J Mol Evol (1994) 38:310–316

Journal of Molecular Evolution



# Genomic Alignment

**Jotun Hein, Jens Støvlbæk**

Institute of Biological Sciences, Aarhus University, DK-8000 Aarhus, Denmark

Received: 5 April 1993 / Revised: 1 September 1993

**Abstract.** As sequencing techniques become increasingly efficient, the average length of a sequence is bound to grow. Traditional sequence-comparison algorithms can either compare DNA or protein, but not a mixture, which is actually a common situation. Most obtained DNA sequences contain coding regions, and it is more reliable to compare the coding regions as protein than just as DNA.

A heuristic algorithm is presented that can compare DNA with both coding and noncoding regions, but that also can compare multiple reading frames and determine which exons are homologous.

A program, GenAl (*Gen*omic *Al*ignment), was developed that implements the algorithm. Its use is demonstrated on two retroviruses.

**Key words:** Sequencing techniques — Genomic alignment — Heuristic algorithm

## Introduction

When genomic DNA sequences longer than 1 kb are compared, they will typically contain both coding and noncoding regions and cannot be analyzed by the traditional dynamical programming algorithm (Sankoff 1972). As protein evolves slower than its coding DNA it will be more reliable to align the protein than the underlying DNA, and an algorithm that compares genomic DNA should incorporate the information from the protein. Presently this problem is solved by separating the sequences into coding and noncoding parts, then analyzing them separately, and finally patching the resulting alignments into a global alignment. This is laborious and cannot be done if the DNA has overlapping reading frames. The objective of this paper is to present an algorithm that solves these problems.

To understand the problem arising, see Fig. 1, in which the basic events that can occur are illustrated. The characteristic of genomic DNA is that it has coding regions embedded in the DNA. These can overlap in the case of many viruses and they can be interrupted by introns in eukaryotic genes. It should be noted that all events happen at the DNA level as proteins don't replicate. The basic events are:

1. Substitutions. In coding regions they can have the additional consequence of changing the coded protein.
2. Insertion-deletions, abbreviated indel, in coding and noncoding regions. In coding regions, they will normally be assumed to have lengths a multiple of three. In pairwise sequence comparisons insertions and deletions are treated symmetrically, as they cannot be distinguished due to the lack of time direction in the inferred history of the sequences.
3. Movement of stop and start codons. This is actually substitutions in the DNA, but it will appear as external (before or after the protein) indels in the protein.

The result of a genomic alignment algorithm is a global alignment of all the DNA, simultaneous alignment of the protein, and matching of overlapping reading frames and the intron-exon structure correctly.

