16

Detecting and characterizing individual recombination events

THEORY

Mika Salminen and Darren Martin

16.1 Introduction

In addition to point mutation, the most important mechanisms whereby organisms generate genomic diversity are undoubtedly nucleic acid recombination and chromosomal ***reassortment***. Although this chapter will mostly deal with the characterization of true recombination events, many of the methods described here are also applicable to the detection and description of reassortment events. Moreover, we will focus on ***homologous recombination***, which is defined as the exchange of nucleotide sequences from the same genome coordinates of different organisms. Although ***heterologous recombination*** (i.e. recombination resulting in either the physical joining of segments in unrelated genes or gross insertion/deletion events) will not be considered, many of the methods described in this chapter may also be applicable to the detection and analysis of this form of genetic exchange.

Whereas the genomes of all cellular organisms and some viruses are encoded in DNA, many viruses use RNA as their genetic material. The importance of recombination in the evolution of life on earth is underlined by the fact that various mechanisms for both DNA and RNA genomic recombination have evolved. Well-studied recombination mechanisms mediating double-stranded DNA break repair and/or chromosomal recombination during meiotic cell division are extremely common amongst cellular organisms and it is believed that DNA viruses also access these mechanisms during nuclear replication within infected host cells.

The Phylogenetic Handbook: a Practical Approach to Phylogenetic Analysis and Hypothesis Testing, Philippe Lemey, Marco Salemi, and Anne-Mieke Vandamme (eds.). Published by Cambridge University Press. © Cambridge University Press 2009.

Conversely, the recombination mechanisms used by RNA viruses and retroviruses (whose genomes pass intermittently through DNA and RNA phases), are generally encoded by the viruses themselves. For example, two features of retrovirus life cycles that greatly facilitate recombination are packaging of two RNA genomes within each virus particle (i.e. they are effectively ***diploid***), and the predisposition of retroviral reverse transcriptases to periodically drop on and off these RNA molecules during DNA synthesis. If the reverse transcriptase drops off one of the RNA molecules and reinitiates DNA strand elongation on the other, and the two RNA molecules are genetically different, then the newly synthesized DNA molecule will have a mixed ancestry and will therefore be effectively recombinant.

Given the biological and evolutionary significance of recombination and the real probability that recombination features in the recent histories of most DNA sequences on Earth, it is perhaps surprising that so many (if not most) evolutionary analysis methods in common use today assume that nucleotide sequences replicate without recombining. The inescapable fact that a recombinant nucleic acid sequence has more than one evolutionary history implies that recombination will have at least some effect on any sequence analysis method that assumes correctly inferred evolutionary relationships (see Box 15.2 in the previous chapter). It is probably for this reason that an enormous number of recombination detection and analysis methods have been devised (Table 16.1 and see *http://www.bioinf.manchester.ac.uk/recombination/programs.shtml* for a reasonably up-to-date list of the computer software that implements most of these methods). This chapter will briefly discuss how some of these methods can be practically applied to identify and characterize evidence of individual recombination events from multiple alignments of recombining nucleotide sequences.

16.2 Requirements for detecting recombination

The vast majority of recombination events leave no trace on the recombinant molecule that is generated. For an individual recombination event to be detectable, the recombining or parental sequences must differ at two or more nucleotide positions. In practice, however, proof that a recombination event has occurred usually involves a statistical or phylogenetic test to determine whether or not a potential recombinant has a non-clonal ancestry. To detect and characterize individual recombination events, such tests demand considerable amounts of sequence data that must meet certain criteria. Generally they require: (1) sampling of a recombinant sequence and at least one sequence resembling one of the recombinant's parental sequences; (2) that the parental sequences are different enough that at least one of the two sequence tracts inherited by the recombinant contains sufficient

Table 16.1 Available software tools for characterizing individual recombination events

Program	Method(s) implemented	References
3Seq	3Seq	Boni *et al.*, 2007
Barce	Barce	Husmeier & McGuire, 2003
DualBrothers	DualBrothers	Minin *et al.*, 2005
Gard	Gard	Kosakovsky Pond *et al.*, 2006
Geneconv	Geneconv	Sawyer, 1989
Jambe	Jambe	Husmeier & Wright, 2001
JpHMM	Jphmm	Shultz *et al.*, 2007
Lard	Lard	Holmes *et al.*, 1999
Maxchi	Maxchi	Posada & Crandall, 2001; Maynard Smith, 1992
Phylpro	Phylpro	Weiller, 1998
Pist	Pist	Grassly & Holmes, 1997
Plato	Plato	Woroby, 2001
Rat	Rat	Etherington *et al.*, 2005
Recpars	Recpars	Hein, 1993
Rega	Rega	de Oliveira *et al.*, 2005
Rdp3	RDP, Geneconv, 3Seq, Bootscan, Maxchi, Chimaera, Dss, Siscan, Phylpro, Lard	Martin *et al.*, 2004
Rip	Rip	Siepel *et al.*, 1995
Simplot	Simplot, Bootscan	Lole *et al.*, 1999; Salminen *et al.*, 1995
Siscan	Siscan	Gibbs *et al.*, 2000
Topal	Dss	McGuire & Wright, 2000
TOPALi	Dss, Barce, Jambe	Milne *et al.*, 2004

polymorphisms to unambiguously trace its origin to a parental lineage; (3) that the distribution of polymorphisms inherited by the recombinant from its parental sequences cannot be credibly accounted for by convergent point mutation (see Box 15.1 in the previous chapter); (4) that the recombination event has not occurred so long ago that the distinguishing pattern of polymorphisms created by the event has not been erased by subsequent mutations.

A good illustration of the importance of each of these factors can be seen when attempting to detect recombination events in HIV sequences. In the case of detecting recombination between HIV-1M subtypes (the viruses responsible for the vast majority of HIV infections worldwide), all of these requirements are met. (1) The over 600 publicly available full HIV-1M genome sequences provide ample data for recombination analysis; (2) following the introduction of HIV-1M into humans, founder effects in the epidemiological history of HIV-1 group M allowed enough distinguishing genetic variation to occur between the subtype lineages

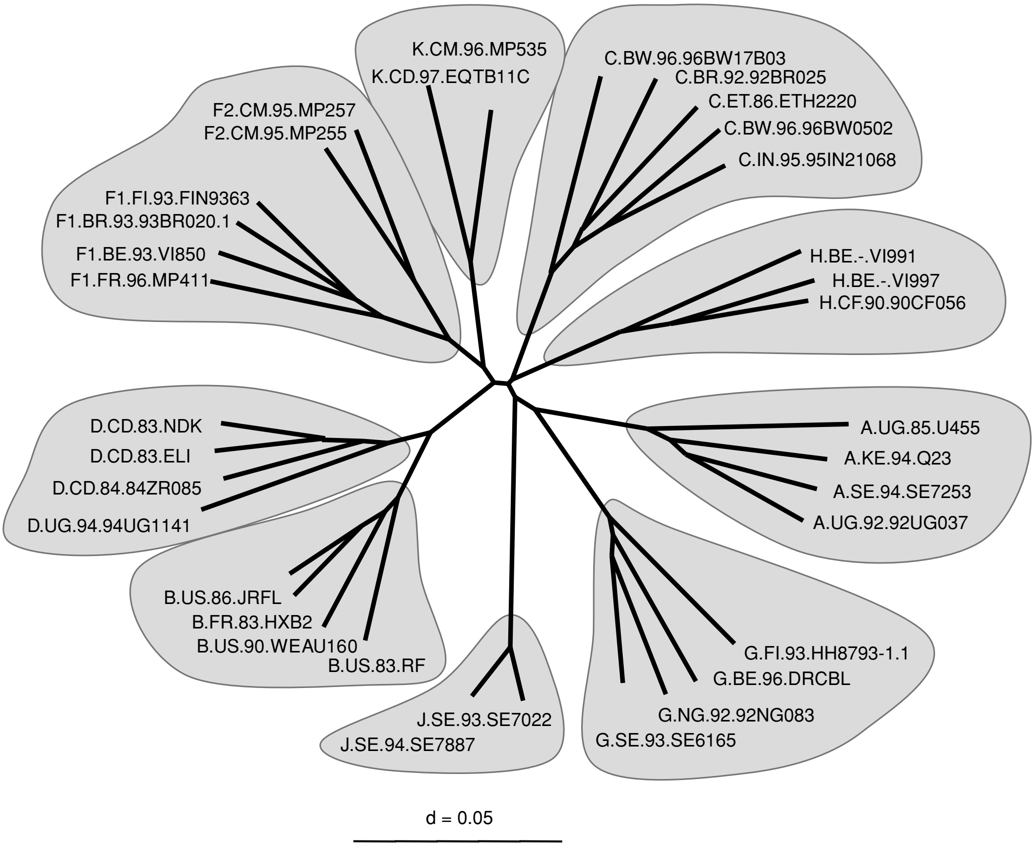


Fig. 16.1 HIV subtypes. K2P model Neighbor-Joining phylogenetic tree using the 1999/2000 Los Alamos HIV database complete genome reference sequence alignment (*http://hiv-web.lanl.gov*). Strain-name coding: X.CC.00.YYYY with X = Subtype, CC = two-letter country code, 00 = year of sampling, YYYY = original isolate identifier. Note the nine discrete groups of sequences.

(Fig. 16.1) that the origins of many sequence tracts found in today's inter-subtype recombinants can be quite easily traced; (3) the recombination mechanism yielding HIV recombinants often results in "exchanges" of sequence tracts large enough to contain sufficient polymorphisms that clusters of polymorphisms characteristic of different subtypes within a single sequence cannot be reasonably explained by convergent evolution; (4) mosaic polymorphism patterns that characterize the many inter-subtype recombinants that have emerged in the wake of relatively recent widespread epidemiological mixing of HIV-1 subtypes are still largely unobscured by subsequent substitutions.

The ease with which inter-subtype HIV-1M recombinants can be detected is starkly contrasted with the difficulty of characterizing individual recombination events that have occurred between sequences belonging to the same HIV-1M subtypes – so-called intra-subtype recombination. While there are many publicly available sequences for most of the subtypes and the sequences within each subtype are sufficiently divergent for intra-subtype recombinants to be detectable, it is still very difficult to accurately characterize intra-subtype recombination events because: (1) within subtypes, phylogenetic trees generally show star-like structures lacking clusters with enough distinguishing genetic variation. Such trees can be expected for exponentially growing viral epidemics. However, also recombination makes structured trees appear more star-like, which aggravates the problem. (2) Many of the mosaic polymorphism patterns of older intra-subtype recombinants may have been obscured by subsequent mutations.

16.3 Theoretical basis for recombination detection methods

Again, using HIV recombination as an example, inter-subtype recombination can be graphically illustrated using phylogenetic analyses. If separate phylogenetic trees are constructed using sequences corresponding to the tracts of a recombinant sequence inherited from its different parents, the recombinant sequence will apparently "jump" between clades when the two trees are compared (Fig. 16.2). Most methods developed to detect specific recombination events and/or map the positions of recombination breakpoints apply distance- or phylogenetic-based methods to identify such shifting relationships along the lengths of nucleotide sequence alignments. We will hereafter refer to these shifts or jumps in sequence relatedness as "recombination signals."

