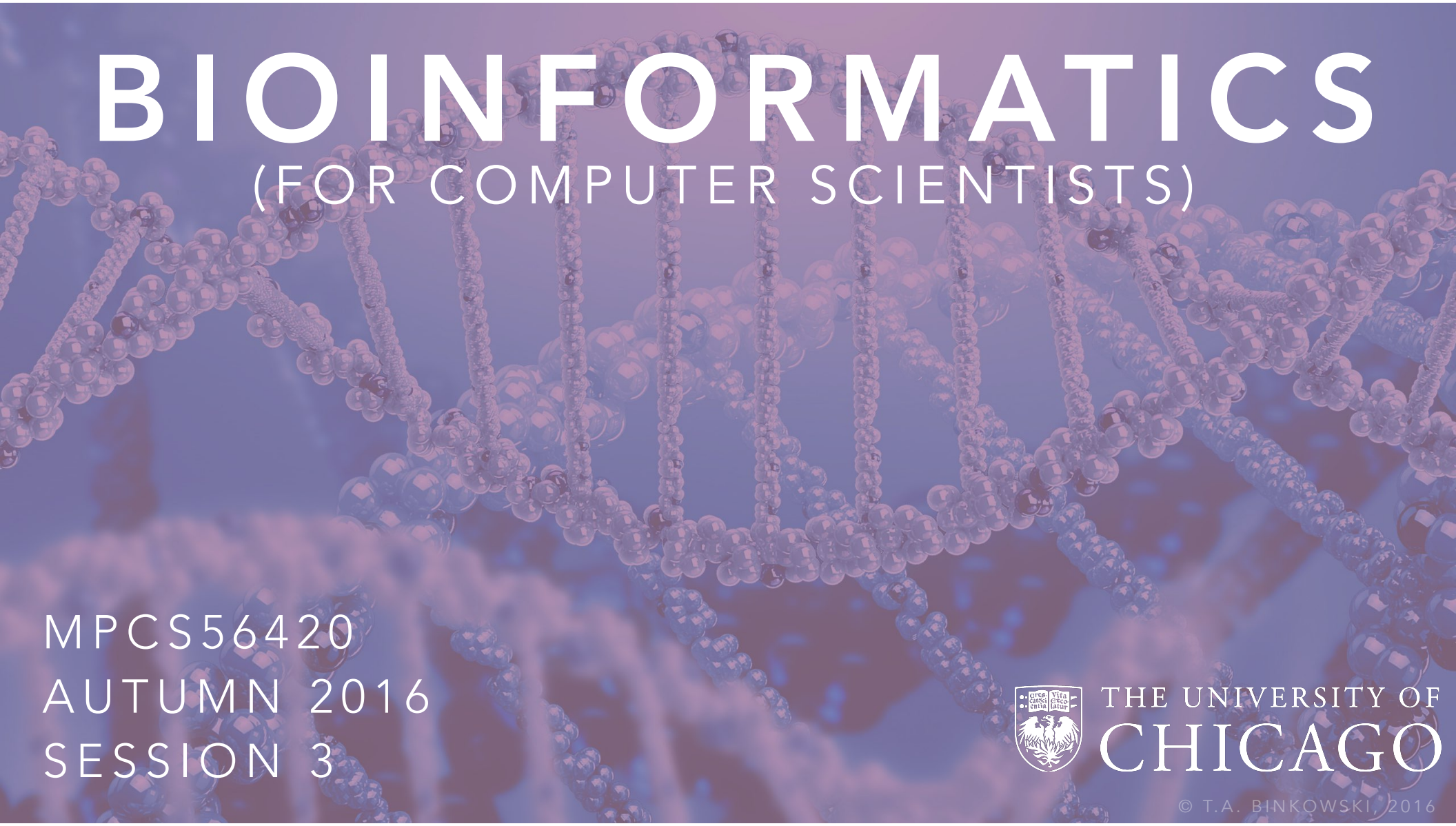


BIOINFORMATICS

(FOR COMPUTER SCIENTISTS)

MPCS56420
AUTUMN 2016
SESSION 3

THE UNIVERSITY OF CHICAGO

© T.A. BINKOWSKI, 2016


BIO NEWS

BIO NEWS

- Promising HIV "cure"
- Kick-and-kill


CLASS NEWS

SEQUENCE ALIGNMENT

- Mutations, substitutions and evolutions
- Sequence similarity
- Pairwise sequence alignment
- Sequence alignment algorithms
- Substitution matrices
- Significance of alignments

BIO NEWS

Image: Umberto Salvagnin/Flickr

Big news: Australian High Court rules that the BRCA1 gene can't be patented

YES!

FIONA MACDONALD 7 OCT 2015

Earlier this morning, the Australian High Court ruled that the BRCA1 gene - which is linked to a significant increase in breast and ovarian cancer risk - isn't a "patentable invention". The decision means that a single company will no longer be able to control research on the gene, or receive all the profits from testing for it.

Prior to this, US biotech company Myriad Genetics held the patent for BRCA1 in

CLASS NOTES

- Question #1
  - Nice to meet you all

CLASS NOTES

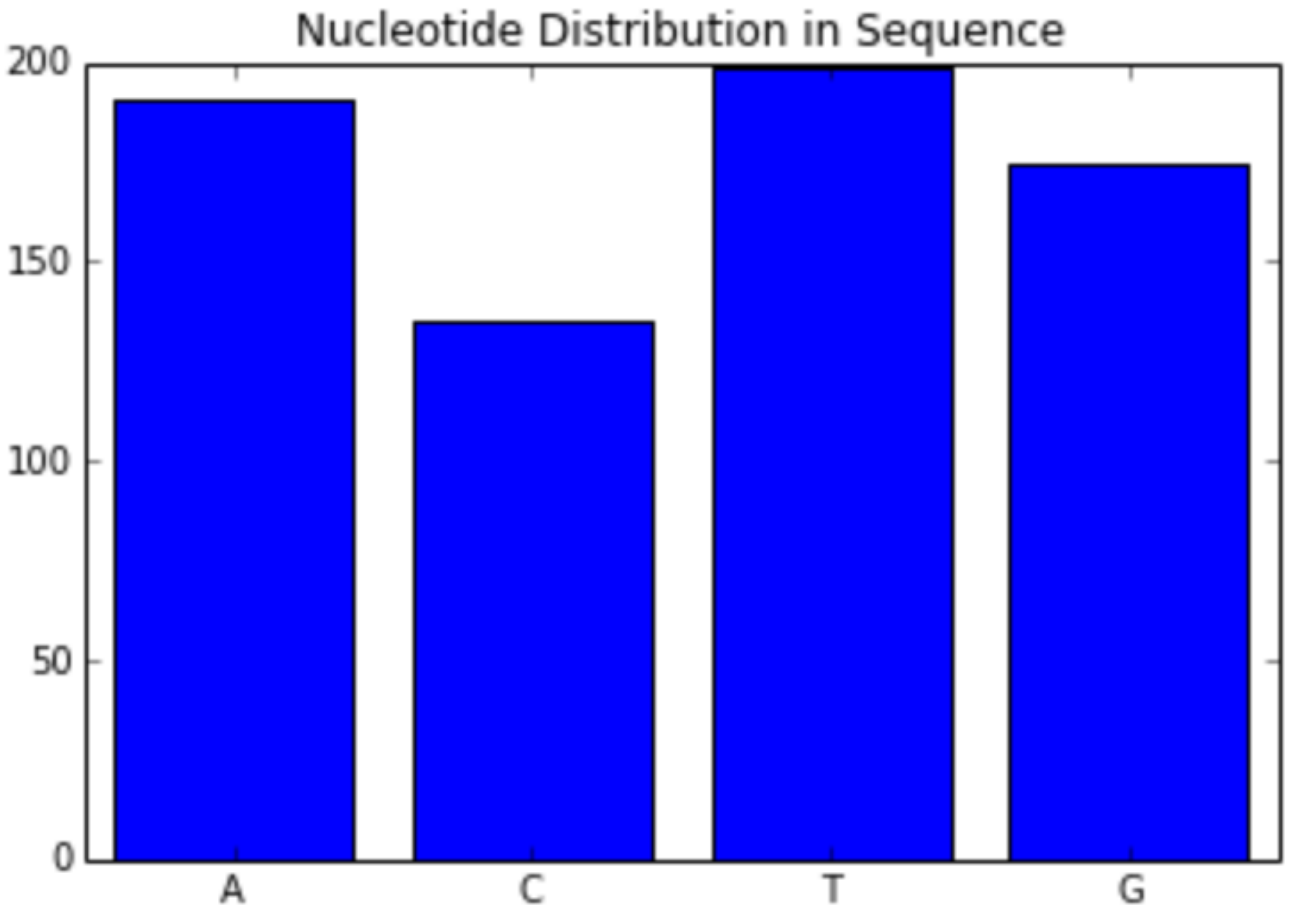
```
- hypothetical protein ARALYDRAFT_487528 [Arabidopsis lyrata subsp. lyrat
a] | MGSTDEVEKSKTRVSCPPALSTSHKILISEEKPRRWSESSLPDVSNRIKLLKFGSASARFKRMAEERD
EVSRSVNSSSSHNFRERISVVFSRKIEWACLMKMGKQWLQNPLNMVLFLWILVVAVSGAILFMVMTGMLNHAL
PKKSQRDVWFEVNNQILNAMFTLMCLYQHPKRFYHLVLLCRWRQDDVTTLRKIYCENGTYKPNEWIHMMVVVL
LLHLNCFAQYALCGLNLGYRRSERPAIGVAICISIAIAAPASAGLYTILSPLGKDYDPQGDEENQVEPVEEGS
VTNHKLSLERRYSFASADVSNPEWRGGVLDIWEDISLAYLSLFCTFCVFGWNMERIGFGNMYVHIATFILFCL
APFFIFNLAAINIDNEMVREALGYTGIVLCLFGLLYGGFWRIQMRKRFKLPGYNFCCGRPAIADCTLWLFCCW
CSLAQEVRTANSYEIVEDKFCKRSEENSKIDDEVVVSSLPRDDGVFDPSCSPKKMTTAIASSSLSPSRQKYET
CLGDKSDEALSPPSPPFIHRS

ERROR: Invalid FASTA format. "!" is not a legal character
ERROR: Invalid FASTA format. "!" is not a legal character
```