*Correspondence to:* J. Hein

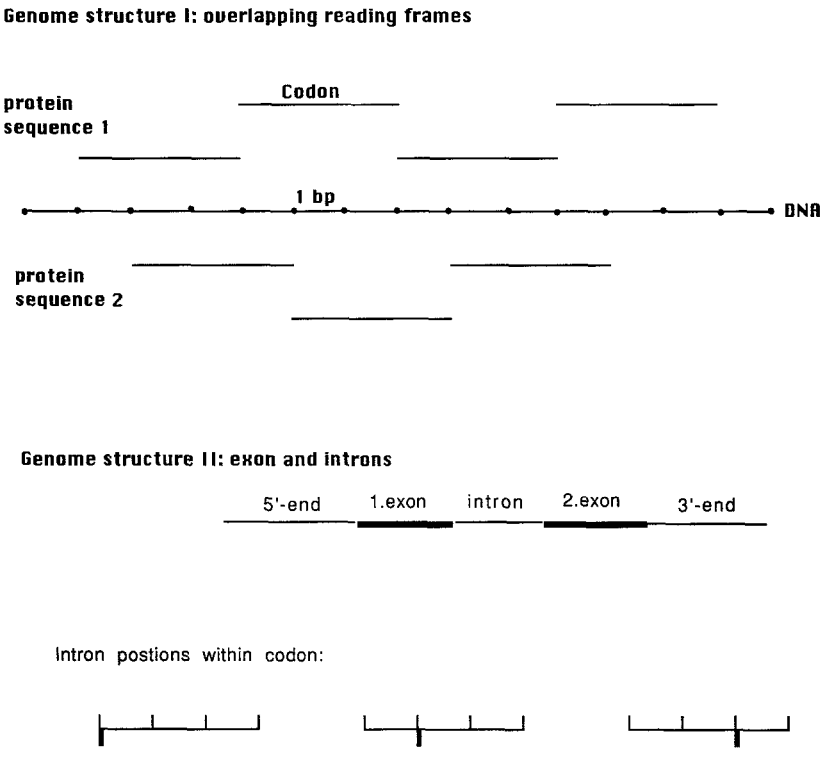


**Fig. 1.** Viruses can have overlapping reading frames, so the same stretch of DNA codes for two or more proteins. If the reading frames have the same orientation (are read in the same direction), then they must be out of phase and their codon boundaries will not coincide. It is then impossible to introduce insertion-deletions that obey codon boundaries for both proteins.

In eukaryotes the reading frames can be interrupted by noncoding sequences—introns. These introns can be inserted at any position within a codon.

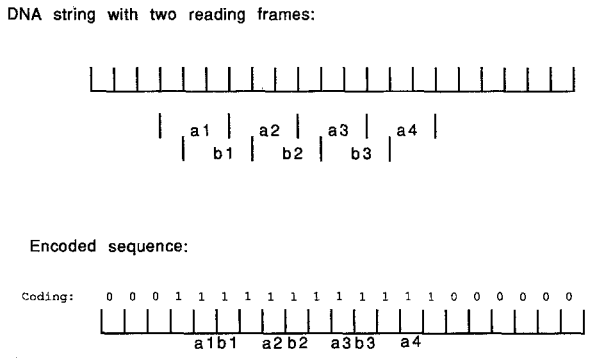


**Fig. 2.** A DNA string with two reading frames is shown. The compressed sequence has the amino acids coded by the middle nucleotide in the codon only. If this middle nucleotide (and amino acid) is matched to another middle nucleotide, then the two amino acids are considered matched. It is still registered for each nucleotide if it belonged to a codon before the sequence was compressed.

This paper is organized as follows: First a heuristic but fast and flexible algorithm is presented that can compare DNA with any coding structure in it, including overlapping reading frames and introns. Then this method is illustrated by applying it to two retroviruses. Lastly, problems and further extensions are discussed.

## The Heuristic Genomic Alignment Algorithm

The true representation of DNA with amino acids being coded by three consecutive nucleotides is changed so an amino acid is only associated to the middle nucleotide of its codon and not all three nucleotides. It will also be registered for all nucleotides whether they are involved in the coding of an amino acid or not. This allows a heuristic algorithm that is very similar to the simple DNA (or protein, but not both)—comparing algorithm, which will have the advantage that it is similar to the traditional algorithms, in contrast to an exact algorithm, which would be complex to implement (Hein, 1994) (Fig. 2).

This representation will create a new string with the same length as the ordinary DNA string, except that a position can now contain not only a nucleotide, but also an amino acid (potentially two amino acids if there is a reading frame in the opposite direction).

A central problem is to determine where indels can have any length and where their lengths must be multiples of three. The problem hinges on the concept of homology. If two matched reading frames code for homologous proteins there must have been a reading frame in the DNA all the way back to the most recent common ancestor, and all indel lengths should have lengths that are multiples of three. In regions where no homologous proteins are matched, indels can have any lengths, as there could have been a period in recent history when the DNA was noncoding (Fig. 3). If the homology relationships between proteins in the two sequences are known beforehand, this could be part of the data and this would solve the problem. This is an option in the program and is called *guiding*. It would be advantageous if the method could deduce the homology relationships itself. As homologous proteins are more similar (less distant), there should be an inherent tendency to match homologous instead of nonhomologous proteins. We therefore introduce the rule that if amino acids are matched, it will be assumed that they belong to homologous proteins. This rule cannot be consistently applied if indels can be of any length. A reading frame could then be homologous to one protein in one region, to a second protein in another region, and to no protein in a third region. Indels must therefore have length multiples of three in regions that match reading frames, irrespective of their homology relationships. On the border between coding and noncoding regions, indels can have any length, as this corresponds to insertion-deletion before and after a protein.

