I-C

Gateway to Tools of the HIV and HCV Databases

Charles Calef[a], Carla Kuiken[a], James J. Szinger[a], Brian Gaschen[a], Werner Abfalterer[a], Ming Zhang[a], Ning Tao[a], Robert Funkhouser[a], Karina Yusim[a], Mark Flynn[a], Anita Dalwani[a], Brian Foley[a], William Bruno[a], Thomas Leitner[a], Bette Korber[a]

I-C-1 Introduction

Over the years the staff of the HIV databases have developed web-based software for working with HIV sequence data. This is a general overview of the tools that are available on the HIV database website. Many of these tools are very simple, and were developed because we wanted to ease frequently-used computational tasks for our colleagues who use the database.

Some tools are tailored for HIV or HCV, and have counterparts developed specifically for the HCV (`http://hcv.lanl.gov/`) or HIV (`http://hiv.lanl.gov`) databases. Others are general and can be used for analysis of any organism. A fast way to understand what these programs do is to click the "Sample Input" button on the input page. This causes an example input file to be loaded into the input page, so you can run the program to get an idea about what the output looks like.

I-C-2 Outline of HIV database tools

This part of this publication provides an outline of these programs organized by their functions. A short description of each tool is provided. If a tool can be applied to any sequence, not just HIV or HCV, it is labeled, "General," while a tool that is applicable only to HIV or HCV sequences is labeled "HIV/HCV".

Formats

Convert between formats

Format Converter Converts sequence files between 18 standard bioinformatics formats. Automatic recognition of input format. (General)

Seq-Convert Converts between 8 standard bioinformatics formats. No automatic recognition of input format. (General)

Formatting for publication

SeqPublish Formats an alignment for publication: identical columns are replaced by dashes, and the sequences are printed interleaved in blocks of user-determined length. (General)

Sequence and alignment manipulation

Translate Converts nucleotide sequences to 1-letter amino acid sequences. (General)

Gapstreeze (Gap strip/streeze) Removes columns containing more than a user-determined percentage of gaps. (General)

Consensus Builds consensus sequences of alignments according to user specifications. (General)

Generation of alignments

Gene Cutter Extracts coding regions from a nucleotide alignment, codon-aligns and translates them, highlighting frameshifts, stop codons, and translates alternatives from IUPAC ambiguity codes. (HIV/HCV)

SynchAligns Synchronizes two alignments that overlap so they are aligned to one another, optionally trimming alignments to the region of overlap. (General)

PrimAlign Retrieves an alignment of a nucleotide sequence fragment (e.g., a primer) from our HIV complete genome alignment to assess variability. (HIV/HCV)

Epilign Retrieves an alignment of a HIV-1 peptide, epitope or functional domain from our web protein alignments to assess variability. (HIV/HCV)

[a] Los Alamos National Laboratory, Los Alamos, NM 87545, seq-info@lanl.gov, `http://hiv.lanl.gov/`

In *HIV Molecular Immunology 2005.* Bette T. M. Korber, Christian Brander, Barton F. Haynes, Richard Koup, John P. Moore, Bruce D. Walker, and David I. Watkins, editors. Publisher: Los Alamos National Laboratory, Theoretical Biology and Biophysics, Los Alamos, New Mexico. LA-UR 06-0036. p. 33–56.

Sequence analysis

Sequence characterization

Sequence locator Determines the beginning and ending positions numbers of sequence fragments in the genome or proteome relative to database reference strains. (HIV/HCV)

HIV/HCV BLAST Finds sequences most similar to your query in the HIV database. Helpful in detecting possible contamination issues. (HIV/HCV)

Sequence subtyping and recombination

SUDI Determines if a newly discovered set of related sequences should be considered a new subtype, according to standards developed by the HIV nomenclature committee. (HIV specific)

RIP Identifies intersubtype recombination by calculating similarity in a sliding window between your query sequence and a set of HIV-1 reference sequences of different subtypes. (General)

CRF-DRAW Maps HIV-1 recombinant breakpoints onto a graphical figure of the HXB2 genome, with parental subtypes indicated by different colors. (HIV/HCV)

Sequence analysis

VESPA Identifies site-specific signature residues that are rare in one group of sequences and common in another and calculates the frequencies of different amino acids in each position. (General)

PCOORD Summarizes the variation in the sequences in 10 dimensions using principal coordinate analysis. (General)

Hypermut Tracks base substitution patterns, and highlights G→A substitution events relative to other mutations, as they dominate in sequences damaged by hypermutation. (General)

N-Glycosite Highlights and tallies potential N-linked glycosylation sites in a protein alignment. (General)

Entropy Quantifies variation in a given position in an alignment using Shannon Entropy, and statistically compare variation in each position in two sets of aligned sequences. (General)

SNAP Calculates synonymous/non-synonymous substitution rates for a set of codon-aligned nucleotide sequences, based on the method of Nei and Gojobori. (General)

ADRA Finds mutations associated with anti-HIV drug resistance in HIV-1 protease, RT, integrase, and envelope sequences. (HIV)

Phylogeny

Phylogenetic trees

TreeMaker Generates a neighbor joining tree which is displayed and downloadable. PHYLIP outfile and Newick-formatted treefiles can also be downloaded. (General)

Search HIV sequence DB and make a tree Combines your sequences with those obtained through a database search, aligns the combined set, and generates a tree. (HIV/HCV)

FindModel Analyzes your alignment to see which evolutionary model best describes the input sequences. Can be used to generate a better phylogenetic tree. (General)

Immunology

PeptGen Generates a set of overlapping peptides according to user specifications from a protein sequence or an alignment. (General)

ELF Identifies potential and known epitopes in immunologically reactive peptides using HLA anchor motifs. (HIV/HCV)

Motif Scan Finds HLA anchor residue motifs within protein sequences for specified HLA serotypes, genotypes or supertypes using two major motif libraries. (General)

Hepitopes Tests for HLA alleles that are enriched in individuals that react with a set of peptides. Useful for population studies. (General, although the output can be combined with ELF and tailored to HIV/HCV.)

Also see, Epilign and Sequence Locator, above, two tools originally developed for mapping epitopes and their diversity.

Database searches

We are listing the HIV/HCV search capabilities here, although they will not be described in detail in this review; a comprehensive review of the databases and search interfaces will be included in the 2006 compendium.

HIV Sequence Database

Website content Google search for content and topics anywhere on our website. Where: small search box at upper left on most pages.

Sequence databases Search for sequences by selecting from numerous criteria such as subtype, genomic region, sequence length, geographic origin, time from infection, etc. From the results page, sequences can be selected, downloaded, used to generate a phylogenetic tree, aligned, translated, etc.

Advanced Search Build your own search criteria by selecting from a more extensive list of search fields than is available on the standard search page.

Search/display by geography Maps geographic distribution of HIV-1 sequences and their subtypes and can be used for sequence retrieval.

Drug Resistance DB Search for mutations that confer resistance to HIV-1 drugs. Search fields include protein, drug class and compound, amino acid position, citation, etc.

HIV Molecular Immunology Database

Website content Google search for content and topics anywhere on our web site. Where: small search box at upper left on most pages.

CTL epitopes Search for known CTL or CD8+ epitopes by protein, sequence, immunogen, vaccines, HLA, author or keywords. Retrieves epitope summaries from the literature, alignments, Medline links and epitope maps.

T-helper epitopes Search for T-helper or CD4+ epitopes, analogous to the CTL database.

Antibodies Search for HIV antibodies by protein, sequence, immunogen, AB type, author, monoclonal antibody name or keywords.

Best-defined epitopes Search for the best-defined CD8+ T-cell epitopes by serotype, genotype or protein.

Vaccine trials Search data from published studies on SIV, HIV and SHIV vaccine trials in nonhuman primates. Search criteria include objective, species, publications, vaccine immunogen, adjuvant and challenge.

I-C-3 Detailed Descriptions of Tools

Seq-Convert—Format conversion

Purpose This interface combines four different sequence alignment format conversion tools.

Background Many tools on the website now are fairly good at automatically recognizing common sequence formats, but in some cases they fail and manual conversion is necessary, or a user may need to change their sequence format to make it compatible with another tool. Seq-Convert is a combination of

1. Seq-Convert, which in turn combines code from an extension of the READSEQ program developed by Don Gilbert [Gilbert] and code developed by the HIV database staff to produce the table, GDE and SLX output formats. The interface can read all formats it writes except for these three.
2. Omniread: This tools attempts to automatically recognize the format of your input file, using a different combination of the programs Fmtseq and Readseq.
3. cf: This tool, developed by Charles Calef at the HIV database, attempts to automatically recognize and convert a total of 18 sequence formats.
4. Readseq2: A web interface to the update of Readseq, Don Gilbert's sequence reformatting tool.

Sequence reformatting is a recurring and difficult problem. Many formats are only very loosely defined, while others are very strictly defined but difficult to parse. Our databases mostly use fasta and table format, but some 50 different formats are used in the sequencing world. The Seq-Convert suite combines enough programs that almost any sequences can be converted to something more common, but it may require some experimentation to find the right tool for unusual formats. The tool shows the resulting sequences, so the user can decide quickly if the conversion has succeeded or not.

History and context Seq-Convert is a combination of efforts of several people. Don Gilbert created the Readseq and Readseq2 programs. Brian Gaschen wrote the code for Seq-Convert, the least flexible but probably the most robust tool; Charles Calef wrote cf, which is very flexible but not extensively tested. Carla Kuiken created Omniread by testing the Readseq and Fmtseq input and output algorithms and combining the best of those. Anita Dalwani combined all tools in one website.

SeqPublish

Purpose Make visually attractive, publication-quality alignments.

Background This interface takes a sequence alignment and replaces residues identical to those in a reference sequence with dashes. Either the first sequence in the input alignment will be used as the reference sequence for the output, or you can create a consensus from the alignment to be used as the reference sequence. This program is useful for making publication quality figures, or for exploratory work that involves visually assessing levels of variation in a region. It can be used in conjunction with alignments created using the search interface.

History and context Implemented by Patrick Rose and Kristina Kommander; designed by Carla Kuiken.

Translate

Purpose This simple program translates nucleotide sequences to amino acids in frame 1 or all frames. Users who retrieve nucleotide alignments from our database but who are unfamiliar with multiple alignment programs can easily obtain an amino acid alignment.

History and context Suggested by a database user. Implemented on the web by Charles Calef using a translation subroutine by Brian Gaschen.

Gapstreeze—Gap Stripping and Squeezing

Purpose Remove columns of gaps from an alignment. Generally useful for preparation of alignments for phylogenetic analysis. Offers various options like removing only positions that contain more than a user-specified percentage of gaps.

Background HIVs and SIVs not only evolve by base substitution, but they also frequently mutate through insertions and deletions (indels), which tend to be imperfect direct repeats focused in hypervariable "hot spots". These regions can be difficult to align, and gaps must be included to compensate for insertions and deletions relative to other sequences in the alignment. While indels are often forced into the same positions in an alignment, it can be difficult to resolve whether they have evolved by base substitution, the baseline assumption of most phylogenetic tree programs, or by insertion and deletion. For example, a single insertion event of 15 bases might suggest unreasonably large evolutionary distances between two otherwise very closely related sequences. A blunt way to resolve this problem is to simply remove all positions from an alignment that have a gap inserted to maintain the alignment. Alignment programs generally use a tilde (˜), or dash (-), to indicate a gap. Positions with missing information in some sequences will also be deleted, so the gene regions compared between all sequences will be the same. For this reason, users may want to remove particularly short sequences from an alignment before gap-stripping, as the alignment will only be as long as the shortest sequence included.

How to use Set the value of tolerance between 0% and 100%. A value of 0% will cause columns to be deleted if they contain any gaps (gapstrip), while a value of 100% will delete only columns that are entirely (100%) gaps (gapsqueeze). An intermediate tolerance value, of say, 10% will delete columns with more than 10% gap characters. You can define multiple gap characters and even specify ordinary letters to be gaps. This latter tactic is useful if, for example, you are interested in removing all columns containing IUPAC ambiguity codes (e.g. R and Y) from your nucleotide alignment, thereby preserving only columns with ATGCU. The "Show deleted columns" feature will include the intact first sequence in the output with marks showing columns that were deleted in the stripped alignment that follows. The default values set for the submission page will cause only columns that are 100% dash (-) characters to be removed.

History and context Many programs enable gap stripping, but Gapstreeze offers more flexibility with regard to specifying which columns are deleted, and retains a record of deleted columns. The record is particularly helpful if it is important that the alignment is codon aligned, as HIV sequences are often not biologically active and contain frameshift mutations. Any contiguous deleted columns that are not divisible by three would cause a frameshift downstream for the entire alignment. Brian Gaschen wrote the original script to facilitate preparing sequences for phylogenetic analysis, implementing features requested by Bette Korber; Charles Calef created an improved version and made a web interface.

Consensus Maker

Purpose Consensus Maker takes an input file of aligned sequences and calculates a consensus sequence for those sequences. Consensus sequences are useful as reference sequences for alignments or for reagent design.

How to use The consensus tools website offers three choices for creating a consensus of your alignment: simple, advanced, and ambiguity:

Simple consensus This option calculates a quick consensus of an alignment based on customary parameter choices.

Advanced consensus This option allows complete control over consensus parameters such as the values to be used for unanimity and majority, what characters to consider when making the consensus, whether to squeeze gaps, etc.