- Question #2

CLASS NOTES

- Question #3



CLASS NOTES

- Question #4
  - CG Content
- Question #5
  - http://web.expasy.org/translate/

5'3' Frame 1

VRRSVIRWEIGYQMKLWIRCKSRQISLKS-VIMFN-RSKAETTLDSVLFMEKAHLRFPYR
PTNRFFIALAAERAAMFSLF-GRWKAILLPSRFLTLLTNTKLIFQMI-QSIPEPGQSLLE
NKKWLRHMSS-RNFTIIC--IQKKVKRHWIICFLGALRKS-LMNFRLAMLLILGTLSRNS
L-RGDLVRRKWKKRVS-SDAKTEADISTASETVSCFRSMIITGLLLLSQAGLL

5'3' Frame 2

YDGVL-DGKSDTR-NCGSGAKVGRYR-SHR-LCSIKEARPKLLWTLSFSWRKHTFVFRIA
RQTDFSLLWLRSGRQCFLFFKADGRLFFCRVGFSPC-QIPN-FSR-YNSPFRSPARVFWR
TKNG-GT-APEEILPSFVNKYKRRSRGTGLSAF-GLYERAD--ISDWLCS-FLGLYHEIP
CKEGI--GANGKSGSPDQTRRRKRIFRPLQKPCHVSDP-SSRGCCCFLRQGSW

5'3' Frame 3

TTECYKMGNRIPDEIVDQVQKSADIVEVIGDYVQLKKQGRNYFGLCPFHGESTPSFSVSP
DKQIFHCFGCGAGGNVFSFLRQMEGYSFAESVSHLADKYQIDFPDDITVHSGARPESSGE
QKMAEAHELLKKFYHHLLINTKEGQEALDYLLSRGFTKELINEFQIGYALDSWDFITKFL
VKRGFSEAQMEKAGLLIRREDGSGYFDRFRNRVMFPIHDHHGAVVAFSGRAL

3'5' Frame 1

PRALPEKATTAP--SWIGNMTRFLKRSKYPLPSSRLIRRPAFSICASLNPLFTRNFVIKS
QESRA-PI-NSLISSFVKPLESR-SSAS-PSFVFINK-W-NFFRSSCASAIFCSPEDSGR
APEWTVISSGKSIWYLSAR-ETDSAKE-PSICLKKEKTLPPAPQPKQ-KICLSGDTENEG
VLSP-KGQSPK-FRPCFFN-T-SPMTSTISADFCT-STISSGIRFPIL-HSVV

3'5' Frame 2

QEPCLRKQQQPRDDHGSET-HGF-SGRNIRFRLRV-SGDPLFPFAPH-IPSLQGIS--SP
KNQEHSQSEIH-SALS-SP-KADNPVPLDLLLYLLTNDGKISSGAHVPQPFFVLQKTLAG
LRNGLLYHLENQFGICQQGEKPTRQKNSLPSALKKRKHCRPLRSQSNEKSVCRAIRKTKV
CFLHEKDRVQSSFGLASLIEHNHL-LQRYLPTFAPDPQFHLVSDFPSYNTPSY

3'5' Frame 3

KSPA-ESNNSPVMIMDRKHDTVSEAVEISASVFASDQETRFFHLRLTKSPLYKEFRDKVP
RIKSIANLKFINQLFRKAPRKQIIQCLLTFFCIY-QMMVKFLQELMCLSHFLFSRRLWPG
SGMDCYIIWKINLVFVSKVRNRLGKRIAFHLP-KRENIAARSAAKAMKNLFVGRYGKRRC
AFSMKRTESKVVSALLL-LNIITYDFNDICRLLHLIHNFIWYPISHLITLRR

CLASS NOTES

- Question #6
  - There are 184, 190, and 64 mutations in positions 1, 2, and 3 respectively
  - #2 is most sensitive
  - It is about 3x more sensitive than the third position.

```
codon = {'ttt':'F', 'tct':'S', 'tat':'Y', 'tgt':'C', 'ttc':'F', 'tcc':'
         'tgc':'C', 'tta':'L', 'tca':'S', 'taa':'-', 'tga':'-', 'ttg':'
         'tag':'-', 'tgg':'W', 'ctt':'L', 'cct':'P', 'cat':'H', 'cgt':'
         'ccc':'P', 'cac':'H', 'cgc':'R', 'cta':'L', 'cca':'P', 'caa':'
         'ctg':'L', 'ccg':'P', 'cag':'Q', 'cgg':'R', 'att':'I', 'act':'
         'agt':'S', 'atc':'I', 'acc':'T', 'aac':'N', 'agc':'S', 'ata':'
         'aaa':'K', 'aga':'R', 'atg':'M', 'acg':'T', 'aag':'K', 'agg':'
         'gct':'A', 'gat':'D', 'ggt':'G', 'gtc':'V', 'gcc':'A', 'gac':'
         'gta':'V', 'gca':'A', 'gaa':'E', 'gga':'G', 'gtg':'V', 'gcg':'
         'ggg':'G'}

# Possible mutations for each base (we are not counting same to same mu
mutation = {'a': 'cgt', 'c': 'agt', 'g': 'act', 't': 'acg'}

def mutateAndCount(pos, type):
    total = 0
    count = 0
    for code, amino in codon.iteritems():
        amino = codon[code]
        nucleotide = code[pos]
        for mutant in type[nucleotide]:
            total += 1
            newCode = ''.join([mutant if (i == pos) else code[i] for i
            #print code+"->"+mutant+"->"+newCode
            if (amino != codon[newCode]):
                count += 1
    return count, total

print "Possible Amino Acid Changes from Mutation"
print "----------------------------------------"
for pos in xrange(3):
    possible, total = mutateAndCount(pos, mutation)
    print "Position %d has %d possible changes from %d mutations" % (po
```


MUTATIONS,
SUBSTITUTIONS
& EVOLUTION

MUTATIONS

- All living organisms are related to each other through evolution
  - Any pair of organisms have a common ancestor from which they evolved
- Evolution involves
  - Inheritance
    - Passing of characteristics from parent to offspring
  - Variation
    - Differentiation between parent and offspring
  - Selection
    - Favoring some organisms (or features) over others