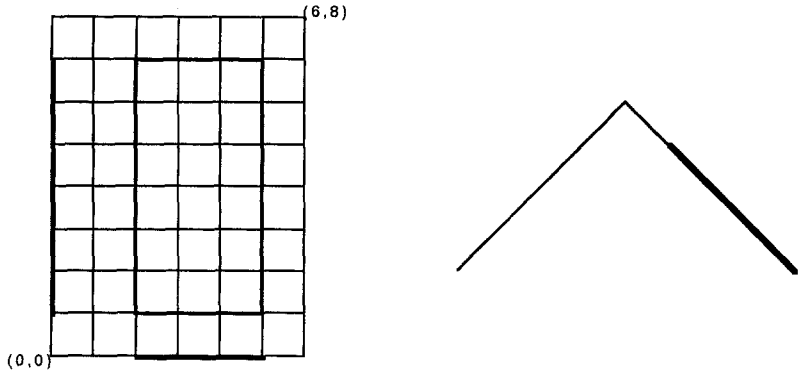


**Fig. 3.** Two matched DNA sequences, each coding for one protein (one and two codons long), that are homologous to each other. The coding nucleotides are shown as *thick bars*. If they are homologous indel lengths must be multiples of three inside the *rectangle of thick bars* but they can be of any length at and outside the thick bars. The *right tree* describes the history of two homologous genomes. Where the *branch* is *thick* a given reading frame was present, and where the *branch* is *thin,* the reading frame was absent. If aligned DNA matches a reading frame (*thickened edge*) with no reading frame, then there is a *period* where the DNA has been noncoding and indels can therefore have any length. The same argument is valid for any combination of matched reading frames: Only if some are homologous must the DNA have been coding all the way back to the most recent common ancestor.

A general algorithm aligning DNA with multiple reading frames can now be formulated as follows: Let $s1$ and $s2$ be two sequences of length l1 and l2, respectively. The substring consisting of the first $i$ elements of $sk$ ($k = 1,2$) is denoted $sk_i$ and $sk[i]$ refers to the $i$'th element of $sk$. Let $D_{i,j}$ be the minimal distance between $s1_i$ *and* $s2_j$ when all insertion-deletions have a length that is a multiple of three and the differences in DNA are weighted by the amino acids that they invoke. Let the indices $i'$ and $j'$ be such that $i$-$i'$ (or $j$-$j'$) are three when only indels of length three are allowed; otherwise they are one. The distance between two elements in the sequences is $d(s1(i),s2(j)) = dn(\text{nuc}(s1(\text{i})),\text{nuc}(s2(j))) + da(aa(s1(\text{i})),aa(s2(j)))$, where $dn(\ ,\ )$ is the distance between two nucleotides. $da(aa(s1(i),aa(s2))$ is the distance between the amino acids associated with that nucleotide. If there is one amino acid at each nucleotide, then it is the traditional amino acid distance; if there is only one amino acid instead of no amino acid, then it will be $g_a$, the cost of deleting one amino acid. ($g_n$ is the cost of deleting one nucleotide.) In contrast to ordinary alignment, the matching term can also contain a weight that corresponds to the insertion-deletion of amino acids. The algorithm is then

Initialization: $D_{0,j}$ (analogous for $D_{i,0}$):

$$D_{0,0} = 0$$
$$D_{0,j} = D_{0,j-1} + d('\text{-}',s2(j))$$
$$D_{i,0} = D_{i-1,0} + d(s1(i),'\text{-}')$$

The $D_{i,j}$ then obeys the recursion:

$D_{i,j}$ = minimum of following three quantities, when $i > 0$ and $j > 0$:

1. (insertion in $s1$) $\{D_{i',j} + g_a{}^*$ (number of *aa* with codons overlapping $(i,i') + g_n\}$
2. (insertion in $s2$) $\{D_{i,j'} + g_a{}^*$ (number of *aa* with codons overlapping $(j,j') + g_n\}$
3. (matching nucleotides) $\{D_{i-1,j-1} + d(s1(\text{i}),s2(j))\}$

The algorithm will thus need two sets of parameters: One set for proteins and one for DNA. Each set has a gap penalty and a distance function on the elements of the sequences—amino acids in the case of proteins and nucleotides in the case of DNA. The guiding principle for the parameters to use is that very frequent events should have a low weight while relative rare events should have a higher weight. Many schemes has been devised to accomplish this (Doolittle 1986). The new problem in this analysis is how to weight events at the protein level relative to events at the DNA level. As proteins are more conserved than DNA and can retain observable similarity over a much longer time, the protein events should have a higher penalty than the DNA events. Since this is a new problem only continued use of the method can determine a satisfactory relative weighting. In principle it would be possible to let the protein level have complete precedence over the DNA level, corresponding to weighing proteins infinitely higher than DNA. DNA would still be a determining factor in the alignment, deciding the precise position of indels within a codon and choosing between equally good protein alignments.