There are a variety of ways in which recombination signals are detected. The most common are variants of an average-over-window-based scanning approach. Some measure of relatedness is calculated for a set of overlapping windows (or alignment partitions) along the length of an alignment (Fig. 16.3) and evidence of recombination is obtained using some statistic that compares the relatedness measures calculated for different windows. Every scanning window approach is, however, a simplification of a more computationally intense analysis-of-every-possible-alignment-partition approach. Scanning window approaches are generally quite fast because relatedness measures are only calculated for a relatively small proportion of all possible alignment partitions and sometimes only adjacent windows are compared. All recombination signal detection approaches that do not employ scanning windows have some other (occasionally quite complex) means of first selecting the partitions (from all those possible) for which relatedness measures are calculated, and then selecting which of the examined partitions are statistically compared with one another. Generalizing, therefore, recombination signal detection involves partitioning of alignments, calculation of relatedness measures for

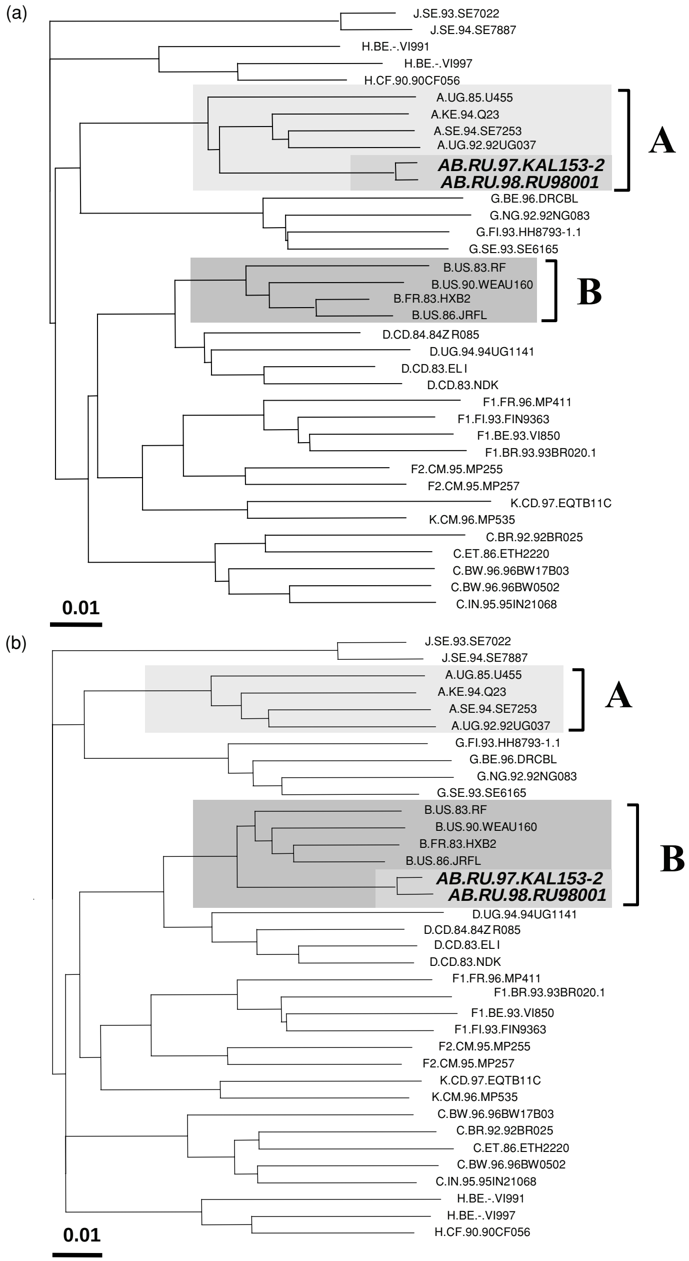


Fig. 16.2 "Jumping" of recombinant isolates. Two representatives (AB.RU.97.KAL153-2 and AB.RU.98.RU98001, in boldface) of a recombinant between HIV-1 M-group subtypes A and B were phylogenetically analyzed in two different genome regions. The trees represent a region clustering the recombinant isolates in (a) with subtype A and in (b) with subtype B. Reference sequences are from the 1999/2000 Los Alamos HIV-1 database.

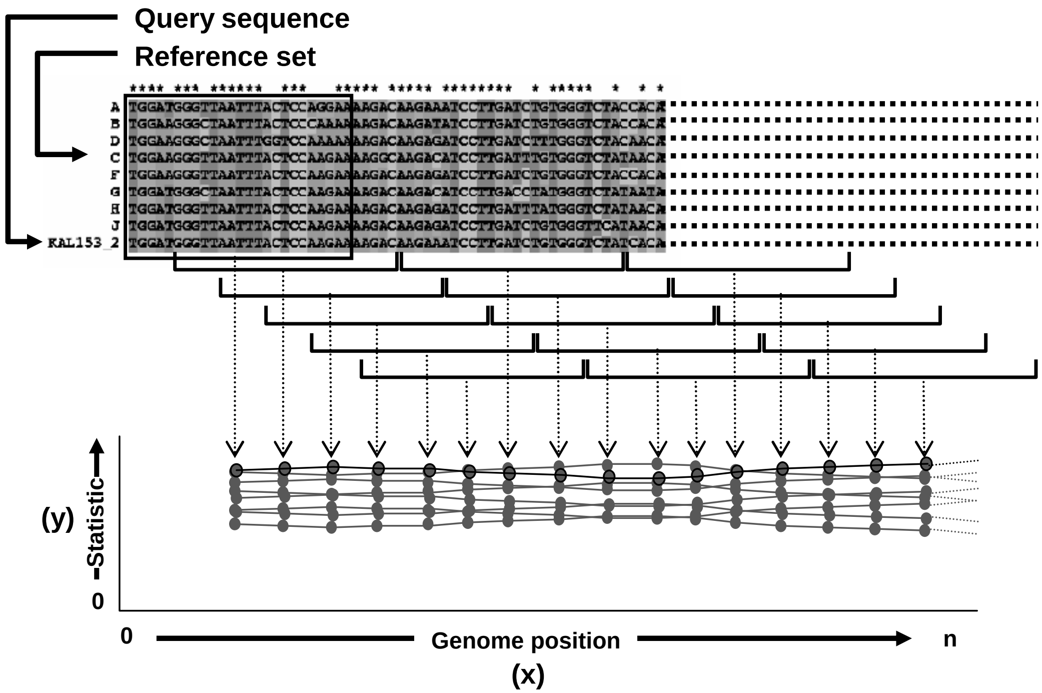


Fig. 16.3 Basic principle of average-over-window scanning methods. An alignment of reference sequences (A–J) and a potential recombinant (the query sequence) are sequentially divided into overlapping sub-regions (i.e. window, box, and brackets) for which a statistic/measure is computed. This statistic/measure is then plotted (broken arrow lines) in an *x*/*y* scatter plot using the alignment coordinates on the *x*-axis and the statistic/measure range on the *y*-axis. Each value is recorded at the midpoint of the window and connected by a line.

each partition, and use of some statistic determined from these measures to identify pairs or sets of partitions where recombination signals are evident.

There are many possible relatedness measures and statistics that could be used to compare partitions. The simplest and computationally quickest to compute measures are pairwise ***genetic distances*** between the sequences in an alignment. These are based on the reasonable (but not always true) assumption that any given sequence will be most similar to whichever sequence it shares a most recent common ancestor with. Therefore, if the genetic distances of every sequence pair are computed along an alignment using, for example, a sliding-window approach, the relative distances between non-recombinant sequences should remain consistent across all alignment partitions (Fig. 16.4). If recombination occurs, however, the relative distances between the recombinant sequence and sequences closely related to one or both of its parental sequences might shift from one partition to the next, with the point at which the shift occurs indicating the recombination breakpoint.

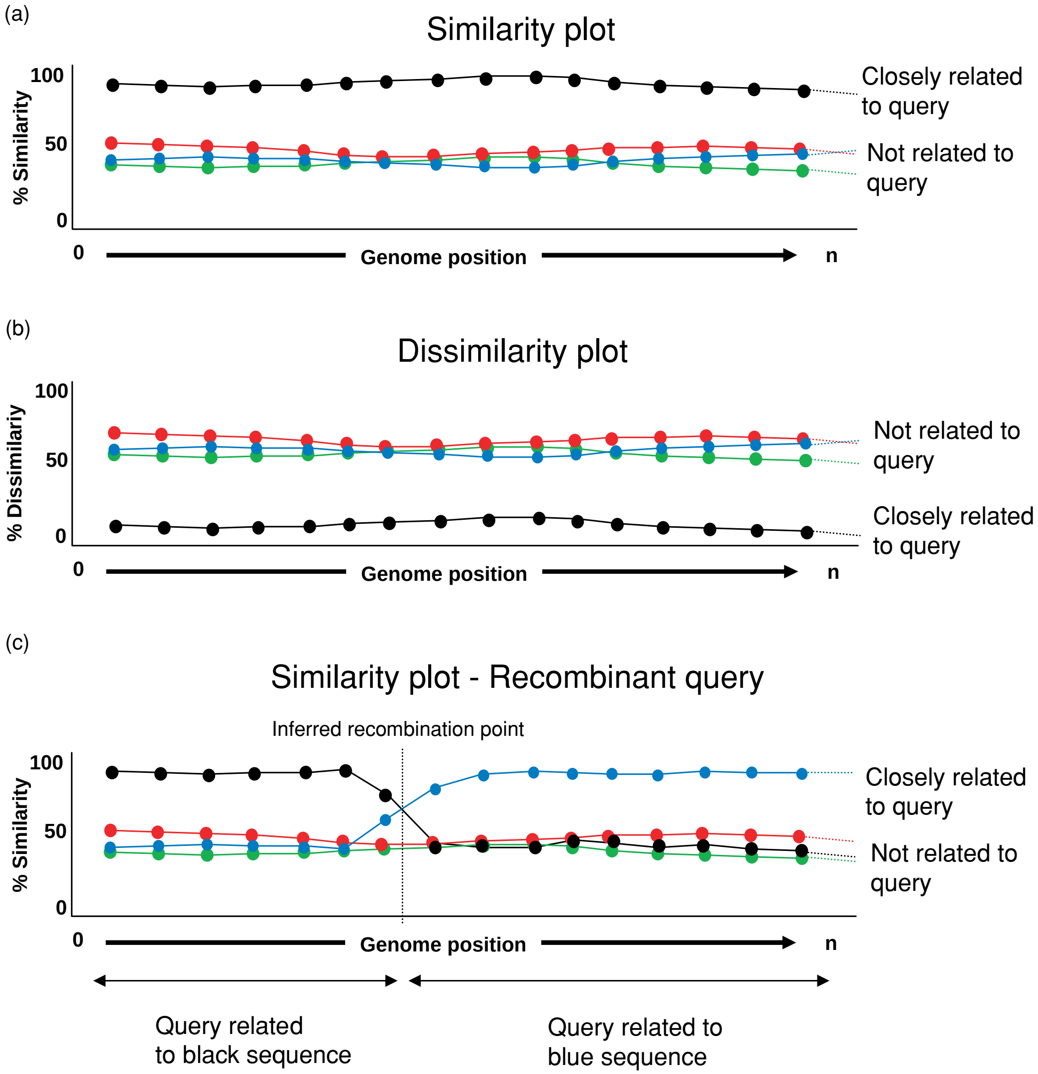


Fig. 16.4 Similarity and dissimilarity methods. (a) Similarity plot. In this type of analysis, the measure recorded is the similarity value (sim) between the query sequence and each of the reference sequences. (b) The same analysis, except that the inverse value to similarity (1–sim) or the dissimilarity is plotted. In various methods, similarity/dissimilarity values corrected by an evolutionary model, i.e. any of the JC69, TN93, K80, F81, F84, HKY85, or GTR models (see Chapter 4 and Chapter 10) may be used. (c) Schematic view of a plot of a recombinant sequence.