MUTATIONS

- Mutation
  - Change of nucleotide or amino acid at a given position
  - Driven by error, evolution or environmental factors
- Selection
  - Favoring of traits
  - Significant divergence from ancestral sequences

MUTATIONS

TYPES OF DNA MUTATIONS

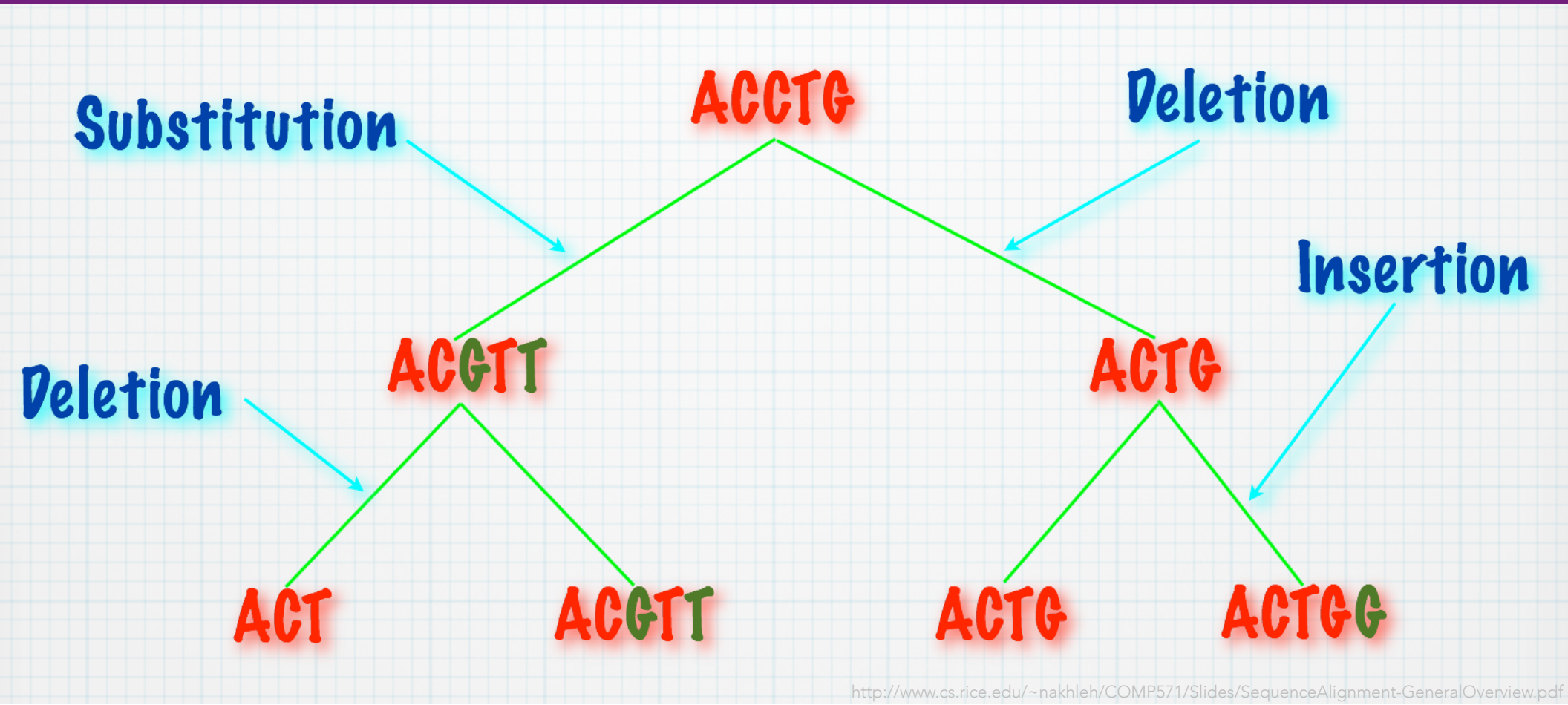
- Substitution
- Insertion
- Deletion
- Frameshift

First Position 5′	Second Position				Third Position 3′
	U	C	A	G	
U	UUU F UUC F UUA L UUG L	UCU S UCC S UCA S UCG S	UAU Y UAC Y UAA stop UAG stop	UGU C UGC C UGA stop UGG W	U C A G
C	CUU L CUC L CUA L CUG L	CCU P CCC P CCA P CCG P	CAU H CAC H CAA Q CAG Q	CGU R CGC R CGA R CGG R	U C A G
A	AUU I AUC I AUA I AUG M	ACU T ACC T ACA T ACG T	AAU N AAC N AAA K AAG K	AGU S AGC S AGA R AGG R	U C A G
G	GUU V GUC V GUA V GUG V	GCU A GCC A GCA A GCG A	GAU D GAC D GAA E GAG E	GGU G GGC G GGA G GGG G	U C A G

http://www.nature.com/scitable/topicpage/genetic-mutation-441

MUTATIONS

TYPES OF DNA MUTATIONS



http://www.cs.rice.edu/~nakhleh/COMP571/Slides/SequenceAlignment-GeneralOverview.pdf

MUTATIONS

DNA SUBSTITUTION TYPES

- Non-synonymous mutation
  - amino acid sequence is changed
- Synonymous (silent) mutation
  - do not change an amino acid sequence
- Nonsense mutation
  - Base changes to “stop” codon

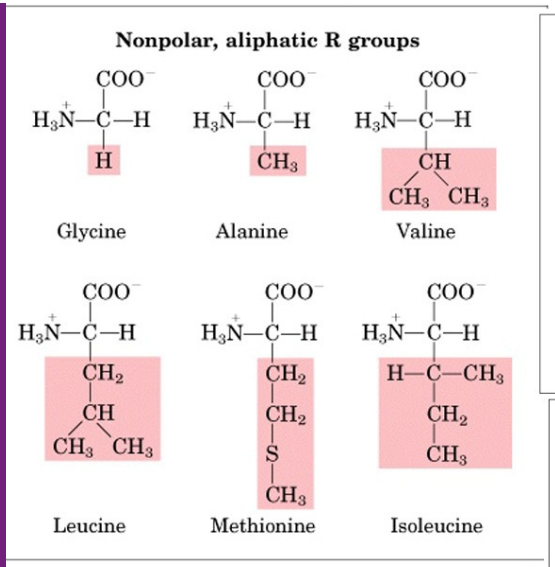
First Position 5′	Second Position				Third Position 3′
	U	C	A	G	
U	UUU F UUC F UUA L UUG L	UCU S UCC S UCA S UCG S	UAU Y UAC Y UAA stop UAG stop	UGU C UGC C UGA stop UGG W	U C A G
C	CUU L CUC L CUA L CUG L	CCU P CCC P CCA P CCG P	CAU H CAC H CAA Q CAG Q	CGU R CGC R CGA R CGG R	U C A G
A	AUU I AUC I AUA I AUG M	ACU T ACC T ACA T ACG T	AAU N AAC N AAA K AAG K	AGU S AGC S AGA R AGG R	U C A G
G	GUU V GUC V GUA V GUG V	GCU A GCC A GCA A GCG A	GAU D GAC D GAA E GAG E	GGU G GGC G GGA G GGG G	U C A G

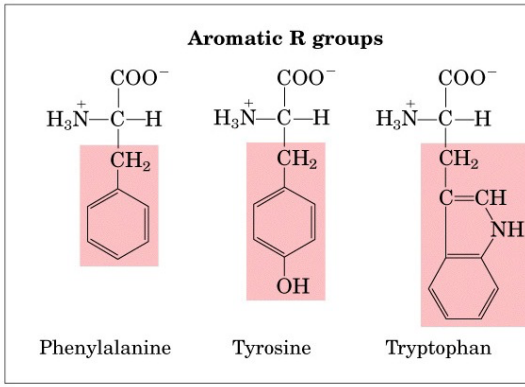
http://www.nature.com/scitable/topicpage/genetic-mutation-441

