. This would be a problem with the original version of TOPAL because the sum of squares calculation (carried out by the FITCH component of the PHYLIP package) often does not produce any result when using trees with only three sequences in them (this occurs, for example, when two of the three branches have identical lengths). To "solve" this problem RDP5 generates a fourth sequence that is a randomised version of all the sequences in the original alignment. The random number seed used to generate the sequence is the same as that used during the rest of the TOPAL analysis. The fourth sequence is generated by moving along the alignment one nucleotide at a time and randomly selecting a nucleotide from one of the sequences in the alignment at that position.

The reason that TOPAL cannot be used for automated analysis of recombination is that the parametric bootstrapping part of the method (required to infer whether DSS peaks represent significant evidence of a recombination breakpoint) is very slow. If there is any interest in the use of TOPAL for the automated detection of recombination I will attempt to upgrade it from a "checking" method to an "automated screening" method.

8.12.2 Potential problems. I am not sure whether my modification of the original TOPAL algorithm is legitimate. Although the tests that I've run indicate that the modification enables confirmation of results derived using other methods, I have no idea what impact the modification has on the validity or significance of the DSS scores that are calculated.

One problem with generating a fourth sequence is that the sequence is in effect a random recombinant of all other sequences in the alignment. Depending on the number of sequences in the alignment and their relatedness to one another the fourth sequence may not be a suitable "average" of all the sequences in the alignment. If, for example, the alignment contains many sequences that are closely related to one another and a few sequences that are more distantly related the averaged sequence will resemble the sequences in the closely related group more than it should. This may present a problem when using TOPAL to examine recombination between the more distantly related sequences. Note, however, that this problem is specific to the "check using" version of TOPAL and is not a problem with the "Manual TOPAL scan" version which should work in the same way as the original method.

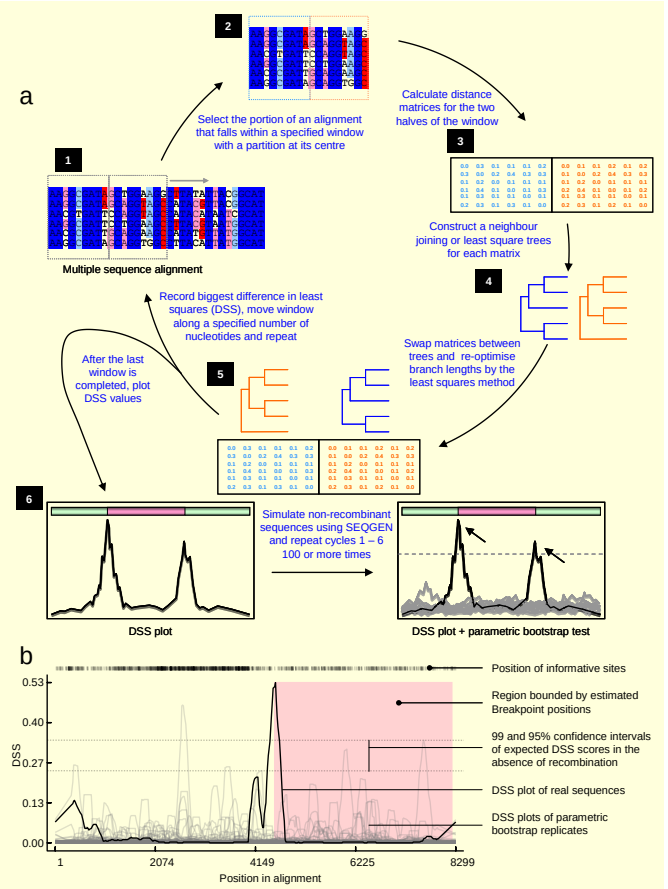

Figure 18. The TOPAL(DSS) method. (a) The analysis procedure. (b) An example DSS plot. This is the same recombination signal depicted in Figures 8 to 18. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

8.13 BURT

8.13.1 The method. Given an alignment of three sequences BURT uses a simple windowless hidden Markov model-based approach to both detect recombination breakpoint positions and, if any of these are identified, determine the approximate confidence intervals associated with their estimated positions (Fig 19). Regardless of the methods that are selected to detect recombination, BURT will be used by RDP5 during automated recombination analyses (whether of the fully exploratory or query vs reference sort) to estimate recombination breakpoint sites whenever the "*polish breakpoints*" setting is used (see section 3.13). Briefly, for every sequence triplet that yields evidence of recombination during the primary recombination screen BURT does the following:

1. All sites within the three sequence alignment are discarded except those at which one of the three sequences differ from the other two.
2. Sites at which sequence 1 & 2 are the same are encoded as "A", sites where sequence 1 & 3 are the same are encoded as "B" and sites where 2 & 3 are the same are encoded as "C".
3. Each distinct recombinant event (corresponding to a hidden state of the HMM) can have a potentially different distribution over A,B,C frequencies, and we use a step up procedure to learn how many hidden states are required to explain the data (ranging from 2 to 20).
4. Viterbi training (which is a fast approximation of the Expectation Maximization algorithm) is then used to estimate model parameters (emission and hidden state transition probability matrices), using 10 random initial conditions to avoid local optima traps.
5. The forwards/backwards algorithm is used to determine the probabilities of individual sites belonging to each of the different hidden states - switches between hidden states occur at recombination breakpoints.
6. The 95% confidence intervals of breakpoint positions are taken as the interval between when emission probabilities drop below 0.95 for one emission state and rise above 0.95 for an alternative emission state.

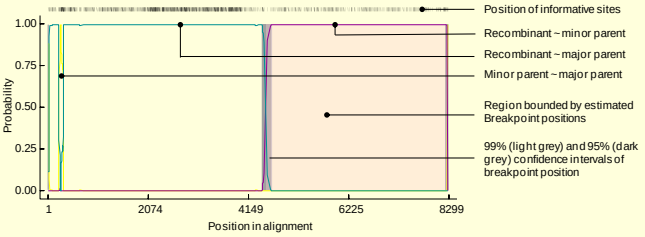

Figure 19. An example BURT plot. This is the same recombination signal depicted in Figures 8 to 19. In this case the left and right bounds of the pink region indicate breakpoint positions suggested by the GENECONV method.

9. SUPPLEMENTARY METHODS

9.1 Breakpoint Distribution Plots (for hot/cold spot detection)

Once a set of unique recombination events has been identified following a automated exploratory recombination scan (see section 4 and section 10 on setting up such scans), it is possible to construct a breakpoint distribution plot either by selecting the "breakpoint distribution plot" option from the "Check using" combo box beside the plot display (Fig. 4) or by selecting the "breakpoint distribution plot" menu option that appears when you press the arrow beside the "Run" button (Fig 1). This plot is made as follows:

1. A breakpoint map is constructed containing the positions of all positively identified breakpoints (i.e. excluding those labeled as "unknown"/"uncertain" during the automated exploratory scan and any subsequent manual checks) for every unique detected recombination event (see section 4 on how these are detected).
2. A breakpoint density plot is then constructed from this map by moving a window of constant user defined width (see section 3.13) one nucleotide at a time along the length of the map and counting all the identified breakpoints falling within each window. Breakpoint counts for each window are plotted at the central window position.

Recombination hot and cold-spots are identified using the following permutation test:

1. Starting with the first recombination event identified, the positions of all variable nucleotide positions (VNPs) between the three sequences used to detect the recombination event are determined. A window of set length is moved a set number of nucleotides at a time along an alignment of the proposed parental and recombinant sequences.
2. The number of VNPs between the breakpoints is counted.
3. The 5' breakpoint position is then randomly changed to one of the VNPs and the 3' breakpoint is placed at a VNP exactly the same number of VNPs away from the randomised 3' breakpoint as the actual 3' breakpoint was from the actual 5' breakpoint. If sequences are linear and the 5' breakpoint either overlaps the end of the sequence or is within 3 variable nucleotide positions of the end of the sequence step (3) is repeated until it is located in a suitable position.
4. Breakpoint positions are then recorded on a linear map of the recombinant sequence.
5. If there is more than one sequence containing evidence of the same recombination event, breakpoints are also recorded on linear maps representing these sequences. These other breakpoints are mapped so that their positions relative to those in the randomized event are preserved.
6. If the newly mapped breakpoints of any of the sequences bound any previously identified breakpoint positions then all the new mapped positions are erased and the process is repeated from step (3).
7. Starting with the next recombination signal identified, the positions of all VNPs between the triplet of sequences used to detect the recombination event are determined and the process is repeated from steps (2) through (6) until the positions of all identified unique events have been randomly shuffled.
8. A breakpoint density plot is generated and stored for these shuffled events. As with the actual breakpoint plots, breakpoint positions labeled as uncertain in the real analysis are recorded but not counted in the shuffled breakpoint map. The maximum number of breakpoint positions detected within a single plot window is recorded.
9. The process repeats from step (1) through (8) however many times is specified under the "Permutations" setting in the "matrices" or "breakpoint distribution plot" options tab (see sections 3.13 or 3.15.5).

Globally significant breakpoint clusters are identified as those windows within the breakpoint density plot that contain more breakpoint positions than the maximum found in more than 95% of the permuted breakpoint density plots. Locally significant breakpoint clusters are identified as those windows at a particular location within the plot that contain more breakpoint positions than more than 99% of windows at the identical location in the permuted density plots.

Although the local test may seem to be more conservative than the global test (due to a higher confidence threshold) one should note that it is in fact considerably less conservative. The reason for this is that whereas the global test is innately corrected for multiple tests, the local test is not.

The design of the permutation test is such that it takes into consideration certain important features of the recombination analysis that might contribute to the incorrect identification of recombination hot- and cold-spots. Probably the most significant of these is that recombination may be far easier to detect in certain parts of an alignment than others. Either too little or too much nucleotide sequence diversity in parts of an alignment can make it difficult or even impossible to detect recombination in these regions. As a result alignment regions of high or low diversity may be incorrectly identified as recombination cold-spots. By permuting the positions of identified recombination events relative to VNPs that are insensitive to the underlying diversity of the sequences used to detect the events (rather than permuting actual alignment positions), the permutation test takes into account variations in breakpoint "detectability" due to variations in l ocal sequence diversity along the length of an alignment.

By explicitly simulating the cyclical detection and signal erasing procedure used to originally identify recombination events (see section 4.1.3), the permutation test also takes into consideration any biases in recombination breakpoint density that may have arisen as a result of this procedure.

9.2 Association Tests

RDP5 can perform two different types of association test to determine whether there is any evidence of breakpoint locations being influenced by specified features of the genomes being analysed.

9.2.1 Binary variable test. This test will indicate whether breakpoints cluster within a pre-specified set of genome sites. These genome sites could, for example, be genes, intergenic regions, secondary structural elements or even just the particular groups of sites within secondary structural elements thata re base-paired. The locations of these genome sites can be specified by either a GenBank file (which must contain a sequence corresponding to one of those included within the dataset being analysed for recombination) or a ORFMap file. When a set of genome regioms have been specified ysing a GenBank file or an ORFMap file RDP5 will automatically test for differences in breakpoint densities between (1) the specified genome regions and the remainder of the genome (if, for example, the specified regions are secondary structural elements then this will test whether there is an association between secondary structures and recombination), (2) different specified genome regions (if, for example, the specified regions are genes this will indicate whether particular genes are more predisposed to recombination than others) , and (3) between the ends of the specified genome regions and the middle of these regions (if, for example, the specified genome regions are different protein domains this test will indicate whether there is a tendency for recombination breakpoints to preferentially fall at the boundaries of these domains; Fig 21; see Lefeuve *et al*., 2009 and Simon-Loriere *et al*., 2010 for examples of how these tests can yield useful information). ORFMap files should be plain text (i.e. not word or rich text format files) with the following structure:

```
[ORF]
Gene_X, 210, 300
Gene_Y, 330, 420
```

RDP5 will interpret this file as Gene_X starting at position 210 and ending at position 300 and Gene_Y starting at position 330 and ending at position 420. When an ORFMap file such as this is loaded by RDP5 it will ask you whether the coordinates are for the currently loaded alignment or whether they instead refer to site positions of a particular sequence within this alignment.

RDP5 will test for associations between breakpoint locations and the genome regions specified in a loaded ORFMap/GenBank file whenever it produces breakpoint distribution plots (see section 9.1). Note, however, that the minimum p-value that can be achieved with these association tests will be determined by the number of permutations that are performed when producing these plots. With 1000 pertmutations the minimum p-value that can be measured will 0.001 whereas the minimum that could be measured with 10 000 permutations would be 0.0001.

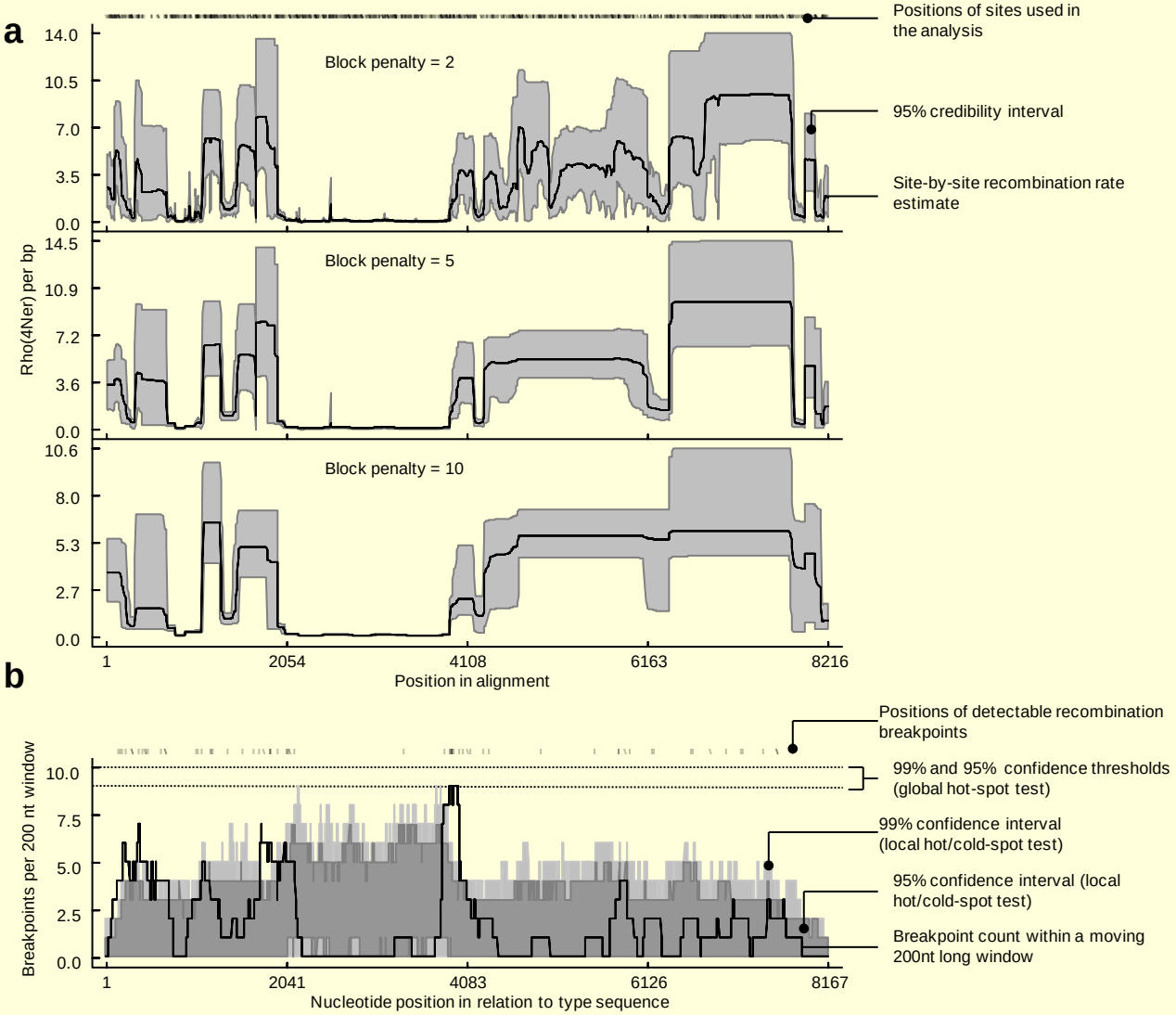

Figure 20. Recombination rate (a) and recombination breakpoint distribution (b) plots. (a) The results of LDHAT INTERVAL analyses with three different block penalty settings (2, 5, and 10; with all other settings identical) are shown to emphasise the influence of this setting on the plots produced. (b) The recombination breakpoint distribution plot for the same data presented in (a). The breakpoint distribution and recombination rate plots have some key similarities and differences that nicely illustrate the fundamental differences between the methods used to produce the plots. Recombination cold-spots in the recombination rate plots should correspond with genome regions containing few detectable recombination breakpoints in the recombination breakpoint plots. Conversely, recombination breakpoint clusters in the recombination breakpoint plots should usually (but not necessarily) correspond with genomic regions that have high recombination rates. However, different degrees of negative selection acting on recombinants in nature means that (1) genomic regions with high estimated recombination rates will not always correspond with recombination breakpoint clusters and (2) genomic regions with few detectable breakpoints will not always have low estimated recombination rates (see Simon-Loriere *et al.*, 2009, Lefeuvre *et al.*, 2007 and Martin *et al* 2005c for examples of how selection can influence detectable recombination breakpoint patterns). Whereas the recombination rate plots in (a) are produced using a model based method that queries over-all patterns of nucleotide substitution and does not rely on the identification of individual recombination breakpoints, the recombination breakpoint distribution plot in (b) provides information on the distributions of actual detectable recombination breakpoints. The key shortcoming of the breakpoint distribution plot is that usually most recombination breakpoints will be undetectable and the analysis therefore only describes the big, easily detectable, recombination signals. While the recombination rate plot incorporates information from more subtle recombination signals that are difficult or impossible to detect individually, its main shortcoming is that it is potentially sensitive to violations of model assumptions (such as random sampling from unstructured populations, all recombination events being neutral) and can only properly be used to describe relatively small (<100 sequences) low-diversity (average pairwise distances <0.1) datasets. Also, the choice of analysis settings can have a large influence on the resultant plots. The data used to produce these plots is the same as that used to produce the matrices in Figure 20.

