CSE 397-497: *Computational Issues in Molecular Biology*

Lecture 7

Spring 2004

LEHIGH UNIVERSITY

CSE seminar on Monday

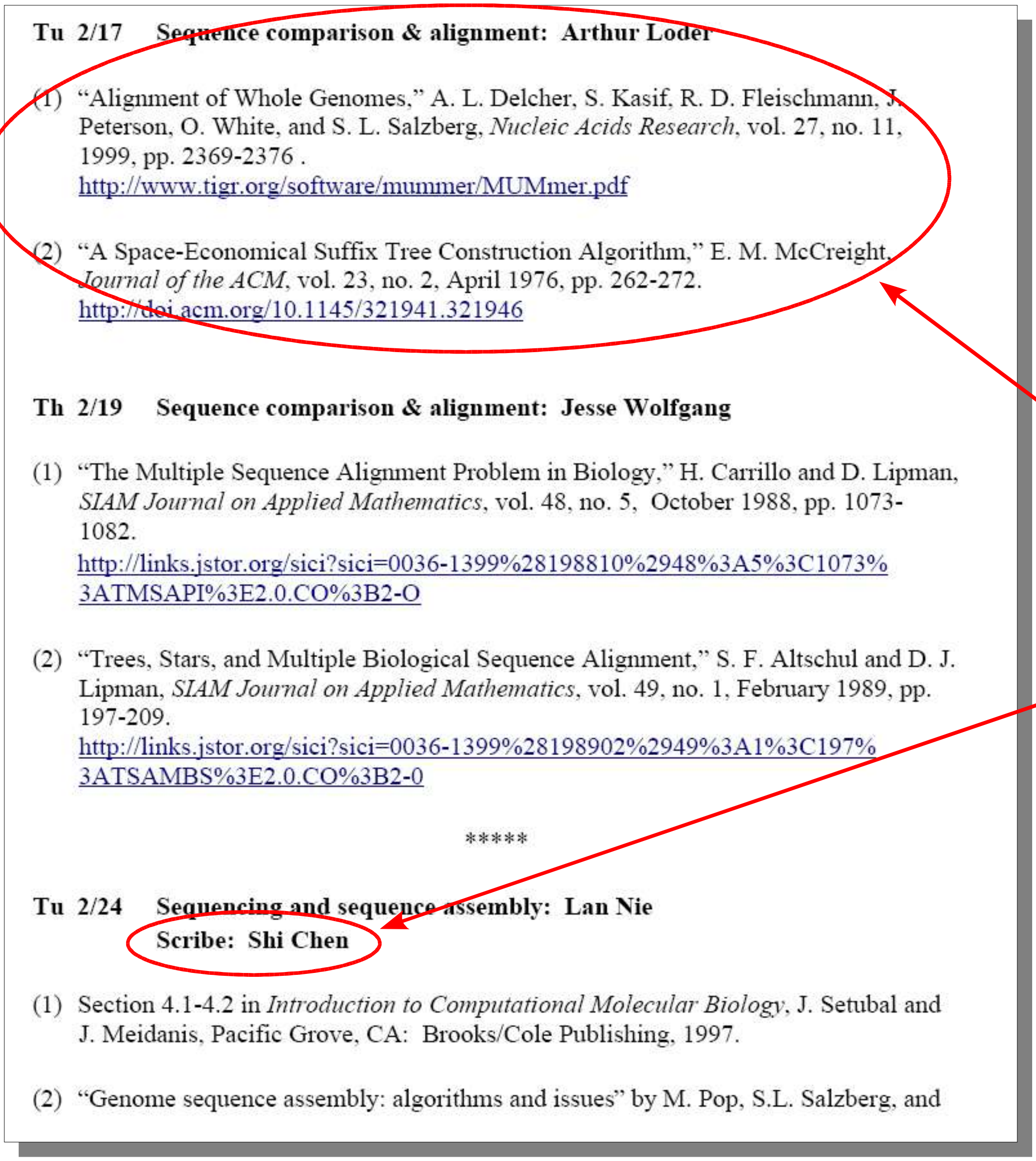
Title: Redundancy Elimination Within Large Collections of Files
Speaker: Dr. Fred Douglis (IBM T.J. Watson Research Center)
Date: Monday, February 16
Time: 4:00 pm
Place: PL 466

From the abstract:

> "Ongoing advancements in technology lead to ever-increasing storage capacities. In spite of this, optimizing storage usage can still provide rich dividends. Several techniques based on delta-encoding and duplicate block suppression have been shown to reduce storage overheads ..."

Could there be a connection to the algorithms we've studied?

Student lectures: reading list, scribe duties are online



Tu 2/17 Sequence comparison & alignment: Arthur Loder

(1) "Alignment of Whole Genomes," A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, *Nucleic Acids Research*, vol. 27, no. 11, 1999, pp. 2369-2376 . http://www.tigr.org/software/mummer/MUMmer.pdf

(2) "A Space-Economical Suffix Tree Construction Algorithm," E. M. McCreight, *Journal of the ACM*, vol. 23, no. 2, April 1976, pp. 262-272. http://doi.acm.org/10.1145/321941.321946

Th 2/19 Sequence comparison & alignment: Jesse Wolfgang

(1) "The Multiple Sequence Alignment Problem in Biology," H. Carrillo and D. Lipman, *SIAM Journal on Applied Mathematics*, vol. 48, no. 5, October 1988, pp. 1073-1082. http://links.jstor.org/sici?sici=0036-1399%28198810%2948%3A5%3C1073%3ATMSAPI%3E2.0.CO%3B2-O

(2) "Trees, Stars, and Multiple Biological Sequence Alignment," S. F. Altschul and D. J. Lipman, *SIAM Journal on Applied Mathematics*, vol. 49, no. 1, February 1989, pp. 197-209. http://links.jstor.org/sici?sici=0036-1399%28198902%2949%3A1%3C197%3ATSAMBS%3E2.0.CO%3B2-0

*****

Tu 2/24 Sequencing and sequence assembly: Lan Nie
Scribe: Shi Chen

(1) Section 4.1-4.2 in *Introduction to Computational Molecular Biology*, J. Setubal and J. Meidanis, Pacific Grove, CA: Brooks/Cole Publishing, 1997.

(2) "Genome sequence assembly: algorithms and issues" by M. Pop, S.L. Salzberg, and

Posted on Blackboard; to be updated periodically as you meet with me to set lecture material.

Be sure to check this for readings (<u>in</u> <u>advance</u>).

CSE 497 students: make sure you know when you will scribe.

Everyone: let me know if you see a mistake ASAP.

Exciting new development

A couple of lectures ago, I noted:

Early literature:

"Binary codes capable of correcting deletions, insertions and reversals," V. Levenshtein, *Soviet Physics Daklady*, 10:707-710, 1966.
→ (So far as I know, this is only available in Russian.) ←

"A general method applicable to the search for similarities in the amino acid sequences of two proteins," S. B. Needleman and C. D. Wunsch, *Journal of Molecular Biology*, 48:443-453, 1970.

"The string to string correction problem," R. A. Wagner and M. J. Fisher, *Journal of the Association for Computing Machinery*, 21(1):168-173, 1974.

Thanks to the ingenuity and initiative of Upmanyu, we now have a copy of Levenshtein's original paper in English!

(To be made available on Blackboard soon, once I've had a chance to review it myself.)

Cross-domain approximate string matching

"Cross-Domain Approximate String Matching," D. Lopresti and G. Wilfong, *Proceedings of the Sixth International Symposium on String Processing and Information Retrieval*, September 1999, Cancún, Mexico, pp. 120-127.

We've seen a number of techniques for comparing two sequences to decide when they are similar.

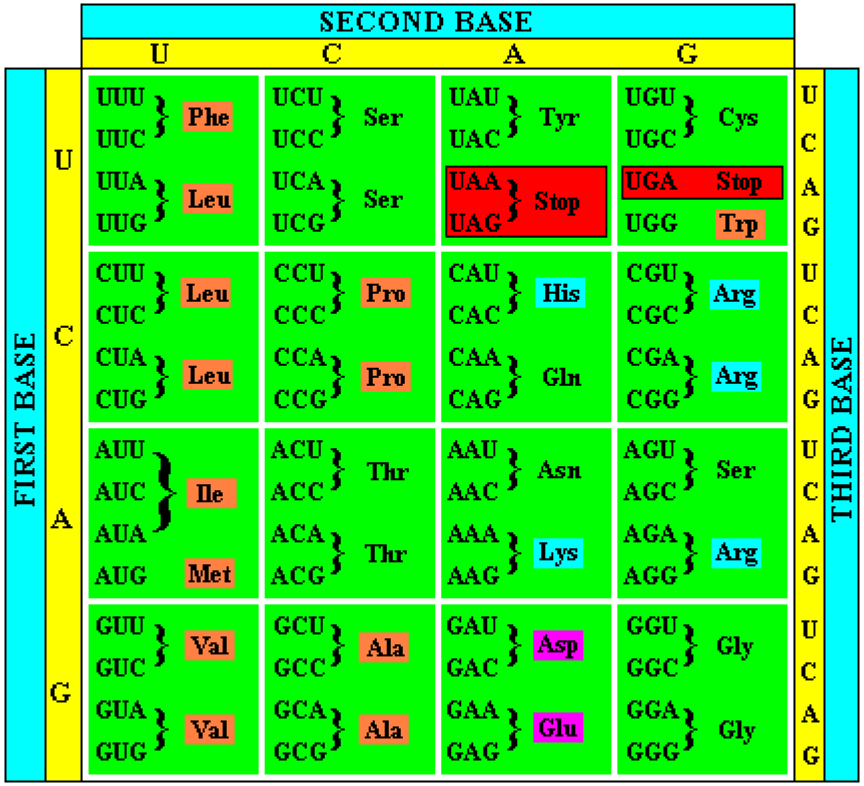
In an abstract sense, all are <u>independent</u> of underlying alphabet.