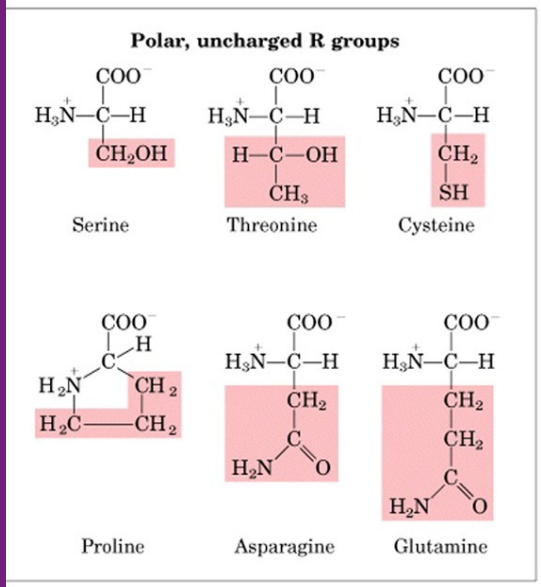
MUTATIONS

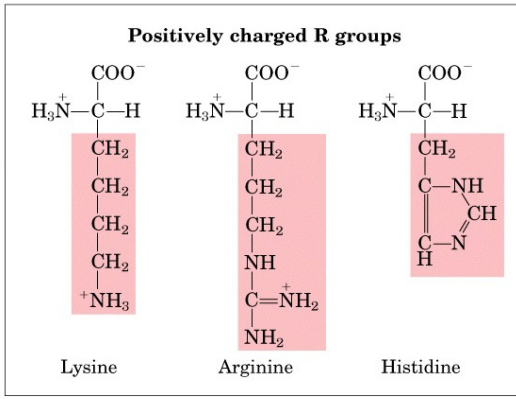
AMINO ACID SUBSTITUTIONS

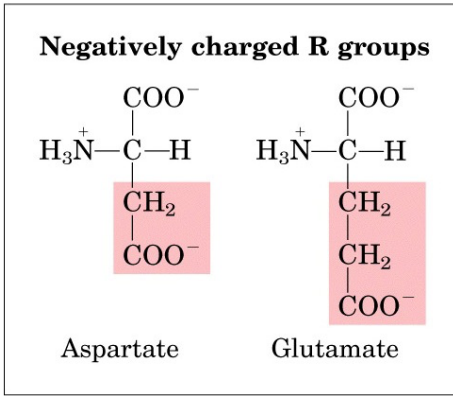
- Amino acids differ by their side chains (R-groups)
- Grouped by properties
  - Hydrophobic or non polar
    - Side chains that repel water
  - Hydrophilic or polar
    - Side chains that are attracted to water
  - Acidic or negatively-charged
    - Contain carboxyl groups (-COO-) as side chains
  - Basic or positively-charged side chains
    - Contain amine groups (-NH3+) as side chains







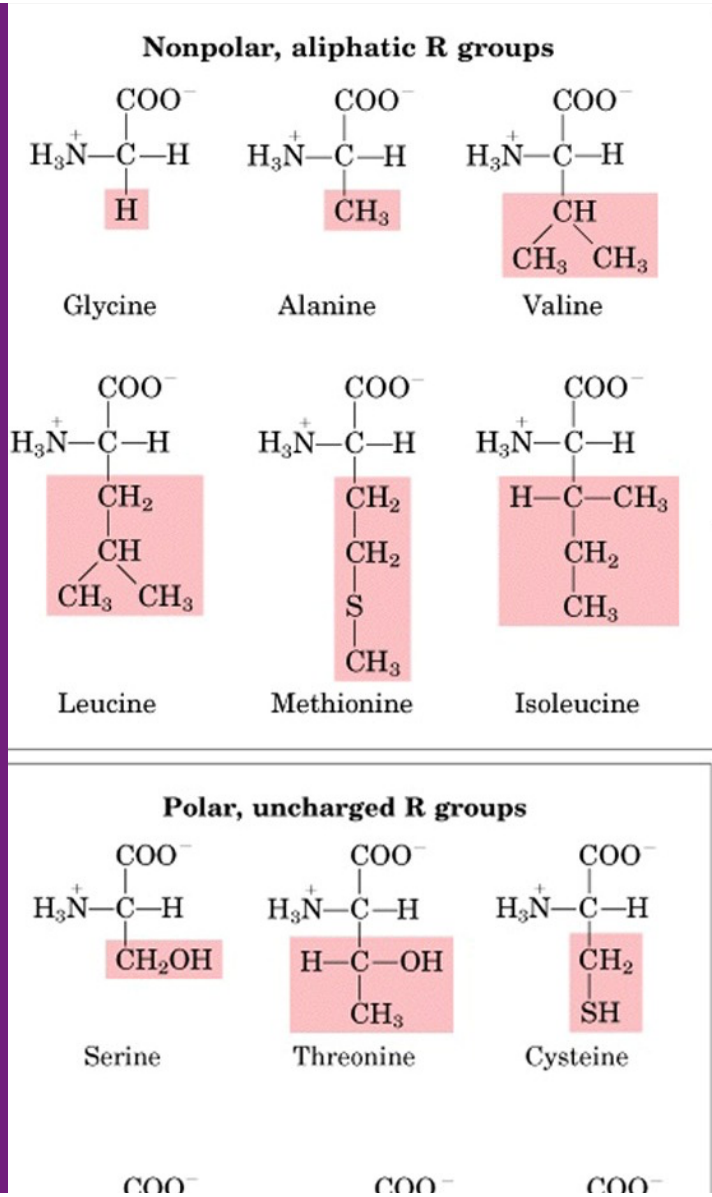




MUTATIONS

AMINO ACID SUBSTITUTIONS

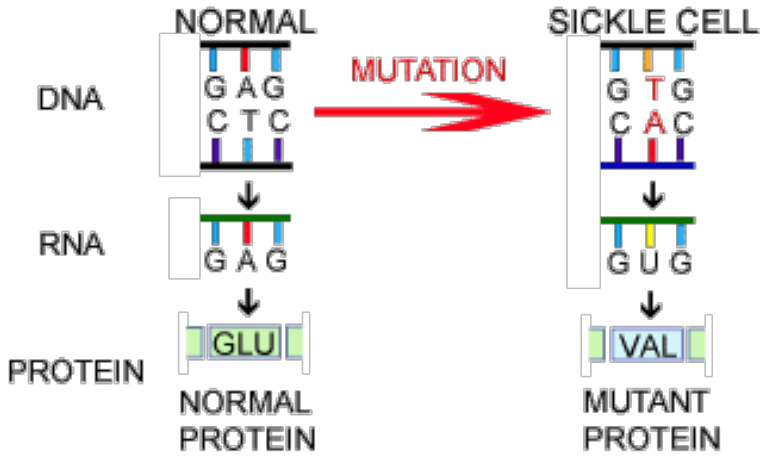
- Mutations in protein sequence have varying biochemical effects
- Non-conserved
  - Results in different side chain
  - Side chains may be in same functional group
  - Side chains may be different functional group
- Conserved
  - Results in same side chain

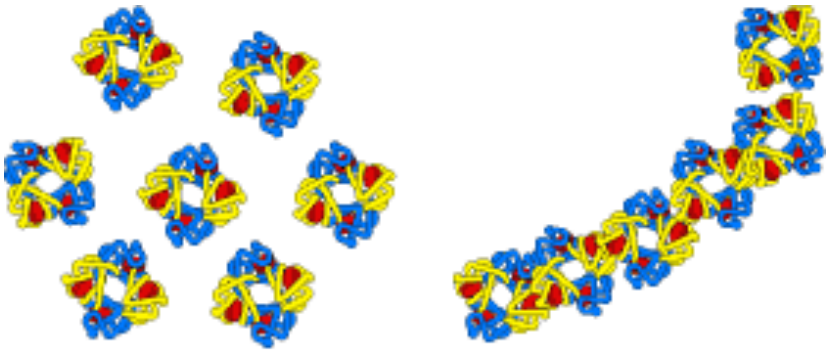


MUTATIONS

SICKLE CELL ANEMIA

- Mutation in hemoglobin
  - Carries oxygen in red blood cells
- Deprived of oxygen cells become sickle-shaped
- Carrier experiences pain and fatigue