Ambiguity consensus A consensus sequence made up of the IUPAC ambiguity codes for each column in a nucleotide alignment can also be computed.
Example ambiguity consensus:

```
CON          AGCTRWMYSK HDBVNA
A.sequence1  AGCTAAACGG aagaAA
A.sequence2  AGCTAAACGG cgcgGA
A.sequence3  AGCTAAACGG tttcCA
A.sequence4  AGCTAAACGG tttcTA
B.sequence5  AGCtAAACGG tttcAA
B.sequence6  AGCtAAACGG tttcGA
B.sequence7  AGCtAAACGG tttcCA
B.sequence8  aGCtGTCTCT tttcTA
B.sequence9  aGCtGTCTCT tttcGA
B.sequence10 aGCtGTCTCT tttcTA
B.sequence11 #$*!?xxyyz zttcCA
```

Input options

Format of input alignment Consensus Maker recognizes most standard alignment formats.

Squeeze gaps If your alignment contains columns that are entirely gaps, they will be removed before a consensus is calculated. Default is squeeze gaps. You can also specify what character is used in your alignment to signify gaps. The default is -.

Output options

Do consensus for each block If the input contains blocks of sequences, such as subtypes, then calculate a consensus for each block, not just a single consensus for the alignment as a whole. Default is false.

Minimum number of sequences for a consensus If a block contains fewer than *n* sequences, then don't calculate a consensus for that block. Default is 3.

Do consensus of consensuses If consensuses are to be computed for each block in the alignment also calculate a consensus of these consensuses. (This would provide an HIV-1 M group consensus weighting all subtypes equally). Default is false.

Consensus + alignment Results will show consensus appended to the top of the user's alignment. Default is true. When false, the output consists of the consensus alone.

Output format A "pretty print" output shows your alignment aligned to the consensus with 50 characters per line and spaces every 10 characters.

Consensus calculation options

Unanimous value The fraction of characters in a column of the alignment needed to establish unanimity (shown as a capital letter) for that column. Default is 1.0.

Majority value The fraction of characters in a column of the alignment needed to establish majority (shown as a lowercase letter) for that column. Default is 0.5.

Use most common character This option determines what symbol to enter in the consensus for a column that has no majority character. Suppose a column contained letters AAAGGTTC. Does the user want that column to be represented in the consensus by a (i.e., the most common letter) or by ? (i.e., no letter forms a majority)? If so, then set this value to false. If multiple blocks are present in the alignment and there is a tie between two letters in one block, the program will try to resolve the tie by looking at that column of the alignment in all other blocks as well.

Characters to count when making consensus This is a set of characters ("letters") that the program considers when making a consensus. The default for nucleotide alignments is the set of valid nucleotide characters and the gap character ACGTU-. Using these defaults, the alignment column AAAAAXAA would have a consensus of A because the X character is ignored—it's not in the set of valid characters.

Use any character when making consensus Finally, if you want to consider *all* characters (including blanks, *, x, $, etc.) when making a consensus, check this box.

Options unique to ambiguity consensus

Characters to count when making consensus The program considers ACGTU when making a consensus.

Character presence percentage If a column of an alignment contained 99 A and 1 G would you want to give this a consensus of A or R, where R is the IUPAC code for purine (A or G)? In other words, if a character is present below a certain "presence percentage" threshold, should it be ignored when making the consensus? You can set this presence percentage threshold in the box provided. The default is 0, which means every occurrence of an A, C, G, T or U counts. If you had set the value to, say 2%, then the G in the above example would be ignored and the consensus would be A.

History and context We make alignments relative to consensus sequences to minimize the changes in the alignments and make it easier to see the differences between sequences. Consensus sequences also are central to circulating strains, and can be synthesized for vaccine design [Korber *et al.*, 2001; Gaschen *et al.*, 2002] or in reagent design (for example, HIV consensus sequence overlapping peptide sets for EliSpot [Korber *et al.*, 2001]). The tie-breaking algorithm and the concept of creating an HIV-1 M group consensus as the consensus of the subtype consensus sequences was developed by Bette Korber for reagent design. Charles Calef, with input from Carla Kuiken, developed this web-based tool to generate consensus sequences. Ready-made consensuses are available in our alignments section, useful in reagent design, and periodically updated.

Gene Cutter

Purpose Gene Cutter extracts pre-defined HIV-1 protein coding regions from a set of nucleotide sequences, then codon aligns and provides translations of the cut regions. It is particularly helpful for processing alignments of full-length HIV-1/SIVCPZ or HIV-2/SIVSMM genome sequences, or long interior regions that contain multiple coding regions.

Background All coding regions are clipped from a nucleotide alignment, and a matched codon-aligned nucleotide and translated protein alignment are created. Gene Cutter translates all codon possibilities in sequences containing IUPAC/IUB multistate characters, and provides a web-based format that allows users to move rapidly between nucleotide and protein alignments, and get details regarding translational properties of multistate characters. This tool is useful for sequence quality control of new sequences, as all stop codons and frameshifts are highlighted so potentially lethal mutations can be rapidly identified and cross-checked. Indels cause problems for multiple alignment programs, and often codons are split in an automated alignment and not readily translated; Gene Cutter will keep codons associated in the sequence. If a lethal frame shift occurs that is not compensated for within five amino acids downstream, the codon is translated as a hash, (#), and the appropriate downstream translation of the sequence beyond the inactivating substitution is thus enabled. The protein translations Gene Cutter creates can also be helpful for generating GenBank submissions.

Input The input file can include either HIV-1 and SIVCPZ sequences or HIV-2, SIVsm and SIVmac, but these sets should not be mixed because of different gene boundaries. Gene Cutter is organism-specific and does not extend to all primate lentiviruses, just the two human HIV lineages and their most closely related SIVs. Sequences can either be aligned, in which case Gene Cutter will modify the alignment to make it codon aligned and split out each codon region, or unaligned, in which case Gene Cutter will create a baseline alignment. The unaligned input option takes longer to run.

Output The matched nucleotide and amino acid alignments can be saved to your computer for further study. Working with the output on the web interface enables rapid switching between DNA and protein alignments and identification of problematic frame shifts and stop codons. A nucleotide alignment could be opened in BioEdit, and the ability to move between nucleotide and protein alignments would be retained. We recommend reviewing your Gene Cutter alignment (or any automatically generated alignment) and hand editing as needed.

History and context Brian Gaschen first developed this tool for internal database work, in response to increasing acquisitions of large numbers of full-length sequences that needed rapid processing. He built the public web interface incorporating suggestions provided by local users Bette Korber, Thomas Leitner, as well as outside users Jean Carr at the Henry M. Jackson Foundation, Rockville, MD, and James Mullins and colleagues at the University of Washington, Seattle, WA.

SynchAligns—synchronize alignments

Purpose Align two different alignments of the same gene region or protein to each other. The two alignments need not cover the identical genomic span but they must overlap. One application of this tool is combining a reference or database alignment with a novel set of study sequences.

Background A SynchAligns option was initially added to the BioEdit sequence editor [BioEdit] at the suggestion of the HIV database. The HIV/HCV database version uses align0 [Myers & Miller, 1988] to align one sequence from each alignment; the gaps that were inserted into each sequence are then applied to the rest of the alignment, and the two alignments are concatenated.

Input options The user may specify a reference sequence common to both alignments to be used in synchronizing. Failing that, the program will select the longest sequences from each alignment to use as references. Gap characters and whether to squeeze them can be specified. The synchronized alignment can be trimmed to the region of overlap between the two component alignments.

Output A single synchronized alignment in the same format as the second input file or "pretty-printed" versions of this alignment.

Example:

```
align1  GSEEL-RSLY-NTVATL
        GSEELMRSLYMNTVATL
        GSEELMRSLY-NTVATL

align2  EELRS-LYNTVATLYCVHQ
        EELRSPLYNTVATLYCVHQ
        EELRSPLYNTVATLYCVHQ
        EELRSPLYNTVATLY-VHQ
```

Result after SynchAligns:

```
      GSEEL-RS-LY-NTVATL-----
      GSEELMRS-LYMNTVATL-----
      GSEELMRS-LY-NTVATL-----
      --EEL-RS-LY-NTVATLYCVHQ
      --EEL-RSPLY-NTVATLYCVHQ
      --EEL-RSPLY-NTVATLYCVHQ
      --EEL-RSPLY-NTVATLY-VHQ
```

History and context This tool was developed at our sister database, the Los Alamos HCV database, by Carla Kuiken and Charles Calef, and is included in the HIV database tools as well.

PrimAlign—explore DNA primer diversity

Purpose PrimAlign generates an alignment of your nucleotide sequence against our complete genome alignment.

Background PrimAlign can be used to rapidly assess variation in primers, functional domains, or any HIV nucleotide sequence of interest. The HIV complete genome alignments are meant to approximate a population survey. They are updated annually and include only a single sequence per person, but still have sampling biases.

If obtaining an alignment of all sequences in the database is desired, or just a subset (for example, all Ugandan D subtype sequences in the database that span the fragment) the Sequence Locator tool can be used to find the

Figure I-C-1: PrimAlign sample output

```
QUERY                   ATGGGTGCGA GAGCGTCAAT ATTAA
A1.BY.97.97BL006        -YR------- --------G- -----
A1.KE.00.KER2008        .....----- --------G- -----
```

boundary positions of the fragment in the HXB2 genome, and these can be used to extract all sequences covering that region directly from the search interface of the HIV database [Gaschen *et al.*, 2001]. But beware, such database searches often can return hundreds of sequences from one subject if, for example, the individual was enrolled in a longitudinal study.

Input The direct sequence or its reverse complement (for primers) can be used as input.

Output A map that shows the position of the query relative to the HXB2 reference strain and an alignment of the fragment to all sequences from the same region in our curated alignment of complete genomes. Sequences whose names are printed in red are identical to the query. A simple fasta version of the alignment is available for downloading. Sequences include the subtype (A1 in the example below), followed by the country where the sample was taken (BY), the year of the sample (97), and finally the sequence name (7BL006).

History and context This tool was developed in parallel with its protein analog Epilign by Satish Pillai and Bette Korber, with support from Charles Calef. The alignment strategy and output options were later improved by Charles Calef and Brian Gaschen.

Epilign—explore epitope diversity

Purpose Epilign generates an alignment of your protein sequence against our web-based protein alignments.

Background Epilign can be used to get a rapid overview of the variability of an epitope, peptide, or protein. The location of the input sequence is automatically determined, and it is aligned to the HIV-1 database protein alignments, which excludes very similar sequences (e.g., multiple clones from one isolate, multiple sequences from one person), and so is meant to be a population survey.

Output A map (not reproduced here—see Figure I-C-5 in Sequence locator, below) is generated that shows the position of the query relative to the HXB2 reference strain. An alignment of the fragment to all sequences from the same region in our curated protein alignments is also created. Sequences whose names are printed in red, are identical to the query. If there are gaps in the main protein alignment, there is an option to squeeze the gaps and shift the sequence towards the C-terminal end. This is how potential T-cell epitopes would be seen by the immune system. The alignment is available for downloading in three simple formats. On the output page are two buttons that summarize the frequency of variants of your query. This analysis can be done for the entire alignment or for each subtype groups in the alignment.

Figure I-C-2: Epilign sample output, alignment results

```
         Query          SLYNTVATL
A1.KE.86.ML170          --F------
A1.KE.94.Q23            --F------
A1.SE.94.SE7253         --F----V-
A1.SE.94.SE7535         ---------
A1.SE.95.SE8538         ---------
```

Figure I-C-2 shows the top of the alignment for the SLYNTVATL epitope. The sequence names indicate the subtype (A1), the two letter country code (KE for Kenya), year of sampling (86 for 1986) and the sequence name.

Figure I-C-3: Epilign sample output, summary by subtype

```
Variant          Count   Percent
SLYNTVATL
---------         7       53.8
--F------         3       23.1
--F----V-         2       15.4
-------V-         1       7.7

   Total sequences = 13
Number of variants = 4
```

Figure I-C-4: Epilign sample output, subtype histogram

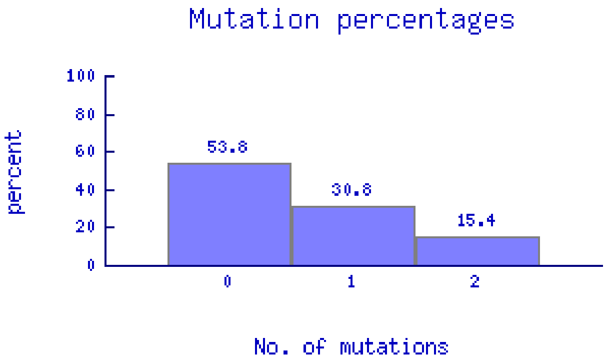


Figure I-C-3 summarizes the variation of the A1 subtype for this epitope. Of 13 A1 subtype sequences, 53.8%

are identical, and 23.1% differ from the query by having only an F substitution at position 3. These data are also presented in histogram form (Figure I-C-4). Further summaries of each kind of variant in every subtype are also provided but not shown here.

History and context This tool was developed in parallel with its nucleotide analog, Epilign, by Satish Pillai and Bette Korber, with support from Charles Calef. The alignment strategy and output was improved by Charles Calef and Brian Gaschen. Richard Koup (NIH) suggested adding the graphical representation of the identities in each subtype.

Sequence locator—HIV/SIV sequence locator tool

Purpose Finds the genomic position of a nucleotide or protein sequence in HIV-1/SIVcpz or HIV-2/SIVsmm/SIVmac relative to the reference strains HXB2 and SMM239.