The use of this algorithm is now illustrated in two very simple cases (Figs. 4 and 5).

The algorithm has been extended in several relevant ways that enhance its applicability to real data:

1. Frameshift indels do occur, but at very low frequency. These can be incorporated by allowing for one- and two-length indels but assigning them a very high cost.
2. Where it is known which reading frames are homologous to which reading frames, this can be supplied as part of the data. If reading frame P is homologous to P′, then any matching of these two reading frames with reading frames beside each other should be interpreted as independent insertions of reading frames. This will create a strong tendency for P and P′ to be aligned with each other. This is presently implemented by assuming that reading frames with the same name are homologous.
3. Early in the history of sequence analysis it was realized that long stretches of gapsigns in an alignment were likely to be due to one long indel and not a series of adjacent small indels. This should be reflected in the gap penalty function. The most used gap penalty function is of the form $a + b*k$ (where $k$ is the length of the indel), as this allows for a fast algorithm (Gotoh 1981). This has been incorporated into GenAl and will improve the alignment at the small scale.
4. The program uses Hirschberg's (1975) algorithm, so its memory

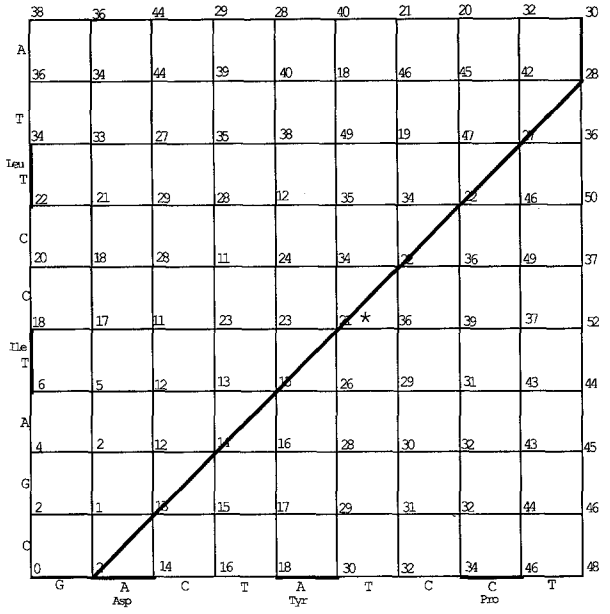


**Fig. 4.** Two DNA sequences with one reading frame each are being aligned. The first sequence is at the x-axis of the matrix and the second is at the y-axis. The weights used are: substitutions between nucleotides is 1, a mutation of an amino acid costs 5, an indel of a nucleotide costs 2, and of an amino acid 10. The distance between $s1_i$ and $s2_j$ is found at the node $(i,j)$. $d(s1i,'-')$ (the cost of deleting the first $i$ elements of $s1$) is 10* (number of amino acids in first i elements) + 2*$i$. The path up through the matrix corresponding to a minimal alignment is shown in *thick lines*. The total weight of the alignment is 30, which corresponds to the deletion of two nucleotides (cost 4), deletion of amino acids (cost 10), the cost of two amino acid replacements (cost 10), and lastly, the substitution of six nucleotides. To calculate the value of the node (5,4) with one asterisk and value 21, three previous cells must be considered: (4,3), and since (5,4) is within both reading frames, indels must come in groups of three and the relevant nodes will be (2,4) and (5,1). The value coming from (2,4) is 11 plus the cost for deleting three nucleotides (6) and one amino acid (10), which will be 27. The value coming from (5,1) is calculated analogously to 45. The value coming from (4,3) is 15 plus the cost for both mutating a nucleotide (1) and an amino acid (5), which is 21. Since 21 is the smallest of the three, this value is assigned (5,4). If a node is not within exons in both sequences, then indels do not have to come in groups of three. To calculate the value associated with node (5,2), which is outside the reading frame in sequence 2, the nodes that must be considered are (4,1), (5,1), and (4,2). The resulting total alignment and protein alignment are shown at the *bottom* of the figure.

requirements is only linear in the length of the sequences and it can be used to analyze very long sequences.