I.e., same basic algorithm can be used for proteins (20 symbol alphabet) and for nucleic acid sequences (4 symbol alphabet).

But what happens when biologist has one kind of sequence and wants to search for matches expressed as the other kind?

Questions and an obvious solution

- Given an mRNA (or DNA) sequence, find protein sequences similar to the protein it codes for.
- Given a protein sequence, find mRNA (or DNA) sequences similar to the mRNA (or DNA) sequence that codes for it.



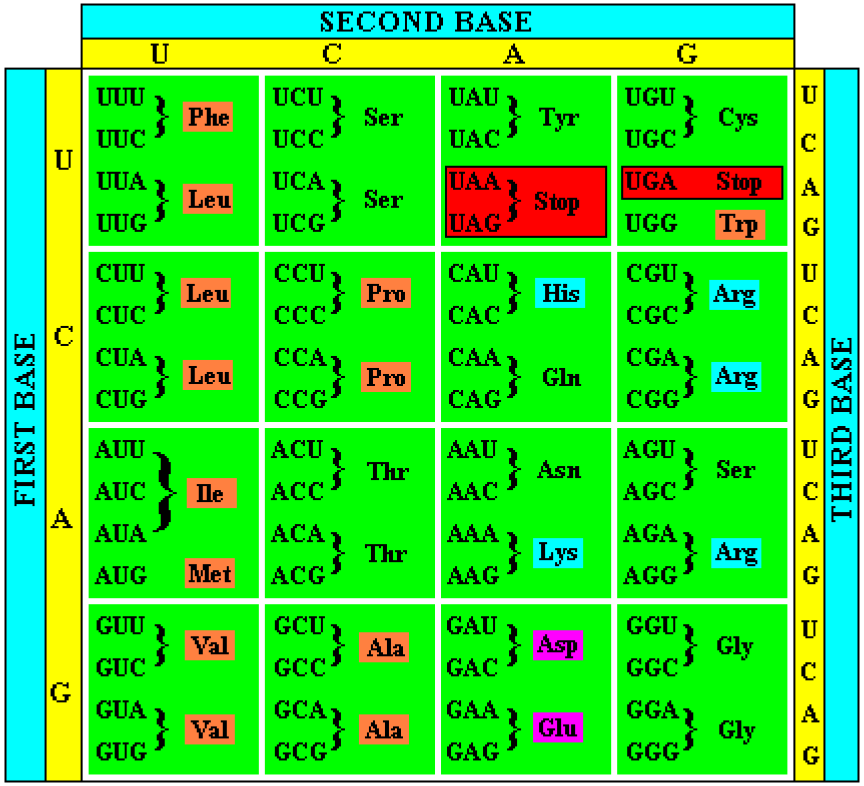
	SECOND BASE				
FIRST BASE	U	C	A	G	THIRD BASE
U	UUU Phe	UCU Ser	UAU Tyr	UGU Cys	U
	UUC	UCC	UAC	UGC	C
	UUA Leu	UCA Ser	UAA Stop	UGA Stop	A
	UUG	UCG	UAG	UGG Trp	G
C	CUU Leu	CCU Pro	CAU His	CGU Arg	U
	CUC	CCC	CAC	CGC	C
	CUA Leu	CCA Pro	CAA Gln	CGA Arg	A
	CUG	CCG	CAG	CGG	G
A	AUU Ile	ACU Thr	AAU Asn	AGU Ser	U
	AUC	ACC	AAC	AGC	C
	AUA	ACA Thr	AAA Lys	AGA Arg	A
	AUG Met	ACG	AAG	AGG	G
G	GUU Val	GCU Ala	GAU Asp	GGU Gly	U
	GUC	GCC	GAC	GGC	C
	GUA Val	GCA Ala	GAA Glu	GGA Gly	A
	GUG	GCG	GAG	GGG	G

We know the Genetic Code. Use it to translate one sequence into the alphabet of the other sequence.

mRNA → protein = six reading frames

protein → mRNA = exponentially many reverse translations (less desirable)

The Genetic Code



		SECOND BASE				
FIRST BASE		U	C	A	G	THIRD BASE
	U	UUU Phe	UCU Ser	UAU Tyr	UGU Cys	U
		UUC	UCC	UAC	UGC	C
		UUA Leu	UCA Ser	UAA Stop	UGA Stop	A
		UUG	UCG	UAG	UGG Trp	G
	C	CUU Leu	CCU Pro	CAU His	CGU Arg	U
		CUC	CCC	CAC	CGC	C
		CUA Leu	CCA Pro	CAA Gln	CGA Arg	A
		CUG	CCG	CAG	CGG	G
	A	AUU Ile	ACU Thr	AAU Asn	AGU Ser	U
		AUC	ACC	AAC	AGC	C
		AUA	ACA Thr	AAA Lys	AGA Arg	A
		AUG Met	ACG	AAG	AGG	G
	G	GUU Val	GCU Ala	GAU Asp	GGU Gly	U
		GUC	GCC	GAC	GGC	C
		GUA Val	GCA Ala	GAA Glu	GGA Gly	A
		GUG	GCG	GAG	GGG	G

So, for example, ...

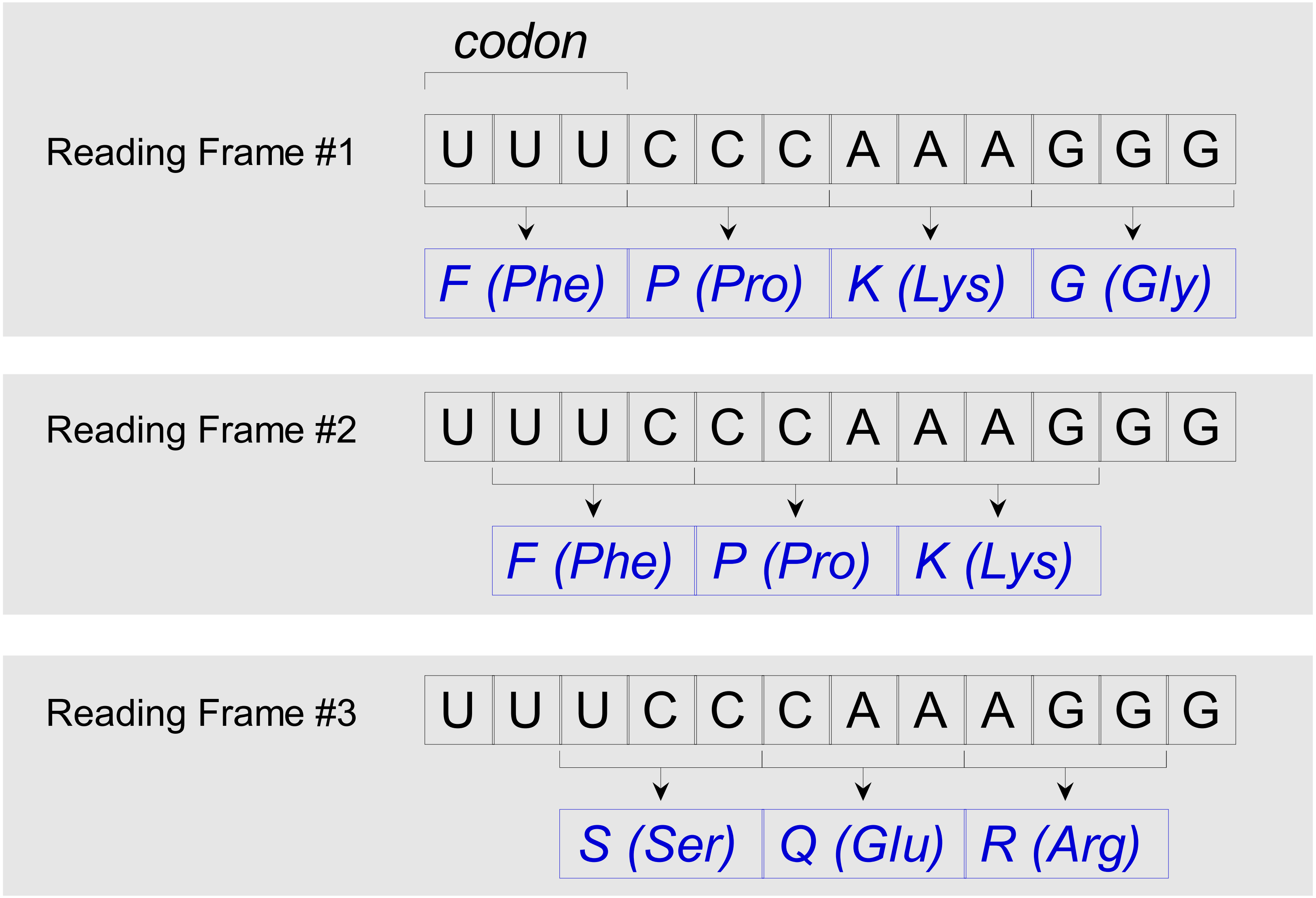
UUU $\rightarrow$ *Phe*
(= *F*, *Phenylalanine*)

CCC $\rightarrow$ *Pro*
(= *P*, *Proline*)

AAA $\rightarrow$ *Lys*
(= *K*, *Lysine*)

GGG $\rightarrow$ *Gly*
(= *G*, *Glycine*)

mRNA to protein translation



NCBI BLAST “translating” options

Translating BLAST searches translate either query sequences or databases from nucleotides to proteins so that protein - nucleotide sequences can be performed.