Background Because HIV sequences vary in length, inconsistent and inaccurate numbering of locations in HIV DNA and protein sequences remains a problem in the literature. Positions published without reference to a strain (for example Gag positions 242–251), are meaningless because insertions and deletions change the length of HIV proteins. Often the numbers are not precise and do not match the reported sequence. This tool enables publication of precise and accurate positions relative to our reference strain HXB2 (GenBank accession number K03455). See the HIV database reviews about HIV and SIV numbering for more details [Korber *et al.*, 1999; Calef *et al.*, 2002a].

Output The query HIV epitope SLYNTVAAL produces the output shown in Figure I-C-5.

The "NA position relative to the HXB2 genome start" can be used as input on our sequence search interface to retrieve sequences of interest that span a given region. A user can also input HXB2 positions (e.g., p17 77–85) and retrieve the corresponding amino acids.

History and context This tool was initially designed and implemented by Bette Korber and Satish Pillai, with input from Joseph Sodroski at Harvard. Improved versions of this code were designed and developed by Charles Calef with input and the addition of the SIV locator from Brian Foley, John Mokili, Bette Korber, and Carla Kuiken.

HIV BLAST

Purpose Performs a BLAST search [Altschul *et al.*, 1997] restricted to the HIV Sequence Database.

Background The interface can handle both nucleotide and amino acid sequences, and calls these searches either BLAST or TBLASTN, respectively. In addition, you can access a smaller BLAST database that excludes sequences whose subtype is unknown; this restricted database can help identify the likely subtype of the query. HIV-1 specific BLAST results can be particularly useful for identifying potential contamination events. If the query perfectly matches a common lab strain, contamination may be indicated. While traditional BLAST searches explore vast databases looking for statistical support of genetic relationships, virtually all HIV and SIV sequences are statistically highly related. BLAST searches are useful simply to identify the closest sequences in the current database.

Output Aside from the standard BLAST scores and query/match alignments, you can also download all or a selection of the sequences your BLAST search finds. If you choose the 'master-slave' output option, the downloaded sequences will be aligned. If you choose 'pairwise', the downloaded sequences will not necessarily be aligned. Output includes the sequence name, sampling country, and subtype, which are not provided by an NCBI search.

History and context This derivative application of the search tool developed at NCBI was suggested by Carla Kuiken and Bette Korber, and implemented by Charles Calef.

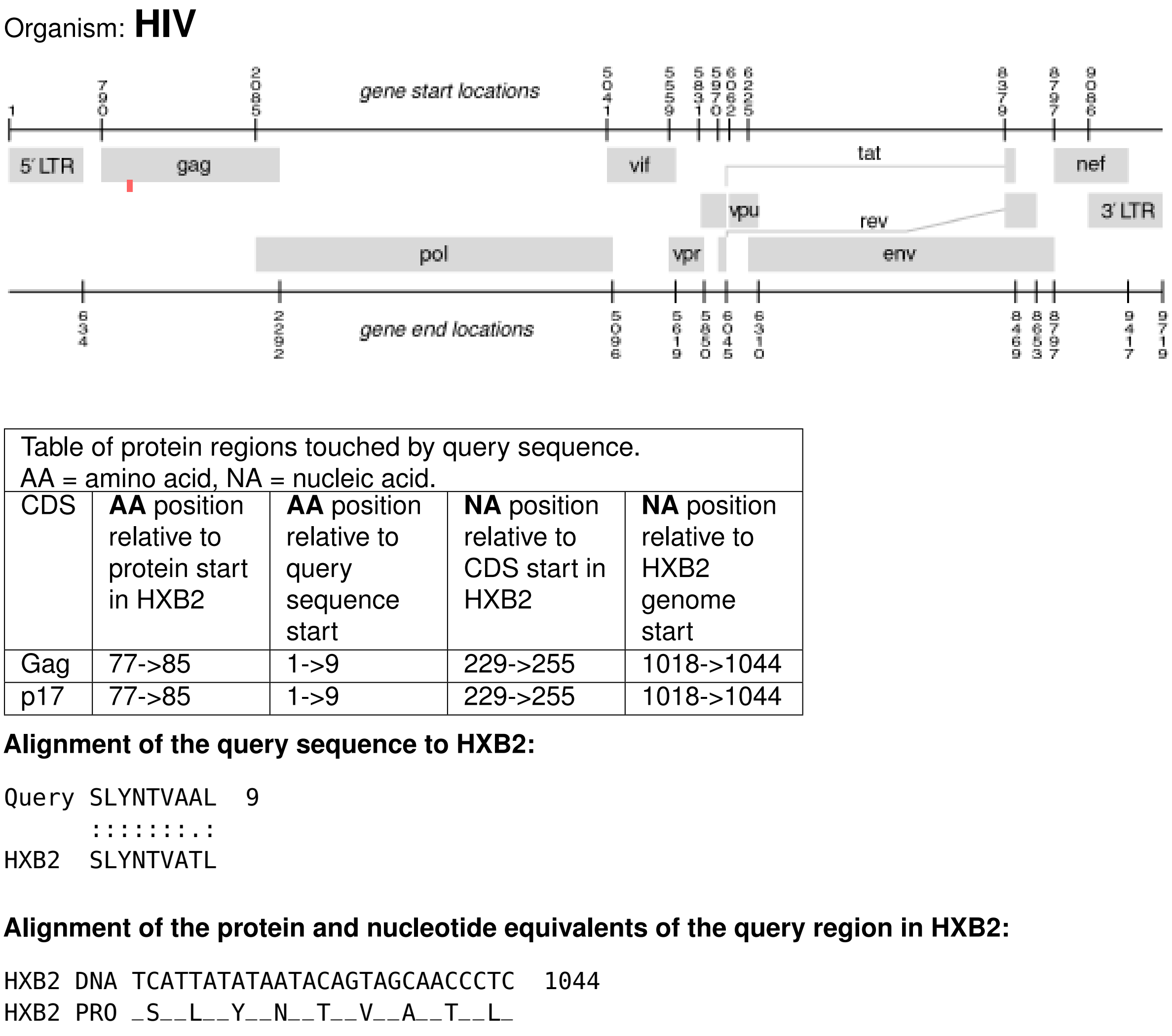
SUDI—Determining if a new subtype or sub-subtype has been identified

Purpose Helps determine if a newly defined clade of related sequences should most appropriately be considered a new subtype, a new sub-subtype, or part of a previously defined subtype.

Background SUDI was created at the request of participants in the 2000 HIV nomenclature committee [Robertson *et al.*, 2000a,b]. SUDI's purpose is to determine tree-based genetic distances for a new cluster relative to known subtypes, and then to compare these distances to typical distances found among pre-existing subtypes. Because absolute levels of similarity will depend on the region under consideration, the time of sampling in an ever-diverging epidemic, and the specific alignment, no absolute criteria for intra- and inter-subtype distances are included.

Input SUDI can use either an alignment or the outfile of a PHYLIP tree building program. The default tree for the program is a PHYLIP neighbor-joining tree built using an F84 model. If users want to base the analysis on a different tree, then a user tree can be created with PHYLIP, and the PHYLIP outfile can be used as the input for SUDI.

Figure I-C-5: Sequence locator sample output



Organism: **HIV**

Table of protein regions touched by query sequence.
AA = amino acid, NA = nucleic acid.

CDS	**AA** position relative to protein start in HXB2	**AA** position relative to query sequence start	**NA** position relative to CDS start in HXB2	**NA** position relative to HXB2 genome start
Gag	77->85	1->9	229->255	1018->1044
p17	77->85	1->9	229->255	1018->1044

Alignment of the query sequence to HXB2:

```
Query SLYNTVAAL  9
      :::::::.:
HXB2  SLYNTVATL
```

Alignment of the protein and nucleotide equivalents of the query region in HXB2:

```
HXB2 DNA TCATTATATAATACAGTAGCAACCCTC  1044
HXB2 PRO _S__L__Y__N__T__V__A__T__L_
```

Output Based on the tree, histograms will be generated showing the range of intra-subtype distances, inter-subtype distances, and sub-subtype distances. The category that a given pairwise distance is assigned to (intra-subtype, inter-subtype, or sub-subtype distances) will depend on how the sequence was labeled (A_, B_, ...) and how the clusters were defined. The cluster of sequences that the user is interested in, those sequences labeled 'U', will be highlighted. The U intra-subtype distances will be shown, and the U inter-subtype distance relative to the subtype closest to U will be shown. This way the user can determine if the novel cluster should be broken into sub-subtypes, or be considered part of a previously defined subtype.

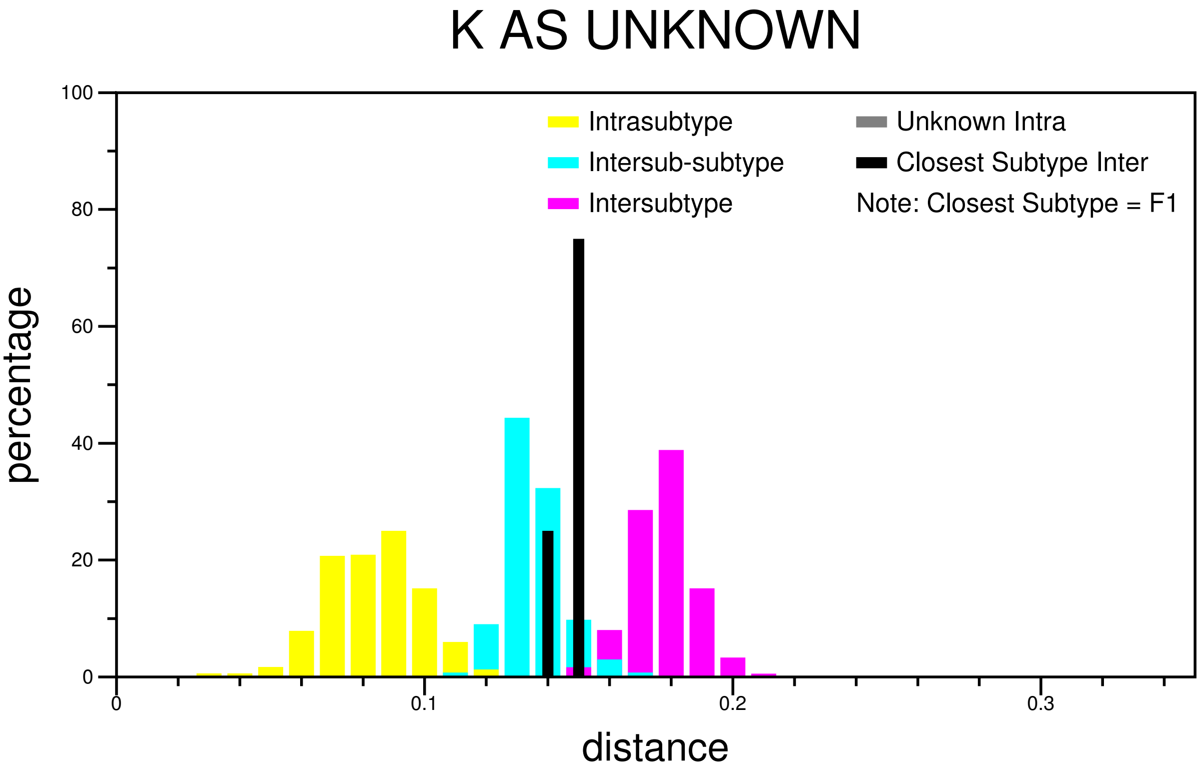
History and context SUDI was written by Bette Korber and Bob Funkhouser. Patrick Rose assisted with the interactive web interface.

RIP

Purpose The Recombinant Identification Program (RIP) is a computer program developed at the HIV Database to identify genetic sequences that appear to be mosaics of members of distinct phylogenetic clades. The idea is that such mosaic sequences are likely recombinants [Siepel *et al.*, 1995].

Background RIP was designed to detect recombinants of sequences belonging to different subtypes of HIV-1, but it can be used for other applications, including analysis of non-HIV sequences. The program moves a "window" of specified length stepwise across an alignment containing a query sequence and several background representatives. For each step in the window's progression across the genome, the query is compared to each of the background representatives within the window, and similarity is quantified as the fraction of identical base pairs. The values are

Figure I-C-6: SUDI sample output



retained, and the window is advanced one position. After the window has traversed the alignment from left to right, the program displays output revealing which background representative the query sequence most resembles at all possible positions. So-called "best matches" are marked if they are significant according to a statistical test.

Input There are three options for creating the alignment that RIP analyzes. 1. You may submit a single sequence, the query, and have RIP align it automatically to the subtype consensus alignment. 2. You may submit a single sequence, the query, and then build a custom background of sequences by selecting from a list provided on the website. 3. You may submit an alignment of your own that you have built with your query as the first sequence in the alignment. This option runs faster than the other two because RIP skips the alignment step. The size of the sliding window and the statistical significance threshold can be adjusted by the user. Gaps in the alignment can be handled in four different ways.

Output The default output consists of graphs showing the distances between the query sequence and the background set for each window position, and an alignment annotated with the best match sequence and whether or not it is statistically significant.

History and context A sliding window approach to identifying recombination events was first developed by Bette Korber and Adam Siepel [Siepel *et al.*, 1995]. In the fall of 1995 we used RIP to scan the HIV Database's env and gag master alignments for intersubtype recombinants [Siepel *et al.*, 1995]. Since its original development, RIP's web interface has been much improved by Thomas Leitner, Carla Kuiken, Brian Gaschen, Bette Korber, and Charles Calef.

Draw CRF—Make a figure to graphically represent recombinant genomes

Purpose Draws maps of HIV-1 genomes that are known to be recombinant. The different subtypes that comprise your genome appear as colored regions in the map.