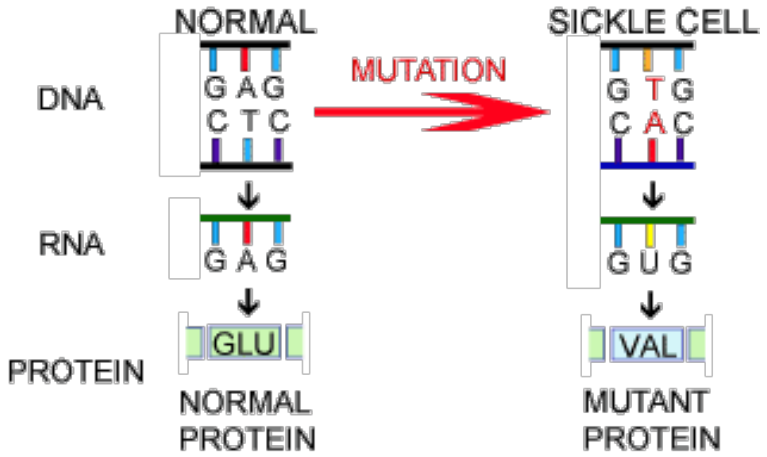
NORMAL HEMOGLOBIN

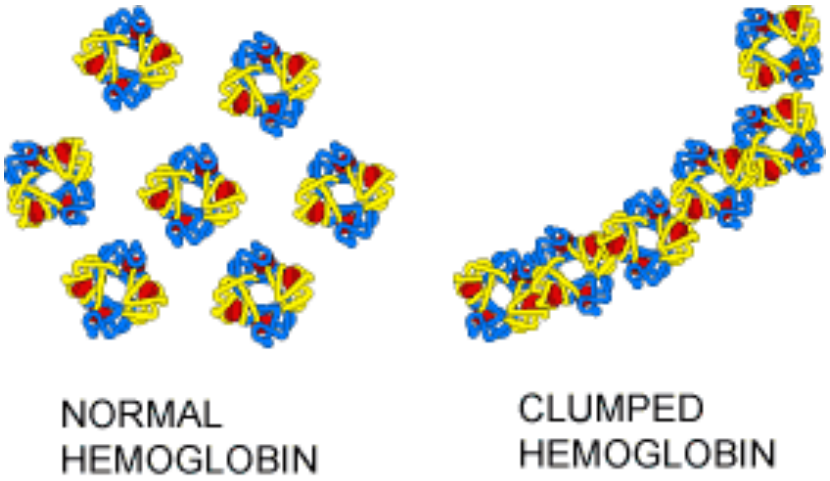
CLUMPED HEMOGLOBIN

MUTATIONS

SICKLE CELL ANEMIA

- Carrier are resistant to malaria
  - Parasites are killed in blood cells

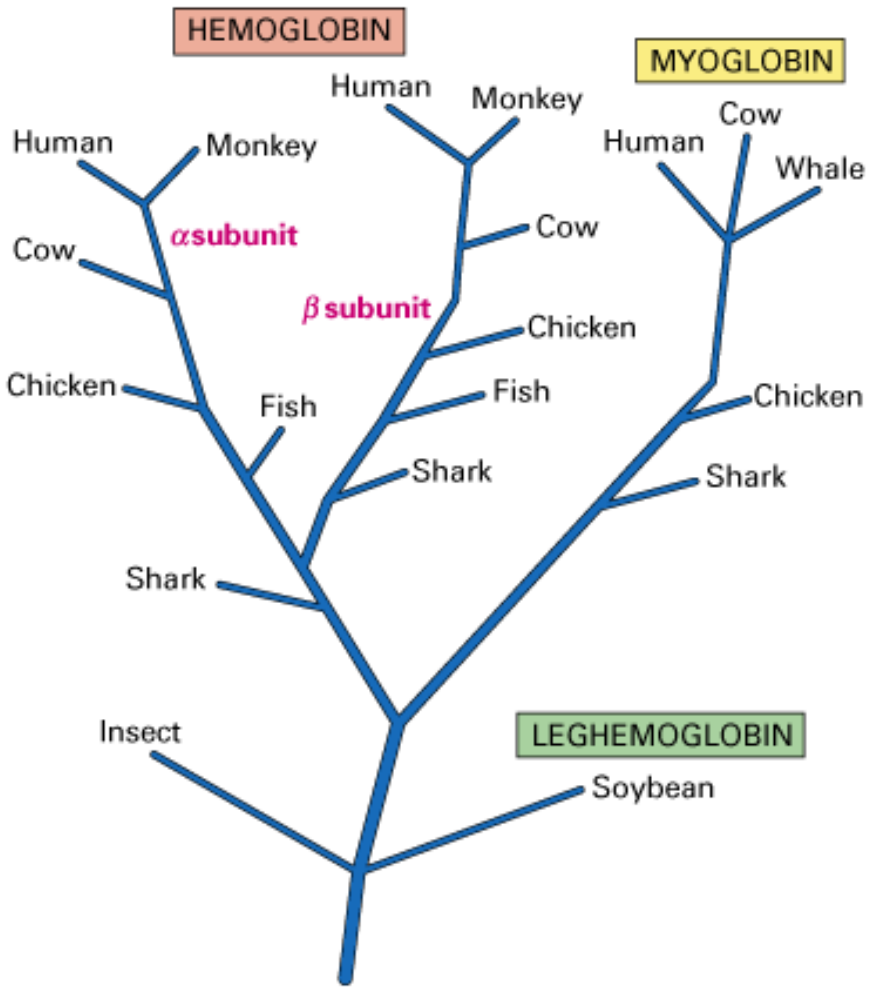





EVOLUTIONARY
RELATEDNESS

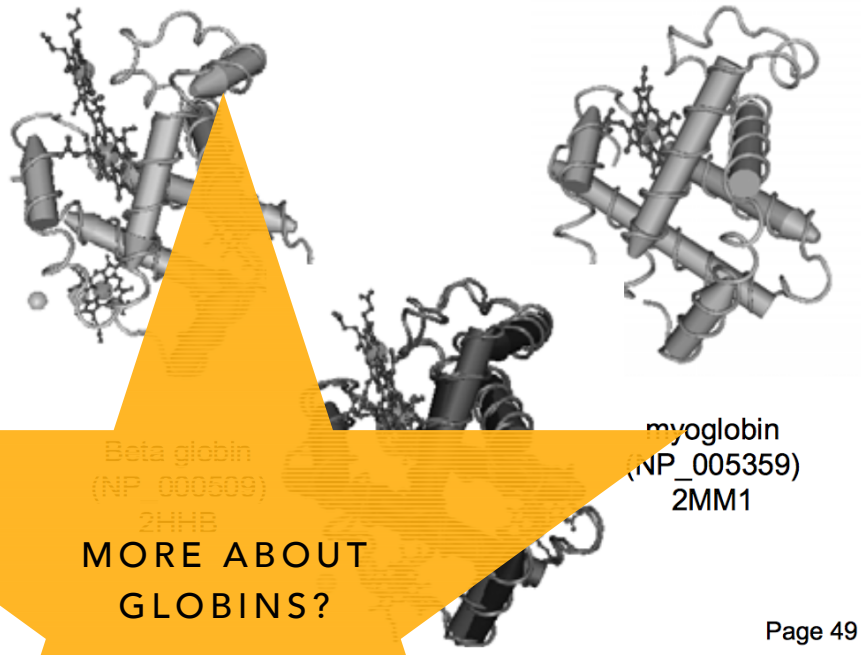
EVOLUTIONARY RELATEDNESS

- Group of evolutionarily related proteins/genes are grouped in to “families”
- Typically share
  - Function
  - Structure
  - Sequence

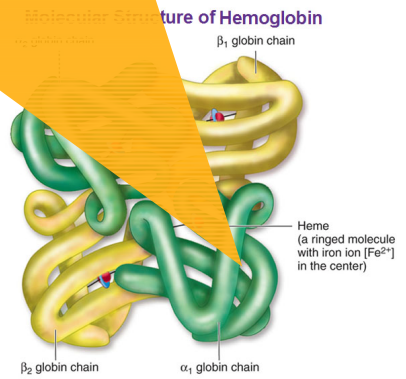


EVOLUTIONARY RELATEDNESS

- Globin family of proteins
  - Heme-containing proteins involved in binding, transporting oxygen
  - Distantly, but significantly related
- Human myoglobin and beta global
  - Diverged 600 million years ago