9.2.2 Continuous variable test. This test can be used to indicate whether recombination breakpoints are associated with, for example, sequence diversity, base pairing probabilities, entropy, GC content or any other measurable variable that might vary along the lengths of the analyzed sequences. RDP5 can read these variables from a plain text SiteSet file (i.e. not in word or rich text format) with the following structure:

[siteset]
1,2032.66
2,2015.32
3,2022,16
..
..
4567,1567.35

RDP5 will associate the value 2032.56 with site position1, 2015.32 with site position 2 etc. and 1567.35 with site position 4567. When a SiteSet file such as this is loaded by RDP5, the program will try to automatically infer whether the site coordinates refer to alignment coordinates (if the largest site-number in the SiteSet file is equal to the alignment length) or whether they refer to coordinates within a particular sequence (if the largest site-number in the SiteSet file is equal to the length of one or more gap-stripped sequences in the alignment). If there are two or more possible sequences that the site-coordinates might refer to then RDP5 will ask you to specify the appropriate sequence.

Once a SiteSet is loaded for a particular variable, RDP5 will test for an association between the variable and recombination breakpoint locations whenever it performs breakpoint distribution plots (see section 9.1). As with the binary variable tests, the minimum p-value that can be achieved with these association tests will be determined by the number of permutations that are performed when producing these plots.

9.3 Recombination Rate Plots (Using LDHat)

RDP5 can function as a graphical user interface for the program INTERVAL (McVean *et al*., 2004) in the LDHat package (McVean *et al*., 2002). INTERVAL can estimate site-by site variations in recombination rates along the lengths of nucleotide sequence alignments and can therefore be potentially used in a similar way to the recombination breakpoint density plots. Care should, however, be taken when interpreting recombination rate plots. It is strongly recommended that if you choose to use these plots you carefully read the LDHat manual at http://www.stats.ox.ac.uk/~mcvean/LDhat/instructions.html. It is particularly important that you appreciate the underlying assumptions made by the analyses performed (such as the random sampling from freely mixing populations of sequences containing predominantly neutral nucleotide polymorphisms) and that you do not over-interpret the output. It is also strongly recommended that you use a range of different block penalty and starting rho settings (see section 3.14) and that you ensure that you have specified enough MCMC updates for the analysis to converge properly.

It is not recommended that you use these plots for any dataset containing any pair of sequences that differ at more than ~10% of their sites because INTERVAL uses approximate likelihood look-up tables that have not been designed to work with datasets containing much more than this degree of diversity.

9.4 Matrices

9.4.1 Ingrid Jacobsen's compatibility matrix. (Jacobsen and Easteal, 1996). A compatibility matrix (Fig 22a) is a graphical representation of the phylogenetic "compatibility" of informative sites in a sequence alignment. Although compatibility matrices in RDP2 could be used to visualise the positions and boundaries of potential recombination events, this is no longer an option on offer in RDP5. Each cell of a compatibility matrix represents a pair-wise comparison between two phylogenetically informative alignment positions (arranged in the order in which they are found along the alignment). If the same tree could be constructed from the nucleotides at both positions assuming the minimum number of substitutions (i.e. if the pair of sites passes the 4-gamete test), then sites are considered compatible and the cell corresponding to those sites in the matrix is shaded white. Matrix entries corresponding to incompatible sites (i.e. pairs of sites failing the 4-gamete test) are shaded black. Using a compatibility matrix, Ingrid Jacobsen's program, Reticulate, will determine whether there is significant evidence of recombination in an alignment using a permutation test. This permutation test is not currently implemented in RDP5 so if you would like to use compatibility matrices to statistically test for evidence of recombination either get Reticulate at **http://jcsmr.anu.edu.au/dmm/humgen/ingrid/ftp/reticulate** or use RDP2.

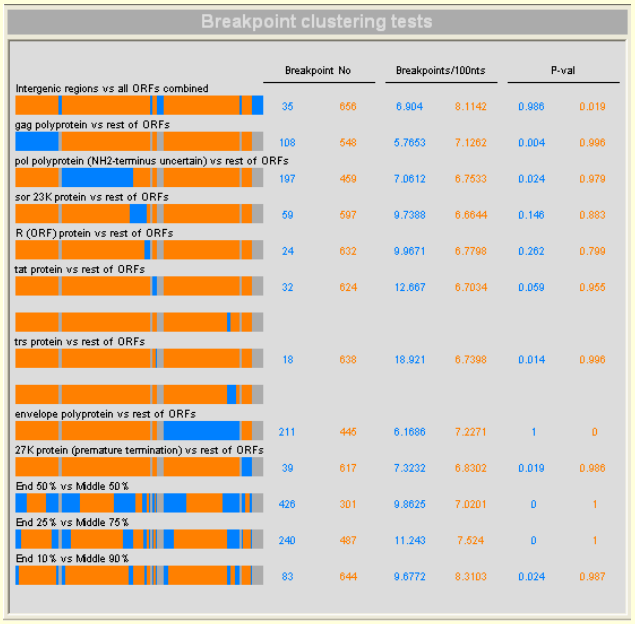

Figure 21. Testing for associations between genome arrangement and breakpoint distributions. Breakpoint clustering is compared between the genome regions represented in blue and orange. In this example (found in the file "HIV Example.rdp") it is clear from the last three rows of the table that that HIV-1M breakpoints tend to cluster far more around the edges of genes (in blue with low associated p-values) than they do within the central parts of genes (in orange).

9.4.2 Robinson-Foulds (RF) compatibility matrix. (Simmonds and Welch, 2006)**.** RF compatibility matrices indicate the degree to which phylogenetic trees constructed from different parts of an alignment differ from one another. Such matrices are therefore useful for visualising the over-all phylogenetic impacts of recombination in a sequence alignment. To produce a matrix such as that represented in the bottom half of Fig 22b, RDP5 moves a sliding window with a specified width (see the *window size* setting in section 3.15.2) along an alignment a specified number of nucleotides at a time (see the *step size* setting in section 3.15.2). For each window RDP5 constructs a neighbour joining tree. Once a tree has been constructed for all windows, RDP5 uses RAxML (Stamatakis, 2014) to determine the normalised Robinson-Foulds distance between each tree and all the others. The Robinson-Foulds distance is simply a measure of how different the branching patterns of two unrooted trees are from one another (i.e. it doesn't look at differences in branch lengths). Whereas a normalised RF distance of 1.0 indicates that the two trees share no bi-partitions (or clades) in common (i.e. the trees are very different branching patterns) a distance of 0.0 indicates that every bi-partition is shared by both threes (i.e. the trees have identical branching patterns). In the RF matrix that is presented in Fig 22b, the blue triangle on the diagonal indicates that whereas trees constructed for nucleotide positions ~2000 to 4000 all have very similar branching patterns. This suggests that recombination breakpoints might be relatively infrequent within this genome region (i.e. it is possibly a recombination cold-spot). However, the matrix also indicates that there are large differences in branching patterns between the trees in this region and those in the remainder of the genome. This pattern reflects the phylogenetic effects of two recombination hotspots (Fig 21) at positions 2000 and 4000 and a recombination cold-spot between these sites.

9.4.3 Shimodaira-Hasagawa (SH) compatibility matrix. (Rousseau *et al*., 2007 ; Shimodaira and Hasagawa, 2001**).** Like RF compatibility matrices (see section 9.4.3), SH matrices can be used to visualise the impacts of recombination on the phylogenetic relationships of the sequences in an alignment. Rather than representing the numbers of topological feature differences between trees constructed from different parts of an alignment (as is the case for RF matrices), SH matrices express degrees of statistical support for differences between trees. To produce a SH matrix such as that represented in the tophalf of Fig 22b, RDP5 moves a sliding window with a specified width (see the *window size* setting in section 3.16.2) along an alignment a specified number of nucleotides at a time (see the *step size* setting in section 3.16.2). For each window RDP5 constructs a neighbour joining tree. Once a tree has been constructed for all windows, RDP5 uses RAxML (Stamatakis, 2014) to maximise the likelihood of all the trees. The nucleotide sequence data used to construct each tree is then swapped with that used to draw every other tree and the likelihoods of each tree+nucleotide sequence data combination are again maximised. The site-specifc likelihoods obtained following these likelihood maximisations are then statistically compared to those obtained for the correct tree+nucleotide sequence data likelihood maximisations using both the approximately unbiased and the Shimodaira-Hasagawa test implemented in the computer program CONSEL (Shimodaira and Hasagawa, 2001). For the SH matrix represented in the upper half of Fig 22b, the blue triangle bounded by the red-box corresponds to the recombination cold-spot in the RF matrix (in the bottom half of the matrix). Note that SH compatibility matrices are more suited to analysing subtle phylogenetic tree differences than are RF matrices. For, example, notice in Fig 22b how only a small proportion of trees that are constructed in the apparent cold-spot region (the blue triangle in the lower RF matrix) are not significantly different from one another in the upper SH matrix (the dispersed blue pixels within the red-triangle indicate these particular tree-pairs): this may indicate that, even in this apparent cold-spot, recombination might still have a measurable impact on the accuracy of phylogenetic trees that are constructed using sequences from this genome region.

It is important to stress here that RF and SH compatibility matrices are not recombination tests. Although useful for visualizing the over-all phylogenetic impacts of recombination, it should always be remembered that recombination is not the only evolutionary process that is capable of causing phylogenetic incompatibility.

9.4.4 Recombination Matrix. This matrix (Fig 22c upper half) is a graphical overview of the recombination events detected during an automated screen for recombination. Only recombination events that are "accepted" (see section 5.1.5) will be added to the matrix. Variations in colour indicate maximum genetic distances between parental sequences exchanging the indicated bits of sequence. It is therefore useful for identifying bits of sequence that always tend to be co-inherited from the same/very similar parental sequences.

9.4.5 Modularity Matrix. This matrix (Fig 22c lower half) is also useful for identifying bits of sequence that always tend to be co-inherited from the same/similar parental sequences. It is essentially a more complicated site-by-site version of the recombination matrix. Whereas the recombination matrix represents recombination events as blocks of colour where the colour represents the relatedness of parental sequences, the modularity matrix delves deeper into the relatedness of parents and represents information on degrees of sequence relatedness within smaller regions of sequence (specified by the windows size setting in the options (see section 3.16.3). As with the recombination matrix the only recombination events represented are those that have been "accepted" (see section 5.1.5).

9.4.6 Recombinant Region Count Matrix. The construction of this type of matrix (upper half of Fig 22e) is described in Lefeuvre *et al*., 2007 and 2009. It is an overview of the unique events detected in an automated recombination analysis (see sections 4 and 10) and indicates how often different parts of the analysed sequences are separated from one another by recombination. Specifically, colours indicate the number of times recombination events have separated pairs of nucleotides. There is also a statistical test associated with this matrix that can be used to indicate whether particular pairs of sites are separated more or less frequently by recombination than can be accounted for by chance (lower half of fig 22e and see Lefeuvre et al., 2009 for a description of this test). The "p-value view" of this matrix can be displayed by clicking on the box besides the "Show values" label to the right of the matrix display. Note, however, that this statistical test should not be over-interpreted. The p-values displayed are not multiple testing corrected. This means that with a p-value cutoff of 0.01 one would expect a 1% false positive rate for any individual pair of sites. All recombination events that are represented in the schematic sequence display (Fig 2; irrespective of whether the events have been accepted or not) will be included when constructing this matrix.

9.4.7 Breakpoint pair matrix. This matrix (Fig 22d) represents the distribution of detectable breakpoint pairs across a set of analysed genomes. It is useful for telling whether breakpoint pairs tend to occur in similar locations. It is, for example, possible that paired recombination hot-spots might facilitate the exchange of discreet modules within genomes such that if a breakpoint occurs at one of the hot-spots, a corresponding breakpoint will generally occur at the other hot-spot. The breakpoints of all recombination events that are represented in the schematic sequence display (Fig 2; irrespective of whether the events have been accepted or not) will be included in this matrix.

9.4.8 Hudson and Kaplan's Rmin Matrix. This matrix (upper half of Fig 22f) displays Hudson and Kaplan's (1985) (over) estimate the minimum number of recombination events (Rmin) separating every pair of nucleotide positions in an alignment. Note that the method underlying this matrix is only applicable to alignments of linear sequences within which recombination events will have involved only single recombination breakpoints i.e. this matrix should not be used to analyse either circular sequences or linear sequences in which breakpoint pairs occur. The Rmin/Distance version of this matrix (see below) is more useful for looking at large changes in RMin that occur over short physical distances (such as would occur in the presence of recombination hot-spots). For extra information on this type of matrix please see the LDHAT manual at http://www.stats.ox.ac.uk/~mcvean/LDhat/instructions.html

9.4.9 Hudson and Kaplan's Rmin/Distance Matrix. This matrix (lower half of Fig 22f) is a distance normalised version of the one above - i.e. it helps visualise large changes in Rmin that occur over short genetic distances (such might occur across recombination hot-spots). I have, however, not had much success using this or the Rmin matrix to demonstrate evidence of recombination hotspots. Also note that, as with the Rmin matrix, this matrix should not be used to analyse either circular sequences or linear sequences between which recombination events have involved breakpoint pairs. For extra information on this type of matrix please see the LDHAT manual at http://www.stats.ox.ac.uk/~mcvean/LDhat/instructions.html

9.4.10 McVean's LD Matrix. This matrix (Fig 22g) indicates pairs of sites that have unusual linkage disequilibrium patterns given the assumption of a constant recombination rate across the analysed genome region. Colours represent marginal likelihood ratios. High marginal likelihood ratios (>4) close to the diagonal are suggestive of recombination hotspots as these indicate a greater degree of deviation from the average recombination rate (i.e. that estimated across the entire alignment) than would be expected if recombination breakpoints were not more likely to occur at some sites than at others. Low marginal likelihood ratios (>-4) close to the diagonal are suggestive of recombination cold-spots (see blue spots in Fig 22, panel G). For extra information on this type of matrix please see the LDHAT manual at http://www.stats.ox.ac.uk/~mcvean/LDhat/instructions.html

9.4.11 MAXCHI Matrix. This is a three dimensional triplet scanning version of the MAXCHI method described by Maynard Smith (1992; see section 8.4 for a description of the MAXCHI method; Fig 23a). A MAXCHI matrix can only be constructed once a specific recombination event is selected within the schematic sequence display (i.e. when you right click on the colored rectangle representing a recombination signal; Fig 2). MAXCHI matrices are useful for identifying the statistically optimal positions of breakpoint pairs (see section 10.4 of the step-by-step guide). Colors represent chi squared values for different pairs of breakpoints. For each represented pair of potential breakpoints three chi squared values are calculated (one for each pair of sequences in the currently selected sequence triplet) to compare patterns of nucleotide similarity between the region bounded by recombination breakpoints and the remainder of the genome. For the given potential breakpoint pair the highest chi square value of all three sequence pairs is plotted.