Input The data used by the program record the points at which each component subtype in the genome begins and ends. If exact breakpoints are not known, there is a mechanism for entering uncertain boundaries. The breakpoint coordinates should be in standard HXB2 coordinates. The program can convert your data to HXB2 coordinates automatically if you select that option. A sample of input data looks like this:

```
1 2677 G
2678 3345 A
```

Figure I-C-7: RIP sample output

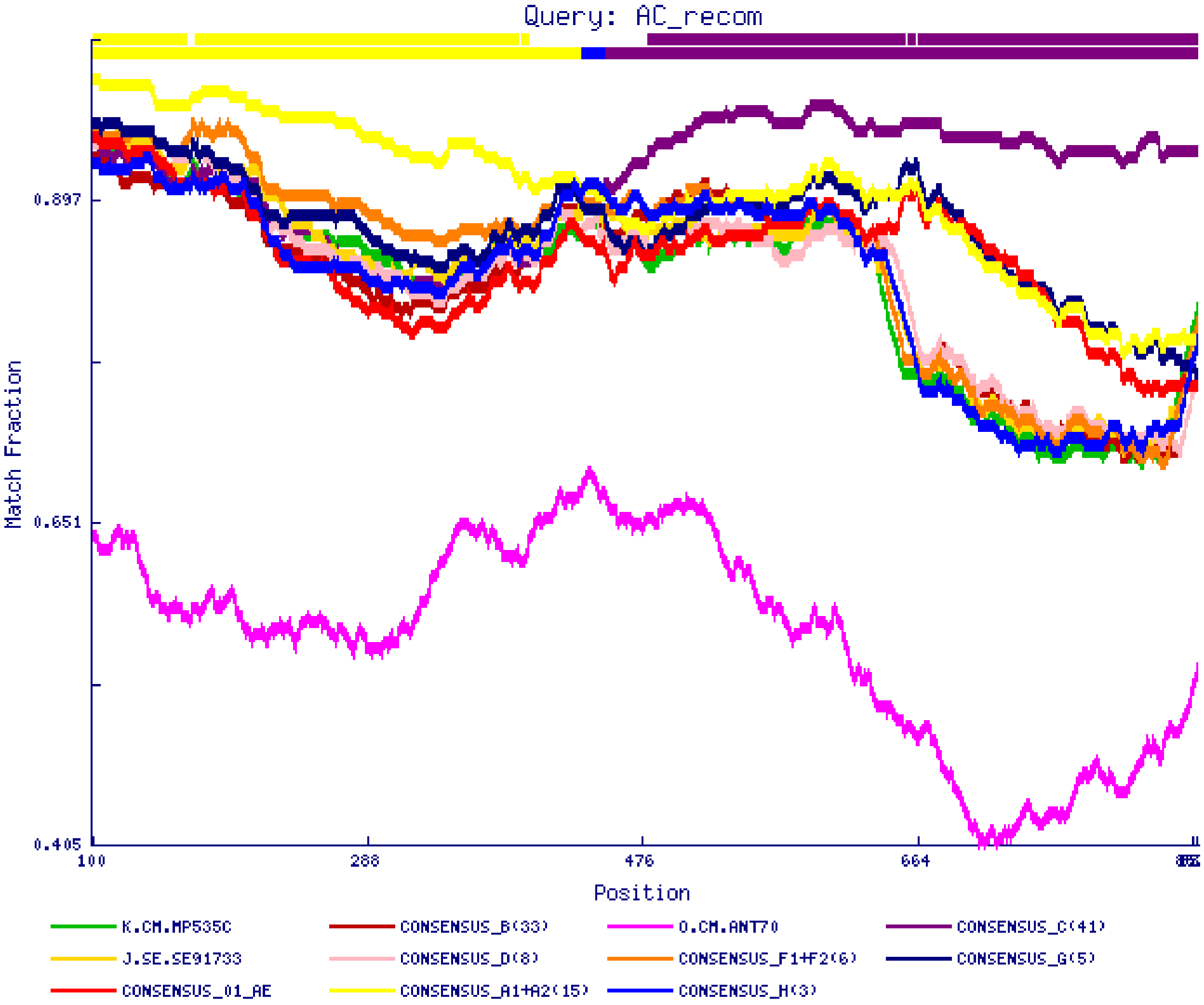
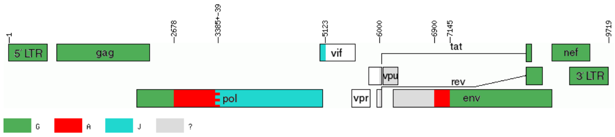


Figure I-C-8: RIP sample output

```
                                                                               180
           AC_recom GAGTCCTGGCTGTGGAAAGATACCTAAAGGATCAACAGCTCCTAGGAATTTGGGGCTGCT
    CONSENSUS_01_AE -----------------------------------A--T--------C------------
CONSENSUS_A1+A2(15) ------------------------------------------------------------
    CONSENSUS_B(33) ----------------------------------------------G--------T----
    CONSENSUS_C(41) -----------A-A--------------------------------G-------------
     CONSENSUS_D(8) --A----------------------------------------------------T----
  CONSENSUS_F1+F2(6) ----------------------------------------------G-------------
     CONSENSUS_G(5) -------------A--------------------------------G-------------
     CONSENSUS_H(3) -------A-----------------------------------G--G-------------
        J.SE.SE91733 ------------------------------------------------------------
        K.CM.MP535C --A----------A--------------------------------G-------------
         O.CM.ANT70 -CC-G--A--CT-A----CC-TA---C--A----G--A------A-CC-A--------TA
         Best Match bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
      Significant ^ ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  ^  ^^^^^^^^^^^
```

```
3346 3423 AJ
3424 5123 J
6000 6899 ?
6900 7144 A
7145 9719 G
```

Figure I-C-9: Map of mosaic HIV-1 genome whose breakpoint data is presented above.



Output The various subtype regions of the genome appear as different colors. Gray is used to illustrate regions of uncertain subtype. Uncertain breakpoint boundaries are illustrated on the map by an interfingering of the two colors that make up the two flanking regions; the breakpoint is shown at the center and expresses the size of the regions using a plus or minus notation. Regions with no data appear uncolored in the map.

History and context This tool was developed by Charles Calef with design suggestions from Thomas Leitner.

VESPA—Viral Epidemiology using Signature Pattern Analysis

Purpose Identifies sites which are common in one group of sequences, and are rare in another group.

Background VESPA detects signature patterns (atypical amino acid or nucleotide residues) in a set of query sequences relative to a set of reference sequences [Ou *et al.*, 1992; Korber & Myers, 1992]. It can be used to detect amino acids that characterize differences between two groups of sequences. It compares two groups of sequences and looks for a "signature" pattern, or the set of amino acids that is conserved among each set, but differing between the sets. VESPA will pick out those distinguishing amino acids, and calculate their frequencies in each set. Nucleotide alignments can also be used; however, amino acids are used as representative examples in the following discussion.

VESPA calculates the frequency of each amino acid (or nucleotide) at each position (column) in an alignment for the query and reference set, and selects the positions for which the most common character in the query set differs from that in the background set. The frequencies of characters at the distinguishing sites are also calculated [Ou *et al.*, 1992; Korber & Myers, 1992]. VESPA can also be used to compare the query and background sets' similarity in amino acid length, total charge, and amino acid content.

The sequences should all be of the same length, so if some sequences are shorter than others, the user should insert stars (∗) to complete the sequences, to indicate that no information was available at those sites. Positions with stars will be discounted from frequency calculations. Insertions made to maintain the alignment should be dashes (-); positions with dashes will be counted, and included in the signature pattern analysis.

The allowed characters for inclusion in an alignment are valid one-letter amino acid codes, -, and ∗; lower case letters are treated the same as uppercase letters. An asterisk is treated as 'missing', as will any other character; these will not be counted in the signature pattern tally. Therefore, if you have a stop codon and you label it as a dollar sign, it will be treated as if you have no information at that site.

The tool can also calculate various statistics, such as the amino acid frequencies in each group, the number of conserved signature amino acids in each sequence, and the amino acid content, amino acid length, and total charge.

History and context VESPA was originally written by by Bette Korber. Mark Flynn created the web interface and implemented a number of suggestions by Bette Korber and Carla Kuiken.

PCOORD—Principal Coordinate Analysis

Purpose PCOORD is a procedure to identify meaningful multivariate patterns in sequence data.

Background The Principal Coordinate Analysis method is very similar to regular Principal Component Analysis. The method was developed by the statistician J. C. Gower [1966]. PCOORD attempts to summarize the variation in the sequences in a limited number of axes or dimensions. A "dimension" is basically a combination of positions in a sequence that behave similarly, for example, "Position 133 usually has an A when position 250 has a G." One way to describe the process of finding these dimensions is as follows. If we have a two-dimensional swarm of datapoints, then we need two dimensions (the x and y axis) to describe the variation in our data. However, if the swarm is very elongated and the points almost lie on a straight line, then we really need only one dimension, although we use two. PCOORD uses a mathematical method to find the best way to describe a multi-dimensional dataset in a smaller number of dimensions, which are linear combinations of the original dimensions.

The dimensions are not necessarily biologically meaningful, but they can be. Quite frequently, some dimensions that are extracted correspond to an epidemiological variable or some other feature of the data. The patterns that are found using PCOORD usually can be seen in a phylogenetic tree as well, but they may be much less pronounced there.

Output Each sequence gets a score on each of the dimensions, and these scores can be plotted pairwise. The coordinates can be downloaded, so that a better-looking graph can be produced with a spreadsheet or graphing program. The PCOORD program can identify each sequence with a character (number, letter, or symbol such as ∗ or ˆ). To use that feature, you need a file with one character for each sequence. In the dimension plot, the point representing each sequence will then be identified by the corresponding character.

History and context The PCOORD program suite was developed by Des Higgins [1992] (then at the EMBL), and adapted for the UNIX platform by Jack Leunissen of the CAOS/CAMM institute in Nijmegen, The Netherlands. The web interface was created by Kersti Rock based on specifications by Carla Kuiken [Kuiken *et al.*, 1993; Potts *et al.*, 1993; Kuiken *et al.*, 1994].

Hypermut

Purpose Hypermut highlights hypermutational changes among other base mutations [Rose & Korber, 2000]. It takes a nucleotide alignment and documents the nature and context of nucleotide substitutions in a sequence population relative to a reference sequence.

Background A retroviral provirus is considered hypermutated if it undergoes an inordinate number of identical transitions, usually guanine to adenine (G←A). Hypermutation most often results in the production of replication-incompetent virus. Several papers were published in 2003 describing a host cellular defense mechanism that induces hypermutation in reverse transcribed nascent retroviral DNA. The Vif protein of HIV seems to be able to counter this activity [Mangeat *et al.*, 2003; Zhang *et al.*, 2003; Lecossier *et al.*, 2003].

Identifying hypermutated sequences in a viral population can be critical when reconstructing viral phylogenies (to assess the effects of drug therapy, immune surveillance, etc.). The apparent rate of viral evolution can be dramatically exaggerated by hypermutated sequences, when in actuality these viruses are evolutionary dead ends; their profound divergence is an artifact of a single aberrant round of replication.

Input The first sequence in the input alignment will be used as the reference sequence for the entire analysis, so this sequence should be chosen carefully. For example, for an intrapatient set, the reference should probably represent the most common form in the first sampled time point. For a set of unrelated sequences, the consensus sequence for the appropriate subtype would be used. Also, you may choose to display a general or region-specific overview of your sequences.

Output Hypermut output consists of

- a data sheet summarizing the hypermutations,
- a graphical overview of all the sequences and their nucleotide changes,
- a graphical overview of all mutations in a selected sequence, and
- a table for allowing quick analysis of mutations resulting in stop codons.

The program allows either an overview of the complete sequence, or a detailed view of a subregion. The hypermutational changes are color coded.

History and context HYPERMUT was originally written by Bette Korber and web development was undertaken by Patrick Rose; improvements were made by Werner Abfalterer. Francine McCutchan, Jean Carr, and Feng Gao offered suggestions for additional analysis. An application of the method to the HIV database is described by Rose & Korber [2000].

N-Glycosite

Purpose This tool highlights and tallies potential N-linked glycosylation sites in an aligned set of protein sequences.

Background The N-linked glycosylation site pattern `N-x[ST]` (where `N` is asparagine, `x` can be any amino acid, and `[ST]` is serine or threonine) is called a sequon. N-Glycosite can be used for any protein alignment, but is particularly helpful for the HIV envelope as it is heavily glycosylated. Sequons vary in position and number, and glycosylation can be critical for protein function and for immune evasion [Zhang *et al.*, 2004]. The extent of actual glycosylation of a sequon depends on the context, which could be expanded to a four amino acid `Nx[ST]y` pattern where the amino acids in the `x` or `y` positions influence the glycosylation efficiency. In particular, proline in position `x` or `y` does not favor N-linked glycosylation. Thus we also provide `Nx[ST]` or `Nx[ST]y` summaries.

Input If you just want to tally the number of N-glycosylation sites, this can be done with unaligned sequences, but to track movement or changes in particular sequons, aligned sequences are necessary.

Output The initial output page contains links to all other output files. These include an alignment with the N-linked sites highlighted (Figure I-C-10), tallies of the number of sequons in every sequence in an alignment, figures showing the fraction of each position in an alignment that contains an asparagine (N) that is part of a sequon (Figure I-C-10), and tallies of the number of sequons in a window of user specified length moving through the protein alignment.

History and context Bette Korber developed a simple version of this code for analysis of acute infection sequences [Derdeyn *et al.*, 2004]. Ming Zhang then made a web interface and added many useful features suggested by Brian Gaschen and Bette Korber.