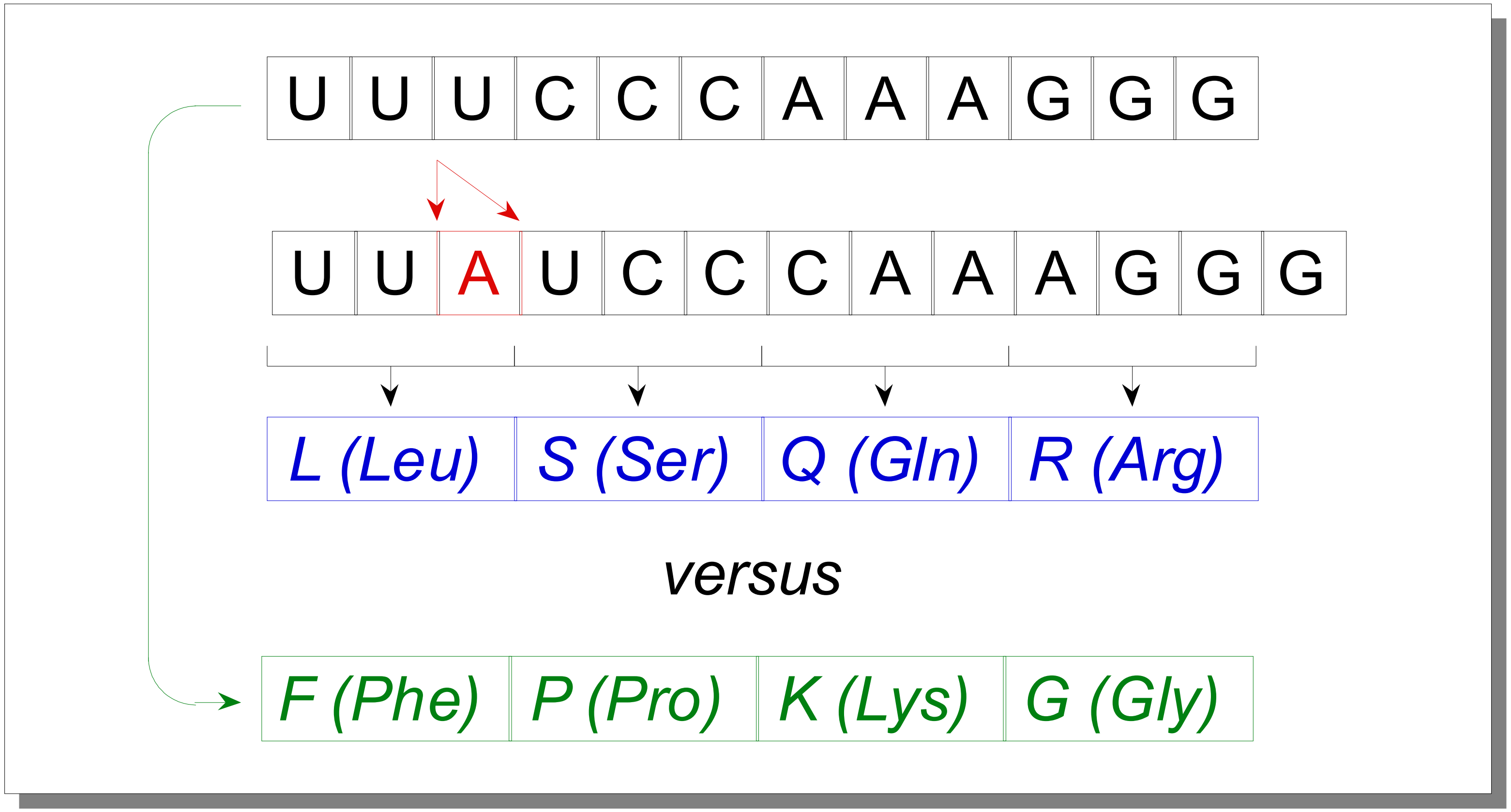
Translated query - Protein db [blastx] - Converts a nucleotide query sequence into protein sequences in all 6 reading frames. The translated protein products are then compared against the NCBI protein databases

Protein query - Translated db [tblastn] - Takes a protein query sequence and compares it against an NCBI nucleotide database which has been translated in all six reading frames.

Translated query - Translated db [tblastx] - Converts a nucleotide query sequence into protein sequences in all 6 reading frames and then compares this to an NCBI nucleotide database which has been translated in all six reading frames.

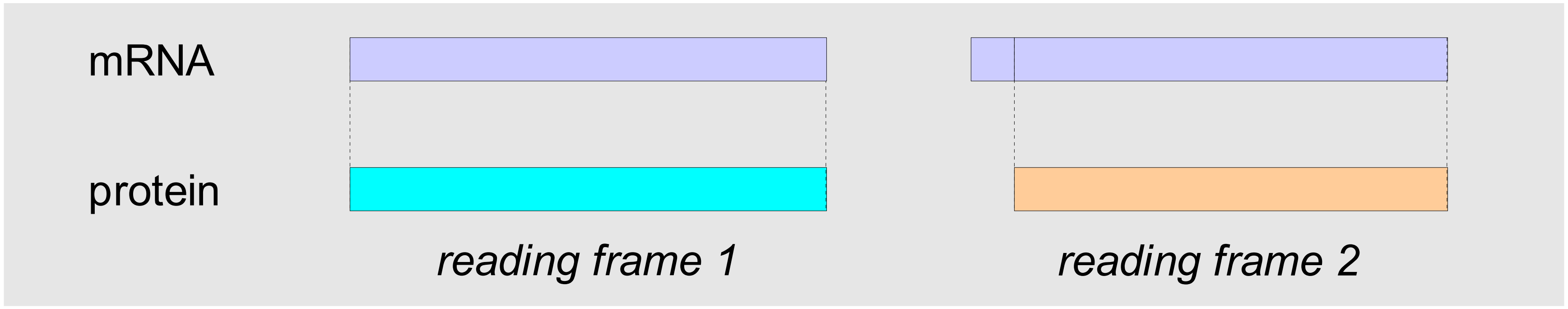
http://www.ncbi.nlm.nih.gov/blast/html/BLASThomehelp.html#TRBLAST

With a single nucleotide insertion

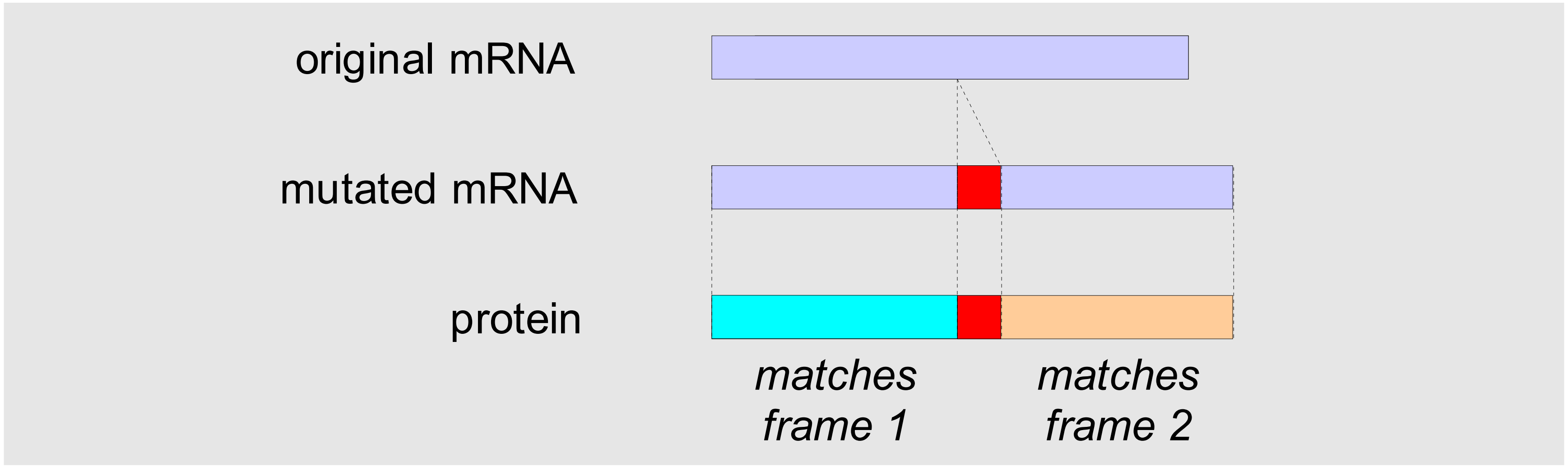


Small change in mRNA (or DNA) sequence leads to large change in protein sequence, assuming a given reading frame.