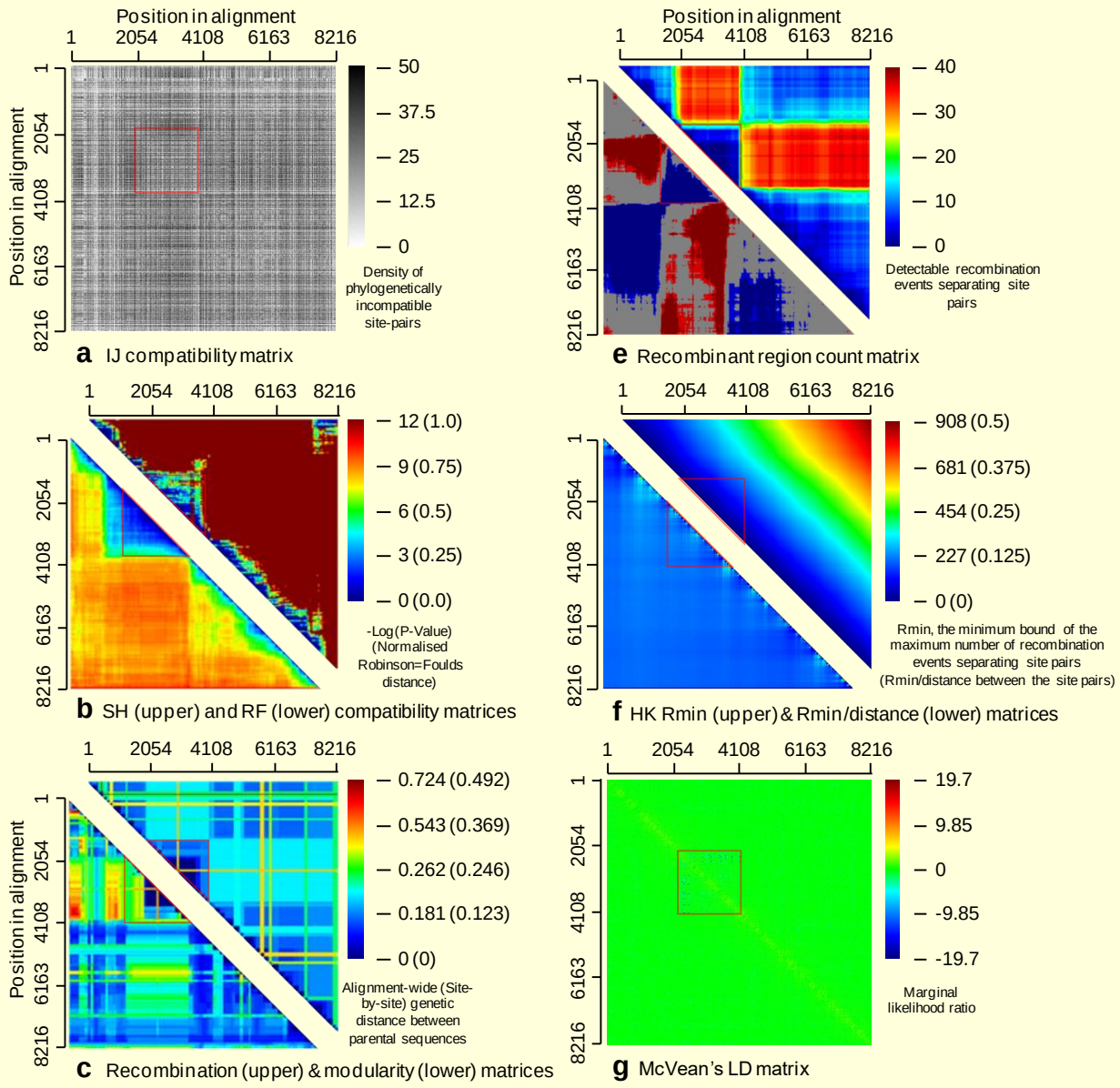

Figure 22. Matrices constructed from the same dataset as that described in Fig 20. Red squares bound corresponding areas on all matrices. The darker regions in the IJ compatibility matrix (a) and the orange/red regions in the SH and RF matrices (b) above, below and to the sides of the red square indicate genomic sites with low degrees of phylogenetic compatibility – a pattern consistent with recombination. The recombination and modularity matrices (c) indicate that recombination events within the more phylogenetically compatible regions of the dataset (e.g. within the red square in (a, and b) tend to have occurred between parental sequences that were quite closely related (indicated by cooler colours in these regions). The breakpoint pair (d) and recombination region count (e) matrices indicate the presence of a recombination breakpoint pair hotspot (note yellow/red patch at the upper right and lower left corners of the red square in (d)) that result in site pairs between the breakpoint positions (indicated by the top left and bottom right corners of the red square) being more frequently inherited from the same parent that can be accounted for by chance if breakpoints were distributed randomly (blue patch within the red square in the bottom half of (e)). Red spots along the diagonal of the HK Rmin/distance matrix (barely discernible in the bottom half of (f)) correspond with recombination breakpoint hotspot positions. In this case only the 3' hotspot (indicated by the red spot at the bottom right corner of the square) corresponds with a hotspot indicated for this data in (d) and Fig 20. The blue spots within the red square in McVean's LD Matrix (g) confirm that recombination rates are lower than expected within the genomic region bounded by the recombination hotspot pairs indicated in (d).

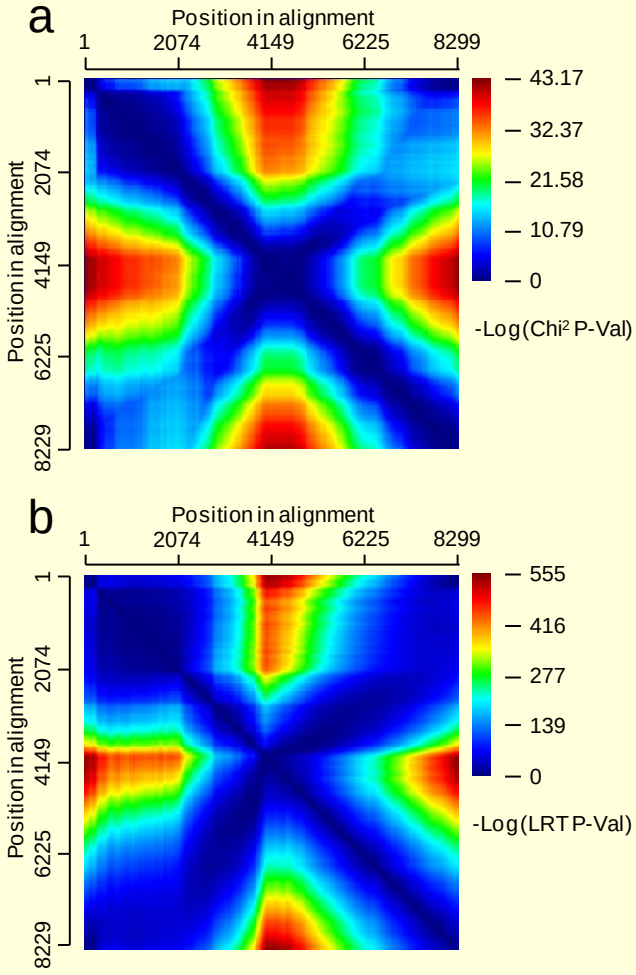

Figure 23. Breakpoint matrices. (a) A MAXCHI matrix and (b) a LARD two partition matrix. Dark red peaks indicate the most probable positions of breakpoint pairs. Both matrices were constructed for the same example recombination event depicted in Figures 8 through 19.

9.4.12 LARD Matrix. This is a matrix representation of the two breakpoint scan implemented in the program LARD (Fig 23b; Holmes et al., 1999; see section 8.10 for a description of the LARD method). As with the MAXCHI matrix (see 9.3.9), it is useful for identifying statistically optimal positions of breakpoint pairs. The advantage of the LARD matrix over the MAXCH matrix is that it can account for various different nucleotide substitution models. The major disadvantage relative to the MAXCHI matrix, however, is that LARD matrices can take a very long time to compute.

9.5 SCHEMA Protein Folding Disruption Test

Once an automated sequence analysis has been performed and you are confident that RDP5 has done a reasonable job detecting recombinants and identifying recombination breakpoints (see Section 10 for information on how to do this), you can test whether the observed recombination events have been less disruptive of protein folding than would be expected if recombination breakpoints were randomly distributed. This is done by pressing the arrow to the left of the "Run" button (see Fig 1) and selecting the "SCHEMA protein fold disruption test" menu option. The test of recombination induced protein folding disruption used by RDP5 is that described by Voigt *et al* (2002; i.e. that implemented in the program SCHEMA). RDP5s implementation of the test is that described by Lefeuvre *et al.*, (2007). Briefly, RDP5 takes the observed recombinants with recombination breakpoints within regions encoding the amino acid sequences represented in loaded up .pdb files and compares the estimated degrees of protein folding disruption in these to those that could have occurred had the observed breakpoints fallen elsewhere within the amino acid sequence encoding region(s). See Lefeuvre *et al* 2007 for a full description of how the method works.

9.6 SCHEMA Nucleic Acid Folding Disruption Test

This test is similar to the protein fold disruption test (section 9.4) and is described in Golden *et al*., 2014. As with the protein fold disruption test the nucleic acid fold disruption test should only be run after you have gon through and are happy with, the results of an automated recombination screen. You can run the test by pressing the arrow to the left of the "Run" button (see Fig 1) and selecting the "SCHEMA nucleic acid fold disruption test" menu option. Briefly, RDP5 takes the observed recombinants that have two identified parental sequences (i.e. it excludes recombinants labelled as having an unknown parent), constructs a mimic of these recombinants from their identified parental sequences such that the mimics have the same breakpoint positions as the real recombinants, infers the minimum free energy folds of the parental and mimic sequences using the program hybridssmin (Markham and Zuker, 2008), and counts (1) the numbers of base pairings that are present in parental sequences but are absent in their respective mimic recombinants (these are referred to as "broken" base-pair counts), and (2) the numbers of base pairings that are present in mimic recombinants but are absent in both of their parents (these are referred to as "aberrant" base-pair counts). RDP5 then compares these collective counts for all the mimic recombinants to those determined for sets of simulated recombinants (which are constructed from the same parental sequences and have the same parental nucleotide proportions as the mimics) to determine whether the mimics have a significantly lower broken and/or aberrant base-pair counts than the simulated recombinants. See Golden *et al* 2014 for a full description of how the method works.

9.7 Inferring Ancestral Sequences

RDP5 can infer ancestral sequences represented by any of the nodes in any of the trees that it constructs. This is achieved by moving the mouse pointer over a node, right clicking n the node and selecting the "Infer ancestral sequence" option. When you attempt to infer the ancestral sequence that is represented by a tree node it is recommended that (1) you ensure that the branch beneath the node has strong bootstrap/p-value support and (2) that the tree on which the node resides was constructed accounting for recombination. You will then be asked whether you would like to (1) using maximum parsimony and a maximum likelihood method alone (with DNAPARS and RAxML) or maximum parsimony, maximum likelihood and Bayesian methods (using MrBayes) and (2) whether you would like to take recombination into account Using the maximum likelihood method alone will yield results quicker but will not take uncertainty in the phylogenetic tree into account. For large datasets (>500 sequences) it could take the Bayesian method months to converge on a good ancestral sequence estimate. If you opt to account for recombination during the ancestral sequence inference, RDP5 will do this in two different ways depending on whether or not detected recombination events occur in an ancestor of the ancestral sequence being inferred. For recombination events that occurred in an ancestor of the inferred ancestor, phylogenetic trees will be separately determined for the region of the alignment between the recombination breakpoints, and for the remainder of the alignment. For other recombination events, in each detected recombinant the regions of sequence derived from the minor parent will be treated as missing data.

During ancestral sequence inference, the sequence display will be partitioned, and inferred ancestral sequences (the parsimony, maximum likelihood and Bayesian estimates) together with the probabilities associated with each inferred nucleotide state will be displayed. If a Bayesian approach is used, RDP5 will give live updates of the inferred state probabilities. Left click on the method names to shuffle between parsimony, maximum likelihood and Bayesian ancestral state probabilities. Right click anywhere in the ancestral sequence display to save the inferred ancestral sequences to a .csv file. This file will contain a "best estimate" of the ancestral sequence taking into account all the site state probabilities yielded by the maximum parsimony, maximum likelihood and Bayesian methods, and will also include all of the site state probability data associated with this best estimate.

10 A STEP-BY-STEP GUIDE TO USING RDP5

Given an input nucleotide sequence alignment, an ideal recombination analysis program would tell you the exact recombination history of

every nucleotide position relative to all others in every sequence in the alignment. Such a program does not and probably never will exist. The reason for this is that it is almost impossible to infer the exact recombination history of sequences in any but the simplest datasets. The best one could hope to achieve is the formulation of a set of consistent recombination hypotheses that describe a plausible series of recombination events that account for all recombination signals detectable in dataset. The step-by-step procedure described below tells you how you can use RDP5 to map and characterise a reasonably plausible recombination history for a group of moderately diverse recombining sequences.

Before you start analysing recombination with RDP5 you will need some things:

1. **A reliable sequence alignment tool**. ClustalX/W (Thompson et al., 1994) from http://www.clustal.org/, MUSCLE (Edgar, 2004) from http://www.drive5.com/muscle/ and POA (Grasso and Lee, 2004) from http://bioinfo.mbi.ucla.edu/poa/ are recommended for small, medium and large datasets respectively.
2. **A good sequence alignment editor with the capacity to realign sections of an alignment.** The Mega (version 2 or higher; Kumar et al., 2008) editor (from http://www.megasoftware.net/) is recommended.
3. **A suitable sequence dataset.** This should include no fewer than three sequences. The optimum size of the dataset is strongly dependant on the degree of diversity in the dataset and the types of recombination event one is attempting to detect. Tips for making a good dataset are given in the next section.

10.1 Compiling a Good Dataset

A wide variety of datasets can be productively analysed by RDP5 as long as care is taken during their assembly. The optimal size of a dataset depends on the degree of sequence diversity present therein. As recombination can only be detected if it occurs in sequences that are not identical to one another, it is important that a dataset contain enough diversity. For datasets with very low degrees of diversity, increasing the lengths of sequences being examined can increase the number of usable variable sites for recombination detection. It is, however, inadvisable to simply choose the largest dataset possible. The reason for this is that exploratory searches for recombination require repetitive statistical testing, with the number of tests performed increasing exponentially with the number of sequences and linearly with the lengths of sequences examined. A multiple test correction is therefore absolutely required to prevent false inference of recombination. Unfortunately, guarding against false positives also almost invariably means discarding some real evidence of recombination. At a certain point, that is dependent on the diversity of the sequences being analysed, the extra recombination signals potentially detectable by increasing either the lengths or numbers of sequences in a dataset will be counterbalanced by the increasing severity of multiple testing correction needed to guard against false positives. Although nobody has yet derived a simple formula for working out the optimal numbers and lengths of sequences needed for optimal recombination detection given a specified degree of diversity, the following procedure should ensure the assembly of a reasonably good dataset.