Entropy

Purpose Assigns a quantitative measure of diversity to every position in an alignment, and compares one alignment to another to see if there is statistically supported evidence for positions with increased diversity in one set relative to another.

Background This code provides one strategy for quantifying sequence diversity, using the information theory concept of Shannon entropy [Shannon, 1948]. This code was originally used to compare blood derived HIV envelope sequences from two data sets, and we found evidence for sites that were more variable in the blood than brain [Korber *et al.*, 1994]. A second application compared the variability of sequence positions to immunologically important regions. Here the Shannon entropy of each position was calculated, and compared to some other biological property that has been characterized for that position. For example, the number of distinct cytotoxic T-lymphocyte (CTL) epitopes that span a position inversely correlates with the variability of that position [Yusim *et al.*, 2002]. In this application, the entropy scores were compared with another score of biological interest, in our case CTL epitope density.

Output Entropy comes in two flavors, called Entropy-one and Entropy-two. To calculate the entropy for positions in a single alignment, use the Entropy-one interface. If you want to compare the entropy in two different sequence sets (they will need to be aligned to each other), use the Entropy-two interface.

Entropy-one also estimates the average entropy of all positions in a given window size, advancing the window by a user-specified length.

Entropy-two compares the entropy in to different sequence sets. To assess statistical significance, a user-specified number of Monte Carlo randomizations of two sequence sets can be performed, and a comparison of the difference in entropy between the real data and the randomized data sets can be used to determine whether a difference in entropy was likely to have been observed by chance alone or is significant.

History and context This code was originally written by Bette Korber [1994] with the Monte Carlo randomization implemented by James Theiler. Ming Zhang adapted it to the web and added features suggested by Brian Gaschen, Carla Kuiken, and Bette Korber.

SNAP

Purpose Calculates synonymous versus non-synonymous base substitutions for all pairwise comparisons of sequences in a codon-aligned nucleotide alignment.

Background SNAP is based on the method of Nei & Gojobori [1986]. You should be familiar with this paper before using this program.

Output The number of synonymous and non-synonymous codon changes are counted, as well as the number of potential synonymous and non-synonymous changes when comparing two sequences. Ambiguous codons or codons with insertions are excluded from the tally of compared codons. The output provides overall sequence distances as well as a codon by codon summary. One must be wary when doing typical statistical analysis of these values. Distributions of values that are far from Gaussian are commonly found, so you should either check to see if you have a Gaussian distribution, or default to the use of non-parametric statistics, like a Wilcoxon rank sum test. Therefore the averages given at the bottom are only meant as a crude guide. Also, if one uses the full column of values for all pairwise comparisons (say all values of dn for one set, compared to all values for another set) there is a non-independence of points issue to be considered. An alternative is the use of a sequence like a consensus or a best estimate of an ancestral sequence as the first sequence in the alignment, and then just use the comparison of the first sequence to all others rather than all pairwise comparisons.

History and context SNAP was written by Bette Korber [2001], and adapted for the web by Satish Pillai. It was one of the earliest attempts to analyze synonymous versus nonsynonymous mutation rates in a way that was not averaged over entire genes; more sophisticated tree-based methods were later developed, although this method is simple in concept and also tracks insertions and deletions, and so still merits consideration. An application of

Figure I-C-10: Section of output from an HIV hypervariable region with N's that might be glycosylated highlighted in red.

```
                   110        120        130        140        150
D.UG.94.94UG1141   LNCTN--WVT DTT------- -N-TT----- ---------- G-MANCSFNI
01_AE.CF.90.90CF11 LHCTK--AKL NDT------- YNGTAKLND- -------TIG DEVRNCSFNV
02_AG.CM.97.97CM8  LDCHD--YNS TSH-NYSSIS NNMTEEM--- -------EMK GEIKNCSFNM
CPZ.CM.-.CAM3      MECRK--VTF NSTSN----- RNKTSTMTTN SPNEKX---D STVKNCTFNM
```

Figure I-C-11: The fraction of each position that is an N embedded in a potential N-linked glycosylation site in an alignment of 8 sequences.

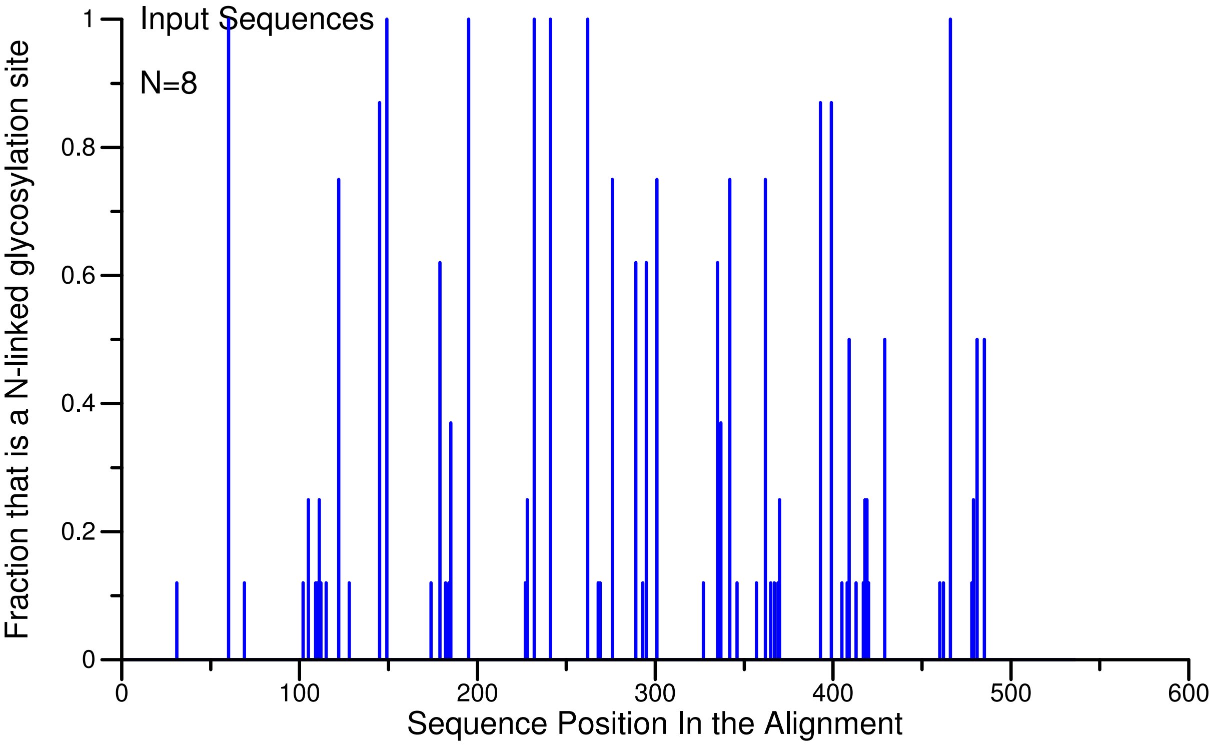
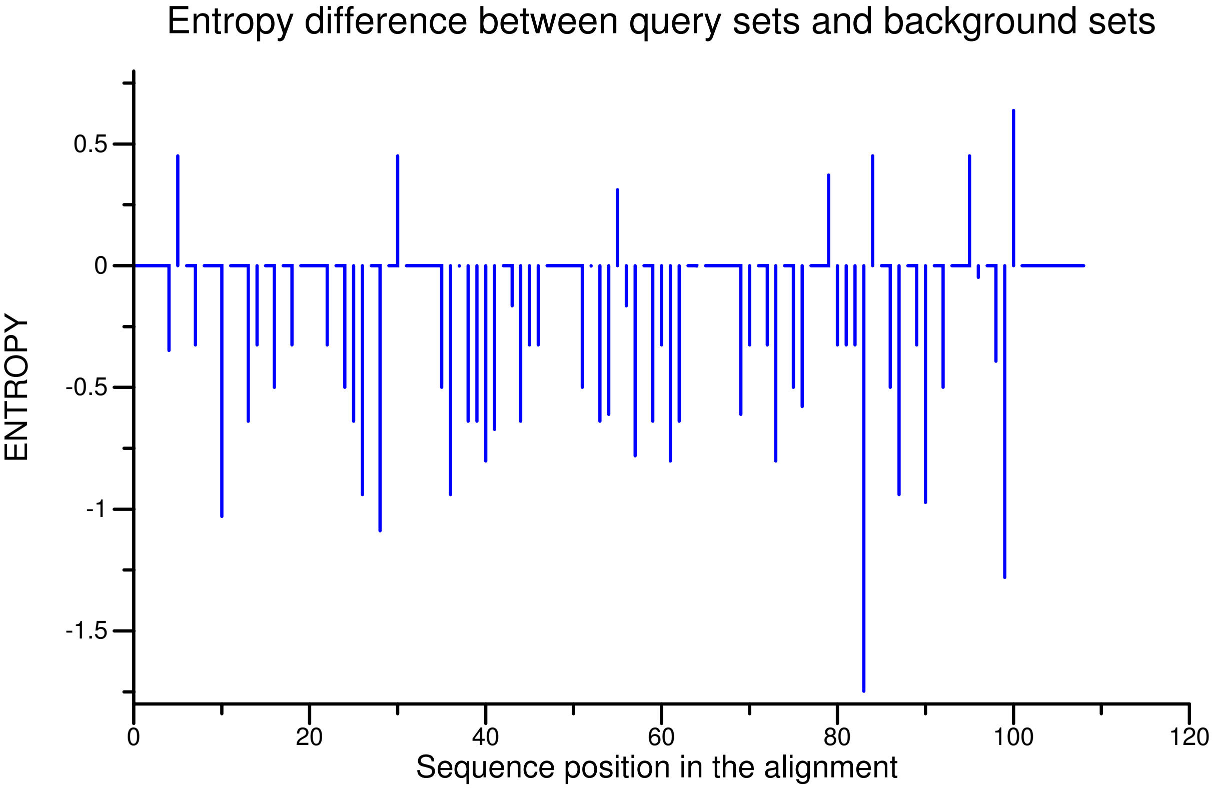


Figure I-C-12: The beginning of the output file that calculates the average entropy for each window of 15 with an overlap of 11. The Entropy tool can be used in conjunction with PeptGen tool to assign an average entropy score to each peptide as investigators are designing a panel of reagents. The top sequence is the consensus from the input.

```
        10        20        30        40
         |         |         |         |
LAEEEVVIRSENFTDNAKTIIVQLNESVEINCTRPNNNTRKSIHI
LAEEEVVIRSENFTD[Average entropy = 0.1778]
           NFTDNAKTIIVQLNE[Average entropy = 0.2795]
                      QLNESVEINCTRPNN[Average entropy = 0.3498]
```

Figure I-C-13: The difference in entropy between the two input files. The background file is more variable in positions that drop below the line, the query file in those that rise above the line.



the SNAP package is described in Ganeshan *et al.* [1997]. Statistical analysis was added at the suggestion of Yumi Yamaguchi-Kabata, following the method described in Ota & Nei [1994].

ADRA

Purpose Finds mutations associated with anti-HIV drug resistance in HIV-1 protease, RT, integrase, and envelope sequences. Accepts both nucleotide or amino acid sequence input.

Output Produces a table of resistance-associated mutations (Figure I-C-14) and an alignment with mutations indicated (Figure I-C-15). Tabulates drugs to which this sequence may show resistance and links to additional information on these mutations in the HIV Drug Resistance Database [Clark *et al.*, 2001].

History and context This tool was designed and written by Patrick Rose and Charles Calef as a way to explore the HIV Drug Resistance Database (`http://resdb.lanl.gov/Resist_DB/default.htm` maintained within our Los Alamos database by John Mellors.

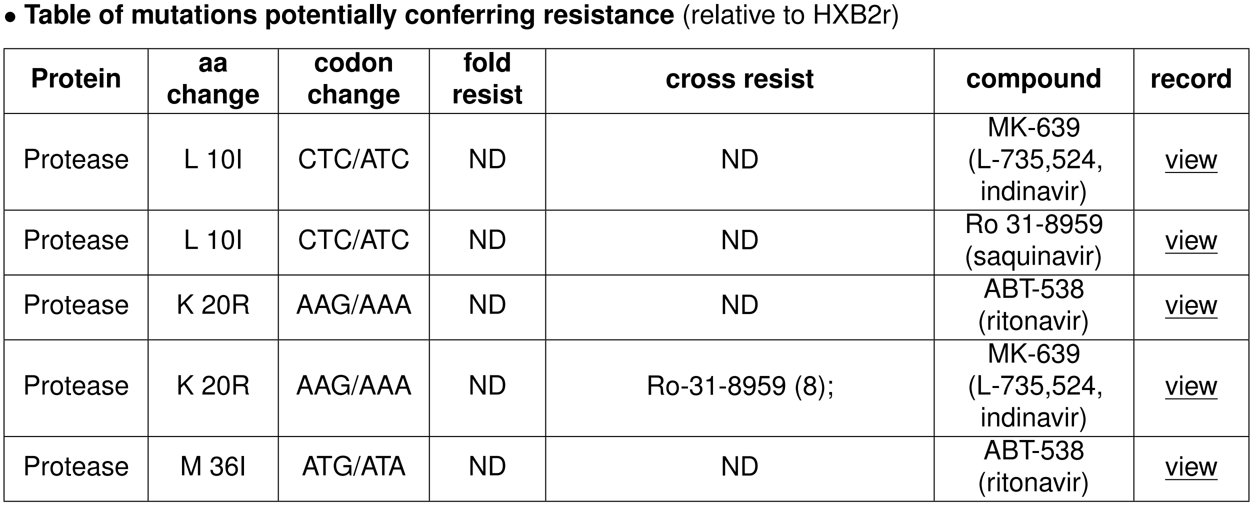
TreeMaker

Purpose To produce "quick and dirty" trees. Our aim is not to make them dirty, but to make them quickly. These trees are generally not publication quality, but are meant to be used in an exploratory framework.