Impact of insertions or deletions



Even allowing for comparing different reading frames ...

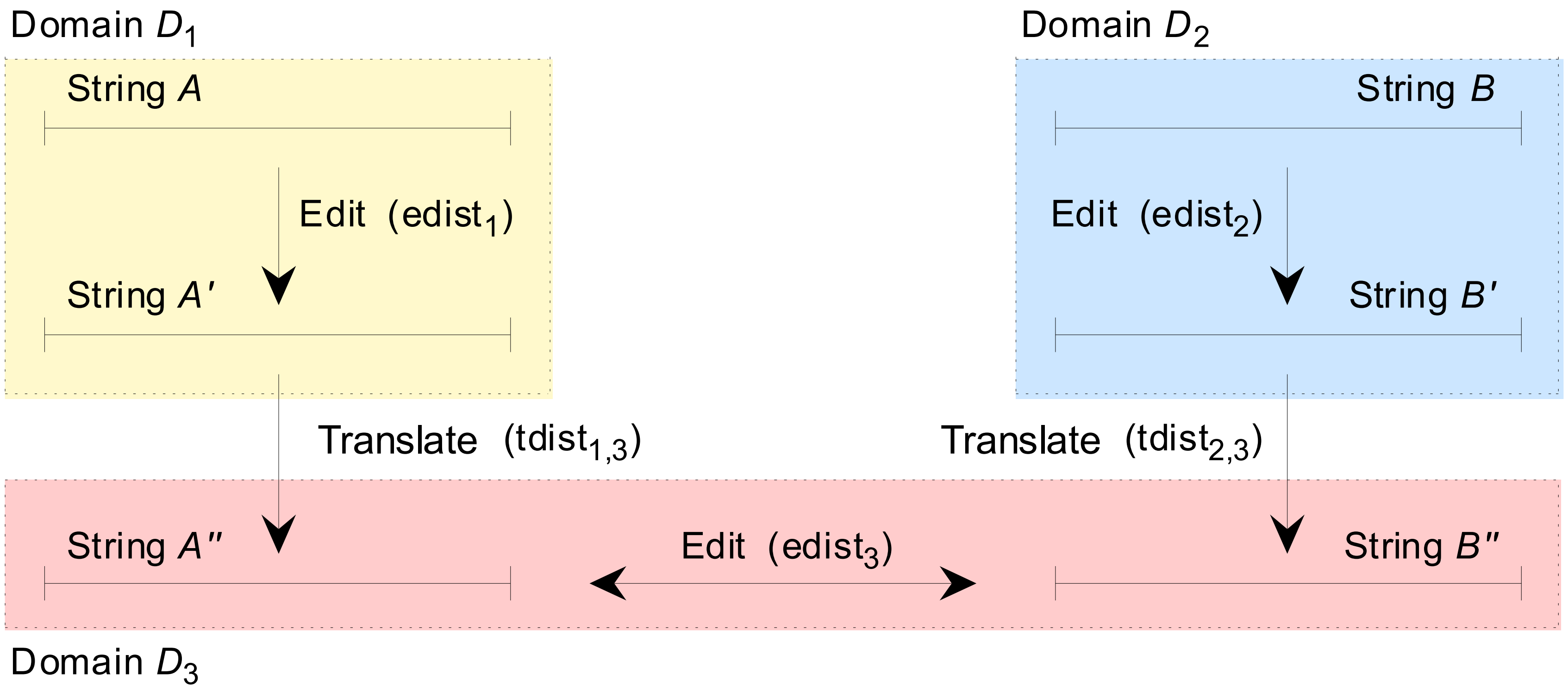


... a single isolated event has far-reaching consequences.

Cross-domain approximate string matching

Problem: sequence comparison is phrased as optimization problem, but translation is oblivious to this.

Solution: cross-domain approximate string matching.



Formal statement of the problem

Given two strings $A \in D_1$ and $B \in D_2$, along with a third domain D_3, find the optimal way to:

(a) Edit A into some other string A' in D_1,

(b) Translate A' to A'' in D_3,

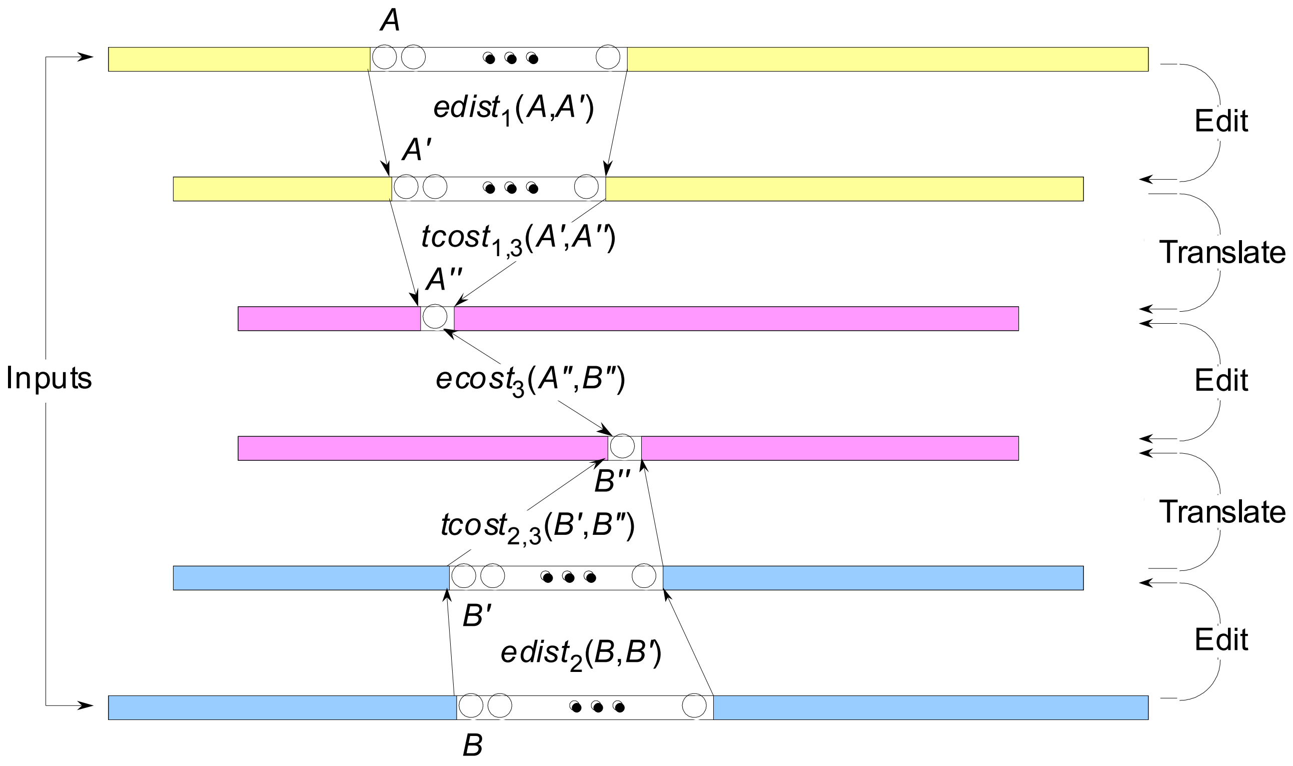
(c) Edit B into some other string B' in D_2,

(d) Translate B' to B'' in D_3,

(e) Compare A'' and B'' in D_3.

$$xdist_{1,2,3}(A,B) \equiv \min_{\substack{A' \in \Sigma_1^*,\, B' \in \Sigma_2^* \\ A'',\, B'' \in \Sigma_3^*}} \left\{ \begin{array}{c} edist_1(A, A') + tdist_{1,3}(A', A'') + \\ edist_2(B, B') + tdist_{2,3}(B', B'') + \\ edist_3(A'', B'') \end{array} \right\}$$