1. Collect all available sequences that share some degree of detectable identity (>50%) with the sequence(s) that are of greatest interest to you.
2. Load up the unaligned sequences in the program SDT1.0 (it should be in the same directory where you installed RDP5; Muhire *et al*., 2014) and perform a pairwise scan with the MUSCLE method.
3. When the run is completed and a matrix is displayed in SDT press the “Save” button and select the “create datasets” option. Saving datasets with a minimum identity of 70% and a maximum of 100% will ensure that the sequences will be properly alignable (see section 3.2 for reasons for not including highly diverged sequences in an analysis). SDT will split the alignment into groups of sequences that are all >70% identical and save these to separate .fas files. Note that the sequences in these .fas files are unaligned and they will need to be aligned before they can be analysed by RDP5.
4. Do a quick alignment of one of the SDT generated datasets (POA and MUSCLE are good for this as they are both >10 times quicker than ClustalW).
5. Discard all but one sequence in groups of sequences that are identical to one another. The tree drawing tools in RDP5 are useful for trimming datasets down to an optimum size. Open a preliminary alignment in RDP5 by pressing the “Open” button at the top of the screen and, once the alignment is loaded, press the “trees” button (also at the top of the screen). You can then visually identify groups of identical sequences on the tree(s) presented. Mark all but one sequence in groups of identical sequences by moving the mouse pointer over the sequence names and pressing the left mouse button. Pressing the right mouse button when the mouse pointer is over the sequence alignment display (the top left panel of the program) will cause a menu to appear which will then enable you to save the marked (or “masked”) and unmarked sequences to separate files.
6. The multiple testing correction that will be carried during screening for recombination signals means that it is pointless to attempt detection of recombination between sequences that are so similar that no recombination events between the sequences will be detectable. The genetic distance threshold at which sequences are so similar that no recombination will be detected between them varies both from one recombination detection method to the next, and with the number of sequences in the dataset. A reasonably conservative minimum genetic distance threshold can be approximated on a hand-calculator with the following formula: $Y = (2 \ln 4X) / L$; where Y is the pair-wise genetic distance threshold below which no recombination between a pair of sequences can be detected, X is the number of sequences in the dataset and L is the length of the alignment. For example if you have a 100 sequence dataset and the lengths of the alignment being analysed is 1000 bp then it will most likely prove fruitless to include groups of sequences in the alignment that have uncorrected genetic distances smaller than 0.012 (i.e. fewer than 12 polymorphisms differentiating them).
7. Once you have worked out a lowest acceptable pair-wise identity threshold you should identify pairs or groups of sequences that all have distances beneath this threshold and remove all but one of these from the dataset. It is very difficult to achieve an absolutely optimal dataset size as different recombination detection methods have different sensitivities and may use slightly different bits of information in the sequences being examined. The guidelines given here are approximate and could be improved upon by identifying the most closely related sequence pair, testing whether their genetic distance is below the threshold and removing one of the pair if it is. If a sequence is removed then a new threshold is calculated based on the new number of sequences in the dataset and the procedure repeated.
8. Don’t get too obsessive with manually getting the optimal dataset size - it is possible to do some automated dataset pruning within RDP5 once you have generated a final multiple sequence alignment and are ready to analyse it for recombination. Whenever you open a sequence alignment file in RDP5, the program will auto-mask some sequences that are very closely related to other sequences in the alignment so as to maximise recombination detection power. When sequences are masked RDP5 will not examine them in its exploratory search for recombination signals (and therefore a less severe multiple testing correction will be used). If the program finds a recombination signal in an unmasked sequence it will then examine all masked sequences to see if they too contain evidence of a similar signal.

10.2 Making a Good Alignment

While the importance of good sequence alignment in most nucleotide sequence analyses cannot be overstressed, good alignment is absolutely essential for recombination analyses. The reason for this is that all recombination signal detection methods are extremely sensitive to misalignment and will usually identify misaligned regions of sequence as having a recombinant origin. Although RDP5 has an inbuilt alignment checking capacity, it will simply eliminate regions of misalignment from the analysis and, in so doing, waste potentially important data.

Because alignment is so critical it is not recommended that datasets analysed for recombination ever contain any pair of sequences that share less than 60% nucleotide sequence identity. Ideally analyses should only be performed on sequences all sharing greater than 70% identity. When using this advice, however, one should be aware that even in cases where sequences are mostly very similar, there might be small regions within the sequences that are highly diverse and therefore nearly impossible to align accurately

Many multiple sequence alignment methods (such as that implemented in the Clustal programs) use a “guide-tree” to determine the order in which sequences are added to the alignment during its

construction. These methods will occasionally make obvious alignment errors when adding recombinant sequences with divergent parents to the alignment. The reason for this is that one or both portions of the recombinant sequence will be added to the alignment at an inappropriate point. Realigning subsections of the preliminary alignment that appear to have low degrees of sequence conservation can often rectify these errors.

It is strongly recommended that any unalignable (or just barely alignable) tracts be either deleted from the alignment or shifted/staggered in relation to one another. Shifting or staggering difficult to align tracts of sequence is a means of preserving as much data as possible while at the same time avoiding false inference of recombination. Often in an alignment you will find that sequences in different blocks of closely related sequences are poorly aligned with one another but are well aligned with other closely related sequences within the block. Such misaligned regions should be staggered in the alignment editor by adding gap characters at the 5' end of the region to sequences in one of the blocks and the same number of gap characters to the 3'end of the region to sequences in the other block. The number of gap characters added should be equal to or greater than the size of the misaligned region. It is also possible to do this with more than two misaligned blocks. Be careful not to knock the rest of the sequence 3' of your edits out of alignment.

To make a good alignment you will need to:

1. Make a preliminary alignment of the sequences using the alignment program of your choice. It is recommended that for sequence alignments involving fewer than 100 sequences you use ClustalX/W or MUSCLE (or any other version of Clustal such as that implemented in the MEGA sequence editor), and for alignments of more than 100 sequences you use POA or MUSCLE. Use default alignment settings for both unless you really know what you are doing.
2. Open the completed alignment in an alignment editor such as MEGA or IMPALE (the latter of which is distributed with RDP5 and should be within the RDP5 installation folder) and check its accuracy by both eye and/or using the sub-sequence realignment tool in MEGA/IPALE with different alignment parameter settings.
3. For small alignments it is often quite easy to visually detect obvious alignment errors. For larger alignments you will need to rely on systematic realignment of subsections of the alignment using different alignment parameter settings. You need to first identify parts of the alignment where sequences are most highly diverged and test to see whether realignment with different parameter settings (usually incrementally decreased gap extension and gap opening penalties) substantially improves the quality of the alignment.
4. In parts of the alignment where there is very little sequence conservation and no improvement in alignment quality is achievable, it is advisable that you either delete these columns or use the alignment shifting/staggering approaches.
5. For alignment of coding regions it is not always good to use codon/amino acid alignments as guides as these can occasionally be quite misleading.

10.3 Setting up a Preliminary Scan for Recombination

Before you start scanning an alignment for recombination it may be necessary to adjust some of RDP5's settings:

1. Start RDP5 and open your sequence alignment file.
2. Press the "Options" button at the top of the screen (Fig 1) and, under the "General" tab, go to the "General Recombination Detection Options" section. Specify whether the sequences being examined are linear or circular.
3. Move to the "Analyse Sequences Using:" section. You can select the methods you wish to use to detect recombination. It is strongly advised that you use the default selections (RDP, GENECONV and MAXCHI). If, however, you are analysing small datasets (<50 sequences) you could also select the CHIMAERA, BOOTSCAN/RECSCAN, SISCAN and 3SEQ methods and the program will still not take overly long to run. Note that there are two possible ways of using the BOOTSCAN and SISCAN methods for detecting recombination in an alignment. By default both BOOTSCAN and SISCAN will be used to automatically check recombination signals detected by all other methods but they will not be used to explore for any new signals. You can force their use for exploratory screening by selecting the left box beside the method name, but be warned that the analysis can become very slow if you use these for exploratory screening of large datasets. The RDP, GENECONV, MAXCHI, 3SEQ and CHIMAERA methods will all automatically be used to check recombination signals detected by all other methods regardless of whether they are selected or not. The LARD (Holmes et al., 1999) method can only be used to check signals detected by other methods and should only be selected if datasets are very small (<20 sequences). A very rough estimate of the anticipated analysis time is given at the bottom of the options menu so that you can judge whether particular selections are computationally viable.
4. Move to the "Data Processing Options" section. Apart from the "Disentangle overlapping events" option you should use the default settings. If the "disentangle overlapping events" option is selected the program will attempt to ensure that the recombination hypothesis it derives does not invoke recombination between recombinant sequences with similar mosaics (such as would be identified as relatively unlikely reciprocal recombination events) to explain the recombination signals it detects. This setting works well when recombination in the dataset is relatively sparse and there is some evidence of recombination hotspots. However, the algorithm used to disentangle overlapping recombination events can get into a circular loop where it is unable to derive a recombination hypothesis that does not involve, for example, reciprocal recombination. It is therefore recommended that if you want to try the "disentangle overlapping events" setting you should keep track of how long the analysis is taking: Specifically, you should note the event numbers in the summary table: if the numbers outside the brackets seem to repeatedly go up a bit and then repeatedly drop back down to the same number it means that the program is likely in a never ending loop and will probably not run to completion. If it gets stuck like this you will need to stop the program and restart your analysis without the disentangle overlapping events setting.
5. For the method-specific options, the only settings that you should occasionally change from their default values are window and step sizes. The optimal window size will vary slightly from method to method and from dataset to dataset. It is important to note that whereas the RDP, 3SEQ, GENECONV, MAXCHI, and CHIMAERA methods only examine variable nucleotide positions in triplets of sequences sampled from the alignment, the BOOTSCAN, SISCAN and LARD methods examine all variable and conserved positions. Also note (a) that the CHIMAERA and MAXCHI windows should be approximately twice as large as the RDP window and (b) that the SISCAN, BOOTSCAN windows should be approximately the same size. Ideally, window sizes should be set small enough to ensure that events involving exchanges of small tracts of sequence (<200bp) are detectable in the most divergent sequences being examined. The optimal window size to detect a recombination event involving a 200 bp exchange of sequence is 200 for BOOTSCAN and SISCAN and varies for the other methods depending on the number of nucleotide differences between the parental sequences in the recombinant region. The MAXCHI and CHIMAERA methods can be set to run with a variable window size that will respectively get bigger and smaller with lower and higher degrees of parental sequence divergence. Although window size settings can have a substantial impact on the preliminary recombination hypothesis formulated during the automated recombination signal screening stage of analysis, the subsequent (and necessary) phase of manually testing and refining analysis results should largely counteract any "settings biases" that have been introduced.
6. Once all settings have been made, press the "Run" button at the top of the screen and wait for the automated analysis to complete. Note that there are two major phases in the automated analysis. The first involves the detection of recombination signals in the alignment and the second involves inference of the number and characteristics of unique recombination events that have generated these signals (see section 4.1.3). If the analysis is taking far longer than anticipated you can press the "STOP" button towards the bottom centre of the screen. If you prematurely stop the automated analysis you will be given the analysis results up till the point where the analysis was stopped. From here you can either decide to restart the analysis with different settings or you can move on to the hypothesis testing and refinement stage of the analysis process. If you choose the latter you will be given the opportunity to complete the automated analysis at a later stage.

10.4 Testing and Refining Preliminary Recombination Hypotheses

The automated output given by the program is nothing more than a preliminary hypothesis probably describing only a small fraction of the

recombination events that have occurred during the histories of sequences you have analysed. It is very important that you be aware that RDP5 can get things horribly wrong. The program's failures will be of four major types:

1. Inaccurate identification of recombination breakpoint positions.
2. Incorrect identification of parental sequences as recombinants.
3. Incorrect inference that groups of identified recombinants are all descended from the same recombinant ancestor.
4. Incorrect inferences that recombinants descended from the same ancestral recombinant contain evidence of unique recombination events.

Unfortunately there is not yet any automated tool that will enable you to definitively judge whether the results you obtain contain any of these errors (as much automation as I am capable of programming has already been programmed). It is very likely that, unless the initial automated run of RDP5 indicates only a few recombinant sequences (<20% of the sequences in the dataset are recombinant), the program will have made some mistakes interpreting the patterns of recombination it has detected. The size and importance of the mistakes will scale with the number of unique recombination events the program detects. It is especially important to realise that mistakes early on in the analysis (such as in the first 10% of unique recombination events the program has characterised) will be more serious than those made in the end stages of the analysis. The reason for this is that RDP5 identifies and characterises the easiest to detect recombination events first and leaves interpretation of the least obvious recombination signals until last. Once, for example, a mistake has been made identifying which of the sequences is recombinant for a specific recombination event, the probability that the program will make additional mistakes of this type during the characterisation of all subsequent recombination events will be increased. To minimise the risk of a largely incorrect analysis result, it is very important that the following hypothesis refinement approach be used.

1. Once the automated analysis has completed, a set of coloured blocks will be displayed in the bottom right panel of the program (see section 5.1, the schematic sequence display in Fig 1 and Fig 2). These graphically represent the recombination events that the program has detected. For each sequence in the dataset the name of the sequence and a coloured strip is displayed. Beneath some of these strips (and corresponding with lightened sections of the coloured strip immediately below the sequence name) are a series of coloured blocks. These blocks each represent a proposed recombination event. If you move the mouse pointer over any of these blocks, information relating to the proposed recombination event will be displayed on the top right panel of the screen (see section 5.2; Fig 3). This information includes possible recombination breakpoints, names of sequences in the dataset that are closely related to the presumed parents of the recombinant sequence, the approximated probability values (both corrected and uncorrected for multiple testing) of observing a recombination signal with the same strength without recombination having occurred, the number of sequences in the dataset with similar signals detected by different recombination detection methods (in the "confirmation table"), and a graph showing evidence used by the program to infer which of the sequences used to detect the recombination signal is the recombinant sequence (see section 4.1.4). The most important bit of information displayed here is, however, the series of warnings that the program gives in capitalised red letters. These will indicate when RDP5 is reasonably unsure about some of the conclusions it has reached. The program will issue a warning if: (a) One or both of the inferred breakpoint positions are probably inaccurate; (b) The wrong sequence may have been identified as the recombinant; (c) It is possible that an alignment error has generated a false positive signal; (d) one of the recombinant's parental sequences has remained unsampled; (e) Only trace evidence (i.e. technically not statistically significant) of recombination is evident within the currently specified sequence. When RDP5 attempts to infer the minimum number of recombination events in a sample, it detects a recombination signal and tries to determine which other sequences in the alignment carry similar recombination signals. Often sequences carrying similar signals will be identified but the signal in these sequences is not sufficiently strong to generate a statistically significant p-value. These signals are referred to as "trace" signals and are listed as such in both the "recombination information" part of the RDP5 display and the phylogenetic trees that the program constructs.
2. It is strongly recommended that you refine the recombination hypothesis the program provides and that you do this one recombination event at a time. As mistakes early on in the analysis are likely to be more serious than mistakes towards the end, you should always start the refinement process with the recombination event that the program characterised first. All the unique recombination events detected by RDP5 are numbered in order from the first to the last that the program characterised. If you press the left mouse button when the mouse pointer is on a background greyed area of the bottom right display panel (the one with the coloured blocks) you will focus the program on this part of the display. Pressing either the "page down", "page up" or "space bar" keys on your computer keyboard will now allow you to navigate through the detected recombination events in an ordered fashion. You can also navigate through the detected recombination events by pressing the arrow keys beneath the schematic sequence display (Fig 1 & 2). Pressing either the page-down button or the right arrow key immediately after an automated analysis is finished, will take you to the first recombination event that the program characterised. Doing this again will take you to the second event and so-on. Pressing the page up button or the left arrow key will take you to the previous event. If you are currently on the first recombination event that the program characterised, when you press the page up or left arrow key you will be taken to the last event that was characterised. Pressing the space-bar will take you to the recombination event with the best associated p-value.
3. Pressing the page-down button/right arrow key and starting with the first event you will see a graph drawn on the bottom left panel beneath the sequence alignment display (Fig 4). The exact information that is plotted in this graph will depend on the recombination method that yielded the best evidence of recombination for the recombination event at hand. Such graphs can be useful for checking the accuracy of recombination breakpoint estimation. Probably the best graphs for this purpose are those generated by the MAXCHI (Fig 11) and CHIMAERA (Fig 12) methods. To see a MAXCHI graph, press on the "check using" button on the right hand side of the plot display (Fig 4). One of the options offered is to draw a MAXCHI plot. Select this option and see whether the peaks on any of the three lines plotted correspond with the borders of the detected recombinant region (shown in pink on the graph see Fig 11b). If the peaks and recombinant region borders do not match, this does not necessarily mean that the inferred breakpoint positions are wrong. It does, however, mean that there is a fair degree of uncertainty regarding the position. Look at graphs for some of the other methods. The boundaries of the recombinant region in pink should match positions in the RDP (Fig 8b), BOOTSCAN/RECSCAN (Fig 9b), SISCAN (Fig 13b), VisRD occupancy (Fig 16b), BURT (Fig 19) and DISTANCE plots where two of the three plotted lines intersect. As with the MAXCHI plot, the boundaries of the pink region should match peaks in at least one of the lines in CHIMAERA (Fig 12b), TOPAL (Fig 18b), PHYLPRO (Fig 15b) or LARD (Fig 17b) plots. For the 3SEQ plot (Fig 14b) one of the boundaries of the pink region should correspond with a peak and the other with a trough. VisRD Highway and GENECONV plots are not particularly useful for identifying potential recombination breakpoint positions.
4. Another tool that can be used to determine the optimal locations of breakpoint pairs are MAXCHI (Fig 23a) and LARD matrices (Fig 23b). These graphically display the probabilities of all potential breakpoint pairs. To see a MAXCHI matrix press the matrices button on the top right hand panel (Figs 3 and 6). Move the mouse pointer into the matrix window and press the rightt mouse button. Select the "Change matrix type" and then the "MAXCHI breakpoint matrix" options from the menus that appear and the MAXCHI matrix will be displayed. Interpretation of the matrix is relatively simple in that the most probable (although not necessarily correct) breakpoint pairs correspond with matrix cells that have the best associated p-values (use the colour key displayed beside the matrix to determine which colour corresponds with the best p-value; see section 9.3.9 and Fig 23).
5. If you are satisfied with the breakpoint positions identified by the program move on to step (6). If, however, you would like to alter the position of one or both of the breakpoints, this can be done via the sequence alignment display window (Fig 1 and Fig 5). You will need to go to the "show relevant sequences" version of this display. You get there by repeatedly pressing on the curled arrow in the top right hand corner of the sequence alignment display (the

"toggle sequence display" button in Fig 5) until the caption beside the arrow reads "show relevant sequences". Once there, you can get to the approximate region of sequence where you think the breakpoint should be by moving the mouse cursor to the corresponding point on the plot display (Fig 4) and pressing the left mouse button twice in quick succession. The "show relevant sequences" version of the alignment display will allow you to decide the best point to place the breakpoint as it colour codes variable nucleotide positions in the alignment according to which of the three sequences used to detect the recombination signal are most closely related. You should ideally place the breakpoint position in-between two variable nucleotides, one indicating a close relationship to one parent and the other a close relationship to the other parent. When you've decided on a position, point the mouse cursor at it and press the right mouse button. On the menu that appears some of the options involve placing breakpoints at this position. Some refer to "beginning/ending breakpoints" and others refer to "ancestral beginning/ending breakpoints." Placing an ancestral breakpoint will automatically adjust the breakpoint in all other sequences descended from the same ancestral recombinant as the currently selected sequence. Placing a breakpoint rather than placing an ancestral breakpoint will modify the breakpoint position only in the currently selected sequence.