In this simple case, the calculated statistic would be something like the differences in genetic distances between the potential recombinant and its two potentially parental sequences in adjacent windows. Using raw pairwise genetic distances as a relatedness measure for identifying potential recombination signals is one component of many recombination detection methods including, for example, those implemented in programs such as SIMPLOT (Lole *et al.*, 1999), RECSCAN (Martin *et al.*, 2005), RAT (Etherington *et al.*, 2005), and RIP (Siepel *et al.*, 1995).

Although the relative pairwise genetic distances between sequences usually correlates well with their evolutionary relatedness, this is not always the case. This is important because more accurate estimates of evolutionary relatedness should enable more accurate identification of recombination signals. Therefore, using phylogenetic methods rather than pairwise distances to infer the relative relatedness of sequences in different alignment partitions has been extensively explored in the context of recombination signal detection. Recombination detection methods that employ phylogeny-based comparisons of alignment partitions include those implemented in programs such as TOPAL (McGuire & Wright, 1998; McGuire *et al.*, 1997), DUALBROTHERS (Minin *et al.*, 2005; Suchard *et al.*, 2002), PLATO (Grassly & Holmes, 1997), RECPARS (Hein, 1993), GARD (Kosakovsky Pond *et al.*, 2006), JAMBE (Husmeier & Wright, 2001; Husmeier *et al.*, 2005), and BARCE (Husmeier & McGuire, 2003). The most popular of these phylogenetic methods is the *bootscan* method (Fig. 16.5). Implementations of bootscan can be found in the programs RDP3 and SIMPLOT, which will be used later to demonstrate how recombination can be detected with the method . The bootscan method involves the construction of bootstrapped ***neighbor joining*** trees in sliding window partitions along an alignment. The relative relatedness of the sequences in each tree is then expressed in terms of ***bootstrap*** support for the phylogenetic clusters in which they occur. Recombination is detected when a sequence, defined as a query, "jumps" between different clusters in trees constructed from adjacent alignment partitions. This jump is detectable as a sudden change in bootstrap support grouping the potential recombinant with different sequences resembling its potential parental sequences in different genome regions.

A third, very diverse, set of recombination signal detection approaches employ a range of different relatedness measures and statistics and often draw on some phylogenetic information to compare relationship measures determined for different alignment partitions. Usually, these are based on identifying shifts in the patterns of sites shared by subsets of sequences within an alignment. These so-called *substitution distribution methods* use a statistic (e.g. a Z-score, a chi square value, Pearson's regression coefficient, or some other improvised statistic) that can be used to express differences in the relative relationship between sequences in different, usually adjacent, alignment partitions. The relatedness measures used

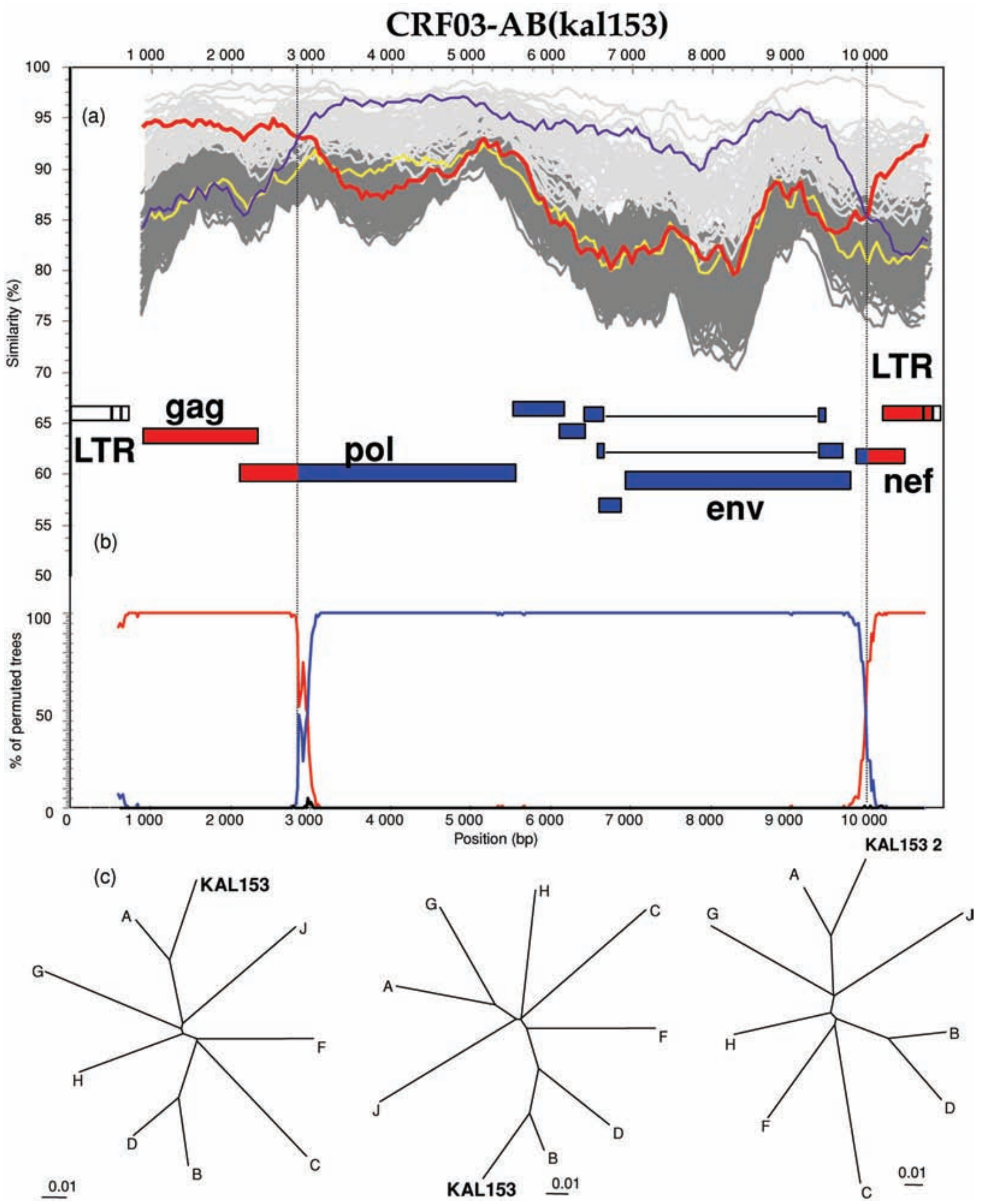


Fig. 16.5 Analysis using SIMPLOT of recombinant isolate KAL153 using (a) the distance-based similarity method and (b) the phylogenetic based bootscanning method. (c) Phylogenetic confirmation (K2P + NJ) of the identified recombination signal. The similarity analysis with subtypes A, B (parental subtypes), and C (outgroup) is superimposed on the ranges of intra- and inter-subtype variation.

are generally counts of nucleotide sites in common and/or different between pairs, triplets or quartets of sequences in the alignment being analyzed. The exact sites that are considered differ from method to method with some counting all sites of the alignment (e.g. the PHYLPRO method; Weiller, 1998) and others examining only those sites that differ in some specified way amongst the sequences being compared (e.g. the SISCAN method; Gibbs *et al.*, 2000). The major advantage of the substitution distribution methods over pure phylogenetic and distance based approaches is that they often allow detection of recombination events that cannot, for example, be visualized as sequences "jumping" between clades of phylogenetic trees constructed using different alignment partitions. By accessing information on overall patterns of nucleotide substitution within an alignment, many substitution distribution methods (such as those implemented in programs like 3SEQ, Boni *et al.*, 2007; GENECONV, Sawyer, 1989; MAXCHI, Maynard Smith, 1992; and CHIMAERA, Posada & Crandall, 2001) can detect when sequences are either more closely related or more distantly related in certain alignment partitions than would be expected in the absence of recombination. These methods are able to detect such recombination signals regardless of whether they have sufficient phylogenetic support.

One of the most powerful substitution distribution methods is the MAXCHI method and it will be demonstrated in the practical exercises later. Like the bootscan method mentioned above, the MAXCHI method involves moving a window across the alignment. However, before scanning the alignment, the MAXCHI method first discards all the alignment columns where all sequences are identical and then moves a window with a partition in its centre along this condensed alignment. For every pair of sequences in the alignment, nucleotide matches and mismatches are scored separately for the two halves of each window and then compared using a 2 × 2 chi square test. Whenever a window partition passes over a recombination breakpoint, the method records peaks in the chi square values calculated for sequence pairs containing the recombinant and one of its "parents" (they are actually usually just sequences resembling the recombinant's parents).

While substitution distribution methods such as MAXCHI are both extremely fast and amongst the most powerful ever devised, many of these methods generally also assume that sequence similarity is perfectly correlated with evolutionary relatedness – an assumption that is often violated and could potentially compromise their accurate inference of recombination.

Given that there are so many different methods with which recombination signals could be detected, it is important to realize that none of these methods has yet emerged as being best under all possible analysis conditions. Some extremely sophisticated and quite powerful methods, such as those implemented in GARD, DUALBROTHERS and TOPALi are also extremely slow and can currently only be applied to relatively small or simplified analysis problems. Some simpler methods, while less powerful, are capable of rapidly scanning enormous, extremely complex data sets. Also, certain methods are incapable of detecting certain types of recombination events, whereas others are prone to false positives if certain of their assumptions are violated. For this reason certain recombination analysis tools such as RDP (Martin & Rybicki, 2000; Martin *et al.*, 2005) and TOPALi (McGuire & Wright, 2000) provide access to multiple recombination signal detection methods that can be used in conjunction with one another. Not only can these methods be used to crosscheck potential recombination signals but they can, in the case of RDP3, also be collectively combined to analyze sequences for evidence of recombination. While this may seem like a good idea, it is still unclear whether detection of any recombination signal by more than one method should afford additional credibility.

16.4 Identifying and characterizing actual recombination events

Many of the methods aimed at identifying recombination signals are also suited to characterize certain aspects of the recombination events – such as localizing recombination breakpoint positions, identifying recombinant sequences and identifying sequences resembling parental sequences. How this information is accessed differs from one method to the next and also relies largely on the amount of prior knowledge one has on the potential recombinant status of the sequences being analyzed.