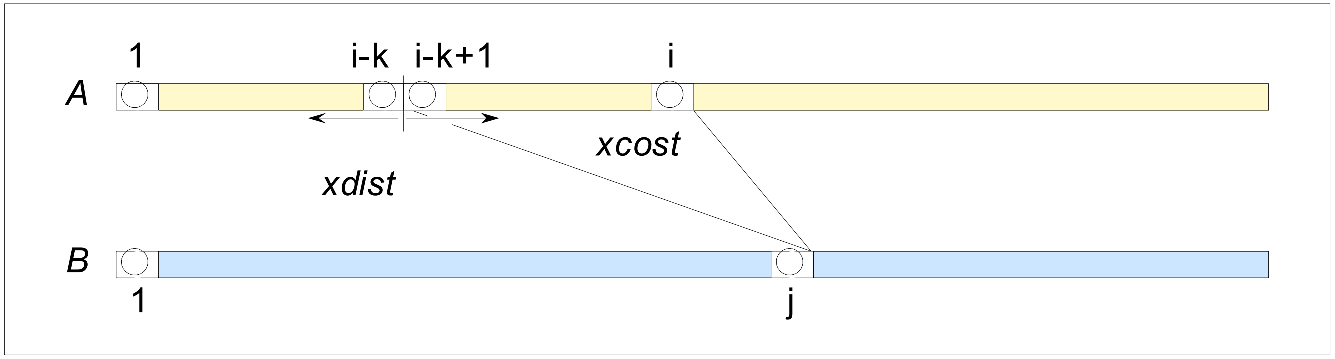
Algorithm: preprocessing



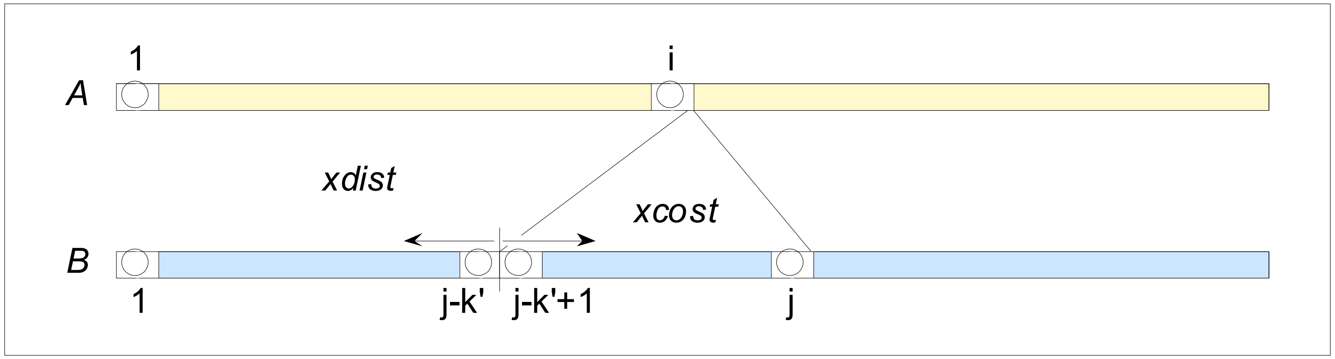
$$xcost_{1,2,3}(A,B) = \min_{\substack{A' \rightarrow A'' \in \tau_{1,3},\\ B' \rightarrow B'' \in \tau_{2,3}}} \left\{ \begin{array}{c} edist_1(A,A') + tcost_{1,3}(A',A'') + \\ edist_2(B,B') + tcost_{2,3}(B',B'') + \\ ecost_3(A'',B'') \end{array} \right\}$$

Algorithm: case analysis

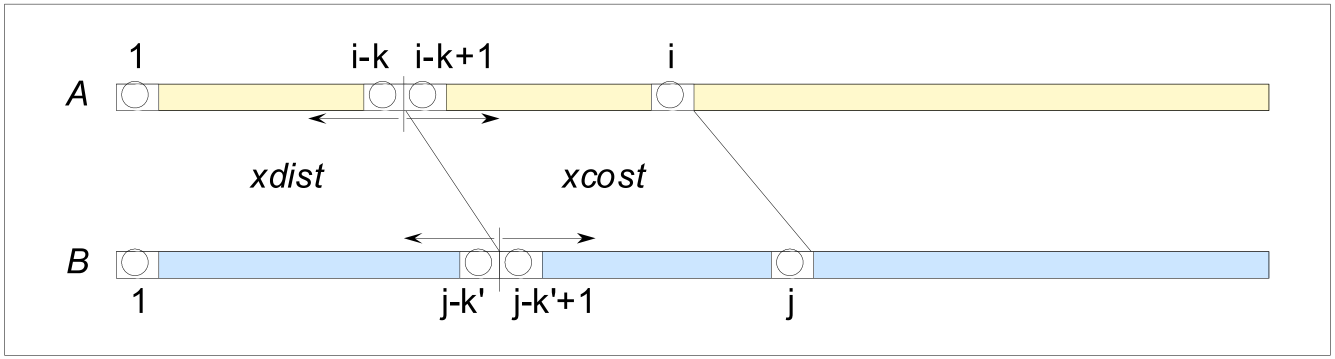
Case 1



Case 2



Case 3



Algorithm: main recurrence

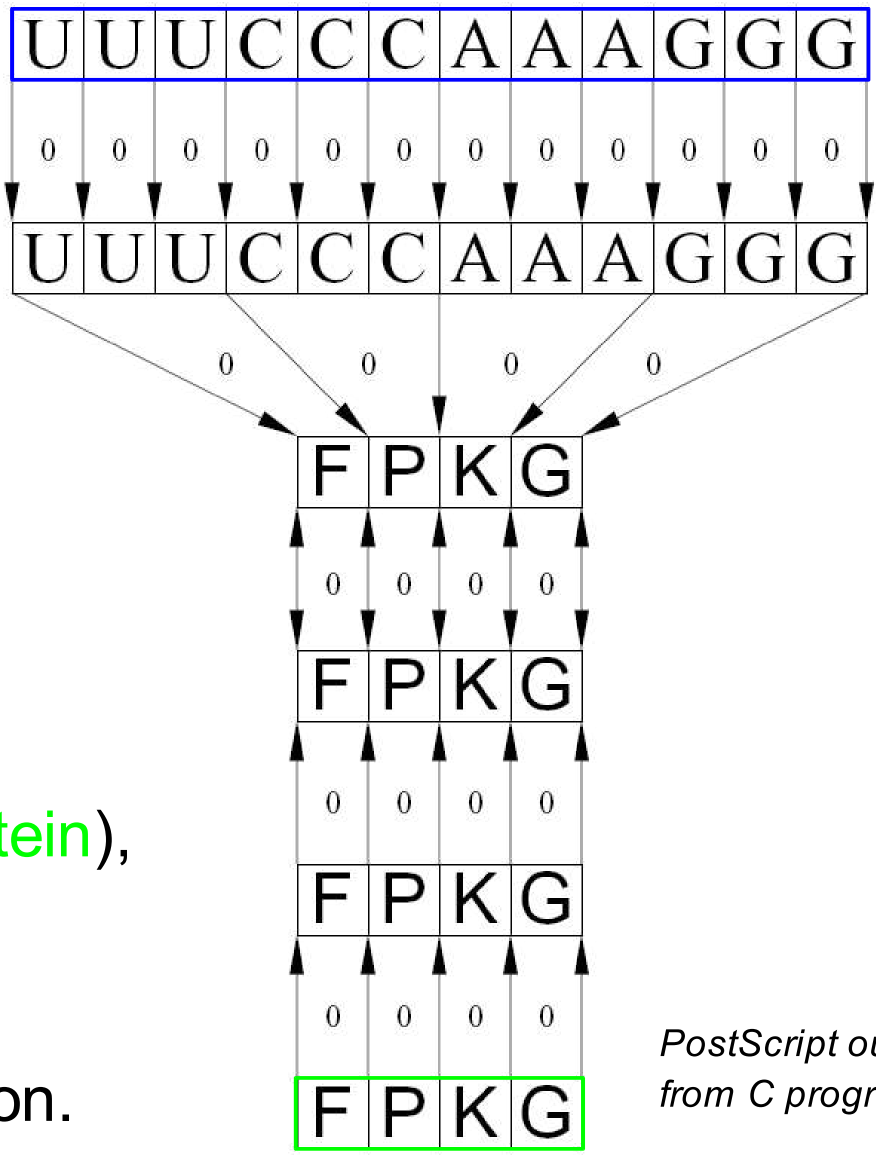
$$xdist_{1,2,3}(A_{1,i}, B_{1,j}) = \min \left\{ \begin{array}{l} \min\limits_{1 \leqslant k \leqslant i} \left[\begin{array}{c} xdist_{1,2,3}(A_{1,i-k}, B_{1,j}) + \\ xcost_{1,2,3}(A_{i-k+1,i}, \epsilon) \end{array} \right] \\ \min\limits_{1 \leqslant k' \leqslant j} \left[\begin{array}{c} xdist_{1,2,3}(A_{1,i}, B_{1,j-k'}) + \\ xcost_{1,2,3}(\epsilon, B_{j-k'+1,j}) \end{array} \right] \\ \min\limits_{\substack{1 \leqslant k \leqslant i, \\ 1 \leqslant k' \leqslant j}} \left[\begin{array}{c} xdist_{1,2,3}(A_{1,i-k}, B_{1,j-k'}) + \\ xcost_{1,2,3}(A_{i-k+1,i}, B_{j-k'+1,j}) \end{array} \right] \end{array} \right\}$$

Time complexity:	$O(m^2 n^2 \lvert\tau_{1,3}\rvert\lvert\tau_{2,3}\rvert)$	to compute $xcost$
	$O(m^2 n^2)$	to compute $xdist$
Bound k and k':	$O(mn\lvert\tau_{1,3}\rvert\lvert\tau_{2,3}\rvert)$	to compute $xcost$
	$O(mn)$	to compute $xdist$

Cross-domain string matching: example 1

Recall from Genetic Code:

UUU $\rightarrow$	F
CCC $\rightarrow$	P
AAA $\rightarrow$	K
GGG $\rightarrow$	G



Inputs to computation:

- two sequences (RNA & protein),
- three edit models,
- two translation models.