6. The next thing to consider is whether the program has correctly identified the recombinant. This can be very difficult to asses. The program uses a range of phylogenetic and genetic distance based tests to infer which sequence is the recombinant in a group of sequences containing a recombination signal (see section 4.1.4 and histograms in Fig 3). Very often different tests tell the program that different sequences are the recombinant and the program therefore uses a weighted consensus of these tests. There is also no guarantee that the relative weighting of the tests is accurate. Tests using recombinant HIV sequences have been used to weight the different tests. However, what may be a reasonably accurate weighting for HIV might not be good for your data. Also, even in tests with HIV the program only has an approximately 90% success rate when it comes to correctly identifying the recombinant sequence. The results of these tests are displayed, together with their weighted consensus, as a series of bar graphs in the "recombination information" panel part of the program display in the top right of the screen (Fig 3 and see section 4.1.4). RDP5 will provide a warning if the tests do not clearly indicate which sequence is recombinant. You must take this warning seriously and determine for yourself whether one of the suggested parental sequences might not be the actual recombinant.
7. The best available tool in RDP5 for assessing whether the recombinant sequence has been correctly identified is to draw and compare two phylogenetic trees, one constructed from the portion of the alignment between the inferred breakpoints, and the other from the remainder of the alignment. The program automatically draws UPGMA trees for each of the two sections of the alignment whenever a particular recombination event is selected for more detailed analysis. You can see these trees side-by-side if you press the "trees" button in the command button panel at the top of the screen (Fig 1). If you would prefer a bootstrapped neighbour joining tree (drawn using PHYLIP, Felsenstein, 1989), least squares tree (also drawn using PHYLIP) or maximum likelihood tree (drawn using PHYML; Guindon and Gascuel, 2003) or Bayesian tree (drawn using MrBayes; Ronquist *et al*., 2012), you can change the tree type that is displayed by (a) moving the mouse cursor over one of the trees, (b) pressing the right mouse button, (c) selecting the "change tree type" option offered at the bottom of the menu that appears, and then (d) choosing the preferred tree type from this sub-menu. If there are more than ~20 sequences in your dataset only select the Least Squares, Maximum Likelihood or Bayesian tree options if you are quite patient – making these trees will take a long time. Comparing the locations of sequences in these trees can indicate which sequence(s) is (are) recombinant. This is because the position(s) of the recombinant sequence(s) should change more between the trees than should the positions of parental sequences. RDP5 allows you to mark particular sequences in the trees that it displays. This can be very useful for tracking the "movement" of particular sequences between the clades of different trees. You can mark a sequence by moving the mouse pointer over the name of the sequence in the tree and pressing the left mouse button. The same sequence is then marked in all of the trees. You can clear markings or automatically colour sequence names (so that they are the same colours as those displayed in the bottom right panel for graphical representations of the sequences) by selecting the appropriate option on the menu that appears whenever you press the right mouse button with the mouse pointer over one of the tree displays. Be very careful when deciding to change the choice of recombinant that the program has made as there are many factors that might seriously complicate the identification of recombinant sequences using phylogenetic trees. As RDP5 uses three sequences at a time to identify recombination signals there is a very high probability that, given enough recombination in a dataset, two or even three of the sequences used to detect a recombination signal will be recombinant (i.e. such as when recombinant sequences recombine). While this is a particularly serious problem if two or three of the recombinant sequences in the sequence triplet have breakpoints close to one another, it can even be a problem if the breakpoints are in completely different parts of the sequence. If the breakpoints in two or three of the sequences in a triplet are close together, the program will possibly infer a recombination event with two breakpoints each of which comes from a different recombinant sequence in the triplet. How this will affect the trees constructed when these breakpoints are used to partition the alignment, will vary from case to case, but it could influence how effectively trees can be used to judge the accuracy of recombinant sequence identification. Even when the correct breakpoints are used to partition the sequences, the fact that one or both of the inferred parental sequences are also recombinant might effect their placement within the trees making it very difficult to identify which sequence(s) is (are) recombinant. Finally, even if both of the inferred parental sequences are non-recombinant, the dataset might contain any number of other recombinant sequences, any of which might confuse identification of a recombinant sequence based on assessing which sequence has "moved" the most between trees. If you are unhappy with the recombinant sequence identified by RDP5 always remember that the program has used phylogenetic approach along with a battery of other tests and has, for some (perhaps very good) reason, come up with its particular choice. Also, if you think the program has made a mistake, be sure that you understand what the trees are telling you. Remember that although trees are presented with a midpoint root for the sake of clarity, they are all actually un-rooted and. therefore the direction of evolution may not be immediately clear. Your greatest chance of doing better than the program at identifying true recombinants is when the program has either identified two very close candidate recombinants (it will tell you which these are), or when you have obviously corrected a badly placed breakpoint position in the preceding step of the analysis. The reason for the latter is that the accuracy of the tests the program initially used to identify the recombinant sequence may have been compromised by the use of badly placed breakpoint positions.
8. If you decide that the program has done a good enough job identifying the recombinant sequence(s), move on to step (9). If, however, you would like to reassign the recombinant you can do so by moving the mouse pointer over the graphical representation of the recombination event in the schematic sequence display (Fig 2) and pressing the right mouse button. A menu will appear and the options offered at the bottom of this will be to "swap the recombinant and either the major or minor parental sequences". Select the appropriate option based on whether you think the recombinant has been misidentified as the major or the minor parent.
9. If the current recombination event occurred in the common ancestor of two or more sequences in the dataset, it is often desirable that the program properly identify these sequences as sharing evidence of the same unique recombination event. However, RDP5 will often mistakenly group sequences that are descendents of different ancestral recombinants. The program could also mistakenly infer that two or more unique recombination events are responsible for recombination patterns detected in sequences that are in fact all descended from the same recombinant. Although you might expect that the descendants of recombinant sequences should all have nearly identical mosaics and "move" together within phylogenetic trees constructed from different parts of an alignment, this is not always the case. For example, some sequences may contain only partial evidence of a particular recombination event because a second, newer recombination event overprinted part of the evidence from the older recombination event. If recombination has occurred frequently within the history of a sample of sequences then there is a good chance that evidence will exist in the alignment of older recombination events being overprinted by newer ones. When events are completely overprinted (i.e the old tract of recombinant

sequence is completely replaced by a newer tract of sequence) it is not too serious a problem, although it can sometimes confuse phylogenetic interpretations of which sequences in the dataset are recombinant. The reason for this is that when trying to use trees to identify sequences that all contain evidence of an ancestral recombination event, sequences descended from the ancestral recombinant but in which the ancient event has been overprinted, will not "move" together with the sequences in which the event was not overprinted. The problem is a lot worse when recombination events are partially overprinted by a more recent event in some sequences but is not overprinted in others. Under these circumstances there is a chance that the program will not group the recombination signals properly and it may infer two recombination events have occurred instead of one. Although the inference of two recombination events is technically correct in such cases, the program will incorrectly identify which sequence exchanges took place. The probability of this happening depends on both (a) whether the partial overprinting of evidence for an older recombination event by a newer one spans one of the breakpoints from the older event, and (b) the proportion of the older event overprinted by the newer event. If the newer event spans one of the older event's breakpoints but only overprints a small proportion of the older tract of recombinant sequence, it is less serious than if it replaces most of the older tract of recombinant sequence. In any case, it is very difficult to interpret the phylogenetic signals caused by partially or completely overprinted events and it will often seem like RDP5 has mistakenly identified sequences as containing partial evidence of a given ancestral recombination event.

10. One way of identifying sequences with evidence of the same ancestral recombination event, is to generate and compare BOOTSCAN or RDP plots with different potential recombinant sequences while using the same parental sequences. To do this in RDP5: (a) Make a list of sequences either that the program has indicated contain evidence of a similar recombination event, or that you have chosen based on, for example, their clustering with identified recombinants in phylogentic trees constructed from different parts of the alignment; (b) Select either the BOOTSCAN or RDP plots in the "check using" dropdown menu on the plot display (Fig 4); (c) In one of the trees move the mouse pointer over one of the sequence names you have on your list, press the right mouse button and select the "Recheck plot with...as recombinant" option. This will compare the plots of the currently selected sequence (usually the sequence containing the "best" evidence of the current recombination event) with one of the sequences on your list. The results of the comparison are displayed graphically in the form of a coloured line above the graph in the plot display. Blue in this plot indicates regions of sequence where recombination signals are highly similar and dark red indicates regions of sequence where recombination signals are completely different. If the portion of sequence spanning one or both recombination breakpoints corresponds with a blue/green signal, then the pattern of sites shared by the sequences being compared and their supposed parental sequences are very similar and are possibly derived at approximately the same time from the same or similar sources.
11. If you are convinced that RDP5 has either missed evidence that a group of sequences have all descended from a common ancestral recombinant, or incorrectly identified that a group of recombinant sequences have all descended from a common recombinant ancestor, the situation can be remedied via the tree display. On one of the trees, move the mouse pointer over one of the sequence names that have been incorrectly included or excluded as sharing evidence of the current recombination event. Press the right mouse button and the first option on the menu that appears will be to mark the sequence you have selected as either having or not having evidence of the current recombination event being analysed. You then select the appropriate option and the sequence will be added to or removed from the list of sequences containing evidence of that event.
12. Having either refined or left the current recombination hypothesis unchanged it is important that you tell the program that you are at least provisionally happy with the way the current event has been characterised. You do this by moving the mouse pointer over the coloured box representing the recombination even in the schematic sequence display (Fig 2 - the block should be flashing blue). Press the right mouse button and select the "Accept all similar" option near the middle of the menu that appears. A red border will be drawn around the coloured box (and all others representing the same event) and this will tell the program that in all subsequent analyses you are happy with the characterisation of this event. If you are not happy with the way either you or the program has characterised the event you can opt to either leave the event unaccepted or you could select the "Reject all similar" option. If events are accepted or rejected the program will then skip these when you are navigating your way through the remaining events that need to be checked.
13. If you did not modify breakpoint positions, change which sequence was identified as the recombinant, and left the list of sequences sharing evidence of the same recombination event unchanged, then you should return to step (3) above and begin checking the next event. If, however, you modified the recombination hypothesis and accepted your modification (as in step (12) above) it is important that you press the rescan button (Fig 2) so that RDP5 can take your corrections into account when it characterises the remaining recombination events. The reason for this is that all the accurate characterisation of all the remaining events will have been at least partially influenced by whatever error you have corrected. Your changes, no matter how small, could influence the remainder of the analysis. To reanalyse the data taking your modifications into account, press the "Rescan" button (Fig 2 – the button may be flashing red) benieth the schematic sequence display. Once the program has reanalysed the sequences, return to step (3) above and check the next event.
14. Save your analysis regularly because RDP5 will occasionally crash (it is quite buggy). Press the "Save" button on the command button panel (Fig 1) and you will be given the opportunity to save the entire analysis in ".rdp" (RDP project file) format – This is the standard format that you should always save tour results in. If you would like a tabulated summary of the results opt to save the data in ".csv" format. This format will allow you to open the results in spreadsheet programs such as Microsoft Excel. Note, however, that RDP5 will not be able to reload your analysis from a ".csv" file.

10.5 Examples

10.5.1 Producing a preliminary recombination hypothesis. Load the example alignment file "PVY Example.fas" (this and all other example files referred to here can be found in the directory where you have installed RDP5) and press the "Options" button (Fig. 1). The example sequences we will be analysing are linear virus genomes, so in the "General Recombination Detection Options" section under the "General settings" tab, change the "Sequences are circular" setting to "Sequences are linear." Besides this change, we will use the default RDP5 settings for this example. Press the "OK" button at the bottom of the options form. Press the "Run" button (Fig. 1) and wait for the automated analysis to complete (it should take less than a minute).

10.5.2 Navigating through the results Press the left mouse button when the mouse pointer is on a background greyed area of the schematic sequence display (Fig. 1). This focuses the program on the display. Pressing either the "Pg Up," "Pg Dn" or "space bar" keys on your computer keyboard will allow you to navigate through the detected recombination events in an ordered fashion (alternatively you can use the arrow buttons beneath the schematic sequence display to do the same thing). Immediately after finishing the automated analysis, pressing the "Pg Dn" button will take you to the first recombination event identified by RDP5. Pressing it again will take you to the second event, and so forth. Pressing the "Pg Up" button will take you to the previous event. Pressing the space bar will take you to the recombination event with the best associated p-value.

Starting with the first event (press the "Pg Up" or "Pg Dn" button until information on "recombination event 1" is displayed in the recombination information panel) you will see a graph drawn on the "plot display" (Fig.1). The exact type of graph that is plotted will depend on the recombination method that was used to detect the recombination event at hand. In this case the plot should be one that is produced using the RDP method. The yellow, purple and green plotted lines each represent comparisons between different pairs of sequences in the sequence triplet (in this case "Q" vs. "T", "I" vs. "T" and "I" vs. "Q" respectively) that was used to detect the recombination event. The part of the graph between alignment positions 1 and 2,250 indicates that the sequence identified as the recombinant, "I", is most closely related to sequence "T" in this region. Over the remainder of the graph, between alignment positions 2,251 and 9,594, sequence "I" is much more closely related to sequence "Q" than it is to sequence "T". The probability that this pattern of relatedness between "I", "Q" and "T" could occur without recombination under neutral selection is approximately 1 in 10198. This is very strong evidence of recombination. Turn your attention to the confirmation table in the recombination information display (Fig. 1). Note that all methods other

than LARD and PHYLPRO [17] have also indicated that the shifting relationship between "I" and these other two sequences is good evidence of recombination. All associated p-values are smaller than 10-28. The LARD and PHYLPRO methods were not used during the automated exploratory scan for recombination signals and this is the reason that they have no associated p-value. The large differences between the reported p-values for the other methods are largely due to the fact that the different methods consider slightly different signals in the alignment and have different approaches to approximating the probability that apparent recombination signals are caused by accidental convergent mutation rather than recombination.

Note that there is a warning (in red capitalised letters) in the recombination information display. Because you are analysing linear sequences, RDP5 is telling you that only the "Ending" (or 3') breakpoint is an actual recombination breakpoint (the other "breakpoint" listed is the beginning of the sequence).