Background TreeMaker generates a neighbor-joining tree based on a sequence alignment. The tree is very basic and quite possibly not optimal for any dataset. The database tool FindModel, described below, can be used to determine the optimal model. We also provide a tutorial that gives some background information about phylogenetic tree construction, and provides further links.

Output The tree is displayed as a PNG file, and can also be downloaded as a PostScript or PDF file. Currently, the PHYLIP outfile and the Newick-formatted treefile can also be downloaded. By default, this interface uses the F84 distance model (also called "ML" because it is used in PHYLIP's maximum likelihood phylogeny program DNAML). This model incorporates different rates of transition and transversion, and also allows for different frequencies of the four nucleotides. Several other distance models are available. TreeMaker will be updated to use

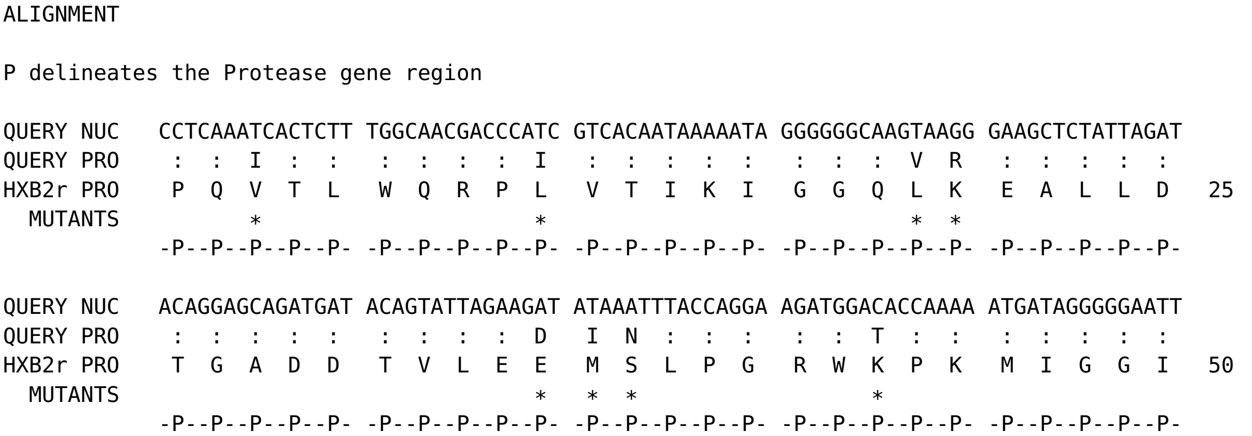
Figure I-C-14: Table of mutations found in user's input that are known to confer resistance to HIV-1 antiretroviral drugs. The column on the right links to the full database record of this mutation.



• **Table of mutations potentially conferring resistance** (relative to HXB2r)

Protein	aa change	codon change	fold resist	cross resist	compound	record
Protease	L 10I	CTC/ATC	ND	ND	MK-639 (L-735,524, indinavir)	view
Protease	L 10I	CTC/ATC	ND	ND	Ro 31-8959 (saquinavir)	view
Protease	K 20R	AAG/AAA	ND	ND	ABT-538 (ritonavir)	view
Protease	K 20R	AAG/AAA	ND	Ro-31-8959 (8);	MK-639 (L-735,524, indinavir)	view
Protease	M 36I	ATG/ATA	ND	ND	ABT-538 (ritonavir)	view

Figure I-C-15: Alignment of user's nucleotide sequence, translated to protein and aligned to equivalent protein regions of HXB2. Mutations are indicated.



```
ALIGNMENT

P delineates the Protease gene region

QUERY NUC   CCTCAAATCACTCTT TGGCAACGACCCATC GTCACAATAAAAATA GGGGGGCAAGTAAGG GAAGCTCTATTAGAT
QUERY PRO    :  :  I  :  :   :  :  :  :  I   :  :  :  :  :   :  :  :  V  R   :  :  :  :  :
HXB2r PRO    P  Q  V  T  L   W  Q  R  P  L   V  T  I  K  I   G  G  Q  L  K   E  A  L  L  D   25
  MUTANTS          *                     *                            *  *
            -P--P--P--P--P- -P--P--P--P--P- -P--P--P--P--P- -P--P--P--P--P- -P--P--P--P--P-

QUERY NUC   ACAGGAGCAGATGAT ACAGTATTAGAAGAT ATAAATTTACCAGGA AGATGGACACCAAAA ATGATAGGGGGAATT
QUERY PRO    :  :  :  :  :   :  :  :  :  D   I  N  :  :  :   :  :  T  :  :   :  :  :  :  :
HXB2r PRO    T  G  A  D  D   T  V  L  E  E   M  S  L  P  G   R  W  K  P  K   M  I  G  G  I   50
  MUTANTS                                *   *  *                  *
            -P--P--P--P--P- -P--P--P--P--P- -P--P--P--P--P- -P--P--P--P--P- -P--P--P--P--P-
```

the PAUP* based trees, similar to our tree building part of the search interface.

History and context This tool was originally developed by Carla Kuiken and Charles Calef. A completely new version of tree making is now part of our search interface (to be described in the 2006 compendium). This version is based on PAUP* and was made by Thomas Leitner, Charles Calef and Werner Abfalterer. We are grateful to Jim Wilgenbush who has given us permission to use PAUP* [Swofford, 2002] to infer our trees.

FindModel

Purpose FindModel analyzes your alignment to see which evolutionary model best describes the input sequences. This model can then be used to generate a better phylogenetic tree.

Background FindModel uses the program Weighbor [Bruno *et al.*, 2000] to generate the guide tree, based on Jukes-Cantor distances. Weighbor is used because it is much faster than maximum likelihood, but less biased and more robust than neighbor joining. Ziheng Yang's PAML [Yang, 1997] is used to calculate the likelihood. The AIC score, a version of the likelihood score that is weighted to compensate for the differences in degrees of freedom (or the number of parameters included) for each model, is calculated using the method described in Posada & Crandall [1998]. The standard log likelihood score is also reported, but the decision of the best fitting model is made based on the AIC. It is intuitively clear that a model that is more 'customizable' to the data, i.e., has more parameters, will usually produce a better fit. This would always result in the most complicated model being selected, even when simpler models would do almost as well. The AIC score compensates for this effect by weighting the likelihood

Table I-C-1: Partial view of the list of models and their AIC and likelihood scores.

Model name	AIC	LnL
JC : Jukes-Cantor (model 1)	3563.252246	-1781.626123
JC+G : Jukes-Cantor plus Gamma (model 3)	3504.693372	-1751.346686
F81 : Felsenstein 1981 (model 5)	3564.440496	-1779.220248
F81+G : Felsenstein 1981 plus Gamma (model 7)	3502.851558	-1747.425779
K80 : Kimura 2-parameter (model 9)	3499.844658	-1748.922329
K80+G : Kimura 2-parameter plus Gamma (model 11)	3423.016278	-1709.508139
HKY : Hasegawa-Kishino-Yano (model 13)	3499.375768	-1745.687884
HKY+G : Hasegawa-Kishino-Yano plus Gamma (model 15)	3412.457576	-1701.228788
TrN : Tamura-Nei (model 21)	3494.642212	-1742.321106
TrN+G : Tamura-Nei plus Gamma (model 23)	3413.187698	-1700.593849
GTR : General Time Reversible (model 53)	3487.768892	-1735.884446
GTR+G : General Time Reversible plus Gamma (model 55)	3411.658894	-1696.829447

score by the number of parameters for each model. FindModel, unlike Modeltest, does not allow invariant sites, because this feature is not implemented in PAML. This was a principled choice by PAML's author, because estimates of the fraction of invariant sites tend to be very sensitive to the number of taxa.

Finding the best evolutionary model is a computationally intensive procedure, both in its original implementation as the Modeltest PAUP* script and in our FindModel implementation. To reduce the computational burden on our servers, we have limited the default runs to a reduced set of models, and excluded those that do not have an obvious biological interpretation. The full set of models can be run, but has to be explicitly specified by checking the checkbox below the input section.

Output The output of FindModel consists of a list of models the program has tested, and their AIC and likelihood scores. The model with the smallest AIC score is shown as 'AIC-selected model'. This model is usually the best, and limited simulations have shown that FindModel shows very little tendency to over-fitting [Tao, in preparation]. In addition to the selected model, the FindModel output also shows a matrix (Figure I-C-16) that indicates which parameters are being estimated from the data in each model. By clicking on the model name, the matrix shows every parameter that is estimated separately in a different color. In Figure I-C-16, the Jukes-Cantor model shows that all transitions and transversions have the same color (orange) and therefore are represented by one parameter. The nucleotide frequencies are all shown as f_N, so they are also all estimated to be the same.

History and context FindModel was developed as a web implementation of the Modeltest script written by David Posada and Keith Crandall [Posada & Crandall, 1998], modified by Bill Bruno with input from Carla Kuiken.

PeptGen

Purpose PeptGen enables design of overlapping peptide sets from single proteins or alignments, with output that allows either visualizing the peptides and differences between them, or produces a list for ordering the peptides.

Background The algorithm to generate the peptides is complex and can be modified by the user in many different ways. For example, "forbidden" amino acids can be excluded from the ends of the peptide because of their inimical effect on binding to the HLA molecule. Peptides beginning with Q (glutamine) are thought to be unreliable, so Q has been made the default for N-term forbidden amino acids. The offset between one peptide and the next, i.e., the "width" of each stairstep, is determined by the "Overlap peptide by" parameter.

Output Figure I-C-17 shows the output for a protein fragment, where 15-mers overlapping by 11 were requested, but the amino acids G, P, E, D, Q, N, T, S and C were all disallowed at the C-terminal position.

When aligned sequences are provided as input, PeptGen creates an output that highlights the difference. This would be convenient for a situation where one wanted to design peptides to compare different subtypes, for example. To create the following peptides sets, no C-terminal amino acids were disallowed (so all peptides are length 15 except the last one, and two aligned sequences were given as input.

To generate a list of peptides ready to order, the set of peptides in Figure I-C-17 can be written out as a list with a unique ID assigned to each peptide, the peptide number, the sequence number, and a list of sequences

Figure I-C-16: Matrix showing free parameter estimation in the GTR (left) and Jukes-Cantor (right) evolutionary models.

General Time Reversible + γ

	T	C	A	G
T	f_T	a	b	c
C	a	f_C	d	e
A	b	d	f_A	f
G	c	e	f	f_G

Jukes-Cantor

	T	C	A	G
T	f_N	a	b	c
C	a	f_N	d	e
A	b	d	f_N	f
G	c	e	f	f_N

Figure I-C-17: Peptgen output for a single sequence. Disallowed C-terminal peptides are bold and underlined; not all peptides are 15 long to accommodate this. Their length is indicated in parentheses after the peptide.

```
MENRWQVMIVWQVDRMRIRTWKSLVKHHMYVSGKARGWFYRHHYESPHPRISSEVHIPL
MENRWQVMIVWQVDR (15) [-0.49]
    WQVMIVWQVDRMRIR (15) [-0.03]
        IVWQVDRMRIRTWK (14) [-0.54]
          WQVDRMRIRTWKSLV (15) [-0.61]
              RMRIRTWKSLVKHHM (15) [-0.92]
                  RTWKSLVKHHMYV (13) [-0.64]
```

that would contain the identical peptides from within the input alignment. One can request all peptides be listed, including duplicates between the two protein sequences, or that identical peptides be excluded so they don't need to be made twice.

History and context This site was designed by Charles Calef and Bette Korber [Calef *et al.*, 2001] in response to multiple requests by immunologists for help in generating peptides for epitope mapping. The request to facilitate peptide generation from alignments came from Richard Koup (NIH). Philip Goulder (Oxford) requested the ability to exclude certain amino acids from C-term positions, and Otto Yang at (UCLA) to forbid N-terminal amino acids. Andrew Bradbury (LANL) suggested calculating hydropathy for the resulting peptides, for antibody studies.

ELF—Epitope Location Finder

Purpose ELF scans a submitted protein sequence for known epitopes in our immunology database whose HLA agrees with the submitted HLAs.

Background ELF was written to identify potential epitopes within larger immunologically reactive target peptides [Calef *et al.*, 2002b]. Based on a peptide and a selection of HLA alleles, any known epitopes in that peptide are retrieved from in the immunology database, with links to the database entries and references. Those epitopes whose HLA presenting molecule agrees with the submitted HLAs are flagged. Anchor residues of potential epitopes that agree with the binding motifs of the submitted HLAs are indicated. Maps can be prepared that highlight every known epitope of the submitted HLA alleles across the HIV proteome. ELF can be used in conjunction with the Hepitope tool, which looks for enriched HLAs among people who make a reaction to the peptide in a population survey.

Output The output from ELF is very rich, so we have tried to make the page as uncluttered as possible. The first graphic on the output page is a map marking the location of the input peptide in the genome (see Figure I-C-5 in the sequence locator tool). Various links go to pages that contain:

- a list of the HLAs associated with your submitted HLA. As anchor motif information is spotty and this tool is exploratory, all related serotypes and genotypes will be incorporated in the search. For example, if the user were to enter either A2 or A*0202, all A2-related serotypes and genotypes with known anchor motifs would be examined.
- a list of all anchor motifs used in the search [Marsh *et al.*, 2000; Rammensee *et al.*, 1997, 1999]. Anchor motifs embedded in epitopes 8–11 amino acids long are considered, but larger epitopes would be missed.
- potential "epitopes" ordered by HLAs. This link takes you to a listing of possible epitopes in your peptide based on the presence of appropriately spaced anchor motifs.