An additional practical improvements is:

5. Alignment algorithms can be sped up and memory requirements can be reduced, without serious loss in reliability, by finding long stretches of common segments to the two sequences. This applies to genomic alignment even more and should be incorporated if it is to be applied to sequences longer than 20 kb. This will also allow terminal indels to be with weight zero, which is realistic, if the sequences compared have been sequenced to different extents.

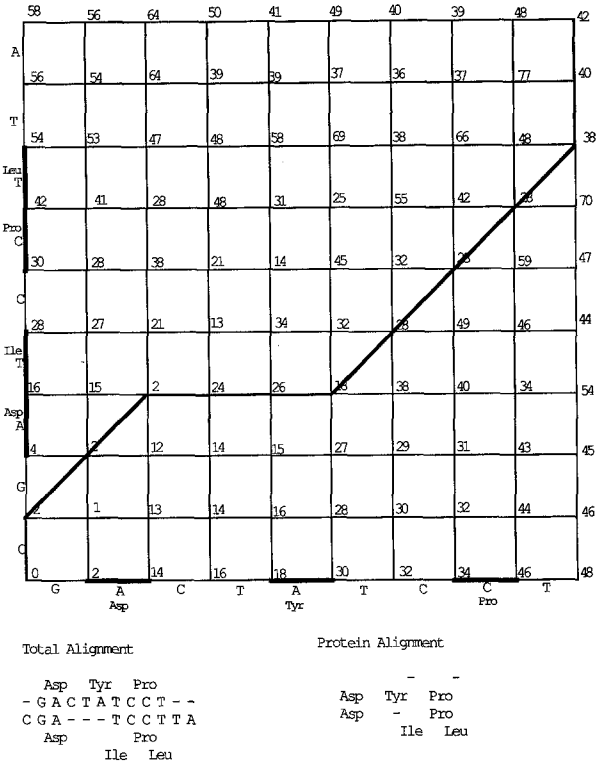


**Fig. 5.** An additional reading frame has been introduced into $s2$ from Fig. 4, so the algorithm has to choose which reading frame the protein in $s1$ is to be matched with in $s2$. It chooses the new reading frame and deletes the old reading frame. The cost of the alignment found is 42, which corresponds to the deletion of six nucleotides (12), the deletion of one reading frame of two amino acids (20), and the deletion of one additional amino acid (10).

## Results

### *Analysis of HIV1 and HIV2*

A program, GenAl, was written implementing the heuristic algorithm. GenAl can be obtained from the Netserver at EMBL. GenAl was used to analyze HIV1 and HIV2. The parameters used were: Nucleotide substitutions 1, amino acid replacements 7, opening an indel 10, deleting-inserting an extra nucleotide 1, deleting-inserting an extra amino acid 5, and a frameshift insertion of one or two nucleotides of cost 100. The analysis took 18 minutes on an HP-705.

The input needed for GenAl is then the complete nucleotide sequences (9,229 bp and 9,636 bp, respectively) and a list of the genes in the two sequences, each gene being a list of exons.