10.5.3 Checking the accuracy of breakpoint identification. It is important that you check the accuracy with which RDP5 has identified the recombination breakpoint positions. The RDP method used to detect this recombination signal has a lower degree of breakpoint inference accuracy than the BURT, MAXCHI, CHIMAERA and 3SEQ methods. To see a MAXCHI graph for event 1, press the "check using" listbox on the right hand side of the plot display (Fig. 1). One of the options listed is to construct a MAXCHI plot. Select this option and see whether the peaks on any of the three lines plotted correspond with the left and right borders of the pink area. Look at graphs for some of the other methods: especially BURT (which seems to be the most accurate breakpoint localisation method of all those implemented in RDP). The left and right boundaries of the pink box should match positions in the BURT, RDP, RECSCAN, SISCAN and DISTANCE plots where two of the three plotted lines intersect. As with the MAXCHI plot, the left and right boundaries of the pink box should match peaks in at least one of the lines in CHIMAERA, TOPAL [18], PHYLPRO and LARD plots. For this recombination event all of the methods seem to indicate the recombination breakpoint at position 2,250 has been correctly identified.

The actual breakpoint position in this example might, however, not be as obvious as you think. Note that in the recombination information display, five different sequences have been identified as descendants of the same recombinant (you can tell this by looking at the confirmation table in the recombination information display). Press the "Trees" button at the top of the screen (Fig. 1). Five of the sequences in the trees displayed are highlighted in red, pink or purple. These are all sequences that potentially also carry evidence of recombination event 1. The sequence in red, "I", is currently selected. Move the mouse pointer over "W" and press the right mouse button. Select the "Go to W" option. This will centre the schematic sequence display on "W". Move the mouse pointer over the left most coloured block representing the recombination event 1 signal in "W". Look at the recombination information display. Note that the "Ending" breakpoint position is identified here as 2,261 and not 2,250. This is a small but important difference. Press the "show relevant sequences button" and use the scroll bar at the bottom of the sequence display to move to position 2,261. The colour-coding of the nucleotides now corresponds with the colours of the lines in the plot below. You will see that at position 2,258, "W" and "T" share an A nucleotide, and also that the breakpoint is inferred to lie three nucleotides to the right of this point in "W" (instead of eight nucleotides to the left of this point as in "I"). Now go back to the corresponding representation of recombination event 1 in "I" and left-click on it. Look at the sequence display and you will see that at position 2,258 sequence "I" has a G residue that is shared with sequence "Q".

Clearly the breakpoint position should be somewhere in the region between 2,250 and 2,261, but its precise location is unknown. Let us, just for the sake of this example, adjust the breakpoint position to nucleotide 2,261. To do this move the mouse pointer to nucleotide 2,261 of the middle sequence in the sequence display and right-click on it. One of the options offered will be to "Place ending breakpoint here". Select this option, so that when you look at the representations of this event in the schematic sequence display you will see that they all report the breakpoint position as 2,261.

10.5.4. Checking the accuracy of recombinant sequence identification. Look at the bar graphs in the recombination information display (Fig. 1). The first set of three bars indicate the consensus "Recombinant scores" of sequences "I" (0.667), "Q" (0.077) and "T" (0.256). These scores are the weighted consensus of a series of tests (each indicated by a set of three bars in the graph) to determine which sequence out of the three is the recombinant. In this case it is apparent that "I" is probably the recombinant.

It is useful to consider the phylogenetic trees in order to validate the status of "I" as the recombinant. Bring up the side-by-side tree display by pressing the "Trees" button in the command button panel (Fig. 1). Right-click on an empty area in one of the two windows. The menu that appears has an option to "Change tree type" - select this and then select the neighbour joining tree type. Now change the tree in the other window to neighbour joining too. You should see the same trees as in Fig. 24. For now focus on the sequences in the tree that are highlighted in red ("I"), green ("Q") and blue ("T") and ignore those highlighted in pink ("W") and purple ("J", "N", and "O"). Comparing the locations of these three sequences in the trees should indicate that the sequence highlighted in red has "moved" from the "Q" clade of the tree to the "T" clade. It seems that the program was correct in its identification of this sequence as the recombinant. You therefore do not need to change anything.

For the sake of this example, however, right-click on the blue flashing block in the schematic sequence display and select the "Make the minor parent (T) the recombinant" option. Observe how the blue flashing block is "moved" to "T". Look at the tree and see how the interpretation of this recombination event has changed. Now make "I" the recombinant again.

10.5.5. Evaluating RDP5's grouping of recombination events. There is apparently some evidence that recombination event 1 may have occurred in the common ancestor of five sequences in the dataset – those sequences currently highlighted in purple/pink/red in the trees ("I", "W", "J", "N" and "O"). It is, however, also apparent that the five identified recombinant sequences neither all cluster within the phylogenetic trees, nor all move together between the phylogenetic trees. This fact suggests that RDP5 may have "over-grouped" these sequences and that they may in fact be carrying evidence of multiple different independent recombination events. If you look at these five sequences in the schematic sequence display you will, however, immediately see the probable reason that these sequences do not move together between the two phylogenetic trees: RDP5 has identified additional recombination events in all of these sequences other than "I". In such cases it is not expected that a group of sequences carrying evidence of the same ancestral recombination event will all cluster together in both of the trees.

RDP5 provides another tool with which you can check whether two sequences carry evidence of the same ancestral recombination event. In either one of the trees right-click on sequence "W" and select the "Recheck the plot with W as the recombinant" option. This will compare the plots produced using the currently selected sequence (in this case sequence "I" – the one in red) with that of sequence "W". The result of this comparison is displayed graphically in the form of a multi-coloured line above the plot in the graph display (Fig. 25). Whereas blue along this line indicates regions of sequence where recombination signals are very similar, dark red indicates regions of sequence where recombination signals are very dissimilar. You will notice when comparing "W" with "I" that the line becomes very red between positions 5680 and 9099. If you look at sequence "W" in the schematic sequence display on the bottom left (click on "W" in the trees and select the "go to W" option) you observe that RDP5 has detected a second recombination event in "W" with breakpoints corresponding to these alignment coordinates. If the portion of sequence spanning one or both recombination breakpoints (i.e. the left and right bounds of the pink area in the plot display) corresponds with a blue/green line (as it does in this case) then the pattern of sites shared by the sequences being compared and their supposed parental sequences are very similar across the breakpoint(s). Thus it is very likely that the two sequences being compared carry evidence of the same recombination event.

For the sake of this example, let us pretend that the very similar recombination signals that are evident in "I" and "W" are not derived from the same ancestral recombination event. To exclude the recombination event detected in "W" from event 1, go to the side-by-side tree display and right-click on "W". Choose the option to "Mark W as not having evidence of this event." If you would like to re-include "W" as having evidence of recombination event 1, then right-click on "W" in the tree and select the "Mark W as having evidence of this event" option.

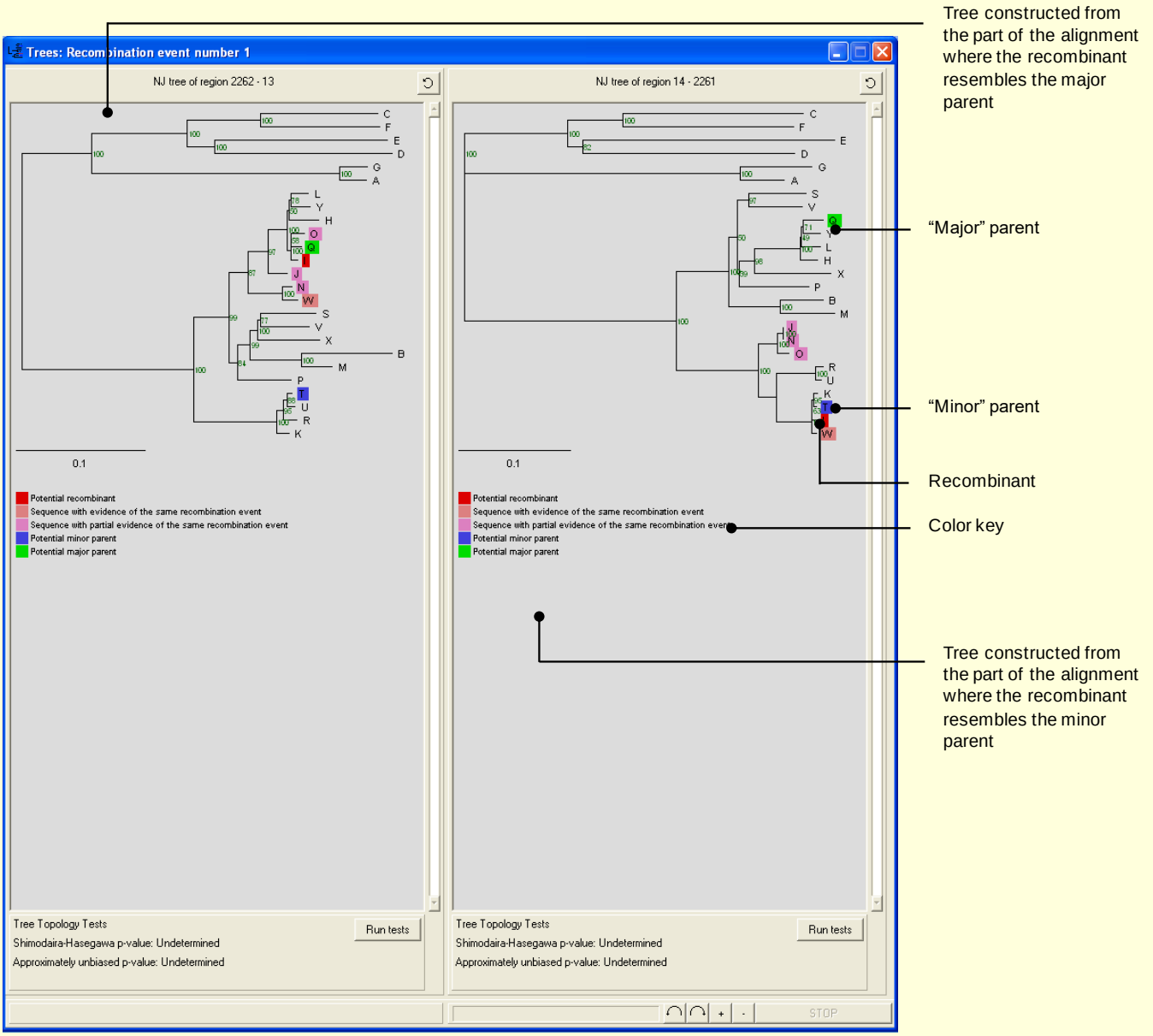

Figure 24. Using phylogenetic trees to determine which sequence(s) is (are) recombinant. In this example RDP4 has inferred that five sequences ("O", "I", "J", "N", and "W") are descended from a common recombinant ancestor with parental sequences resembling the "Q" and "T" sequences. Note that these five sequences do not form a monophyletic group in either tree. This indicates either that RDP4 has "over-grouped" the sequences or that other recombination events elsewhere in these sequences may be obscuring the phylogenetic relationships between them (as turns out to be the case in this example).

.

10.5.6 Completing the analysis. Because you have changed the way that RDP5 has interpreted event 1, you need to let the program reformulate its characterisation of all the other recombination events detected.. First, however, it is important to inform RDP5 that you are content with the current interpretation of recombination event 1. To do this, right click on the blue flashing block in the schematic sequence display. Select the "Accept this event in all 5 sequences where it is found" option. You should notice that a series of red borders have been drawn around the blocks representing the recombination event 1 signals in sequences "I", "W", "J", "N" and "O". Now either click on the flashing "Re-scan" button beneath the schematic sequence display or right-click anywhere in the schematic sequence display and select the "Re-Scan and re-identify recombinant sequences for all unaccepted events" option.

When RDP5 has finished reanalysing the remaining recombination signals press the "Pg Dn" button on your keyboard and you can start evaluating recombination event 2. Continue until you reach the end of the analysis. You may notice that the program skips event 2. The recombination signal corresponding to recombination event 2 has been identified by RDP5 as being attributable to sequence misalignment. To see recombination event 2 you will need to click on the "options" button at the top of the screen, move to the "General" tab, in the "Data Processing Options" section, press the button besides the "list events detected by >1 method" label until the label reads "list all events." If you now look at sequence "B" in the schematic sequence display, you should notice a grey block labelled "unknown" under the line representing this sequence: this block is representative of "recombination event 2".

Recombination event 3 is detected in sequences "J", "N" and "W" but it is clear that the sizes of the recombinationally derived fragments in these three sequences differ substantially. Whereas all three of the sequences have similar recombination signals across the beginning (or 5') breakpoint (approximately at position 5680), they all have completely different recombination signals across the ending (or 3') breakpoint. This likely indicates that following an initial recombination event subsequent recombination events have "overprinted" the 3' breakpoint (identified here as recombination events 5 and 8 in "J" and "W" respectively).

Recombination events 9 and on become progressively more difficult to interpret. This is primarily because they involve either recombination between very closely related sequences (such as recombination event 11 which has a 5' breakpoint with a high degree of uncertainty), or recombination between parental viruses that are only distantly related to sequences within the analysed dataset (such as recombination event 9 for which there is no sequence in the dataset that closely resembles the major parent).

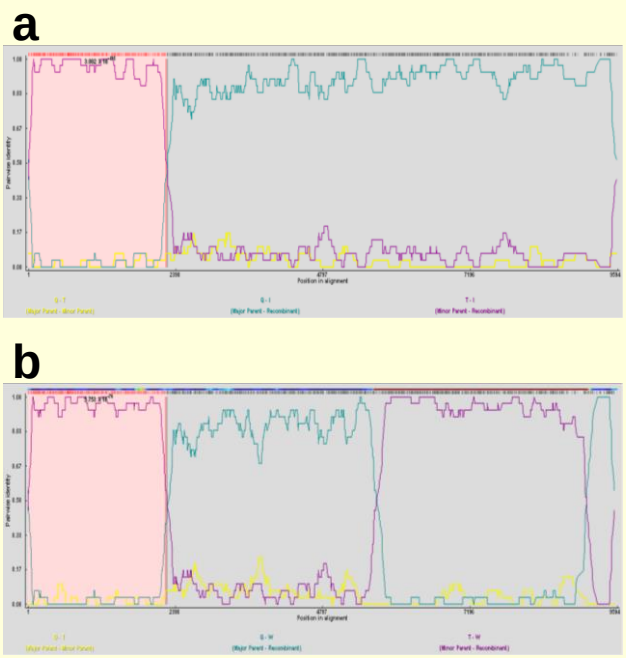

Figure 25. Comparing recombination signals to determine whether two recombinants are descended from a common recombinant ancestor. (**a**) An RDP method plot for recombination event number 1 in the example dataset. (**b**) A similar RDP method plot to that in A but with sequence "W" replacing sequence "I" in the scanned sequence triplet. The coloured line above the plot is a graphical representation of how closely the plot in B resembles that in A. Note that across the recombination breakpoints the two plots are nearly identical (the blue colour in the bar expresses this similarity) implying that "I" and "W" probably both descended from the same recombinant ancestor. Note also that in B the deep red colour in the part of the coloured line corresponding to sequence coordinates ~5000 to ~9000 clearly indicates that "W" likely carries evidence of a second large recombination event that is not shared with "I".