Currently available analysis tools use two main approaches to identify and characterize recombination events. Choosing an approach and the tools that implement it depends firstly on the types of recombination events one wants to analyze and, secondly, on knowledge of which of the sequences available for analysis are probably not recombinant. We will again use analysis of HIV-1 recombination as an example. Analysis of inter-subtype HIV-1M recombination is vastly simplified by the existence of a large publicly available collection of so-called "pure-subtype" full genome sequences. While not definitely non-recombinant (many of these sequences may be either intra-subtype recombinants or ancient inter-subtype recombinants in which the recombination signals have been degraded by subsequent mutations), this collection of sequences can be used to reliably scan potentially recombinant sequences for evidence of recent inter-subtype recombination events. Many recombination analysis tools such as SIMPLOT, REGA, RIP, DUALBROTHERS, JPHMM, and RDP3 will allow one to analyze a potential recombinant, or *query sequence*, against a set of known non-recombinants, or *reference sequences*, and identify: (1) whether the query sequence is recombinant; (2) the locations of potential breakpoint positions; (3) the probable origins of different tracts of sequence within a recombinant (Fig. 16.5).

In cases where no reliable set of non-recombinant reference sequences is available, such as when attempting to analyze intra-subtype HIV-1 recombination, one is faced with two choices. Either a set of reference sequences must be constructed from scratch before applying one of the query vs. reference scanning methods, or use must be made of one of the many exploratory recombination signal detection methods that do not rely on a reference sequence set. Construction of a reference sequence data set will not be discussed here, but see Rosseau *et al.* (2007) for an example involving analysis of intra-subtype HIV recombination.

The exploratory recombination signal detection methods, such as those implemented in **Rdp3**, **Geneconv**, **3Seq**, **Siscan**, and **Phylpro**, all accept sequence alignments as input and, without any prior information on which sequences might be recombinant, will attempt to identify signals of recombination. Although these methods are completely objective and their use might seem more appealing than that of methods relying on a largely subjective query vs. reference scanning approach, one should be aware that there are two serious drawbacks to the exploratory analysis of recombination signals. First, when enumerating all the recombination signals evident in an alignment, the exploratory methods will often compare thousands or even millions of combinations of sequences. This can create massive multiple testing problems that must be taken into account when assessing the statistical support of every recombination signal detected. In extreme cases, such as when alignments containing hundreds of sequences are being analyzed, statistical power can become so eroded by multiple testing corrections, that even relatively obvious recombination signals are discounted.

The second, and probably most important, problem with exploratory recombination detection is that, even if one is provided with a very clear recombination signal and a set of sequences used to detect the signal (something that many of the exploratory methods will provide), it is often extremely difficult to determine which sequence is the recombinant. As a result, "automated" exploratory scanning of recombination often still requires a great deal of manual puzzling over which of the identified sequences is "jumping" most between clades of phylogenetic trees constructed using different alignment partitions. This can be a particularly serious problem because the exploratory methods do not, for better or worse, exclude the possibility of detecting recombination events between parental sequences that are themselves recombinant. The apparent objectivity of exploratory recombination detection is therefore ultimately compromised by a largely subjective process of recombinant identification.

PRACTICE

Mika Salminen and Darren Martin

16.5 Existing tools for recombination analysis

Software for analyzing recombination is available for all major operating systems (see *http://www.bioinf.manchester.ac.uk/recombination/programs.shtml* for a reasonably comprehensive list). The two programs that will be used here for demonstrating the detection and characterization of recombination events are the Windows programs SIMPLOT (downloadable from *http://sray.med.som.jhmi.edu/RaySoft/SimPlot/*) and RDP3 (downloadable from *http://darwin.uvigo.es/rdp/rdp.html*). Both programs will run on Apple Macs under Virtual PC emulation software. Both programs are capable of reading alignments in a wide variety of formats and performing similarity/dissimilarity plots of any of the sequences against all others in the alignment. They allow multiple analysis parameters to be varied, including window sizes, window overlaps and evolutionary-distance corrections. Both produce graphical outputs that can be exported in both bitmap (.bmp) and windows metafile formats (.wmf or .emf). The programs enable reasonably precise mapping of inferred recombination breakpoints and allow trees to be constructed for different alignment partitions; RDP3 implements various tree reconstruction methods to this purpose. SIMPLOT also allows sequence partitions to be exported for phylogenetic analysis by other programs. Both RDP3 and SIMPLOT also implement the popular bootscanning method (Salminen *et al.*, 1995), the use of which will be described in some detail here using SIMPLOT.

To install SIMPLOT, go to *http://sray.med.som.jhmi.edu/SCRoftware/simplot/* and download the zip-compressed installation file; for this exercise, we used `SIMPLOT` version 3.5.1. Place the file in a temporary file folder (e.g. the C:\temp folder found on most systems) and uncompress the file. Install SIMPLOT using the installer file `SETUP.EXE`. By default, SIMPLOT is installed in `Program Files/RaySoft/` folder and a link is added to the `Start` menu. More detailed instructions can be found on the Simplot website. Installation of RDP3 follows a similar process. Download the RDP3 installation files from *http://darwin.uvigo.es/rdp/rdp.html* to the temporary folder and uncompress it. Run the `SETUP.EXE` that is unpacked and RDP3 will add itself to the `Start` menu.

To successfully perform the SIMPLOT exercises, another program, TREEVIEW (see Section 5.5 in Chapter 5), is also needed.

16.6 Analyzing example sequences to detect and characterize individual recombination events

Several sets of aligned sequences will be used for the exercises. These are available at the website *http://www.thephylogenetichandbook.org* and have the following features:

(1) File A-J-cons-kal153.fsa: A single recombinant HIV-1 genome (KAL153) aligned with 50% consensus sequences derived for each HIV-1 subtype (proviral LTR-genome-LTR form).
(2) File A-J-cons-recombinants.fsa: two recombinant HIV-1 sequences in FASTA format aligned with 50% consensus sequences derived for each HIV-1 subtype (proviral LTR-genome-LTR form).
(3) File 2000-HIV-subtype.nex: A reference set of virtually complete HIV genome sequences aligned in FASTA format (virion RNA R-U5-genome-U3-R form).

In the first three exercises SIMPLOT will be used to demonstrate the query vs. reference approach to detecting and characterizing recombination events. RDP3 will then be used in the last three exercises to demonstrate some exploratory approaches to recombination detection and analysis that can be used if one has no prior knowledge of which sequences in a data set are non-recombinant.

16.6.1 Exercise 1: Working with SIMPLOT

This exercise shows the use of SIMPLOT and its basic properties. To start SIMPLOT, select the program from its group on the `Start` Menu in Windows or double-click on the SIMPLOT icon in the SIMPLOT folder. The program will start with a window from which it is possible to select the `file` menu to open an alignment file. Open the file A-J-cons-kal153.fsa (the .fsa extension may or may not be visible, depending on the settings of the local computer). This file contains an alignment of the CRF03-AB strain Kal153 and 50% consensus reference sequences for subtypes A through J. The tree-like graph shown in Fig. 16.6 appears, which has all the sequences contained in a group with the same name as the sequence (by default). Groups can be automatically defined using any number of characters in the sequence names or using characters prior to a separator in the sequence names by clicking on "`Use first character to identify groups`". To reveal the sequences in each group, select "`Expand Groups`". Groups have a color code and are primarily used to quickly generate consensus sequences; they are discussed in another exercise.

Groups or individual sequences can be excluded from the analysis by deselecting them. Groups can be moved or individual sequences can be moved between groups using the options in the right panel. Select the KAL153 group (not the sequence) and try to move it to the top of the tree by pressing the "Move up" button repeatedly. Under the `File` and `Help` menus are four tabs; click on `SimPlot` to go to the page

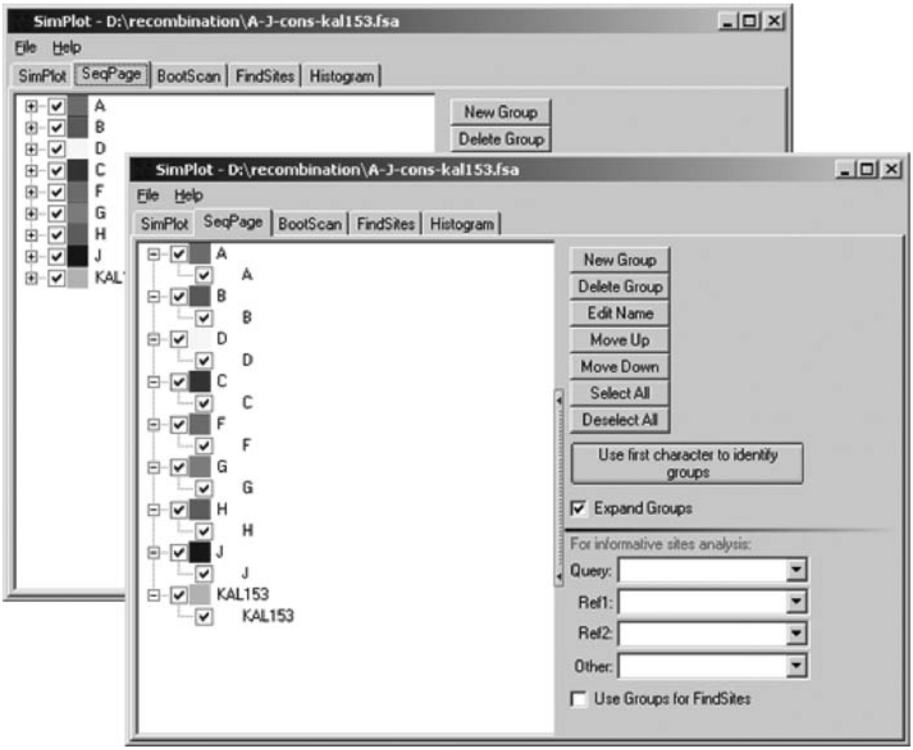
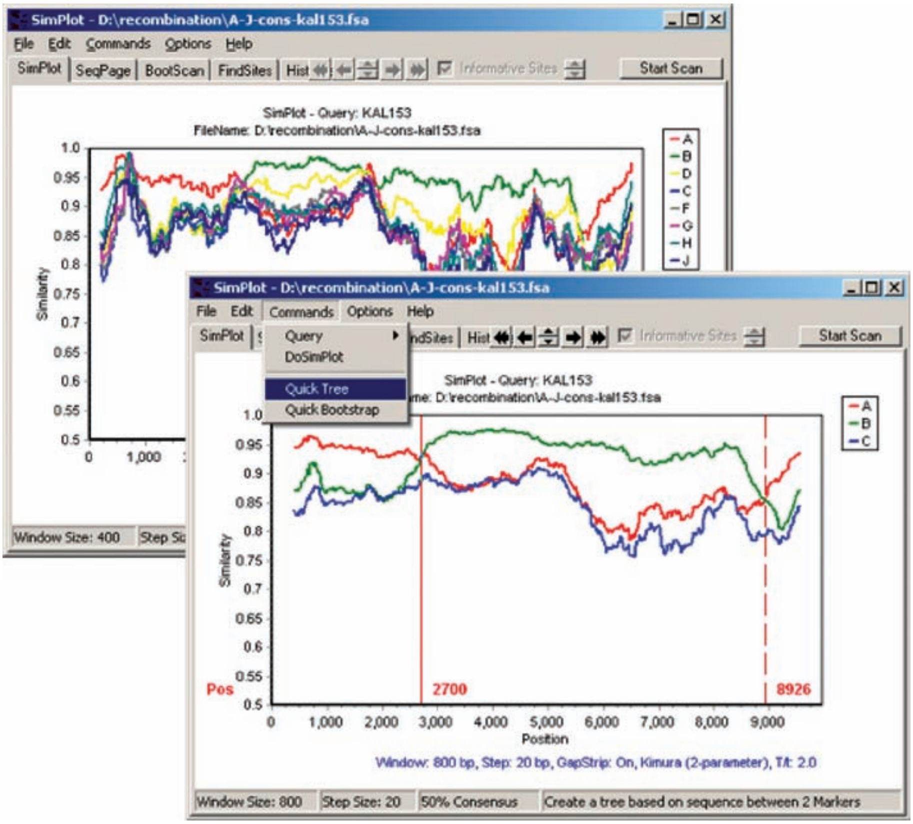