HIV1 has a 300-bp (approximately) deletion around

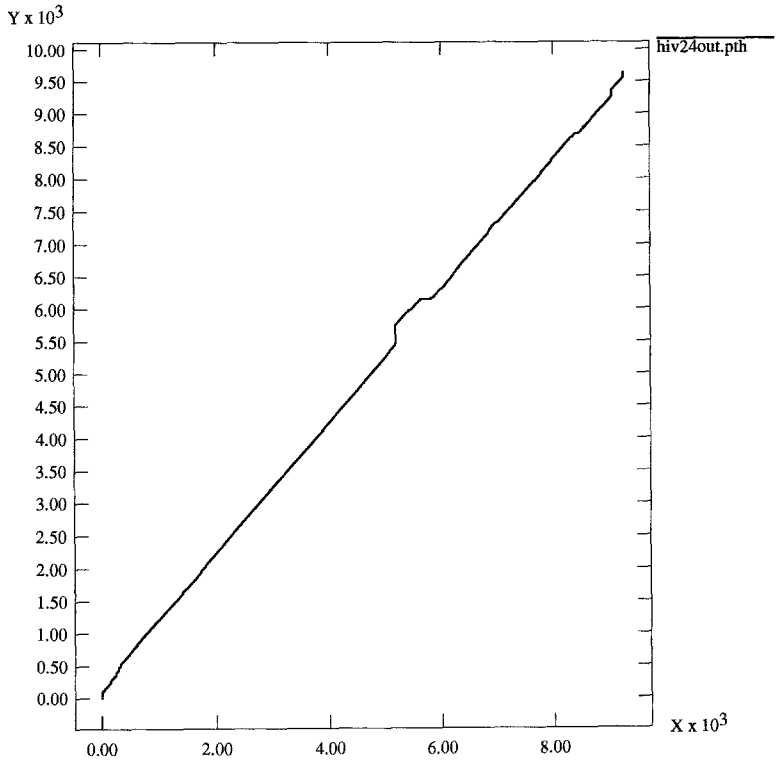


**Fig. 6.** Alignment paths of HIV1 and HIV2. This figure shows the optimal alignment of the two viruses with the present parameters. It stays very close to the diagonal from start to end of the viruses, except around positions 5000–6000, where two indels give it a jump away from the diagonal.

position 5500 and a similar insertion around position 6900 (Fig. 6). There was also a frameshift, so frameshifts had to be allowed to give a correct alignment.

An overview of the resulting alignment is shown in Fig. 7. Homologous exons are drawn the same number of lines away from the center line representing the sequence. The weight of the alignment is 19,725.

A small part of the total alignment is shown in Fig. 8.

The information from the analysis is quite extensive—the total alignment is 40 pages and many questions would be of interest, like: What is the percentage of homology? How are the selective restraints different in noncoding, singly coding, doubly coding regions, etc.? If a region has one exon in one virus and two in the other, it should in principle be possible to evaluate the age of the new reading frame by a method very similar to the one used to evaluate the silencing of a gene. But these questions are detours from the main contribution of this paper and will be addressed in a separate paper (Hein and Støvlbæk 1993a).

A table of the most basic statistics of the homologous reading frames is calculated (Fig. 9).

## Conclusion

A method has been proposed that allows analysis of large sequences, strongly reducing the labor needed from the researcher.

The basic idea is to combine the protein alignment problem with the DNA alignment problem and then solve them simultaneously. The alignment will then also align homologous exons with homologous exons, because this is more parsimonious. The introduction of guiding will be useful when comparing sequences where the homology relationship between the exons are much in doubt and different possibilities should be investigated.

The method presented with be useful in analyzing DNA in the range 1–20 kb (possible larger) but will most likely need to be augmented by alternative methods in megasequencing projects. The evolution of very large DNA segments will also experience genetic events not included in traditional alignment methods such as duplications, inversions, translocations, and recombinations.

The method has been extended to multiple genomic alignment by adding an algorithm that braids pairwise alignments into one multiple alignment (Hein and Støvlbæk 1993b).

*Acknowledgments.* J.H. was supported by the Japanese Society for the Promotion of Science, the Carlsberg Foundation and the Danish Research Council, grants 11-8916-1 and 11-9639-1. J.S. was supported by the Danish Research Council, grants 11-8916-1 and 11-9639-1. Bernt Guldbrandtsen is thanked for comments on the manuscript.