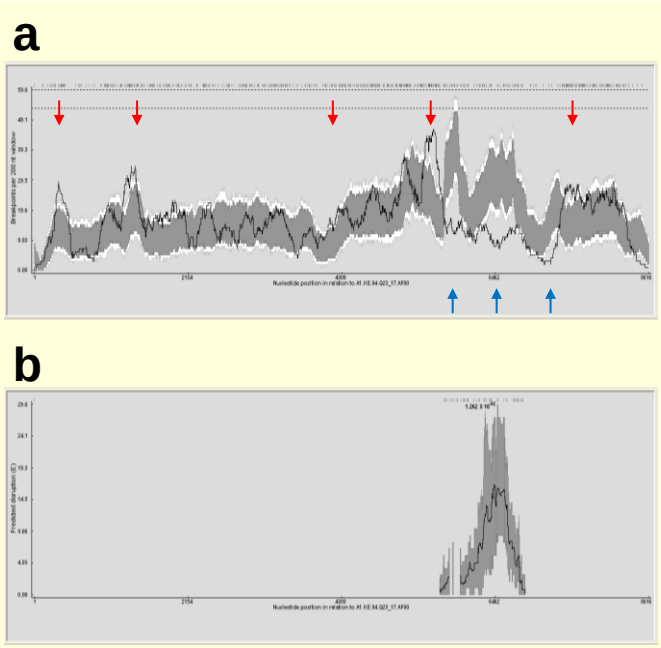

Figure 26. Recombination breakpoint distribution and recombination-induced protein folding disruption plots. (**a**) Recombination breakpoint hot- (red arrows) and cold-spots (blue arrows) detectable within HIV-1M genomes represented in the file, HIV Example.rdp (after Simon-Loriere wt al., 2009). The black plot represents the inferred clustering of recombination breakpoints identified within the HIV-1M sequences analysed here. Dark and grey areas respectively represent the 95% and 99% bounds of the expected degrees of breakpoint clustering under random recombination. Vertical lines above the plot indicate the positions of recombination breakpoints. (**b**) Expected degrees of folding disruption within chimaeric envelope proteins of simulated HIV-1M recombinants.(After Golden et al., 2014). Whereas the black line represents the mean folding disruptions inferred for the simulated chimaeric envelope proteins, the grey area represents the range of folding disruptions observed in these simulated proteins. The vertical black lines above the plots represent the locations of recombination breakpoint sites that were detected in the envelope genes of HIV genomes represented in file "HIV Example.rdp".

10.5.7. Further analyses. If large numbers of recombination breakpoints have been detected, you may want to test whether the distributions of these breakpoints indicate the presence of recombination hot- or cold-spots within the sequences that have been analysed. To demonstrate this, open the file "HIV Example.rdp" (it can be found in the directory where you have installed RDP5). Press the arrow beside the "Run" button and select the "breakpoint distribution plot" menu option. After a minute or two the program will display the plot indicated in (Fig. 26a). The black line in this plot represents the numbers of recombination breakpoints (individually indicated by vertical lines above the plot) that fall within 200 nucleotides of the genome coordinates indicated on the x-axis (in this case corresponding to nucleotide sites within the first sequence in the analysed dataset, A1.KE.94). The grey and white areas respectively represent the 95% and 99% confidence intervals of the expected degrees of breakpoint clustering under random recombination. Whereas genome coordinates at which the black line spikes up above the white/grey area are statistically supported recombination hot-spots, those where the black line dips below the white/grey area are statistically supported recombination cold-spots.

If you have a GenBank file on hand that corresponds with one of the sequences in a dataset that has been analysed for recombination, and this file contains information on the locations of gene boundaries, then it is also possible to test for associations between recombination breakpoint distributions and genome organisation. Once again, open the file "HIV Example.rdp". When it is loaded press the "Open" button again and select the file "HXB2 Genbank File.txt" (it can be found in the folder where you installed RDP5). This file simply contains a plain text version of the GenBank file for sequence "B.FR.83.HXB2" that is accessible using the following URL: http://www.ncbi.nlm.nih.gov/nuccore/K03455.1. When this file is loaded into RDP5 you should notice that a series of arrows are added to the coloured similarity map above the sequence display. Press the arrow beside the "Run" button and again select the "breakpoint distribution plot" menu option. This time, in addition to displaying the breakpoint distribution plot, extra information is tabulated in the recombination information panel in the top right. This extra information relates to the relative clustering of breakpoints between (1) coding and non-coding regions, (2) between different genes, and (3) between different parts of genes (Fig. 21). In this table the specific genome regions that are being compared in each row are graphically depicted in orange and blue. Numbers in the table are coloured according to the genome regions that they relate to. By examining the table, it is evident that: (a) detectable recombination breakpoints are significantly more clustered within coding regions than they are within non-coding regions ($p = 0.019$ in row one), (b) the gag and pol polyproteins contain significantly lower, and the env polyprotein significantly higher, densities of detectable recombination breakpoints than other HIV genes (see rows 2, 3 and 10) , and (c) detectable recombination breakpoints tend to occur significantly more frequently on the edges of genes than they do in the middle parts of genes (see the last three rows).

It is also possible to test whether the observed recombination breakpoints have tended to fall at locations within genes that have less impact on protein folding than would be expected under random recombination. However, this type of analysis requires that high resolution 3D protein structures are known for some of the proteins encoded by an analysed set of sequences. Once again, open the file "HIV Example.rdp". When it is loaded, press the arrow besides the "Run" button and select the "SCHEMA (protein folding disruption

test)" option. You will be prompted for a ".pdb" file. Select the file "HIV env structure (2B4C).pdb" (this file is in the directory where you installed RDP but can also be downloaded from Protein Data Bank at http://www.rcsb.org/pdb/home/home.do). You will then be asked whether you "would only like to consider accepted recombination events". Press the "Yes" button. A plot resembling that in Fig. 25b will be displayed on the bottom left and a table will be given on the top right program panel. In this case the p-value of ~0.013 indicates that recombination events that are found in HIV envelope genes have significantly less impact on the folding of HIV envelope proteins than would be expected under random recombination in the absence of any selection disfavouring the survival of recombinants with misfolded envelope proteins. The vertical lines above the plot indicate the locations of recombination breakpoints that fall within the parts of the envelope gene that correspond with the analysed structures. The plot itself represents the range of folding disruptions inferred for chimaeric envelope proteins that were randomly generated from pairs of parental sequences which resemble the parents of the actual recombinants considered during this analysis. Gaps in the plot indicate genome sites encoding amino acids that were not included in the 2B4C envelope structure. The peak in the plot between positions 6300 and 6650 indicates that recombinants with breakpoints falling between these genome coordinates are much more likely to express misfolded envelope proteins than recombinants with breakpoints falling in the remainder of the envelope gene. The low p-value indicated by this test implies that there is a significant tendency for breakpoint sites in the envelope genes of actual recombinant HIV genomes to fall outside the region where they will maximally disrupt protein folding.

11 RUNNING RDP5 FROM A COMMAND LINE

In the directory where you installed RDP5 is a version of the program, called RDP5CL.exe that can be used to run RDP5 from the command line. At the command prompt in the directory where you installed RDP5 (or in another directory but pointing to the directory where you installed RDP5) type:

RDP5CL.exe –f<inputfileame> -<option 1> -<option 2> -<option n>

Type the name of the file you want to analyse and not <inputfilename> and for <option x> you can use the following options:

Option	What it does
-am or -o	Optimize automasking for maximum recombination detection sensitivity
-nor	Do not output detailed analysis results in .rdp5 format (outputting a .rdp5 file is default)
-noc:	Do not output summary analysis results in .csv format (outputting a summary .csv file is default)
-dist	Save a recombination-free version of the input alignment where recombinant sequences are split into their constituent parts
-da	Save a recombination-free version of the input alignment where recombinant sequences are split into their constituent parts
-rsr	Save a recombination-free version of the input alignment with recombinant sequences removed
-rbr	Save a recombination-free version of the input alignment with recombinationally derived fragments of sequence removed
-sa	Split alignment at detected recombination breakpoint positions into recombination-free subregion alignments
-rbdp	Make recombination breakpoint distribution plots (tests for recombination hot-/cold-spots)
-rfmlt	Remove recombination-derived fragments of sequences and make a maximum likelihood tree (with the GTR-CAT model using RaXML)
-distml	Split recombinant sequences into constituent parts and make a maximum likelihood tree (with the GTR-CAT model using RaXML)

.

To change all other analysis settings, you need to start the graphical user interface version of RDP5 (i.e. RDP5.exe), change whatever settings you would like changed and then exit the program with the "exit" button. When you close the program in this way the analysis settings will automatically get stored in the RDP.ini file and these settings will thereafter be used whenever RDP5CL.exe is run from the same directory as the RDP5.ini file. RDP5CL.exe will give some indication of progress in the console window while it is running. By Default the program will output: (1) an RDP5 project file (".rdp5" extension) that can be opened in the graphical user interface version of RDP5 and will allow interactive exploration of recombination analysis results; and (2) a "comma separated value" file (.csv extension) that summarises the results and can be opened in a spreadsheet program like Microsoft Excel or Google sheets.

Note that it is also possible to run the graphical user interface version of RDP5 (i.e. the rdp5.exe) from the command line with the same command line parameters as RDP5CL.exe. However when using RDP5.exe from the command line it will not run in the same way that most other programs do. Immediately after executing RDP5.exe from the command line the program will immediately pass control back to the command prompt and will give no indication of its progress – a situation that for most other command-line programs indicates that the executed program has finished running. In order to determine when RDP5.exe has finished running it is necessary to check for the .rdp project file having been written to the program directory. This file will appear when the program has finished running. This behaviour enables one to run multiple instances of RDP5.exe in parallel from a standard windows batch file (i.e. a file with a .bat extension)

12 POSSIBLE PROBLEMS WITH USING RDP5

12.1 Poor Alignments

Badly aligned sequences will probably result in incorrect identification of recombinants. By default all recombination signals identified by RDP5 are checked for evidence of their being alignment artefacts. The tests that are used to do this are not perfect and they could potentially miss some false positive recombination signals that have arisen through poor alignment. It is always important, therefore to make sure that sequence alignments are of the best possible quality (see section 10.2 on how to make a good alignment). Whenever poor alignment cannot be avoided because the sequences being analysed are simply too divergent, it is very advisable that steps such as those outlined in Varsani et al. (2006) be taken to avoid overwhelmingly high false positive recombination detection rates.

12.2 Recombinants of Recombinants

If sequences that are used as references during analyses are themselves recombinant, RDP5 may incorrectly identify parental and recombinant sequences. RDP5 will, however, most likely still identify the correct region in which recombination has occurred. This error can be detected if the supposedly recombinant sequence is in the same tree position regardless of which part of the sequence has been used to draw the tree. Looking for changes in the tree position of one or a group of possible parental sequences will identify the recombinant parental sequence. In certain instances this "indirect" evidence for recombination in the parental sequence may be the only evidence RDP5 is able to find that the parental sequence is recombinant (i.e. it will not be able to give any probability measures, descriptions of parents and precise break-points). Carrying out the recommended supplementary analysis (see section 10.4) will be the only means of certifying whether sequences identified in this way are recombinant. Since RDP version 1.07 I have included various checks to detect incorrectly identified recombinant sequences. RDP now gives a warning if there is a fair chance that the recombinant indicated is not the correct recombinant. The checks are, however, themselves fallible and incorrect identification of recombinant sequences is still possible. If you notice that results obtained with the same analysis setting using versions of the program before the current one differ slightly/substantially from those obtained with the current version, it is likely that the current version has now correctly identified recombinant sequences that it had formerly misidentified as parental sequences in previous versions. See steps 6 to 8 in section 10.4 of the step-by-step guide above on how to deal with misidentification of recombinants.

12.3 Over-Grouping of Recombinants

RDP5 will tend to be overly conservative when it comes to counting the number of ancestral recombination events that have yielded the recombination signals that are detectable within a dataset. Specifically, it will frequently infer that recombinants with breakpoints in similar locations and/or similar parental sequences have descended from a common recombinant ancestor when in fact such recombinants

Table 2. Files included in the RDP5 installation.

File Name	Destination directory	Description
RDP5.x.exe	RDP	RDP5 by Darren Martin (x = update number)
RDP5.x.exe.manifest	RDP	Allows RDP5 to run as an administrator
RDP5Manual.pdf	RDP	This document
geneconv.exe	RDP	GENECONV 1.81. by Stanley Sawyer
SDTv1.exe	RDP	SDT1.0 by Brejnev Muhiri
SDTv1.exe.manifest	RDP	Allows SDT 1.0 to run as an administrator
padre.jar	RDP	Network drawing program by Martin Lott and
padre2.jar	RDP	Vincent Moulton
consense.exe	RDP	
consense2.exe	RDP	Phylip 3.5 components by Joe Felsenstein
fitch.exe	RDP	
dnapars.exe	RDP	
neighbour.exe	RDP	
lard.exe	RDP	LARD 2.2 (for Win95) by Andrew Rambaut
clustalw.exe	RDP	Alignment program by Thompson *et al.* (1994)
clustalw2.exe	RDP	
impale.zip	RDP	Alignment editor by Arjun khoosal
lkgen.exe	RDP	
convert.exe	RDP	
pairwise.exe	RDP	LDHat2.0 components by Gil McVean
interval.exe	RDP	
stat.exe	RDP	
phyml_win32.exe	RDP	Phyml 1.0 by Simon Guindon and Oliver Gascuel
phyml_3.0_win32.exe	RDP	Phyml 3.0 by Simon Guindon and Oliver Gascuel
Fasttree.exe	RDP	FastTree2.0 by Morgan Price
mrbayes.exe	RDP	MrBayes by Ronquist and Huelsenbeck
hybrid-ss-min.exe	RDP	UNAFOLD component by Nick Markham
catpv.exe	RDP	Consel0.2 components by Shimodaira &
consel.exe	RDP	Hasegawa
makermt.exe	RDP	
raxmlHPC.exe	RDP	RAxML 8.1 by Alexandros Stamatakis
raxmlHPC-pthreads.exe	RDP	
	RDP	
seq-gen.exe	RDP	Seq-Gen (for Win95) by Andrew Rambaut
LF0100	RDP	
LF250	RDP	Approx. likelihood lookup tables by Gil McVean
LF1100	RDP	
3seqTable	RDP	3Seq p-value lookup table by Maciej Boni
DNA5.dll	Win\Sys	RDP5 c++ command library by Darren Martin
DNA.dll	Win\Sys	RDP5 c++ command library by Darren Martin
MSVCRT40.dll	Win\Sys	Microsoft C runtime library
MFC40.dll	Win\Sys	Microsoft Foundation Class Library 4.1
cygwin1.dll	Win\Sys	UNIX emulator By Red Hat, Inc.
Eg GenBank.seq	RDP	Example GenBank file
HIV env structure.pdb	RDP	Example PDB file
HIV Example	RDP	Example .rdp file
PVY Example.exe	RDP	
Example1(PotySeqs).fas	RDP	Example Alignments
Example2(A-J-cons-kal153).fas	RDP	
Cour.ttf	Win'Font	Courier true type font
Comct232.ocx	Win\Sys	An ActiveX control by Microsoft Corp.
Comctl32.ocx	Win\Sys	An ActiveX control by Microsoft Corp.
Comdlg32.ocx	Win\Sys	An ActiveX control by Microsoft Corp.
Threed32.ocx	Win\Sys	32 bit OLE control by Microsoft Corp.
MSVBVM50.dll	Win\Sys	Microsoft Visual Basic Virtual Machine 5.0
StdOle2.tlb	Win\Sys	
OleAut32.dll	Win\Sys	Microsoft OLE 2.40
OlePro32.dll	Win\Sys	
AsycFilt.dll	Win\Sys	
Ctl3d32.dll	Win\Sys	3D Windows Controls 2.31 by Microsoft Corp.
ComCat.dll	Win\Sys	Microsoft Component Category Manager 5.0

likely (and often obviously) arose from two or more unique recombination events. See steps 9 to11 in section 10.4 of the step-by-step guide above on how to deal with over/under grouping of recombination events.

12.4 Degeneracies

RDP5 does not handle degeneracies. When loading sequences any characters other than A,C,G and T will be replaced with "-" characters. The main reason for this is that handling these seriously slows down many of the analysis methods. If this is a big problem you should deleted the affected columns of the alignment.

12.5 Software Crashes/File Incompatibility

While RDP5 in my hands is relatively stable (I've corrected all the bugs that I've encountered during its use) there are a lot of settings that have not been thoroughly tested and I cannot guarantee its stability in the hands of others. Also, while I am able to load files in all the alignment formats that I frequently use, I cannot be certain that the formatting of files produced by other programs (or even versions of the software that I use but am unfamiliar with) will work with RDP5. Should you encounter any technical problems with the software I would really appreciate you telling me at darrenpatrickmartin@gmail.com. I can only fix the problems that I know about and I promise to sort them out as quickly as I can. Remember that program crashes could occur at any time so you should regularly save your results.

12.6 Crashes When Using Windows VISTA/7/8

If RDP persistently crashes in Windows VISTA/7/8 try doing the following:

1. Copy a shortcut to the desktop
2. Right click on the shortcut icon and select the "properties" option on the menu that appears.
3. Select the "compatibility" tab.
4. Tick the box which will give RDP5 administrator rights.