SimPlot - D:\recombination\A-J-cons-kal153.fsa
File Help
SimPlot SeqPage BootScan FindSites Histogram
A
B
D
C
F
G
H
J
KAL
New Group
Delete Group
SimPlot - D:\recombination\A-J-cons-kal153.fsa
File Help
SimPlot SeqPage BootScan FindSites Histogram
A
A
B
B
D
D
C
C
F
F
G
G
H
H
J
J
KAL153
KAL153
New Group
Delete Group
Edit Name
Move Up
Move Down
Select All
Deselect All
Use first character to identify groups
☑ Expand Groups
For informative sites analysis:
Query:
Ref1:
Ref2:
Other:
☐ Use Groups for FindSites

Fig. 16.6 Starting SIMPLOT. The back window represents the view after opening a sequence-alignment file. The front window represents the view after expanding the groups. Each sequence belongs to a group, which by default is named the same as the sequence itself.

where the similarity plots are performed. The other tabs, `SeqPage`, `Bootscan`, `FindSites` and `Histogram`, are the input page, the bootscanning-analysis page, the phylogenetically informative site-analysis page and the histogram page; the bootscanning analysis is discussed later, and the SIMPLOT documentation describes the `FindSites` and `Histogram` analysis.

To do the first analysis, go to the `Commands` menu and select `Query` and KAL153 in the pop-up list, which reflects the sequences in the order determined in the `SeqPage` window. This first operation determines which of the aligned sequences is compared to all the others (Fig. 16.7). To set the window size for the similarity scanning to 400 bp, click on "`Window Size`" in the lower left corner and choose 400 in the pop-up list. Keep the default settings for "`Step Size`". To perform the analysis, go to the `Commands` menu again and click on `DoSimplot`; the result should be similar to Fig. 16.7. The analysis indicates that the KAL153 strain is a recombinant of subtypes A and B, as reflected by the legend key. The parameters for this analysis can be changed in the `Options` (`Preferences`)



Window: 800 bp, Step: 20 bp, GapStrip: On, Kimura (2-parameter), T/t: 2.0

Window Size: 400 | Step Size

Window Size: 800 | Step Size: 20 | 50% Consensus | Create a tree based on sequence between 2 Markers

Fig. 16.7 Comparing KAL153 to the reference sequences with SIMPLOT. The back window represents the similarity plot for KAL153 as query compared to all reference sequences. The front window represents the similarity plot for KAL153 compared to subtypes A, B, and C. In this plot, the approximate positions of the breakpoints are indicated using red lines that appear by double clicking on the appropriate positions in the plot. When two breakpoint positions are indicated, a quick tree can be inferred for the alignment slice in between the two red lines.

menu. These include the options for the distance calculation (see Chapter 4 for models of evolution), the bootstrap options and miscellaneous options. Change the window size to 800 bases and compare the plot to the composite similarity plot shown in Fig. 16.5. Another useful option is whether to exclude gap regions from the analysis (usually recommended). Consult the SIMPLOT documentation, accessible from the `Help` (`Contents`) menu, for more information about the different options.

Return to the `SeqPage` tab and deselect groups so that only A, B, C, and KAL153 will be included. Go back to the `SimPlot` tab. When doing the analysis, it should be easy to pinpoint the recombination breakpoints. In fact, the user can click on the breakpoint positions in the plot (but not on the distance lines themselves) and a red line will appear with the position in the alignment where the breakpoint is approximately located. A second position, e.g. representing the second breakpoint position can also be selected and a quick tree can be reconstructed for the alignment segment in between the red lines using the “`Quick Tree`” command (Fig. 16.7). The red lines can be removed by double clicking on them. Clicking on the similarity lines will result in a pop-up window with the name of the relevant reference sequence, the position, the similarity score and the possibility to change the color of the line.

The next analysis we will perform is bootscanning. Make sure that in the `SeqPage` tab only the four sequences are still included for this analysis. Click on the `BootScan` tab, select again KAL153 as query sequence and start the bootscan procedure by selecting `DoBootscan` from the `Commands` menu. The default options for this analysis are the Kimura-two-parameter model (see Chapter 4), a ***transition/transversion ratio*** of 2 and stripping positions that have over 50% of gaps in a column. These settings can be changed in the `Options` (`Preferences`) menu. As the bootscanning begins, a running dialog appears in the bottom right corner that shows for which window a tree and bootstrap analysis is being performed; simultaneously, the plot grows on the screen. The analysis is fairly slow but can be stopped using the `Stop Bootscan` button in the upper right corner (it is not possible to continue the analysis after prematurely stopping it).

16.6.2 Exercise 2: Mapping recombination with Simplot

In this exercise, we will map the structure of two other HIV-1 recombinant strains. The sequences under study are in the file A-J-cons-recombinants. fsa. Open the file A-J-cons-recombinants.fsa and use the skills learned in the previous exercise to map the recombination breakpoints in the sequences. First, analyze the recombinants individually by deselecting the other group including a recombinant. Identify the subtypes of the parental sequences using all reference sequences and a relatively large window size (i.e. 600–800). Finally, map the recombination breakpoints with only the parental sequences and a smaller window size (i.e. 300–400). To verify the results, quick trees can be reconstructed for the putative recombinant regions identified by Simplot or those regions can be exported and analyzed running separate phylogenetic analyses with Phylip or Paup*. To export a region of an alignment from Simplot, first select the region of interest (i.e. map the breakpoints) using two red lines. Go to the File menu and select “`Save Marked Slice of Alignment as ...`”. The format of the new alignment file can be chosen; for example, the `PHYLIP` interleaved format. Table 16.2 shows the correct result of the analysis.

Table 16.2 Key to recombinant strain structures of Exercise 2

Strain	Parental subtypes	Breakpoints[a]
UG266	AD	5430, 6130, 6750, 9630
VI1310	CRF06-DF	2920, 3640, 4300, 5260, 6100

[a] There may be some variation in the exact coordinates depending on the window settings used.

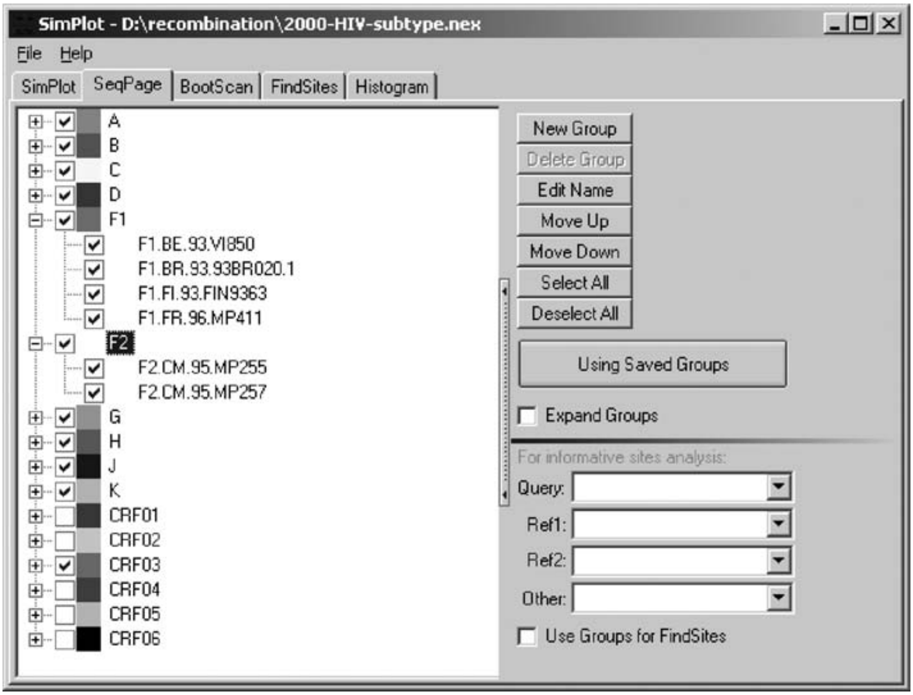
16.6.3 Exercise 3: Using the "groups" feature of SIMPLOT

To focus on recombinant events between major clades, e.g. HIV-1 group M subtypes, the "group" feature provides a way to specify groups of reference sequences and perform quick analyses on their consensus sequences. This is done by adding an extra NEXUS block at the end of a NEXUS-formatted alignment file (see Box 8.4 in Chapter 8 for an explanation on the NEXUS format). This block specifies groups identified by users or written by the SIMPLOT software. The following is an example of a group-specifying block at the end of an alignment:

begin RaySoft; [!This block includes information about groups identified by users of software written by Stuart Ray, M.D.] groups group1 = '"@A":clRed(@0, @1, @2, @3), "@B":clGreen(@4, @5, @6, @7), "@C":clYellow(@8, @9, @10, @11, @12), "@D":clBlue(@13, @14, @15, @16), "@F":clGray(@17, @18, @19, @20, @21, @22), "@G":clFuchsia(@23, @24, @25, @26), "@H":clTeal(@27, @28, @29), "@J":clNavy(@30, @31), "@K":clSilver(@32, @33), "@CRF01":clWhite(@37, @36, @35, @34), "@CRF02":clWhite(@41, @40, @39, @38), "@CRF03":clWhite(@43, @42), "@CRF04":clWhite(@46, @45, @44), "@CRF05":clWhite(@48, @47), "@CRF06":clMaroon(@49, @50)'; end;

The NEXUS-formatted alignment contains 51 sequences. The sequences in the group definitions are represented by "@" and a number according to their occurrence in the alignment (starting with 0). Sequences are grouped by bracket notation and the group is preceded by a "@name" and a color code.