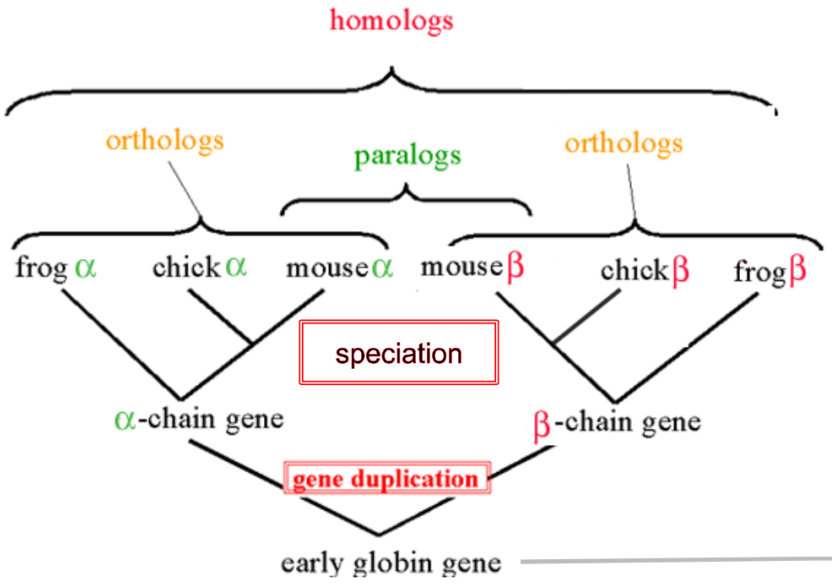


MORE ABOUT GLOBINS?



EVOLUTIONARY RELATEDNESS

- Evolution
  - New genes are generated from preexisting genes
- Intragenic mutation
  - Modified by errors in DNA replication

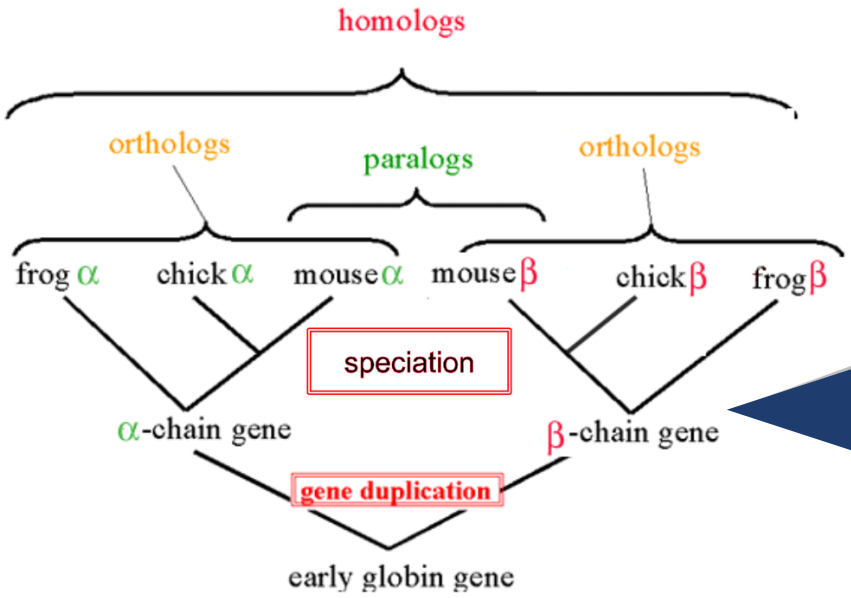


Just one copy of globin in ancient organism

from http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Orthology.html

EVOLUTIONARY RELATEDNESS

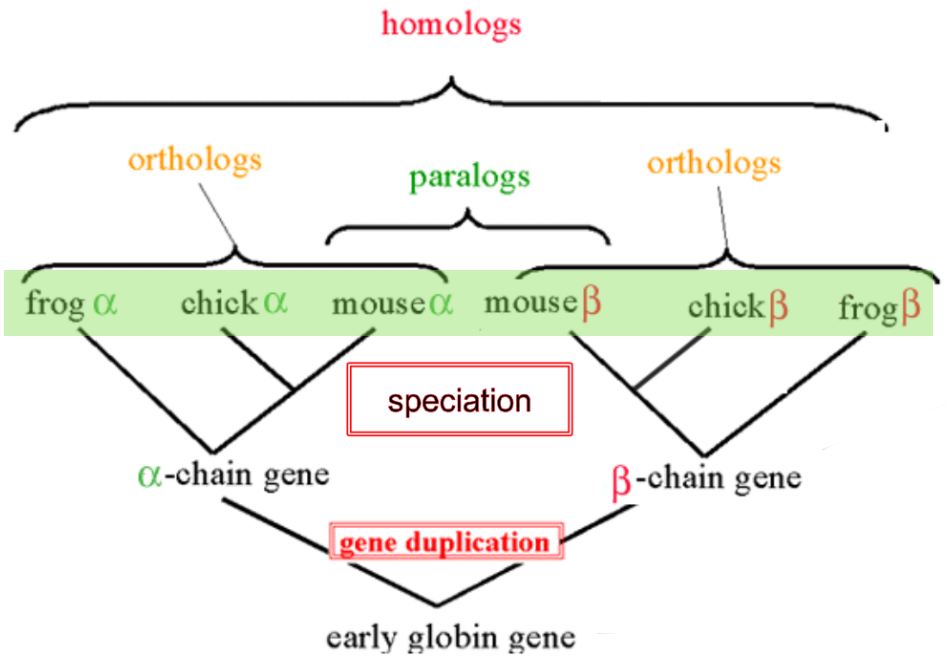
- Gene duplication
  - Two copies of genes may diverge in the course of evolution



GENE DUPLICATION RESULTS IN TWO COPIES IN A COMMON ANCESTOR OF FROG, CHICK, AND MOUSE

EVOLUTIONARY RELATEDNESS

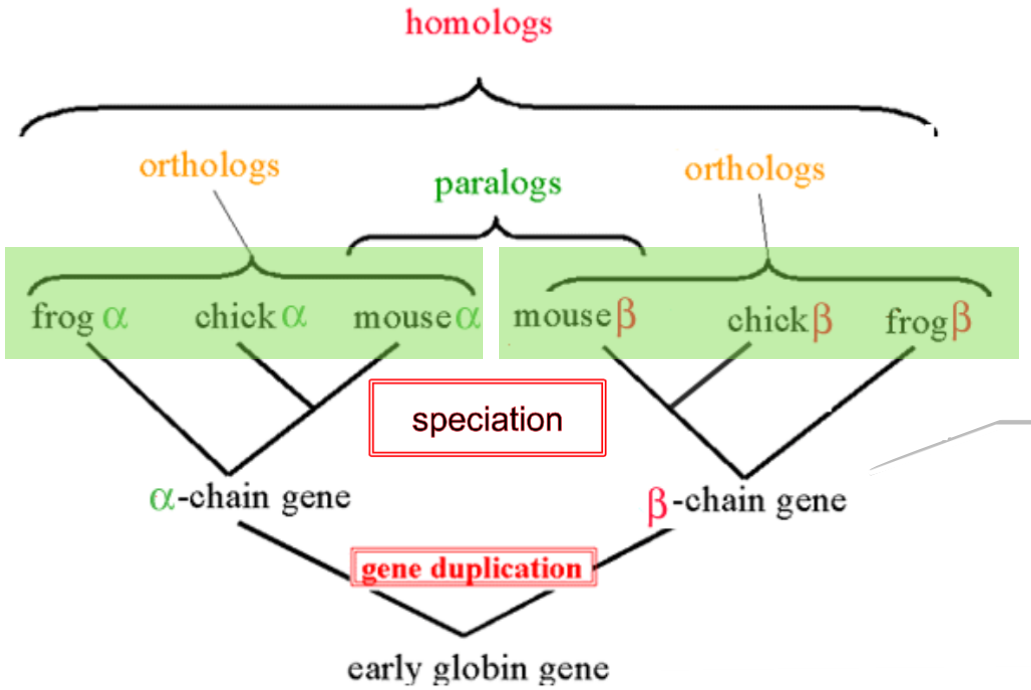
- Homolog
  - A gene related to a second gene by descent from a common ancestral DNA sequence