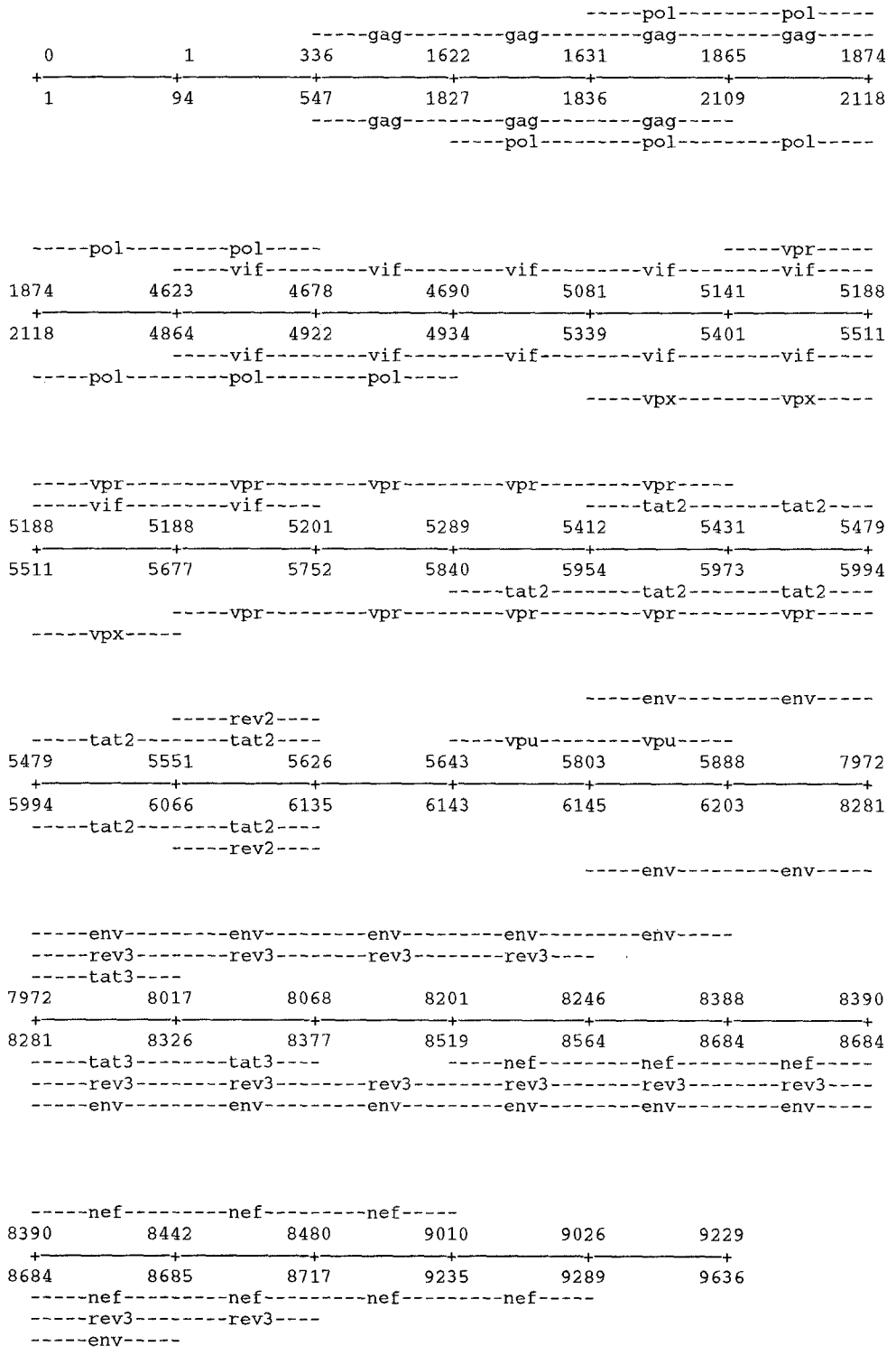


**Fig. 7.** A compact overview of the total alignment. HIV1 and HIV2 both have the reading frames *vif, vpr, tat2, tat3, rev2, rev3, env,* and *nef.* Besides these, HIV1 has *vpu* and HIV2 has *vpx* close to the two large indels. Homologous reading frames are put the same number of lines away from the central line representing the RNA. This representation should be very easy to interpret: Both viruses start with pieces of noncoding RNA (335 bp and 546 bp, respectively). Then the *gag* gene starts at homologous positions in both viruses. HIV2 has an insertion at the very beginning relative to HIV1. The *pol* gene in HIV2 starts 9 bp before the *pol* gene in HIV1, corresponding to 3 amino acids.

```
    Ala   Phe   Leu   Gln   Gly   Lys                 Ala   Arg   Glu   Phe
      Pro   Ser   Tyr   Lys   Gly   Arg                   Pro   Gly   Asn   Phe
1650 C C T T C C T A C A A G G G A A G - - - - - - G C C A G G G A A T T T T
1855 T G G G G A A A G A A G C C C C G C A A C T T C C C C G T G G T C C C A
      Trp   Gly   Lys   Lys   Pro   Arg   Asn   Phe   Pro   Val   Val   Pro
    Met   Gly   Lys   Glu   Ala   Pro   Gln   Leu   Pro   Arg   Gly   Pro

    Ser   Ser   Glu   Gln   Thr   Arg   Ala   Asn   Ser   Pro   Thr   Ile
      Leu   Gln   Ser   Arg   Pro   Glu   Pro   Thr   Ala   Pro   Pro   Phe
1680 C T T C A G A G C A G A C C A G A G C C A A C A G C C C C A C C A T T -
1891 A G T T C G C A G G G G C T A A C A C C A A C A G C A C C C C C A A T G
      Ser   Ser   Gln   Gly   Leu   Thr   Pro   Thr   Ala   Pro   Pro   Met
    Lys   Phe   Ala   Gly   Ala   Asn   Thr   Asn   Ser   Thr   Pro   Asn

                Ser   Ser   Glu   Gln   Thr   Arg
                  Leu   Gln   Ser   Arg   Pro   Glu
1715 - - - - - T C T T C A G A G C A G A C C A G A - - - - - - - - - - - - - G
1927 G A T C C A G C A G T G G A C C T A C T G G A G A A G T A C A T G C A G
      Asp   Pro   Ala   Val   Asp   Leu   Leu   Glu   Lys   Tyr   Met   Gln
    Gly   Ser   Ser   Ser   Gly   Pro   Thr   Gly   Glu   Val   His   Ala

    Ala   Asn   Ser   Pro   Thr         Arg   Arg   Glu   Leu   Gln   Val
      Pro   Thr   Ala   Pro         Pro   Glu   Glu   Ser   Phe   Arg   Ser
1734 C C A A C A G C C C C A C - - - C A G A A G A G A G C T T C A G G T C T
1963 C A A G G G A G A A A A C A G A G A G A G C A G A G A C A A A G A C C A
      Gln   Gly   Arg   Lys   Gln   Arg   Glu   Gln   Arg   Gln   Arg   Pro
    Ala   Arg   Glu   Lys   Thr   Glu   Arg   Ala   Glu   Thr   Lys   Thr
```