To start the exercise, open the file 2000-HIV-subtype.nex in SIMPLOT and compare the SeqPage view to the example NEXUS block; this file contains the same groups as defined in the block. Expand the groups to see how the program groups all the sequences. When the user scrolls to the bottom of the window, six groups are visible: CRF01, CRF02, CRF03, CRF04, CRF05, and CRF06. They contain reference sequences for six currently recognized circulating recombinant forms of HIV-1 (CRF01_AE, CRF02_AG, CRF03_AB, CRF04_cpx, CRF05_DF, and CRF06_cpx). The F subtype actually consists of two groups, the F1 and F2 subsubtypes; split them into two groups. First, rename the F group to F1. Next, create a new group using the buttons on the right, name it F2, and move it right under the F1 group. Expand the F1 group and move the F2 sequences one by one to



SimPlot - D:\recombination\2000-HIV-subtype.nex

☑ A
☑ B
☑ C
☑ D
☑ F1
- ☑ F1.BE.93.VI850
- ☑ F1.BR.93.93BR020.1
- ☑ F1.FI.93.FIN9363
- ☑ F1.FR.96.MP411

☑ F2
- ☑ F2.CM.95.MP255
- ☑ F2.CM.95.MP257

☑ G
☑ H
☑ J
☑ K
□ CRF01
□ CRF02
☑ CRF03
□ CRF04
□ CRF05
□ CRF06

□ Expand Groups

For informative sites analysis:

Query:
Ref1:
Ref2:
Other:

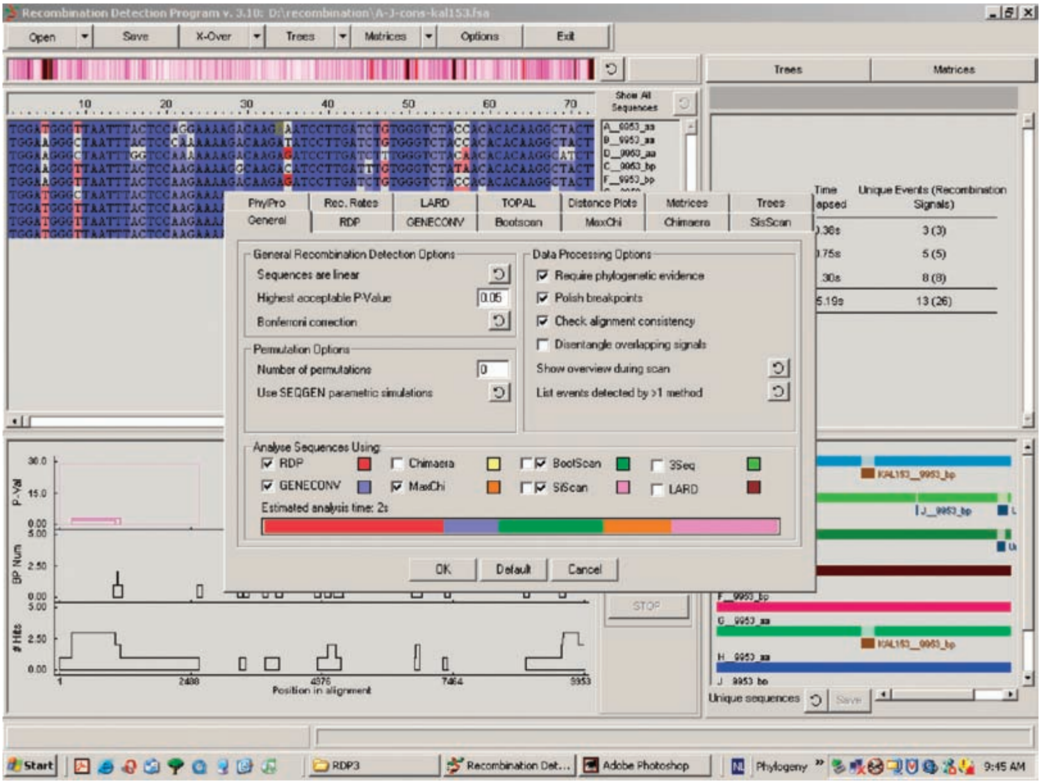
□ Use Groups for FindSites

Fig. 16.8 Using and rearranging groups specified in the alignment file. In the 2000-HIV-subtype.nex file groups have been specified by the inclusion of a NEXUS block. The F group has been renamed to F1 and a new group, F2, has been created. Both groups are expanded. F2 sequences were moved to the F2 group.

the new F2 group using the buttons to the right or by dragging and dropping (Fig. 16.8). Deselect all other CRF groups except the CRF03 group. Switch to the `SimPlot` tab and perform a default parameter analysis with window settings of 400 nucleotides/20 steps and CRF03 as the query. The program calculates by default a 50% consensus for the groups and plots the similarity values to that consensus. By clicking on the consensus panel in the lower part of the window, it is possible to change the type of consensus used in the analysis. Explore the effect of different types of consensus models (see the SIMPLOT documentation for more details).

16.6.4 Exercise 4: Setting up RDP3 to do an exploratory analysis

This exercise demonstrates how to modify RDP3 analysis settings and do a basic exploratory search for recombination signals in a simple data set. Start RDP3 and open the file A-J-cons-kal153.fsa. Certain default RDP3 settings are not ideal for the analysis of HIV sequences and should be changed. Press the "`Options`" button at the top of the screen. All options for all methods implemented in RDP3



PhylPro | Rec. Rates | LARD | TOPAL | Distance Plots | Matrices | Trees
General | RDP | GENECONV | Bootscan | MaxChi | Chimaera | SisScan

General Recombination Detection Options
- Sequences are linear
- Highest acceptable P-Value: 0.05
- Bonferroni correction

Permutation Options
- Number of permutations: 0
- Use SEQGEN parametric simulations

Data Processing Options
- ☑ Require phylogenetic evidence
- ☑ Polish breakpoints
- ☑ Check alignment consistency
- ☐ Disentangle overlapping signals
- Show overview during scan
- List events detected by >1 method

Analyse Sequences Using
- ☑ RDP 🟥
- ☑ GENECONV 🟦
- ☐ Chimaera 🟨
- ☑ MaxChi 🟧
- ☐☑ BootScan 🟩
- ☐☑ SiScan 🟪
- ☐ 3Seq 🟩
- ☐ LARD 🟫

Estimated analysis time: 2s

OK | Default | Cancel

Time elapsed	Unique Events (Recombination Signals)
.38s	3 (3)
.75s	5 (5)
30s	8 (8)
5.19s	13 (26)

Fig. 16.9 Working with RDP3. The top left panel represents the alignment. The boxes in the bottom left panel indicate the positions of the breakpoints and the sizes of recombinant tracts that have been detected. The top right panel provides information on the duration of the analysis and the number of recombination events identified. The bottom right panel shows the recombinant tracts identified in the different sequences. The middle window is the options window in which the sequences have been set to linear.

(including tree drawing, matrix drawing and recombination detection methods) can be changed using the form that is displayed (Fig. 16.9). The main page of the options form contains details of the general exploratory recombination detection settings. The only thing that needs to be changed here is that sequences should be handled as though they are linear. Although it would not invalidate the analysis if sequences were handled as though they were circular, this setting will make analysis results a little harder to interpret. Press the button with the circular arrow besides the "`Sequences are circular`" caption.

At the bottom of the general settings form are a series of colored rectangles with names next to them and a colored strip beneath them (Fig. 16.9). Note that some of the names next to the rectangles ("RDP", "GENECONV", "MaxChi" etc.) have ticks next to them. Each colored rectangle represents a different recombination signal detection method. These are all of the methods implemented in RDP3 that can be used to automatically explore and enumerate recombination signals in an alignment. If you wish to explore an alignment for recombination with any of the methods you should click on the "check box" next to the method's name. A tick in this box means it will be used, along with all the other methods with ticks next to them, to explore for recombination signals in the alignment. Note that the BOOTSCAN and SISCAN methods each have two boxes, only one of which is ticked at the moment. These methods are far slower than all the other methods (except LARD) and the option is given to use these as either primary or secondary exploration methods. Primary exploration methods will thoroughly examine an alignment searching for and counting all detectable recombination signals. Whenever signals are found by the primary exploration methods, secondary exploration methods will be used to thoroughly re-examine sequences similar to those in which the initial recombination signal was found. All of the listed methods except BOOTSCAN, SISCAN and LARD will automatically be used as secondary exploration methods. The LARD method is so slow that RDP3 only permits its use as a secondary exploration method. The colored strip at the bottom of the form gives an estimate of the relative execution times of the different methods. An estimate of total analysis time is given above the bar.

For the moment all analysis options on this form will be left to their default values. However, some analysis settings need to be changed for some of the particular exploratory recombination detection methods. Click on the "RDP" tab and change the window size setting from 30 to 60. Click on the "MAXCHI" tab and change this window size setting to 120. Click on the "CHIMAERA" tab and also change this window size setting to 120. Click on the "BOOTSCAN" tab and change the window size setting to 500. Click on the "SISCAN" tab and also change this window size setting to 500. Window sizes were increased from their default settings because HIV sequences are relatively unusual in that they experience mutation rates that are so exceptionally high that recombination signals are rapidly obscured. Increasing window sizes increases the ratio of signal relative to mutational "noise." It is important to note that increasing window sizes also makes detection of smaller recombination events more difficult.

Now that the analysis settings have been made, press the "OK" button at the bottom of the form. If you shut the program down now these settings will be saved. The saved settings will be used whenever you start the program and you will not have to reset them at the start of every analysis.

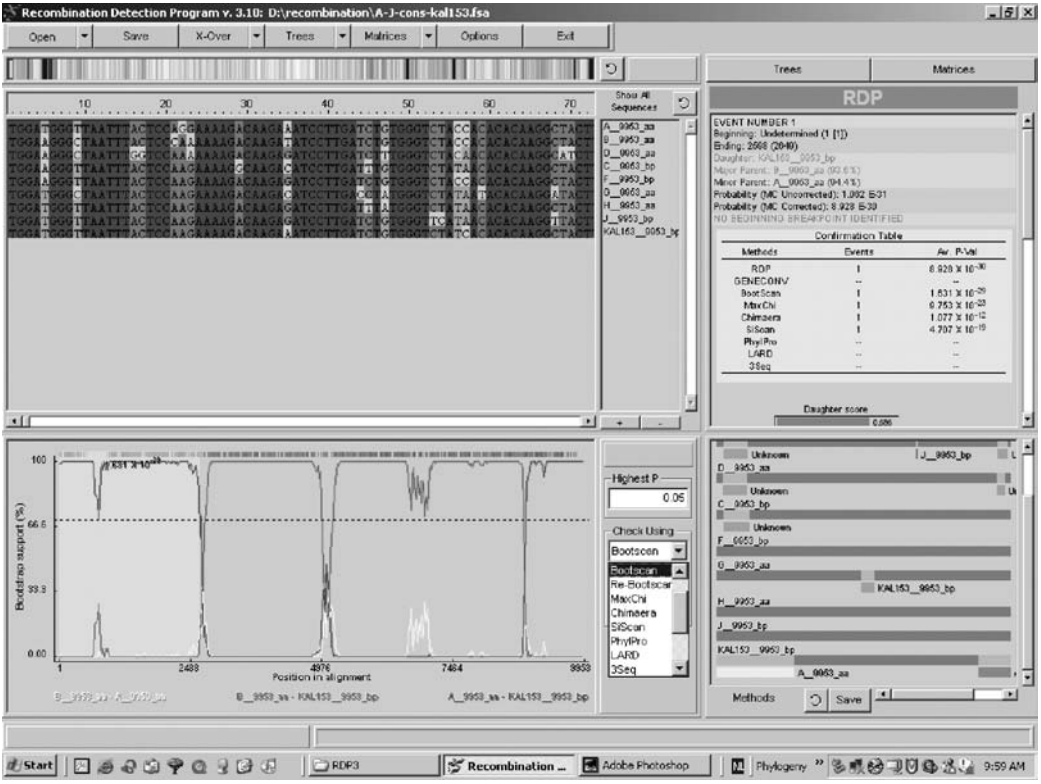
16.6.5 Exercise 5: Doing a simple exploratory analysis with RDP3

To start an exploratory analysis with the RDP, GENECONV, and MAXCHI methods in primary exploratory mode and the CHIMAERA, SISCAN, BOOTSCAN, and

3Seq methods in secondary exploratory mode press the "`X-Over`" button at the top of the screen. A series of colored lines will flash past on the bottom of the screen and will be almost immediately replaced by a set of black and pinkish boxes (Fig. 16.9). These are indicating the positions of breakpoints and sizes of recombinant tracts that have been detected and provide some indication of the *p*-values associated with detected events. These graphics are intended to give you something to look at when the analysis is taking a long time, so don't worry about exactly what they mean right now. In the top right-hand panel you will notice that information is given on how long each analysis method took to explore the data (Fig. 16.9). You will also notice that under the "`Unique events (recombination signals)`" heading there are some numbers. The first number indicates the number of unique recombination events detected by each method and the second number (in parentheses) indicates the number of unique recombination signals detected. The second number will always be equal to or larger than the first number. If the second number is larger it will indicate that two or more sequences in the alignment carry traces of the same ancestral recombination event(s).