12.7 Crashes When Pressing the "Options" Button

If RDP crashes whenever you press the "Options" button it may be because you are using a version of Windows where commas (",") are used as decimal separators rather than points (".") This is standard for many European versions of windows. To run RDP5 you may need to change your language settings to English. To do this:

1. Go to the control panel and select the "Regional and Language" options icon.
2. On the regional options tab either (a) under the "number" heading change "123.456.789,00" to "123,456,789.00" or (b) change the total language setting to an "English" one.

13 ACKNOWLEDGEMENTS

A lot of people have been either peripherally or directly involved in the development of RDP5. These include Ed Rybicki, Carolyn Williamson, Brejnev Muhiri, Ben Murrell, Pierre Lefeuvre, Philippe Lemey, Martin Lott, Vincent Moulton, David Posada, Maciej Boni, Arvind Varsani, Adrian Gibbs, Mark Gibbs, Andrew Rambaut, Oliver Gascuel, Joe Felsenstein, Fredrick Ronquest, George Weiller, Livio Heath, Morgan Price, Robert Beiko Nick Markham and Hidetoshi Shimodaira.

14 APPENDIX

14.1 Program Files

When you install RDP5 you will notice a lot of files being copied onto your computer. This may worry certain people. Many of the "unusual" files are simply common windows components that you may/may not already have and the so-called Visual Basic "run time" files. The installation will not overwrite windows component files that are more recent versions than the ones shipped with RDP5. The files that are included in the RDP5 installation and their destination directories are listed in Table 2. Some users have expressed concern that files are copied to the windows/system directory. I have now set up the installation so that 16 files will be copied to the system directory. For RDP5 to be successfully installed it is unfortunately absolutely required that these files reside in the system directory.

14.2 Citing RDP5 and the Methods Implemented Therein

If RDP5 is used to generate publishable graph's or other results it is important that you cite the appropriate publications. The publications you should cite will depend on the parts of RDP5 that you use. You should always cite the paper describing RDP5:

Martin DP, Murrell B, Golden M, Khoosal A, & Muhire B (2015) RDP4: Detection and analysis of recombination patterns in virus genomes. Virus Evolution 1: vev003 doi: 10.1093/ve/vev003

If you use any of the following methods in RDP5 to obtain publishable results you should cite the indicated references:

The RDP method: Martin, D. & Rybicki, E. (2000). RDP: detection of recombination amongst aligned sequences. Bioinformatics 16, 562-563.
The GENECONV method: Padidam, M., Sawyer, S. & Fauquet, C. M. (1999). Possible emergence of new geminiviruses by frequent recombination. Virology 265, 218-225.

The BOOTSCAN/RECSCAN method: Martin, D. P., Posada, D., Crandall, K. A. & Williamson, C. (2005). A modified BOOTSCAN algorithm for automated identification of recombinant sequences and recombination breakpoints. AIDS Res Hum Retroviruses 21, 98-102.

The MAXCHI method: Maynard Smith, J. (1992). Analyzing the mosaic structure of genes. J Mol Evol 34, 126-129.

The CHIMAERA method: Posada, D. & Crandall, K. A. (2001). Evaluation of methods for detecting recombination from DNA sequences: Computer simulations. Proc Natl Acad Sci 98, 13757-13762.

The SISCAN method: Gibbs, M. J., Armstrong, J. S. & Gibbs, A. J. (2000). Sister-Scanning: a Monte Carlo procedure for assessing signals in recombinant sequences. Bioinformatics 16, 573-582.

The 3Seq method: Lam H.M., Ratmann O., Boni M.F. (2018). Improved algorithmic complexity for the 3SEQ recombination detection algorithm. Mol Biol Evol, 35, 247-251.

The LARD method: Holmes E.C., Worobey, M. & Rambaut,A. (1999). Phylogenetic evidence for recombination in dengue virus. Mol Biol and Evol 16, 405-409.

The Topal/DSS method: McGuire, G. & Wright,F. (2000). TOPAL 2.0: Improved detection of mosaic sequences within multiple alignments. Bioinformatics 16, 130-134.

The PHYLPRO method: Weiller, G.F. (1998). Phylogenetic profiles: a graphical method for detecting genetic recombinations in homologous sequences. Mol Biol Evol 15, 326-335.

The VisRD method: Lemey P, Lott M, Martin DP, Moulton V. (2009). Identifying recombinants in human and primate immunodeficiency virus sequence alignments using quartet scanning. BMC Bioinformatics 10, 126.

The PHI test: Bruin TC, Philippe H, Bryant D. (2006). A simple and robust statistical test for detecting the presence of recombination. Genetics 172, 2665-2681.

Recombination count matrices/Protein SCHEMA: Lefeuvre, P., Lett, J.M., Reynaud, B., Martin, D.P. (2007). Avoidance of protein fold disruption in natural virus recombinants. PLoS Pathog 11:e181.

Nucleic acid SCHEMA: Golden M, Muhire BM, Semegni Y, Martin DP. 2014. Patterns of recombination in HIV-1M are influenced by selection disfavouring the survival of recombinants with disrupted genomic RNA and protein structures. PLoS One. 9:e100400.

Recombination breakpoint hotspot plots: Heath, L., van der Walt, E., Varsani, A. & Martin D.P. (2006). Recombination patterns in aphthoviruses mirror those found in other picornaviruses. J Virol 80, 11827-11832.

Recombination rate plots: McVean, G. A. T., Myers, S. R., Hunt, S., Deloukas, P., Bentley, D. R. & Donnelly, P. (2004). The Fine-Scale Structure of Recombination Rate Variation in the Human Genome. Science 304, 581-584.

RMin/LD matrices: McVean, G., Awadalla, P. & Fearnhead, P. (2002). A Coalescent-Based Method for Detecting and Estimating Recombination From Gene Sequences. Genetics 160, 1231-1241.

Neighbor joining or least squares trees: Felsenstein, J. (1989). PHYLIP – Phylogeny inference package (version 3.2). Cladistics 5, 164–166.

Maximum likelihood trees (PHYML): Guindon, S. & Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst Biol 52, 696-704.

Maximum likelihood trees (RAxML): Stamatakis, S. (2006). RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22, 2688–2690.

Maximum likelihood trees (FastTree): Price, M. N., Dehal, P. S., Arkin, A.P. (2010) FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. PLoS ONE 5, e9490.

Bayesian trees: Ronquist F, Teslenko M, van der Mark P, Ayres DL, Darling A, Höhna S, Larget B, Liu L, Suchard MA, Huelsenbeck JP. 2012. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. Syst Biol. 61:539-542.

15 REFERENCES

Beiko RG, Hamilton N. 2006. Phylogenetic identification of lateral genetic transfer events*. BMC Evol Biol*. **6:**15.

Bruin TC, Philippe H, Bryant D. 2006. A simple and robust statistical test for detecting the presence of recombination. *Genetics* **172:** 2665-2681.

Drouin G, Prat F, Ell M, Clarke GDP. 1999. Detecting and characterizing gene conversions between multigene family members. *Mol Biol Evol.* **16:** 1369-1390.

Edgar RC. 2004. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics.* **5:** 113.

Fang F, Ding J, Minin VN, Suchard MA, Dorman KS. 2007. cBrother: relaxing parental tree assumptions for Bayesian recombination detection. *Bioinformatics.* **23:**507-508.

Felsenstein J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics.* **5:**164-166.

Felsenstein J. 1984. Distance methods for inferring phylogenies: A justification. *Evolution* **38:**16-24.

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol.* **17:**368-376.

Gibbs M.J., Armstrong JS and Gibbs AJ. 2000. Sister-Scanning: a Monte Carlo procedure for assessing signals in recombinant sequences. *Bioinformatics.* **16:** 573-582.

Golden M, Muhire BM, Semegni Y, Martin DP. 2014. Patterns of recombination in HIV-1M are influenced by selection disfavouring the survival of recombinants with disrupted genomic RNA and protein structures. *PLoS One*. **9**:e100400.

Guindon S, Gascuel O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol.* **52**: 696-704.

Hasegawa M, Kishino H, Yano T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol.* **22:**160-174.

Heath, L., van der Walt, E., Varsani, A. & Martin D.P. (2006). Recombination patterns in aphthoviruses mirror those found in other picornaviruses. *J Virol*. **80:**11827-11832.

Holmes EC, Worobey M, Rambaut A. 1999. Phylogenetic evidence for recombination in Dengue virus. *Mol Biol Evol.* **16:**405.

Hudson RR, Kaplan N. 1985 Statistical properties of the number of re-combination events in the history of a sample of DNA sequences. *Genetics.* **111:**147–164.

Jakobsen IB, Easteal S. 1996. A program for calculating and displaying compatibility matrices as an aid in determining reticulate evolution in molecular sequences. *CABIOS.* **12:**291-295.

Jakobsen IB, Wilson SR, Easteal S. 1997. The partition matrix: exploring variable phylogenetic signals along nucleotide sequence alignments. *Mol Biol Evol.* **14:**474-484.

Jukes TH, Cantor CR. 1969. in Mammalian Protein Metabolism, ed. H. N. Munro. (Academic Press, New York) Vol. III, pp. 21-132.

Karlin S, Altschul SF. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Nat Acad Sci USA.* **87:**2264-2268.

Kimura M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol.* **16:**111-120.

Kumar S, Nei M, Dudley J, Tamura K. 2008. MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences. *Brief Bioinformatics.* **9:**299-306.

Lam HM, Ratmann O, Boni MF. 2018. Improved algorithmic complexity for the 3SEQ recombination detection algorithm. *Mol Biol Evol*, **35**: 247-251.

Lanave C, Preparata G, Saccone C, Serio G. 1984. A new method for calculating evolutionary substitution rates. *J Mol Evol.* **20:**86-93.

Lemey P, Lott M, Martin DP, Moulton V. 2009. Identifying recombinants in human and primate immunodeficiency virus sequence alignments using quartet scanning. *BMC Bioinformatics*. **10**:126.

Lefeuvre P, Lett JM, Reynaud B, Martin DP. 2007. Avoidance of protein fold disruption in natural virus recombinants. *PLoS Pathog.* **3:**e181.

Lefeuvre P, Lett JM, Varsani A, Martin DP. 2009. Widely conserved recombination patterns among single-stranded DNA viruses. *J Virol.* **83**:2697-2707.

Lole KS, Bollinger RC, Paranjape RS, Gadkari D, Kulkarni SS, Novak NG, Ingersoll R, Sheppard HW, Ray SC. 1999. Full-length human immunodeficiency virus type 1 genomes from subtype C-infected seroconverters in India, with evidence of intersubtype recombination*. J Virol*. **73:**152-160.

Markham NR, Zuker M. 2008. UNAFold: software for nucleic acid folding and hybridization. Methods in Molecular Biology 453:3-31.

Martin D, Rybicki E. 2000. RDP: detection of recombination amongst aligned sequences. *Bioinformatics.* **16**:562-563.

Martin DP, Williamson C, Posada D. 2005a. RDP2: recombination detection and analysis from sequence alignments. *Bioinformatics.* **21**: 260-262.

Martin DP, Posada D, Crandall KA, Williamson C. 2005b. A modified BOOTSCAN algorithm for automated identification of recombinant sequences and recombination breakpoints. *AIDS Res Hum Retroviruses.* **21**:98-102.

Martin DP, van der Walt E, Posada D, Rybicki EP. 2005c. The evolutionary value of recombination is constrained by genome modularity. *PLoS Genet.* 1:e51.

Martin DP, Lemey P, Posada D. 2011. Analysing recombination in nucleotide sequences. *Mol Ecol Resour*. **11**:943-955.

Maynard Smith J. 1992. Analyzing the mosaic structure of genes. *J Mol Evol.* **34:**126-129.

McLeod D, Charlebois RL, Doolittle F, Bapteste E. 2005. Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement. *BMC Evol. Biol.* **5:**27-37.

McGuire G, Wright F. 2000. TOPAL 2.0: improved detection of mosaic sequences within multiple alignments. *Bioinformatics.* **16:**130-134.

McGuire G, Wright F. 1998. TOPAL: Recombination detection in DNA and protein sequences. *Bioinformatics*. **14:**219-220.

McVean G, Awadalla P, Fearnhead P. 2002. A Coalescent-Based Method for Detecting and Estimating Recombination From Gene Sequences. *Genetics* **160**:1231-1241.

McVean, GAT, Myers SR, Hunt S, Deloukas P, Bentley DR and Donnelly P. 2004. The Fine-Scale Structure of Recombination Rate Variation in the Human Genome. *Science.* **304**:581-584.

Muhire BM, Varsani A, Martin DP. 2014. SDT: a virus classification tool based on pairwise sequence alignment and identity calculation. *PLoS One*.**9**:e108277.

Padidam M, Sawyer S, Fauquet CM. 1999. Possible emergence of new geminiviruses by frequent recombination. *Virology.* **265:**218-225.

Page RD. 1996. TreeView: an application to display phylogenetic trees on personal computers. *Comput Appl Biosci* 12:357-358.

Posada D, Crandall KA. 1998. MODELTEST: testing the model of DNA substitution. Bioinformatics.14:817-188.

Posada D. 2002. Evaluation of methods for detecting recombination from DNA sequences: empirical data. *Mol Biol Evol.* **19:**708-717.

Posada D, Crandall KA. 2001. Evaluation of methods for detecting recombination from DNA sequences: Computer simulations. *Proc. Natl Acad Sci USA.* **98:**13757-13762.

Price MN, Dehal PS, Arkin AP. 2010 FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. PLoS ONE **5**, e9490.

Rambaut A, Grassly NC. 1997. Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput Appl Biosci* **13:**235-238.

Rodrıguez F, Oliver J L, Marin A, Medina JR. 1990. The general stochastic model of nucleotide substitution. *J Theor Biol* **142:**485–501.

Ronquist F, Teslenko M, van der Mark P, Ayres DL, Darling A, Höhna S, Larget B, Liu L, Suchard MA, Huelsenbeck JP. 2012. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Syst Biol*. **61**:539-542.

Rousseau CM, Learn GH, Bhattacharya T, Nickle DC, Heckerman D, Chetty S, Brander C, Goulder PJ, Walker BD, Kiepiela P, Korber BT, Mullins JI. 2007. Extensive intrasubtype recombination in South African human immunodeficiency virus type 1 subtype C infections. *J Virol.* **81**:4492-4500.

Salminen MO, Carr JK, Burke DS, McCutchan FE. 1995. Identification of breakpoints in intergenotypic recombinants of HIV type 1 by BOOTSCANning. *AIDS Res Hum Retroviruses* **11:**1423-1425.

Sawyer S. 1989. Statistical tests for detecting gene conversion. *Mol. Biol. Evol.* **6:**526-538.

Shimodaira, H. & Hasegawa, M. 2001. CONSEL: for assessing the confidence of phylogenetic tree selection. *Bioinformatics* **17**:1246-1247.

Simmonds P, Welch J. 2006. Frequency and dynamics of recombination within different species of human enteroviruses. *J Virol*. **80**:483-493.

Simon-Loriere E, Galetto R, Hamoudi M, Archer J, Lefeuvre P, Martin DP, Robertson DL, Negroni M. 2009. Molecular mechanisms of recombination restriction in the envelope gene of the human immunodeficiency virus. *PLoS Pathog.* 5:e1000418.

Simon-Loriere E, Martin DP, Weeks KM, Negroni M. 2010. RNA structures facilitate recombination-mediated gene swapping in HIV-1. *J Virol.*.**84**:12675-12682.

Stamatakis S. 2006. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* **22**:2688–2690.

Stamatakis A. 2014 RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 30:1312-1313.

Strimmer K, Forslund K, Holland B, Moulton V. 2003. A novel exploratory method for visual recombination detection. *Genome Biol.* **4**:R33.

Tamura K, Nei M. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol Biol Evol*. **10:**512-526.

Tavare S. 1986. Some probabilistic and statistical problems in the analysis of DNAsequences. *Lect Math Life Sci* **17:**57-86.

Thompson JD, Higgins DG, Gibson TJ. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22:**4673-4680.

Varsani A, van der Walt E, Heath L, Rybicki EP, Williamson AL, Martin DP. 2006 Evidence of ancient papillomavirus recombination. *J Gen Virol.* **87:**2527-2531.

Voigt CA, Martinez C, Wang ZG, Mayo SL, Arnold FH. 2002. Protein building blocks preserved by recombination. *Nat Struct Biol*. **9:**553-558.

Weiller GF. 1998. Phylogenetic profiles: a graphical method for detecting genetic recombinations in homologous sequences. *Mol. Biol. Evol.* **15:**326-335.