A list of known CTL epitopes in the peptide (regardless of HLA type) can be useful for searching for unanticipated cross-presentation. The epitopes are linked to the corresponding records in the immunology database; these records provide information regarding escape mutations, clade specific reactions, immunodominance, etc., among epitope variants. Substitutions in the epitope relative to the query peptide are highlighted with red, and epitopes presented by the requested HLAs are marked with a green arrow.

History and context This tool was first developed by Charles Calef, Rama Thakalapally, James Szinger, and Bette Korber [Calef *et al.*, 2002b] to attempt to define epitopes within reactive peptides to support experimental epitope mapping conducted at the University of Alabama by Richard Kaslow and Paul Goepfert [Bansal *et al.*, 2003]. Charles Calef has implemented improvements over time, incorporating new suggestions made by Carla Kuiken, Karina Yusim and Bette Korber, and Christian Brander at Harvard/MGH.

Motif Scan

Purpose Motif Scan is an HLA binding motif scanner that finds HLA anchor residue motifs within protein sequences for specified HLA serotypes, genotypes, or supertypes.

Background Two major motif libraries were used [Marsh *et al.*, 2000; Rammensee *et al.*, 1999] and the literature was surveyed for additional anchor motifs. The supermotifs incorporate anchor residues that are recognized by multiple alleles within the supertype [Sette & Sidney, 1999]. We store only anchor motifs in our libraries; to incorporate auxiliary amino acids you must input your own custom motif. The motif dictionaries we use are listed on the web, as is an abbreviated list of associations between HLA genotypes and serotypes.

Input The input for Motif Scan is obtained in two steps. The first step determines what anchor motifs are of interest. If you are interested in a functional motif or auxiliary and anchor motifs, you can input that instead, using the syntax `x[LM]xxx[K]xx[V]` where `x` allows any amino acid and determines the spacing, and locations where more than one amino acid is allowed are indicated by brackets: `L` or `M` in the second position, `K` in the sixth. The second step selects the sequences to be scanned. Predefined HIV protein sequences can be used, or you can upload your own sequences. Sequences are stripped of gaps before processing.

Output All motifs with identical search patterns are grouped together. C-terminal anchor amino acids are shown in magenta and anchor amino acids in the other positions are shown in cyan. If a given amino acid is matched by more than one motif, then it is highlighted as a C-terminal anchor amino acid. All anchor amino acids are shown in uppercase and non-anchors are lowercase. Following the sequences is a list of potential epitopes showing their positions in the input sequences. You can also view and download the resulting sequences in fasta format where the anchor amino acids are presented in uppercase and all the remaining ones in lowercase. The potential epitopes can be also downloaded in CSV (comma-separated value) format, which can be read into a spreadsheet.

History and context This tool was first developed by Warren Kibbe, Rama Thakallapally and Bette Korber [Thakallapally *et al.*, 2001]. Since the initial publication, the tool and the motif libraries were much improved by Karina Yusim and James Szinger [Yusim *et al.*, 2004].

Hepitope

Purpose Hepitope tests for HLA alleles that are enriched in individuals that react with a set of peptides.

Background This tool can be used in the context of a population study where HLAs and Elispot reactivity are available for a set of patients. To find HLA types that may be more frequent with certain reactivity patterns, a Fisher's exact test is used to look for enriched HLAs with a two-by-two contingency table tally for each subject tallying whether each HLA is present or absent, and whether they reacted to the peptide or not. This can be used in conjunction with our ELF program, which will scan a peptide for known epitopes in the database and for anchor motifs for HLAs that are found to be enriched, thus helping to identify epitopes within a larger peptide fragment (Hopeful Epitopes, or Hepitopes). This tool is not HIV or HCV specific, except when it is used in conjunction with ELF. The output is organized by peptide, and these can be returned either in the order entered or in alphabetical order. You can have all of the data returned, including summaries of every person's HLA that did not react with the peptide in question, but the default is to display only positive reactions.

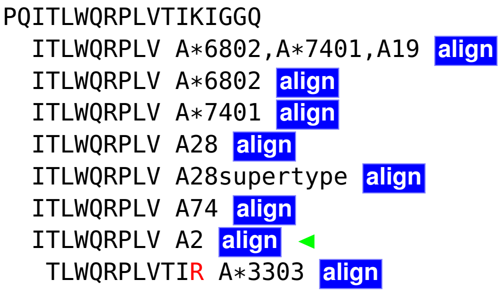
Input This tool requires two inputs. The first input (Figure I-C-22) is a text format table of patients and their HLAs (note that many patients are needed to get statistical significance). The allele can be written as a serotype (A2) or a genotype (A*0201), but if both are used then they will be treated separately in the analysis. If an HLA type is unknown, it should be written as a single character. For example, if the C alleles had not yet been determined in Patient 1, then the HLA could be written as:

```
Patient1  A*0201 A*0201 B*5703 B*1701 C C
```

The second input is a list of reactive peptides, and the patients that reacted:

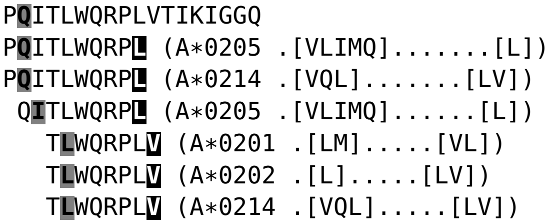
```
Gag1 MGARASVLSGGELDRWEK Patient1
Gag2 SGGELDRWEKIRLRPGGK Patient2 Patient3
Gag3 EKIRLRPGGKKKYKLKHI Patient4
```

Figure I-C-18: Known epitopes that are found within PQITLWQRPLVTIKIGGQ, the query peptide. Clicking on the aligned peptides links you to all of the database entries for the combination of peptide and the HLA presenting molecule. Clicking on the align button takes you to an alignment of this epitope extracted from the main database alignment. The green arrow denotes an epitope from the defined HLA set.



```
PQITLWQRPLVTIKIGGQ
  ITLWQRPLV A*6802,A*7401,A19 align
  ITLWQRPLV A*6802 align
  ITLWQRPLV A*7401 align
  ITLWQRPLV A28 align
  ITLWQRPLV A28supertype align
  ITLWQRPLV A74 align
  ITLWQRPLV A2 align ◄
   TLWQRPLVTIR A*3303 align
```

Figure I-C-19: Highlighting anchor motifs in the epitope. Identification of potential epitopes within the reactive peptide based on the anchor residues described for any HLAs related to the HLA of interest. C terminal anchors are marked in magenta, second position anchors in blue. No B44-related motifs were found.



```
PQITLWQRPLVTIKIGGQ
PQITLWQRPL (A*0205 .[VLIMQ].......[L])
PQITLWQRPL (A*0214 .[VQL].......[LV])
 QITLWQRPL (A*0205 .[VLIMQ]......[L])
   TLWQRPLV (A*0201 .[LM].....[VL])
   TLWQRPLV (A*0202 .[L].....[LV])
   TLWQRPLV (A*0214 .[VQL].....[LV])
```

Output Four columns of data that form the 2 by 2 contingency table used to compute the p-value. The output is arranged by peptide, and if the ELF integration is requested, anchor motifs and known epitopes are also summarized for each epitope.

a The number of individuals that carry the HLA allele and react with the peptide.

b The number of individuals that carry the HLA allele and do not react with the peptide.

c The number of individuals that do not carry the HLA allele and react with the peptide.

d The number of individuals that do not carry the HLA allele and do not react with the peptide.

The one-sided Fisher's exact test p-value is calculated to see if category *a* (number of individuals that both carry the HLA allele and react with the peptide) is higher than one would expect by chance alone. These are uncorrected p-values, and obviously multiple tests are being done, so these values should be evaluated with appropriate corrections or else the enriched HLAs for a give peptide should be considered as a hypothesis forming guideline for a suggestion of a likely HLA presenting molecule.

History and context This web-based tool and strategy for enabling epitope prediction analysis was developed by James Szinger and Bette Korber for a large epitope mapping and HLA typing project run by Christian Brander and Bruce Walker (Harvard University and Massachusetts General Hospital) [Kiepiela *et al.*, 2004].

I-C-4 Acknowledgements

We thank many of our past graduate research assistants, postdoctoral fellows, and technical support staff who contributed directly to these tools and databases, including Satish Pillai, Adam Siepel, Ashish Agrawal, Russell Richardson, Kristina Kommander, Dorothy Lang, John Mokili, Shaun Geer, Una Smith, Kersti Rock, Sampath Billikanti, Rama Thakallapally and Patrick Rose. James Theiler has often contributed valuable bits of code to this project. We particularly thank our colleagues who have made suggestions to improve these tools and the database in general.

I-C-5 References

S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, & D. J. Lipman, 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res* **25**(17):3389–3402. On p. 40

A. Bansal, S. Sabbaj, B. H. Edwards, G. D. Ritter, C. Perkins, J. Tang, J. J. Szinger, H. Weiss, P. A. Goepfert, B. Korber, C. M. Wilson, R. A. Kaslow, & M. J. Mulligan, 2003. T cell responses in HIV type 1-infected adolescent minorities share similar epitope specificities with whites despite significant differences in HLA class I alleles. *AIDS Res Hum Retroviruses* **19**(11):1017–1026. On p. 52

BioEdit. Computer program. http://www.mbio.ncsu.edu/BioEdit/bioedit.html. On p. 38

W. J. Bruno, N. D. Socci, & A. L. Halpern, 2000. Weighted neighbor joining: A likelihood-based approach to distance-based phylogeny reconstruction. *Mol Biol Evol* **17**(1):189–197. On p. 49

Figure I-C-20: Map of epitopes within the protease protein, in which this peptide was embedded. All known epitopes are indicated, with A2 and B44 known epitopes highlighted. More information regarding these epitopes could be obtained through the search page.

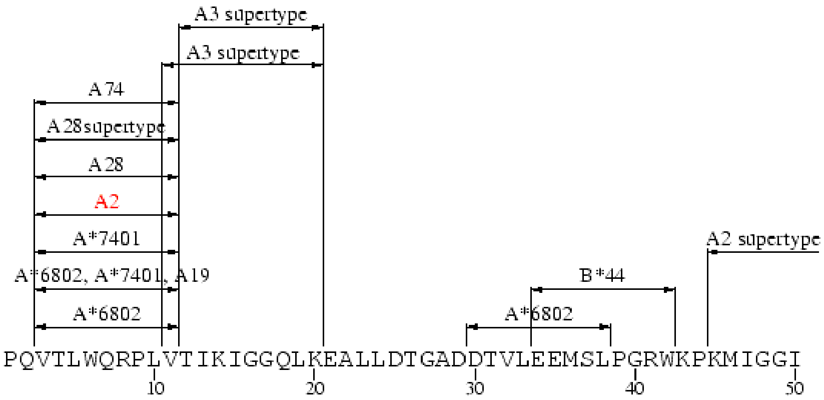


Figure I-C-21: Results of searching HXB2 Tat for HLA*0205 motifs that could give rise to epitopes of length 8, 9 or 10. The motifs that were scanned are listed first, in this case with spacing to give rise to 8, 9 or 10 amino acid long potential epitopes. A*0205 anchor residues are highlighted and capitalized in the Tat sequence, and the same possible epitopes are listed with their position number and spacing following the sequences.

```
>HXB2 x-[LQV]-x-x-x-x-x-[L] x-[LQV]-x-x-x-x-x-x-[L] x-[LQV]-x-x-x-x-x-x-x-[L] A*0205
mepvdprlep wkhpgsqpkt actncyckkc cfhcQVcfit kaLgisygrk   50
krrqrrrahq nsQthqasLs kqptsqprgd ptgpkekkkv eretetdpfd  100

Protein Position Sequence   Anchors
HXB2    62-69    SQTHQASL   .Q.....L
HXB2    35-43    QVCFITKAL  .V......L
HXB2    34-43    CQVCFITKAL .Q.......L
```

Figure I-C-22: Hepitope patient HLA sample input.

```
Patient1  A*0201 A*0201 B*5703 B*1701 Cw*0701 Cw*0705
Patient2  A*0201 A*0701 B*1202 B*0801 Cw*0701 Cw*0401
Patient3  A*1101 A*2403 B*0801 B*5801 Cw*0701 Cw*1501
Patient4  A*3002 A*3002 B*5802 B*5802 Cw*0602 Cw*0602
```

Figure I-C-23: Example of Hepitope output using a representative peptide. All HLAs found in reactive patients that recognize the peptide are listed. The full HLA type of the patients that react with the peptide is also listed. If the integration with ELF is selected, under each peptide will be a summary of known epitopes, links to references, and potential anchor motifs for the HLAs of interest within the epitope.