In the bottom right panel you will notice a series of colored rectangles (Fig. 16.9). Move the mouse pointer around this panel a bit. You will notice that when it moves over some of the rectangles, information flashes onto the top right-hand panel. The information displayed here relates to the rectangle that the mouse pointer is positioned over. The rectangles that are "sensitive" to the mouse pointer are graphical representations of individual recombination signals detected in the alignment. Use the scroll-bar on the right-hand side of the bottom right panel to move down so that the rectangles representing the sequence, KAL153, are visible (move the scroll bar right down to the bottom). This is the recombinant sequence analyzed using the query vs. reference method in exercise 1 above.

Click on the button with the circular arrow at the bottom of the bottom right panel displaying the graphical representation of recombination signals. The caption beside the button should now read "`Methods`" (Fig. 16.10). You will notice that the color of all the rectangles has changed. They are now either grey, red, orange or blue. The long rectangle at the bottom, that is intermittently dark and light gray, represents sequence KAL153. The dark parts represent the "background" sequence and the light bits represent tracts of sequence that possibly have a recombinant origin (Fig. 16.10). As mentioned before, the colored rectangles directly beneath the light gray bits represent the recombination signal. These rectangles also represent recombination hypotheses delimiting the bounds of tracts of horizontally inherited sequence. The labels to the right of the colored rectangles indicate sequences in the alignment resembling the donor parents. As the "`Methods`" coloring scheme is selected, the red, blue and orange rectangles denote recombination events detected by the Rdp, Geneconv, and Maxchi methods, respectively. Pressing the left mouse



EVENT NUMBER 1
Beginning: Undetermined (1 [1])
Ending: 2668 (2040)
Daughter: KAL153__9953_bp
Major Parent: B__9953_aa (93.6%)
Minor Parent: A__9953_aa (94.4%)
Probability (MC Uncorrected): 1.082 E-31
Probability (MC Corrected): 8.928 E-30
NO BEGINNING BREAKPOINT IDENTIFIED

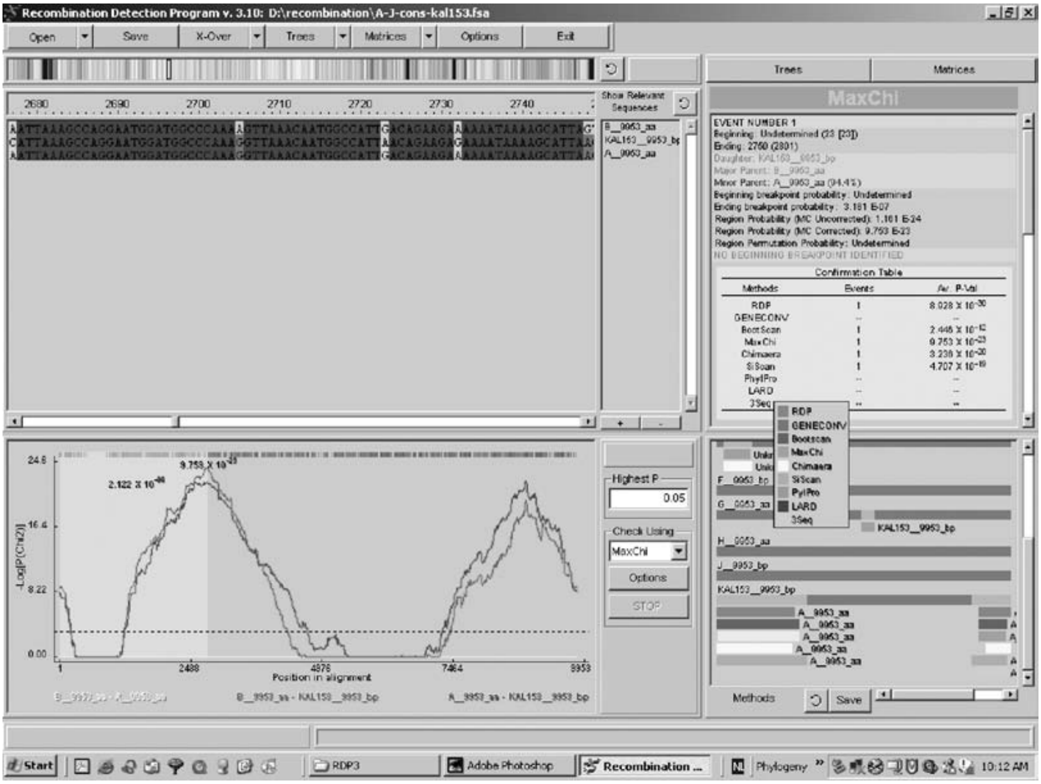
Confirmation Table

Methods	Events	Av. P-Val
RDP	1	8.928×10^{-30}
GENECONV	--	--
BootScan	1	1.631×10^{-29}
MaxChi	1	6.753×10^{-23}
Chimaera	1	1.077×10^{-12}
SiScan	1	4.707×10^{-19}
PhylPro	--	--
LARD	--	--
3Seq	--	--

Fig. 16.10 Detecting recombination events using RDP3. The top right panel provides information that relates to the recombination signal, the rectangle in a sequence in the lower right panel, over which the mouse has been positioned. In the bottom right panel, the graphical information representing recombination signals has been changed from "`Unique sequences`" to "`Methods`." The bottom left panel shows the bootscan plot for the recombination event identified for KAL153. (After clicking on the first recombinant tract identified by RDP in KAL153 at the bottom of the lower right panel, the "`Check Using`" option was changed to "`Bootscan`".)

button in the window when the mouse pointer is not over a colored rectangle gives you a legend explaining the color coding.

Move the mouse pointer over the red rectangle to the left of the panel and press the left mouse button. What should now be displayed in the bottom left panel is a graphical representation of the actual recombination signal used to detect the recombination event depicted by the red rectangle. This plot can be interpreted in much the same way as the BOOTSCAN plots described in earlier exercises. To see a BOOTSCAN version of this plot look for the label "`Check using`" to the right of the plot and press the little arrow besides the label, "`RDP`". On the menu that appears select either "`Bootscan`" or "`Re-Bootscan`" (Fig. 16.10). Now



Recombination Detection Program v. 3.10: D:\recombination\A-J-cons-kal153.fsa

```
AATTAAAGCCAGGAATGGATGGCCCAAAAGTTAAACAATGGCCATTGACAGAAGAAAAAATAAAAGCATTAG   B__9953_aa
CATTAAAGCCAGGAATGGATGGCCCAAAGGTTAAACAATGGCCATTAACAGAAGAGAAAATAAAAGCATTAA   KAL153__9953_bp
AATTAAAGCCAGGAATGGATGGCCCAAAGGTTAAACAATGGCCATTGACAGAAGAAAAAATAAAAGCATTAA   A__9953_aa
```

```
EVENT NUMBER 1
Beginning: Undetermined (23 [23])
Ending: 2760 (2801)
Daughter: KAL153__9953_bp
Major Parent: B__9953_aa
Minor Parent: A__9953_aa (94.4%)
Beginning breakpoint probability: Undetermined
Ending breakpoint probability: 3.181 E-07
Region Probability (MC Uncorrected): 1.161 E-24
Region Probability (MC Corrected): 9.753 E-23
Region Permutation Probability: Undetermined
NO BEGINNING BREAKPOINT IDENTIFIED
```

Confirmation Table

Methods	Events	Av. P-Val
RDP	1	8.028×10^{-30}
GENECONV	--	--
BootScan	1	2.446×10^{-12}
MaxChi	1	9.753×10^{-23}
Chimaera	1	3.238×10^{-20}
SiScan	1	4.707×10^{-19}
PhylPro	--	--
LARD	--	--
3Seq	--	--

Highest P: 0.05

Check Using: MaxChi

Fig. 16.11 Investigating breakpoint positions using RDP3. The top right panel provides information that relates to the recombination signal identified by MAXCHI for KAL153. The bottom right panel shows "`All Events for All Sequences`", the option selected after right clicking in the panel. The color key for the different methods is shown (by pressing the left mouse button on an open part of the panel). The bottom left panel shows the MAXCHI plot for KAL153. By clicking on the right border of the first pink region, the alignment in the top left panel jumps to around position 2800. By clicking "`Show all sequences`" next to the alignment panel, the option changes to "`Show relevant sequences`", which results in the differently colored three sequences in the alignment panel.

try the GENECONV, MAXCHI, CHIMAERA, SISCAN, PHYLPRO, 3SEQ, TOPAL, and DISTANCE options. (Be warned not to select the LARD option unless you are prepared to wait a few minutes.) The distance plot is similar to the "simplots" described in previous exercises.

Notice that most of the plots involve three lines (Fig. 16.11). This is because most of the exploratory methods scan through the alignment successively examining all possible combinations of three sequences. These three-sequence sub-alignments are examined in two basic ways, which are reflected in the two different color schemes used in the plots. The green–purple–yellow schemes indicate pairwise comparisons of sequences in the three sequence sub-alignments whereas the green–blue–red schemes indicate triplet analyses of the three sequences in the sub-alignment where the pattern of sites in each sequence is compared to the other two. For LARD and TOPAL a single line is plotted because only a single statistic is used to compare partitions of the three-sequence sub-alignment.

Point the mouse cursor at a region of the bottom right panel that has no colored rectangles in it and press the right mouse button. One of the options on the menu that appears is "`Show All Events for All Sequences.`" Select this option and use the scroll bar to get back to the representation of sequence KAL153. You should see that some more colored rectangles have appeared (Fig. 16.11). These include light green, dark green, yellow, and purple ones. Use the color key to see what methods these represent (Fig. 16.11, press the left mouse button on an open part of the bottom right panel).