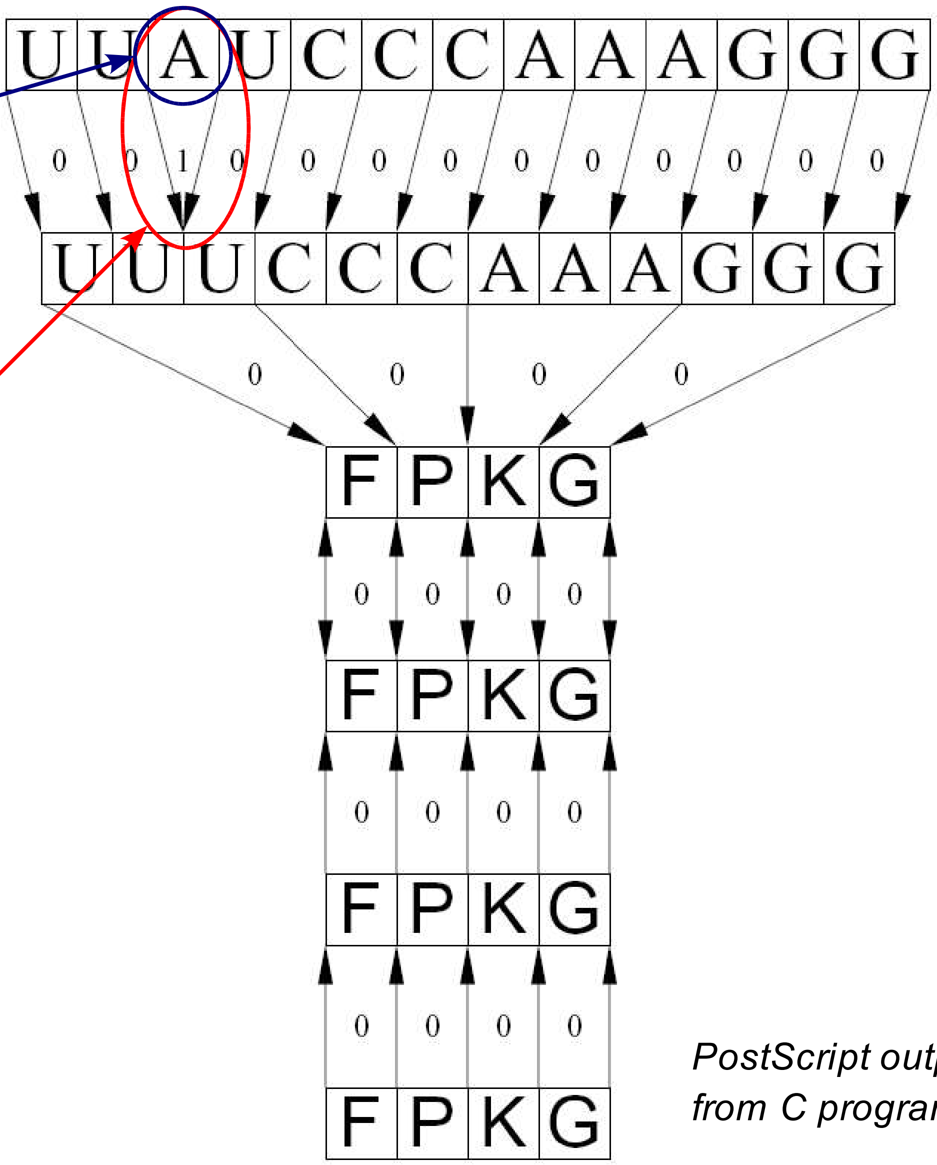
All else arises from optimization.

PostScript output from C program

Cross-domain string matching: example 2

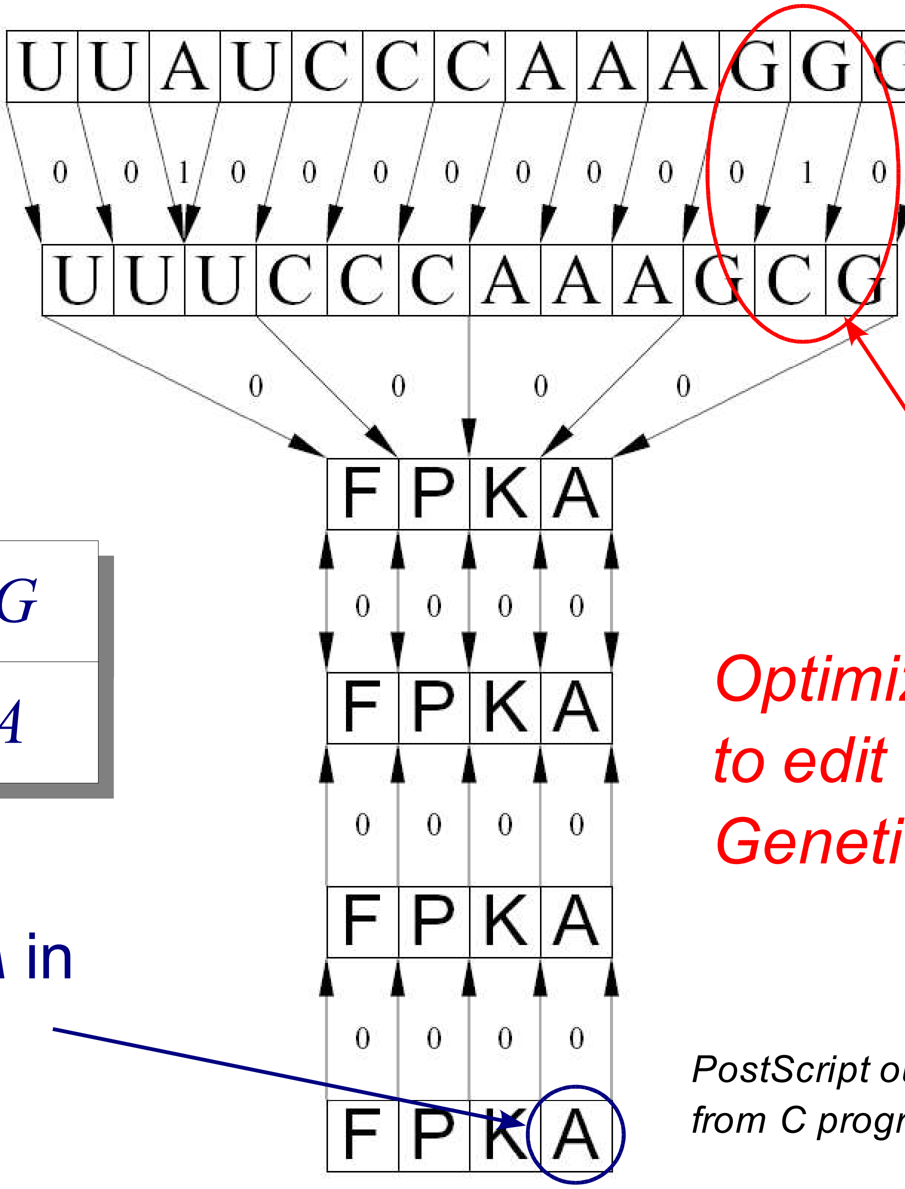
Insert *A* into original RNA sequence as we saw before:

Optimization detects insertion and corrects for it.



PostScript output from C program

Cross-domain string matching: example 3



Genetic Code:

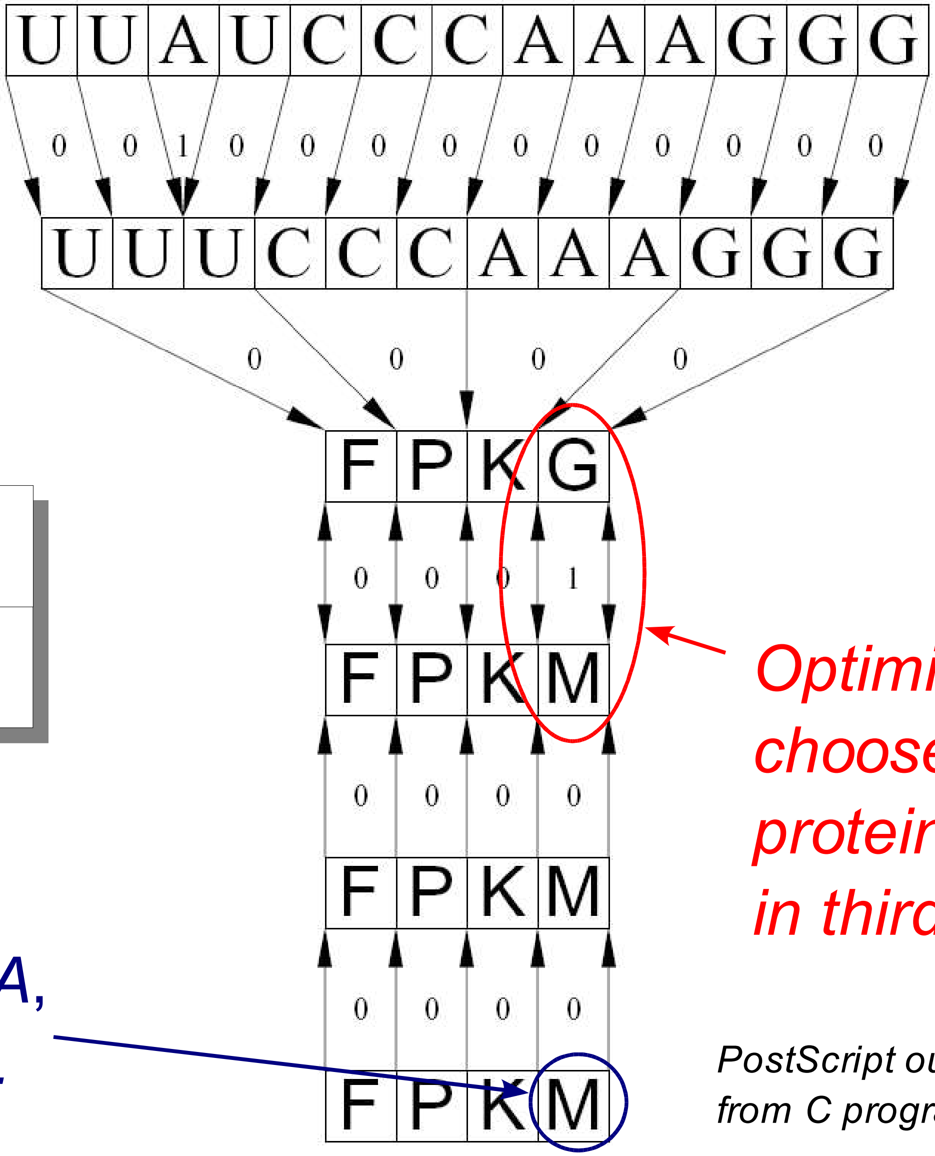
GGG → *G*	
GCG → *A*	

Optimization chooses to edit RNA, exploiting Genetic Code.

Change *G* to *A* in original protein sequence.

PostScript output from C program

Cross-domain string matching: example 4



Genetic Code:

GGG $\rightarrow$	*G*
AUG $\rightarrow$	*M*

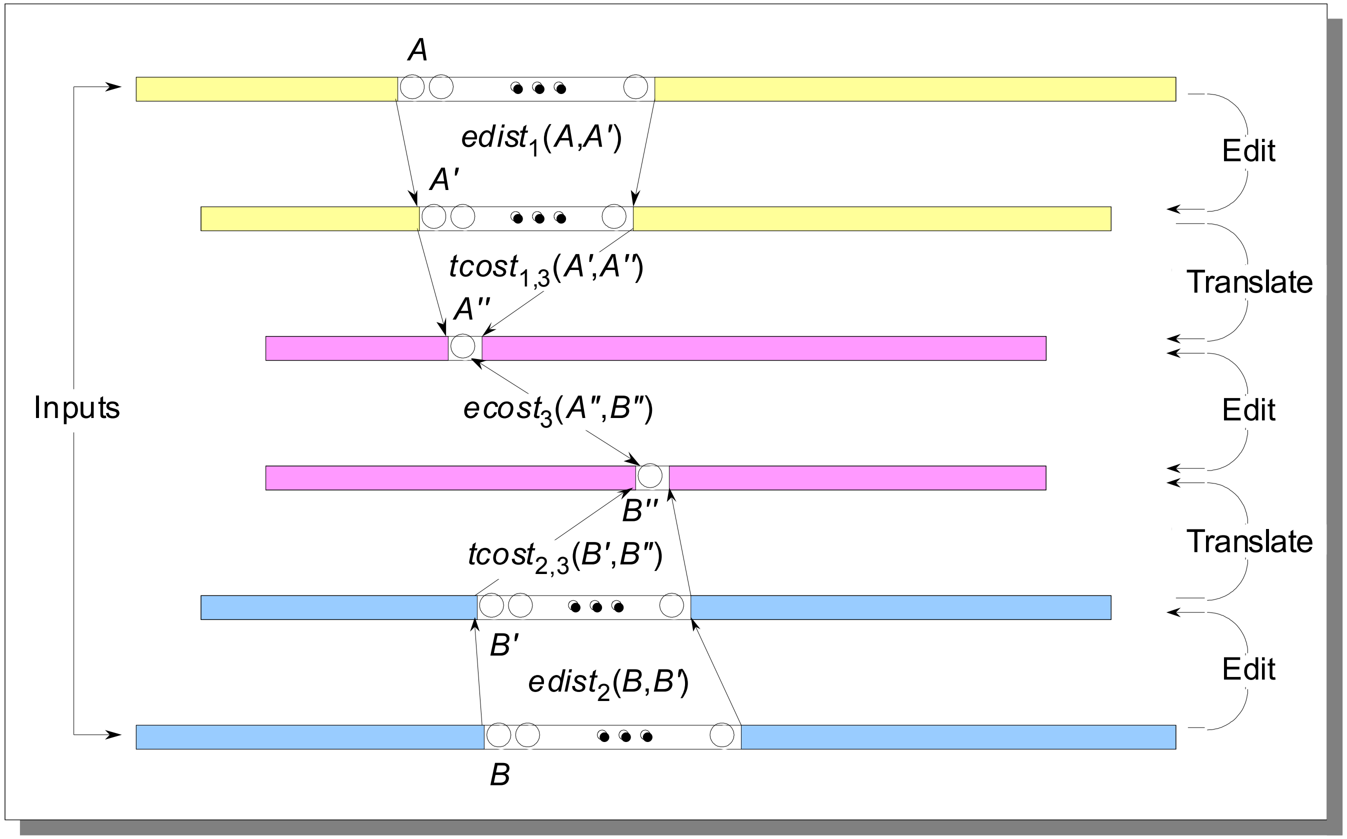
Optimization chooses to edit protein sequences in third domain.

Instead of changing *G* to *A*, change *G* to *M.*

PostScript output from C program

Important point to keep in mind

In construction $xcost$, we depended on the fact that all translations, e.g., $A' \rightarrow A''$, terminated in at most one symbol:



This is true for mRNA→protein translations (the Genetic Code is a 3:1 mapping).

When not true, problem is NP-complete (proof unpublished).

Nasty sequences

So far, we have treated our sequences as though they were (relatively) well-behaved.

There are, however, at least two phenomena that arise to make our work much more challenging:

- reversals (or inversions),
- repeats.

These issues will arise again several times later in the course, but it makes sense to introduce them now as they are basic concepts that have a broad impact.

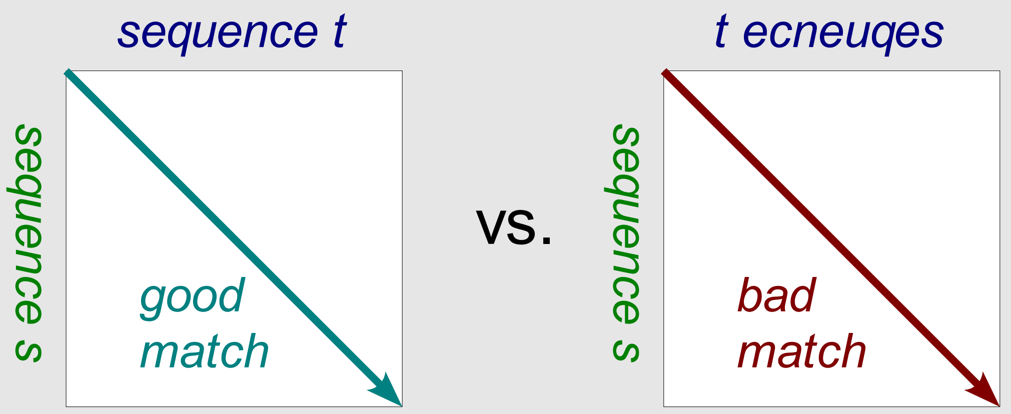
Reversals

As we know, DNA is read directly or as reverse complement:

original sequence	$\longrightarrow$	**ACGTATG**
reverse	$\longrightarrow$	**GTATGCA**
reverse complement	$\longrightarrow$	**CATACGT**

If we're not careful, we might have one sequence backwards:

Run a global sequence alignment algorithm:



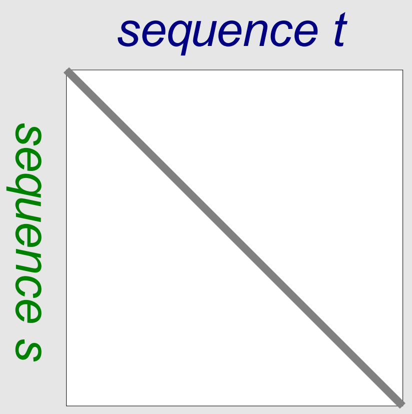
But this is easy enough to fix – just do both directions.