<table>
<tr><th>Peptide</th><th>Sequence</th><th>HLA Type</th><th>a</th><th>b</th><th>c</th><th>d</th><th>P</th></tr>
<tr><td></td><td></td><td>B*1701</td><td>1</td><td>0</td><td>0</td><td>3</td><td>0.25000000</td></tr>
<tr><td></td><td></td><td>B*5703</td><td>1</td><td>0</td><td>0</td><td>3</td><td>0.25000000</td></tr>
<tr><td></td><td></td><td>Cw*0705</td><td>1</td><td>0</td><td>0</td><td>3</td><td>0.25000000</td></tr>
<tr><td>Gag1</td><td>MGARASVLSGGELDRWEK</td><td>A*0201</td><td>1</td><td>0</td><td>1</td><td>2</td><td>0.50000000</td></tr>
<tr><td></td><td></td><td>Cw*0701</td><td>1</td><td>0</td><td>2</td><td>1</td><td>0.75000000</td></tr>
<tr><td></td><td></td><td colspan="6"><table><tr><th>Patient</th><th>HLA</th></tr><tr><td>Patient1</td><td>A*0201 A*0201 B*1701 B*5703 Cw*0701 Cw*0705</td></tr></table></td></tr>
</table>

References

C. Calef, J. Mokili, D. H. O'Connor, D. I. Watkins, & B. Korber, 2002a. Numbering positions in SIV relative to SIVMM239. In C. Kuiken, B. Foley, B. Hahn, P. Marx, F. McCutchan, J. Mellors, S. Wilonsky, & B. Korber, eds., *HIV Sequence Compendium 2001*, pp. 171–181. Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, Los Alamos, New Mexico. LA-UR 02-2877. On p. 40

C. Calef, R. Thakallapally, R. Kaslow, M. Mulligan, & B. Korber, 2002b. ELF: An analysis tool for HIV-1 peptides and HLA types. In B. Korber, C. Brander, B. Haynes, R. Koup, C. Kuiken, J. P. Moore, B. D. Walker, & D. I. Watkins, eds., *HIV Molecular Immunology 2001*, pp. I-21–I-25. Los Alamos National Laboratory, Theoretical Biology & Biophysics, Los Alamos, New Mexico. LA-UR 02-4663. On p. 51, 52

C. Calef, R. Thakallapally, D. Lang, C. Brander, P. Goulder, & B. Korber, 2001. PeptGen: Designing peptides for immunological studies and application to HIV consensus sequences. In B. Korber, C. Brander, B. Haynes, R. Koup, C. Kuiken, J. P. Moore, B. D. Walker, & D. I. Watkins, eds., *HIV Molecular Immunology 2000*, pp. I-63–I-100. Los Alamos National Laboratoy, Theoretical Biology & Biophysics, Los Alamos, New Mexico. LA-UR 01-2430. On p. 51

S. Clark, J. W. Mellors, & C. E. Calef, 2001. Hiv drug resistance database. Web site at `http://resdb.lanl.gov/Resist_DB/default.htm`. On p. 48

C. A. Derdeyn, J. M. Decker, F. Bibollet-Ruche, J. L. Mokili, M. Muldoon, S. A. Denham, M. L. Heil, F. Kasolo, R. Musonda, B. H. Hahn, G. M. Shaw, B. T. Korber, S. Allen, & E. Hunter, 2004. Envelope-constrained neutralization-sensitive HIV-1 after heterosexual transmission. *Science* **303**(5666):2019–2022. On p. 46

S. Ganeshan, R. E. Dickover, B. T. M. Korber, Y. J. Bryson, & S. M. Wolinsky, 1997. Human immunodeficiency virus type 1 genetic evolution in children with different rates of development of disease. *J Virol* **71**(1):663–677. On p. 48

B. Gaschen, C. Kuiken, B. Korber, & B. Foley, 2001. Retrieval and on-the-fly alignment of sequence fragments from the HIV database. *Bioinformatics* **17**(5):415–418. On p. 39

B. Gaschen, J. Taylor, K. Yusim, B. Foley, F. Gao, D. Lang, V. Novitsky, B. Haynes, B. H. Hahn, T. Bhattacharya, & B. Korber, 2002. Diversity considerations in HIV-1 vaccine selection. *Science* **296**(5577):2354–2360. On p. 37

D. Gilbert. Readseq. Computer program. `http://bioweb.pasteur.fr/docs/seqio/fmtseq_doc.html`. On p. 35

J. C. Gower, 1966. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* **53**:325–338. On p. 44

D. G. Higgins, 1992. Sequence ordinations: A multivariate analysis approach to analysing large sequence data sets. *Comput Appl Biosci* **8**(1):15–22. On p. 45

P. Kiepiela, A. J. Leslie, I. Honeyborne, D. Ramduth, C. Thobakgale, S. Chetty, P. Rathnavalu, C. Moore, K. J. Pfafferott, L. Hilton, P. Zimbwa, S. Moore, T. Allen, C. Brander, M. M. Addo, M. Altfeld, I. James, S. Mallal, M. Bunce, L. D. Barber, J. Szinger, C. Day, P. Klenerman, J. Mullins, B. Korber, H. M. Coovadia, B. D. Walker, & P. J. R. Goulder, 2004. Dominant influence of HLA-B in mediating the potential co-evolution of HIV and HLA. *Nature* **432**(7018):769–775. On p. 53

B. Korber, 2001. HIV signature and sequence variation analysis. In A. G. Rodrigo & G. H. Learn, eds., *Computational Analysis of HIV Molecular Sequences*, pp. 55–72. Kluwer Academic Publishers. On p. 46

B. Korber, B. Gaschen, K. Yusim, R. Thakallapally, C. Kesmir, & V. Detours, 2001. Evolutionary and immunological implications of contemporary HIV-1 variation. *Br Med Bull* **58**:19–42. On p. 37

B. Korber & G. Myers, 1992. Signature pattern analysis: A method for assessing viral sequence relatedness. *AIDS Res Hum Retroviruses* **8**(9):1549–1560. On p. 44

B. T. Korber, B. T. Foley, C. L. Kuiken, S. K. Pillai, & J. G. Sodroski, 1999. Numbering positions in HIV relative to HXB2CG. In B. T. Korber, C. L. Kuiken, B. T. Foley, B. Hahn, F. McCutchan, J. W. Mellors, & J. Sodroski, eds., *Human Retroviruses and AIDS 1998*, pp. III-102–III-111. Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, Los Alamos, New Mexico. LA-UR 99-1704. On p. 40

B. T. Korber, K. J. Kunstman, B. K. Patterson, M. Furtado, M. M. McEvilly, R. Levy, & S. M. Wolinsky, 1994. Genetic differences between blood- and brain-derived viral sequences from human immunodeficiency virus type 1-infected patients: Evidence of conserved elements in the V3 region of the Envelope protein of brain-derived sequences. *J Virol* **68**(11):7467–7481. On p. 46

C. L. Kuiken, K. Nieselt-Struwe, W. G. F., & J. Goudsmit, 1994. Quasispecies behavior of human immunodeficiency virus type 1: Sample analysis of sequence data. In K. W. Adolph, ed., *Methods in Molecular Genetics*, vol. 4. Academic Press. On p. 45

C. L. Kuiken, G. Zwart, E. Baan, R. A. Coutinho, J. A. R. van den Hoek, & J. Goudsmit, 1993. Increasing antigenic and genetic diversity of the V3 variable domain of the human immunodeficiency virus envelope protein in the course of the AIDS epidemic. *Proc Natl Acad Sci USA* **90**(19):9061–9065. On p. 45

D. Lecossier, F. Bouchonnet, F. Clavel, & A. J. Hance, 2003. Hypermutation of HIV-1 DNA in the absence of the Vif protein. *Science* **300**(5622):1112. On p. 45

B. Mangeat, P. Turelli, G. Caron, M. Friedli, L. Perrin, & D. Trono, 2003. Broad antiretroviral defence by human APOBEC3G through lethal editing of nascent reverse transcripts. *Nature* **424**(6944):99–103. On p. 45

S. G. E. Marsh, P. Parham, & L. D. Barber, 2000. *The HLA FactsBook*. Academic Press, San Diego. On p. 51, 52

E. W. Myers & W. Miller, 1988. Optimal alignments in linear space. *Comput Appl Biosci* **4**(1):11–17. On p. 38

M. Nei & T. Gojobori, 1986. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Mol Biol Evol* **3**(5):418–426. On p. 46

T. Ota & M. Nei, 1994. Variance and covariances of the numbers of synonymous and nonsynonymous substitutions per site. *Mol Biol Evol* **11**(4):613–619. On p. 48

C. Y. Ou, C. A. Ciesielski, G. Myers, C. I. Bandea, C. C. Luo, B. T. Korber, J. I. Mullins, G. Schochetman, R. L. Berkelman, & A. N. Economou, 1992. Molecular epidemiology of HIV transmission in a dental practice. *Science* **256**(5060):1165–1171. On p. 44

D. Posada & K. A. Crandall, 1998. MODELTEST: Testing the model of DNA substitution. *Bioinformatics* **14**(9):817–818. On p. 49, 50

B. J. Potts, K. G. Field, Y. Wu, M. Posner, L. Cavacini, & M. White-Scharf, 1993. Synergistic inhibition of HIV-1 by CD4 binding domain reagents and V3-directed monoclonal antibodies. *Virology* **197**(1):415–419. On p. 45

H.-G. Rammensee, J. Bachmann, N. P. N. Emmerich, O. A. Bachor, & S. Stevanović, 1999. SYFPEITHI: Database for MHC ligands and peptide motifs. *Immunogenetics* **50**(3-4):213–219. On p. 51, 52

H. G. Rammensee, J. Bachmann, & S. Stevanovic, 1997. *MHC Ligands and Peptide Motifs*. Chapman & Hall. On p. 51

D. L. Robertson, J. P. Anderson, J. A. Bradac, J. K. Carr, B. Foley, R. K. Funkhouser, F. Gao, B. H. Hahn, M. L. Kalish, C. Kuiken, G. H. Learn, T. Leitner, F. McCutchan, S. Osmanov, M. Peeters, D. Pieniazek, M. Salminen, P. M. Sharp, S. Wolinsky, & B. Korber, 2000a. HIV-1 nomenclature proposal. *Science* **288**(5463):55–56. On p. 40

D. L. Robertson, J. P. Anderson, J. A. Bradac, J. K. Carr, B. Foley, R. K. Funkhouser, F. Gao, B. H. Hahn, M. L. Kalish, C. Kuiken, G. H. Learn, T. Leitner, F. McCutchan, S. Osmanov, M. Peeters, D. Pieniazek, M. Salminen, P. M. Sharp, S. Wolinsky, & B. Korber, 2000b. HIV-1 nomenclature proposal. In C. Kuiken, B. Foley, B. Hahn, P. Marx, F. McCutchen, J. W. Mellors, J. Mullins, S. Wolinsky, & B. Korber, eds., *Human Retroviruses and AIDS 1999*, pp. 492–505. Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, Los Alamos, New Mexico. LA-UR 00-. On p. 40

P. P. Rose & B. T. Korber, 2000. Detecting hypermutations in viral sequences with an emphasis on G –> A hypermutation. *Bioinformatics* **16**(4):400–401. On p. 45

A. Sette & J. Sidney, 1999. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. *Immunogenetics* **50**(3-4):201–212. On p. 52

C. E. Shannon, 1948. A mathematical theory of communication. *Bell System Tech J* **27**:623–656. Reprinted with corrections. On p. 46

A. C. Siepel, A. L. Halpern, C. Macken, & B. T. Korber, 1995. A computer program designed to screen rapidly for HIV type 1 intersubtype recombinant sequences. *AIDS Res Hum Retroviruses* **11**(11):1413–1416. On p. 41, 42

D. L. Swofford, 2002. PAUP* phylogenetic analysis using parsimony (*and other methods). Computer program. Version 4.0b10. On p. 49

R. Thakallapally, W. Kibbe, D. Lang, & B. Korber, 2001. Motifscan: A web-based tool to find HLA anchor residues in proteins or peptides. In B. Korber, C. Brander, B. Haynes, R. Koup, C. Kuiken, J. P. Moore, B. D. Walker, & D. I. Watkins, eds., *HIV Molecular Immunology 2000*, pp. I-101–I-102. Los Alamos National Laboratoy, Theoretical Biology & Biophysics, Los Alamos, New Mexico. LA-UR 01-2430. On p. 52

Z. Yang, 1997. PAML: A program package for phylogenetic analysis by maximum likelihood. *Comput Appl Biosci* **13**(5):555–556. On p. 49

K. Yusim, C. Kesmir, B. Gaschen, M. M. Addo, M. Altfeld, S. Brunak, A. Chigaev, V. Detours, & B. T. Korber, 2002. Clustering patterns of cytotoxic T-lymphocyte epitopes in human immunodeficiency virus type 1 (HIV-1) proteins reveal imprints of immune evasion on HIV-1 global variation. *J Virol* **76**(17):8757–8768. On p. 46

K. Yusim, J. J. Szinger, I. Honeyborne, C. Calef, P. J. R. Goulder, & B. T. M. Korber, 2004. Enhanced motif scan: A tool to scan for HLA anchor residues in proteins. In B. T. M. Korber, C. Brander, B. F. Haynes, R. Koup, J. P. Moore, B. D. Walker, & D. I. Watkins, eds., *HIV Immunology and HIV/SIV Vaccine Databases 2003*, pp. 25–36. Los Alamos National Laboratory, Theoretical Biology & Biophysics, Los Alamos, New Mexico. LA-UR 04-8162. On p. 52

H. Zhang, B. Yang, R. J. Pomerantz, C. Zhang, S. C. Arunachalam, & L. Gao, 2003. The cytidine deaminase CEM15 induces hypermutation in newly synthesized HIV-1 DNA. *Nature* **424**(6944):94–98. On p. 45

M. Zhang, B. Gaschen, W. Blay, B. Foley, N. Haigwood, C. Kuiken, & B. Korber, 2004. Tracking global patterns of N-linked glycosylation site variation in highly variable viral glycoproteins: HIV, SIV, and HCV envelopes and influenza hemagglutinin. *Glycobiology* **14**(12):1229–1246. On p. 45