Going back to the left-hand recombinant tract of sequence KAL153, you may notice that, of the five methods detecting this recombination signal, only the BOOTSCAN and MAXCHI methods (dark green and orange rectangles, respectively) agree on the position of the breakpoint to the right (or "ending breakpoint"). This indicates that there is some degree of uncertainty associated with the placement of this breakpoint. Click on the orange rectangle. The recombination signal detected by the MAXCHI method is displayed (Fig. 16.11). Notice that the purple and green peaks in the plot coincide with the right breakpoint position. It is worthwhile pointing out here that the peaks in MAXCHI, CHIMAERA, TOPAL, and PHYLPRO plots all indicate estimated breakpoint positions (note, however that for PHYLPRO plots the peaks face downward). These methods are geared to breakpoint detection.

Look at the upper left panel where the alignment is displayed and look for the caption "`Show all sequences.`" Beside the caption there is a button with a circular arrow on it. Press this until the caption says "`Show relevant sequences`" (Fig. 16.11). You will notice that the color-coding of the alignment has changed. On the MAXCHI plot double click on the right border of the pink region. You will notice that this causes the alignment to jump to around position 2800. Use the horizontal scroll bar beneath the alignment to scan backwards. Whereas most nucleotides are grayed out, some are colored green, purple, and yellow (Fig. 16.11). The grayed positions indicate nucleotides ignored by the MAXCHI analysis. Nucleotide pairs labeled yellow, green, and purple represent nucleotide positions contributing to the recombination signal plots in the same colors below. As the peaks in the plot below are green and purple, one would expect to see more purple colored nucleotide pairs on one side of the breakpoint (in this case it is the left) and more green nucleotide pairs on the other side of the breakpoint (in this case it is the right). Scan the sequence left of the breakpoint and you will notice that although purple sites are

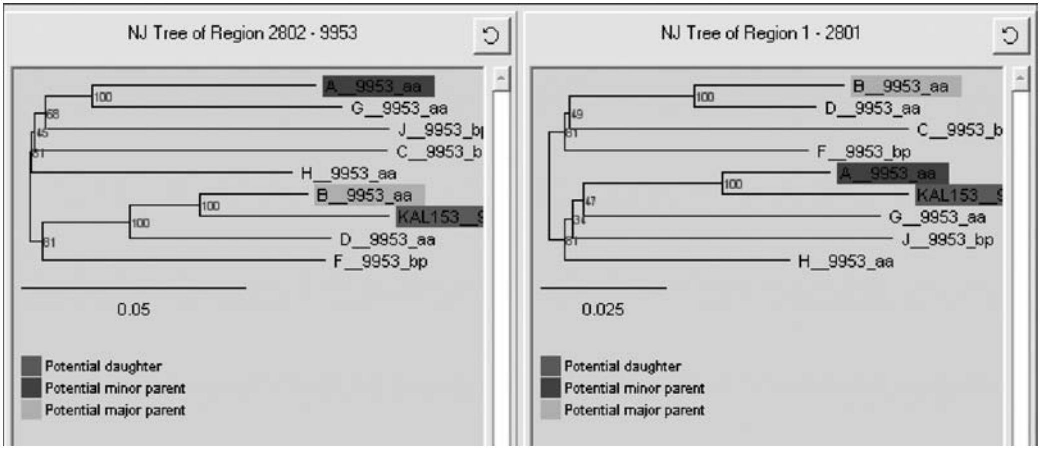


Fig. 16.12 Tree reconstruction using RDP3. On the right, the neighbor-joining (NJ) tree is shown for the first 2801 nucleotides in the alignment. In the NJ tree on the left, the KAL153 is clustering with subtype B.

most common there are also a lot of yellow sites. There is a particularly high ratio of yellow : purple sites between alignment positions 2570 and 2780. These yellow sites may represent either post-recombination mutations or sites at which the sequence in the alignment identified as resembling a parent (A in this case) differs from the KAL153's actual parent. RDP3's uncertainty over the breakpoint position is due to sequence A being a poor match for KAL153 in the immediate vicinity of the breakpoint, as indicated by the different breakpoint positions estimated by the different methods.

At the top of the screen press the "`Trees`" button. A window should appear with two phylograms or trees in it (Fig. 16.12). Now whenever the left mouse button is pressed on one of the colored rectangles in the bottom right panel of the main program window, two trees will be automatically drawn. The left tree is drawn using an alignment partition corresponding with the "background" sequence of the recombinant sequence being analyzed (in this case it is KAL153). The right tree is drawn from the alignment partition corresponding with the currently selected "recombinant region" – i.e. the alignment partition bounded by the ends of the tract of sequence represented by the colored rectangle currently selected. The default tree that is displayed is an ***UPGMA*** without any bootstrap replicates. While not necessarily the best type of tree for describing evolutionary relationships, it is very fast to construct and allows a quick preliminary assessment of whether there is any obvious phylogenetic evidence of recombination, i.e. you should look and see whether the proposed recombinant does indeed "jump" between clades of the two trees.

To more rigorously explore the phylogenetic evidence in favor of a recombination hypothesis you should at the very least draw a bootstrapped neighbor-joining tree. To do this, right click on each of the trees and select the "`Change tree type > Neighbor-joining`" option that is displayed on the menu that appears. Notice that whereas there is 100% bootstrap support for a KAL153-B clade in the left tree, there is 100% support for a KAL153-A clade in the right tree (Fig. 16.12). This is good phylogenetic support that KAL153 has arisen following a recombination event between A- and B-like viruses. These trees are constructed using the neighbor joining method found in the NEIGHBOR component of PHYLIP (Felsenstein, 1996). You could also have constructed least squares trees (using the FITCH component of PHYLIP; see Chapter 5), maximum likelihood trees (using PHYML; Guindon & Gascuel, 2003; see Chapter 6) or Bayesian trees (using MRBAYES; Ronquist & Huelsenbeck, 2003; see Chapter 7), but be warned that large trees can take a very long time to construct with these methods.

16.6.6 Exercise 6: Using RDP3 to refine a recombination hypothesis

In this exercise we will use what you have learned to complete our analysis of the A-J-cons-KAL153.fsa data set. It should become clear that the exploratory analyses carried out by RDP3 yield a set of recombination hypotheses that are not necessarily consistent among different methods or not necessarily the absolute truth about the recombination events that have left detectable traces in a set of aligned sequences.

RDP3 formulates its recombination hypotheses in a stepwise fashion. It firstly scans the alignment for all the recombination signals detectable with the primary exploratory methods selected, and then, starting with the clearest signal (i.e. the one with the best associated *p*-value), it attempts to identify which of the sequences used to detect the recombination signal is the recombinant. This is not a trivial process and the program uses a variety of phylogenetic and distance-based tests to try to figure this out. While the results of these tests are a completely objective assessment of which sequence is recombinant, the program will incorrectly identify the recombinant from time to time. When it has identified a putative recombinant sequence, RDP3 looks for other sequences in the alignment that might share evidence of this same recombination event (i.e. the recombination event may have occurred in some common ancestor of two or more sequences in the alignment and RDP3 tries to see if there is any evidence for this). From time to time it will miss a sequence, but RDP3 will usually be a bit overzealous grouping sequences it thinks are descended from a common recombinant ancestor. It then takes the recombinant sequence (or family of recombinant sequences) and splits it into two sections – one piece corresponding to each of the bits inherited from the recombinant's two parents. The smaller bit (from the "minor" parent) is then added to the alignment as an extra sequence. The alignment of the split sequences with the rest of the alignment is maintained by marking deleted bits of sequence with a "missing data" character (as opposed to the conventional "A,C,G,T,-" characters). This process effectively "erases" the recombination signal from the alignment and ensures it is not counted again when RDP3 next rescans the alignment from scratch for the remaining signals. Once the remaining signals are detected, it again selects the best and restarts the process.

The most important point of this explanation of how RDP3 explores and counts unique recombination signals is that the program does it in a stepwise fashion. You can see the order in which signals were analyzed by moving the mouse cursor over one of the colored rectangles at the bottom right of the screen. On the first line of the information appearing in the top right panel you will see the caption "EVENT NUMBER." This number tells you the order in which this event was analyzed. The number is very important because if the program has made any judgment errors when analyzing earlier events, it might affect the validity of the conclusions it has reached regarding the currently examined event.

When refining RDP3's recombination hypotheses after the automated phase of its exploratory analysis, it is strongly advisable to trace the exact path the program took to derive its hypotheses. You can do this by first clicking the left mouse button when the mouse pointer is over an empty part of the bottom right panel and then using the "pg up" (page up) and "pg dn" (page down) buttons on your computer keyboard to go through events in the same order that the program went through them. Do this and navigate to event 1 using the "pg up" and "pg dn" buttons. Don't be surprised if some events are skipped – these represent recombination signals that were detected by fewer methods than the cut-off specified in the "general" section of the options form you looked at earlier.

Event one is the recombination signal we analyzed earlier and persuaded ourselves was, indeed, genuine evidence of a recombination event. Move the mouse cursor over one of the colored rectangles representing the event and press the right mouse button. On the menu that appears select the "Accept all similar" option. This allows you to specify to RDP3 that you are happy to accept that this is genuine evidence of recombination and that it does not need to concern itself with this event during any subsequent reanalysis cycles. Press the "pg dn" button and you will be taken to event 2. Assess the evidence for this event the same way that you did event 1. Notice the blue area of the recombination signal plot. This specifies that one of the three sequences being examined in the plot contains missing data characters – these missing data is in the "blued-out" region of the plot. The missing data characters are those introduced by RDP3 into the background KAL153 sequence to keep it in alignment with the rest of the sequences after the recombination signal yielding the event 1 was analyzed.

After looking at event 2, move on to subsequent events. One of the following events has red capitalized text in the top right panel, "`POSSIBLE MISSALIGNMENT ARTIFACT`", which is particularly worrying. In the plot window for this event double click on the pinkish region specifying the recombinant tract. If you are not on the "`Show relevant sequences`" view of the alignment display, change the alignment display to this setting now (see the previous exercise). It is quite plain that there has been unreliable alignment of the three displayed sequences in the area identified as being a potential recombinant tract. Recombination signal detection is particularly error prone in poorly aligned sequences and it would be advisable to mark this evidence of recombination as being of dubious quality. To do this, move the mouse pointer over the flashing rectangle, press the right mouse button and select the "`Reject all similar`" option on the menu that appears. The rectangle and its associated caption should become gray.

If you navigate through the remainder of the signals you will notice both that their associated *p*-values are just barely significant and that they are only detectable by the CHIMAERA and MAXCHI methods (see the RDP3 manual for specific differences between both methods). These two methods are the most similar of the methods implemented in RDP3 and supporting evidence for one by the other should perhaps not carry as much weight as if it was provided by another of the methods. Nevertheless these remaining signals may indicate some genuine evidence of recombination. It is not entirely improbable that trace signals of ancient recombination events might be detectable in the sequences identified.

If you are feeling particularly brave, you may want to attempt an exploratory analysis of the more complicated recombinants in A-J-cons-recombinants.fsa. Besides some obvious inter-subtype HIV recombination events, an exploratory analysis will reveal that these contain some pretty good evidence of other, currently uncharacterized, recombination events.