**Fig. 8.** Alignment showing part of the small region involved in the coding of both *gag* and *pol*. The *pol* amino acids are shown two above and two below the DNA, while the *gag* amino acids are one above and below. The percentage identity of matched amino acids is 53.6 for the *gag* and 56.0 for the *pol* sequences. The first indel deletes (or inserts) exactly two codons in the *pol* exon, but does not match codon boundaries in the *gag* exon. In this case the indel could be skidded 1 bp 3′-ward without additional cost to the alignment, making the reverse statement true. This is not generally true. Sometimes an indel doesn't match any reading frame. It is also clear that associating the amino acid with the middle nucleotide creates a protein alignment from the DNA alignment.

| reading frame: | gag | pol | vif | vpr | tat2 | tat3 | rev2 | rev3 | env | nef |
|---|---|---|---|---|---|---|---|---|---|---|
| HIVBRU length | 1539 | 3048 | 579 | 291 | 215 | 46 | 76 | 275 | 2586 | 621 |
| HIV22ISY length | 1563 | 3048 | 648 | 318 | 296 | 97 | 70 | 437 | 2685 | 627 |
| terminal indels | 9 | 21 | 13 | 68 | 114 | 51 | 0 | 151 | 1 | 219 |
| internal indels | 69 | 51 | 612 | 330 | 48 | 6 | 6 | 96 | 297 | 105 |
| base identity | 61.9 | 63.6 | 54.7 | 64.1 | 63.3 | 67.3 | 57.9 | 52.3 | 55.7 | 60.9 |
| amino acid identity | 53.6 | 56.0 | 25.9 | 40.4 | 40.3 | 26.7 | 32.0 | 33.7 | 37.5 | 35.3 |

**Fig. 9.** Each pair of homologous reading frames is scored for the length of the DNA sequences being matched, insertion, deletions, substitutions at the DNA level, replacements the protein level, and percentage similarity at both DNA and protein levels. An element is called inserted if it is present in the lower sequence and not the upper sequence and vice versa. It is seen that *pol* has the highest percentage similarity and *tat3* has the lowest level of similarity. The level of similarity is high enough to be regarded as significant for all homologous reading frames.

## References


Doolittle RF (1986) Of URFs and ORFs. University Science Books

Gotoh O (1981) An improved algorithm for matching biological sequences. J Mol Biol 162:705–708

Hein JJ (1994) An algorithm combining DNA and protein alignment. J Theor Biol (in press)

Hein JJ, Støvlbæk J (1993a) A method to analyze aligned genomic DNA sequences. (submitted to J Mol Evol)

Hein JJ, Støvlbæk J (1993b) Multiple genomic alignment. (in preparation)

Hirschberg DS (1975) A linear space algorithm for computing maximal common subsequences. Comm ACM 18(6):341–343

Needlemann SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequences of two proteins. J Mol Biol 48:444–453

Sankoff D (1972) Matching sequences under deletion/insertion constraints. Proc Natl Acad Sci USA 69:4–6