from http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Orthology.html

EVOLUTIONARY RELATEDNESS

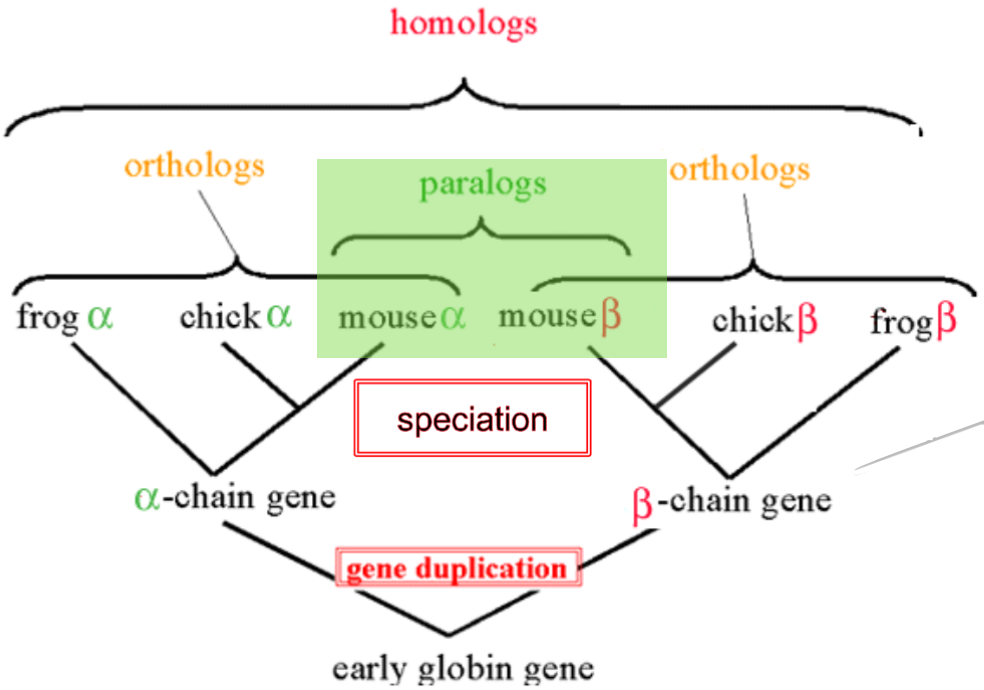
- Ortholog
  - Genes in different species that evolved from a common ancestral gene by speciation
  - Same function



from http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Orthology.html

EVOLUTIONARY RELATEDNESS

- Paralog
  - Genes related by duplication within a genome
  - Different function



from http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Orthology.html


SEQUENCE
SIMILARITY

SEQUENCE SIMILARITY

- Fundamental problem in bioinformatics is to determine if sequences are similar
- The most cited paper in modern biology
  - BLAST by S. Altschul,1990
  - h-index 45

Basic Local Alignment Search Tool

Stephen F. Altschul[1], Warren Gish[1], Webb Miller[2]
Eugene W. Myers[3] and David J. Lipman[1]

[1]*National Center for Biotechnology Information*
National Library of Medicine, National Institutes of Health
Bethesda, MD 20894, U.S.A.

[2]*Department of Computer Science*
The Pennsylvania State University, University Park, PA 16802, U.S.A.

[3]*Department of Computer Science*
University of Arizona, Tucson, AZ 85721, U.S.A.

(Received 26 February 1990; accepted 15 May 1990)

A new approach to rapid sequence comparison, basic local alignment search tool (BLAST), directly approximates alignments that optimize a measure of local similarity, the maximal segment pair (MSP) score. Recent mathematical results on the stochastic properties of MSP scores allow an analysis of the performance of this method as well as the statistical significance of alignments it generates. The basic algorithm is simple and robust; it can be implemented in a number of ways and applied in a variety of contexts including straightforward DNA and protein sequence database searches, motif searches, gene identification searches, and in the analysis of multiple regions of similarity in long DNA sequences. In addition to its flexibility and tractability to mathematical analysis, BLAST is an order of magnitude faster than existing sequence comparison tools of comparable sensitivity.

1. Introduction

The discovery of sequence homology to a known protein or family of proteins often provides the first clues about the function of a newly sequenced gene. As the DNA and amino acid sequence databases continue to grow in size they become increasingly useful in the analysis of newly sequenced genes and proteins because of the greater chance of finding such homologies. There are a number of software tools for searching sequence databases but all use some measure of similarity between sequences to distinguish biologically significant relationships from chance similarities. Perhaps the best studied measures are those used in conjunction with variations of the dynamic programming algorithm (Needleman & Wunsch, 1970; Sellers, 1974; Sankoff & Kruskal, 1983; Waterman, 1984). These methods assign scores to insertions, deletions and replacements, and compute an alignment of two sequences that corresponds to the least costly set of such mutations. Such an alignment may be thought of as minimizing the evolutionary distance or maximizing optimal, based on the given scores. Because of their computational requirements, dynamic programming algorithms are impractical for searching large databases without the use of a supercomputer (Gotoh & Tagashira, 1986) or other special purpose hardware (Coulson *et al.*, 1987).

Rapid heuristic algorithms that attempt to approximate the above methods have been developed (Waterman, 1984), allowing large databases to be searched on commonly available computers. In many heuristic methods the measure of similarity is not explicitly defined as a minimal cost set of mutations, but instead is implicit in the algorithm itself. For example, the FASTP program (Lipman & Pearson, 1985; Pearson & Lipman, 1988) first finds locally similar regions between two sequences based on identities but not gaps, and then rescores these regions using a measure of similarity between residues, such as a PAM matrix (Dayhoff *et al.*, 1978) which allows conservative replacements as well as identities to increment the similarity score. Despite their rather indirect approximation of minimal evolution measures, heuristic tools such as

SEQUENCE SIMILARITY

WHAT CAN SEQUENCE SIMILARITY TELL US?

A

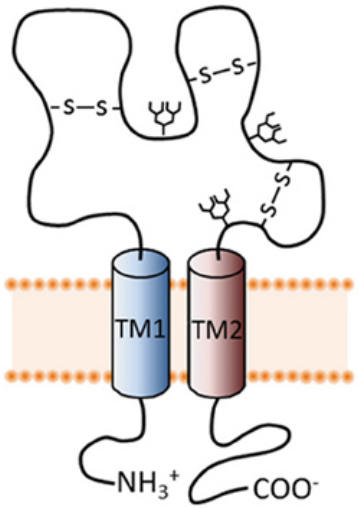
TM1

```
rP2X1R    29 VGVIFRLIQLVVLVYVIGWVFVYEKGYQTSS-DLISSVSVKLKGLAVTQ-------LQGLGPQVWDVADYVFPAHGDSSFVVMT 104
hP2X1R    29 VGVIFRLIQLVVLVYVIGWVFLYEKGYQTSS-GLISSVSVKLKGLAVTQ-------LPGLGPQVWDVADYVFPAQGDNSFVVMT 104
rP2X2R    29 LGFVHRMVQLLILLYFVWYVFIVQKSYQDSETGPESSIITKVKGITMSE-----------DKVWDVEEYVKPPEGGSVVSIIT 100
hP2X2R    41 LGVLYRAVQLLILLYFVWYVFIVQKSYQESETGPESSIITKVKGITTSE-----------HKVWDVEEYVKPPEGGSVFSIIT 112
rP2X3R    23 IGIINRAVQLLIISYFVGWVFLHEKAYQVRDTAIESSVVTKVKGFGRYA-----------NRVMDVSDYVTPPQGTSVFVIIT  94
hP2X3R    23 IGIINRVVQLLIISYFVGWVFLHEKAYQVRDTAIESSVVTKVKGSGLYA-----------NRVMDVSDYVTPPQGTSVFVIIT  94
rP2X4R    28 VGLMNRAVQLLILAYVIGWVFVWEKGYQETD-SVVSSVTTKAKGVAVTN-------TSQLGFRIWDVADYVIPAQEENSLFIMT 103
hP2X4R    28 VGLMNRAVQLLILAYVIGWVFVWEKGYQETD-SVVSSVTTKVKGVAVTN-------TSKLGFRIWDVADYVIPAQEENSLFVMT 103
rP2X5R    29 VGLLYRVLQLIILLYLLIWVFLIKKSYQDIDTSLQSAVVTKVKGVAYTN-------TTMLGERLWDVADFVIPSQGENVFFVVT 105
hP2X5R    29 VGLLYRLLQASILAYLVVWVFLIKKGYQDVDTSLQSAVITKVKGVAFTN-------TSDLGQRIWDVADYVIPAQGENVFFVVT 105
rP2X6R    40 VGISQRLLQLGVVVYVIGWALLAKKGYQEWDMDPQISVITKLKGVSVTQ-------VKELEKRLWDVADFVRPSQGENVFFLVT 116
hP2X6R    38 VGALQRLLQFGIVVYVVGWALLAKKGYQERDLEPQFSIITKLKGVSVTQ-------IKELGNRLWDVADFVKPPQGENVFFLVT 114
rP2X7R    26 YGTIKWILHMTVFSYV-SFALMSDKLYQRKE-PLISSVHTKVKGVAEVTENVTEGGVTKLVHGIFDTADYTLPLQG-NSFFVMT 106
hP2X7R    26 YGTIKWFFHVIIFSYV-CFALVSDKLYQRKE-PVISSVHTKVKGIAEVKEEIVENGVKKLVHSVFDTADYTFPLQG-NSFFVMT 106
```