Reversals

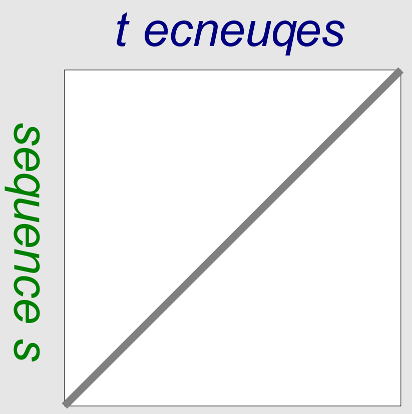
A *dot plot* looks a bit like a dynamic programming matrix, but instead of computing optimal similarity values, we place a “dot” at each cell (*i*,*j*) where *s*[*i*] matches *t*[*j*].

Note: dot plots are intended for visual inspection.

Build a dot plot:



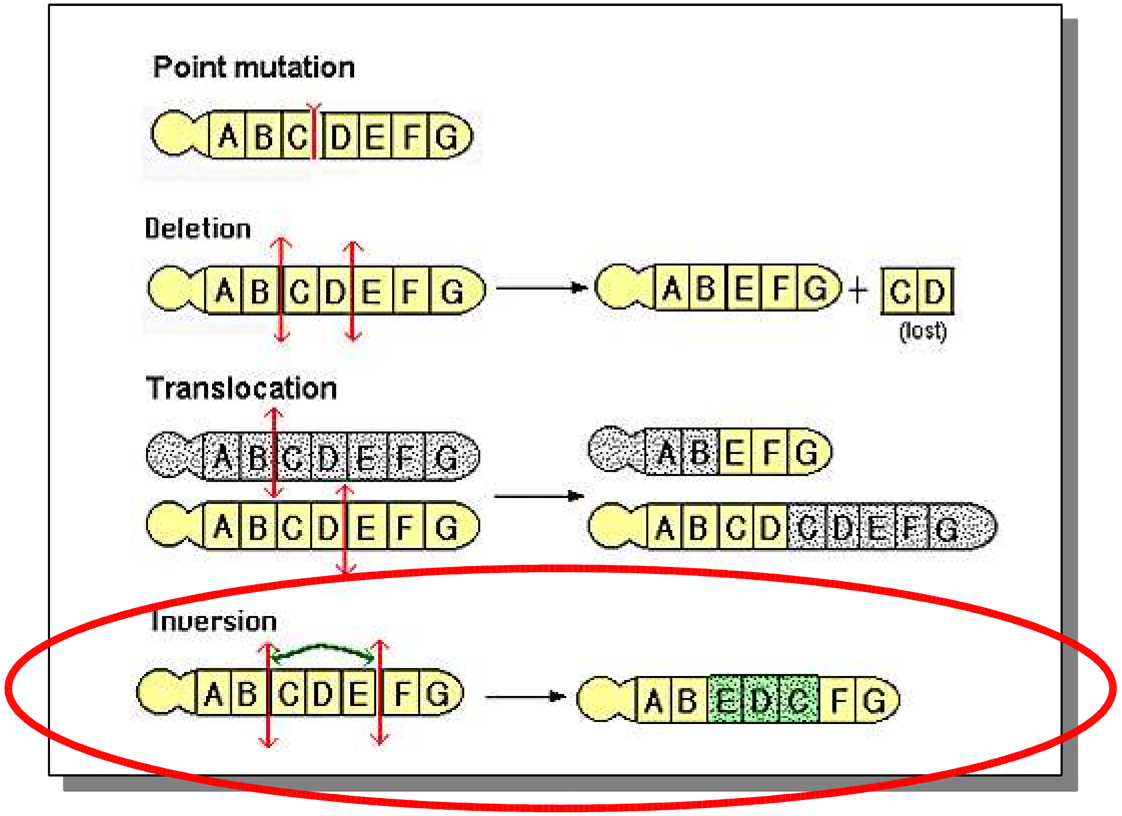
vs.



Hmmm ... that's useful – let's keep it in mind.

Reversals

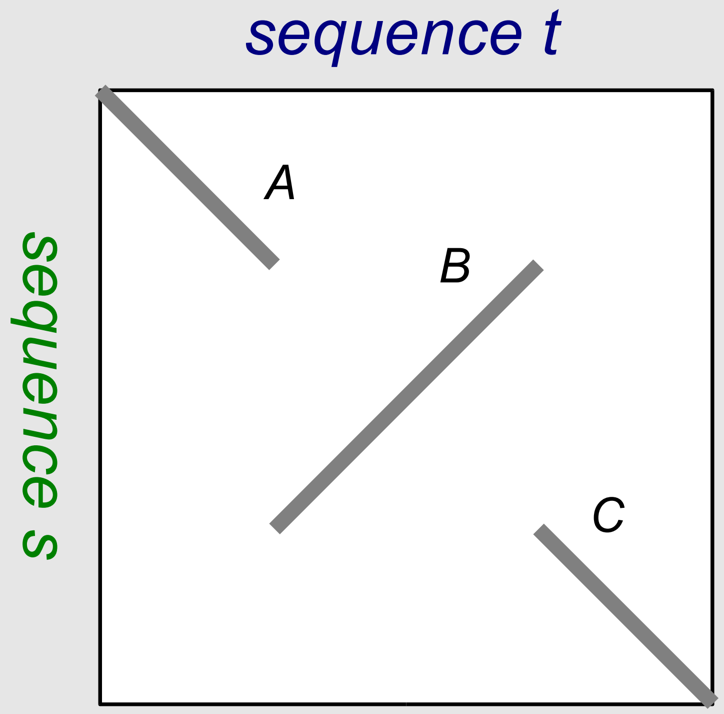
Reversals cause a problem not when the whole sequence is involved, but when an evolutionary event causes a reversal within a sequence. Recall this figure from an earlier lecture:



http://www.accessexcellence.org/AB/GG/mutation.html

Reversals

A dot plot might look something like this then:



In case of dynamic programming, what do we end up with?

Two subsequences that match well (*A* and *C*), and one region that doesn't (*B*) unless we recognize that it is reversed.

Repeats

Repeats are another problem. We would like to believe that the only reason regions look similar is due to homology, and that unrelated regions are random. This is false, unfortunately.

A *tandem repeat* in DNA is two or more adjacent, approximate copies of a pattern of nucleotides. Some (real) examples:

Period = 1

CGAGACTCCGTCTCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACCGCTGACCAGAGATTGCTC

Period = 2

ATGACAACTACCAAAGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGTGGATACCTTTC

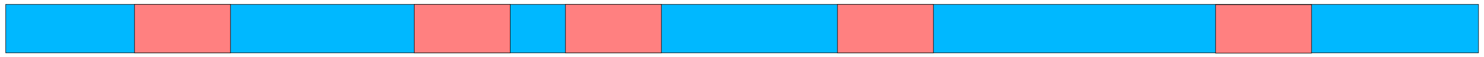
Period = 5

TCACCCCATGTGGGTTTTGTTTTGTTTTGTTTTGTTTTGTTTTGTTTTGTTTTAGGAGAAGAGAA

http://c3.biomath.mssm.edu/trf.html

Repeats

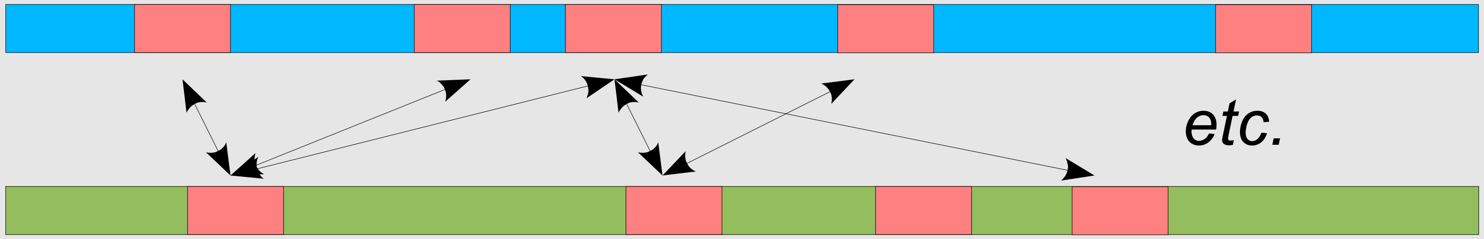
So what's the problem?



🟦 = *stuff we care about*

🟥 = *repeats – misc. "junk" we don't care about*

What are the implications for methods like what BLAST does?



Lots of apparent matches that just serve to confuse us.

Some repeats are useful in forensics

Polymorphisms are variations in DNA sequence between individuals.

Short tandem repeats (STRs) are short sequences of DNA, normally of length 2-5 base pairs, that are repeated numerous times in a head-tail manner. The polymorphisms in STRs are due to the different number of copies of the repeat element that can occur in a population of individuals.

D7S280, found on human chromosome 7, has a tetrameric repeat sequence "gata". Different alleles of this locus have from 6 to 15 tandem repeats of the "gata" sequence:

```
 61 tattttaagg ttaatatata taaagggtat gatagaacac ttgtcatagt ttagaacgaa
121 ctaacgatag atagatagat agatagatag atagatagat agatagatag atagacagat
181 tgatagtttt tttttatctc actaaatagt ctatagtaaa catttaatta ccaatatttg
```

http://www.biology.arizona.edu/human_bio/activities/blackett2/str_description.html

Wrap-up

Readings for next time: see Blackboard.

Remember:

- Come to class prepared to discuss what you have read.
- Check Blackboard regularly for updates.