TM2

```
rP2X1R   323 AGKFDIIPTMTTIGSGIGIFGVATVLCDLLLL 354
hP2X1R   323 AGKFDIIPTMTTIGSGIGIFGVATVLCDLLLL 354
rP2X2R   322 AGKFSLIPTIINLATALTSIGVGSFLCDWILL 353
hP2X2R   333 AGKFSLIPTIINLATALTSVGVGSFLCDWILL 364
rP2X3R   313 AGKFNIIPTIISSVAAFTSVGVGTVLCDIILL 344
hP2X3R   313 AGKFNIIPTIISSVAAFTSVGVGTVLCDIILL 344
rP2X4R   327 AGKFDIIPTMINVGSGLALLGVATVLCDVIVL 358
hP2X4R   327 AGKFDIIPTMINIGSGLALLGMATVLCDIIVL 358
rP2X5R   328 AGKFSIIPTVINIGSGLALMGAGAFFCDLVLI 359
hP2X5R   328 AGKFSIIPTIINVGSGVALMGAGAFFCDLVLI 359
rP2X6R   331 AGKFALIPTAITVGTGAAWLGMVTFLCDLLLL 362
hP2X6R   329 AGKFGLIPTAVTLGTGAAWLGVVTFFCDLLLL 360
rP2X7R   325 GGKFDIIQLVVYIGSTLSYFGLATVCIDLIIN 356
hP2X7R   325 GGKFDIIQLVVYIGSTLSYFGLATVFIDFLID 356
```

B



SIMILARITY IN SEQUENCE CAN IMPLY SIMILARITY IN FUNCTION

SEQUENCE SIMILARITY

WHAT CAN SEQUENCE SIMILARITY TELL US?

A

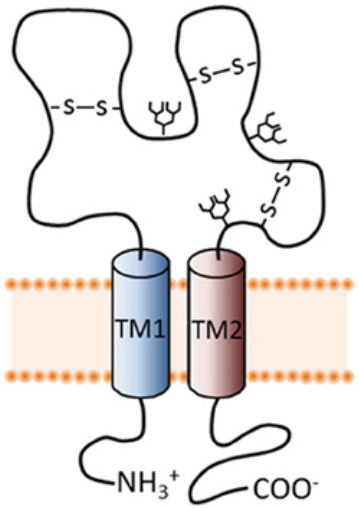
TM1

```
rP2X1R    29 VGVIFRLIQLVVLVYVIGWVFVYEKGYQTSS-DLISSVSVKLKGLAVTQ-------LQGLGPQVWDVADYVFPAHGDSSFVVMT 104
hP2X1R    29 VGVIFRLIQLVVLVYVIGWVFLYEKGYQTSS-GLISSVSVKLKGLAVTQ-------LPGLGPQVWDVADYVFPAQGDNSFVVMT 104
rP2X2R    29 LGFVHRMVQLLILLYFVWYVFIVQKSYQDSETGPESSIITKVKGITMSE-----------DKVWDVEEYVKPPEGGSVVSIIT 100
hP2X2R    41 LGVLYRAVQLLILLYFVWYVFIVQKSYQESETGPESSIITKVKGITTSE-----------HKVWDVEEYVKPPEGGSVFSIIT 112
rP2X3R    23 IGIINRAVQLLIISYFVGWVFLHEKAYQVRDTAIESSVVTKVKGFGRYA-----------NRVMDVSDYVTPPQGTSVFVIIT  94
hP2X3R    23 IGIINRVVQLLIISYFVGWVFLHEKAYQVRDTAIESSVVTKVKGSGLYA-----------NRVMDVSDYVTPPQGTSVFVIIT  94
rP2X4R    28 VGLMNRAVQLLILAYVIGWVFVWEKGYQETD-SVVSSVTTKAKGVAVTN-------TSQLGFRIWDVADYVIPAQEENSLFIMT 103
hP2X4R    28 VGLMNRAVQLLILAYVIGWVFVWEKGYQETD-SVVSSVTTKVKGVAVTN-------TSKLGFRIWDVADYVIPAQEENSLFVMT 103
rP2X5R    29 VGLLYRVLQLIILLYLLIWVFLIKKSYQDIDTSLQSAVVTKVKGVAYTN-------TTMLGERLWDVADFVIPSQGENVFFVVT 105
hP2X5R    29 VGLLYRLLQASILAYLVVWVFLIKKGYQDVDTSLQSAVITKVKGVAFTN-------TSDLGQRIWDVADYVIPAQGENVFFVVT 105
rP2X6R    40 VGISQRLLQLGVVVYVIGWALLAKKGYQEWDMDPQISVITKLKGVSVTQ-------VKELEKRLWDVADFVRPSQGENVFFLVT 116
hP2X6R    38 VGALQRLLQFGIVVYVVGWALLAKKGYQERDLEPQFSIITKLKGVSVTQ-------IKELGNRLWDVADFVKPPQGENVFFLVT 114
rP2X7R    26 YGTIKWILHMTVFSYV-SFALMSDKLYQRKE-PLISSVHTKVKGVAEVTENVTEGGVTKLVHGIFDTADYTLPLQG-NSFFVMT 106
hP2X7R    26 YGTIKWFFHVIIFSYV-CFALVSDKLYQRKE-PVISSVHTKVKGIAEVKEEIVENGVKKLVHSVFDTADYTFPLQG-NSFFVMT 106
```

TM2

```
rP2X1R   323 AGKFDIIPTMTTIGSGIGIFGVATVLCDLLLL 354
hP2X1R   323 AGKFDIIPTMTTIGSGIGIFGVATVLCDLLLL 354
rP2X2R   322 AGKFSLIPTIINLATALTSIGVGSFLCDWILL 353
hP2X2R   333 AGKFSLIPTIINLATALTSVGVGSFLCDWILL 364
rP2X3R   313 AGKFNIIPTIISSVAAFTSVGVGTVLCDIILL 344
hP2X3R   313 AGKFNIIPTIISSVAAFTSVGVGTVLCDIILL 344
rP2X4R   327 AGKFDIIPTMINVGSGLALLGVATVLCDVIVL 358
hP2X4R   327 AGKFDIIPTMINIGSGLALLGMATVLCDIIVL 358
rP2X5R   328 AGKFSIIPTVINIGSGLALMGAGAFFCDLVLI 359
hP2X5R   328 AGKFSIIPTIINVGSGVALMGAGAFFCDLVLI 359
rP2X6R   331 AGKFALIPTAITVGTGAAWLGMVTFLCDLLLL 362
hP2X6R   329 AGKFGLIPTAVTLGTGAAWLGVVTFFCDLLLL 360
rP2X7R   325 GGKFDIIQLVVYIGSTLSYFGLATVCIDLIIN 356
hP2X7R   325 GGKFDIIQLVVYIGSTLSYFGLATVFIDFLID 356
```

B

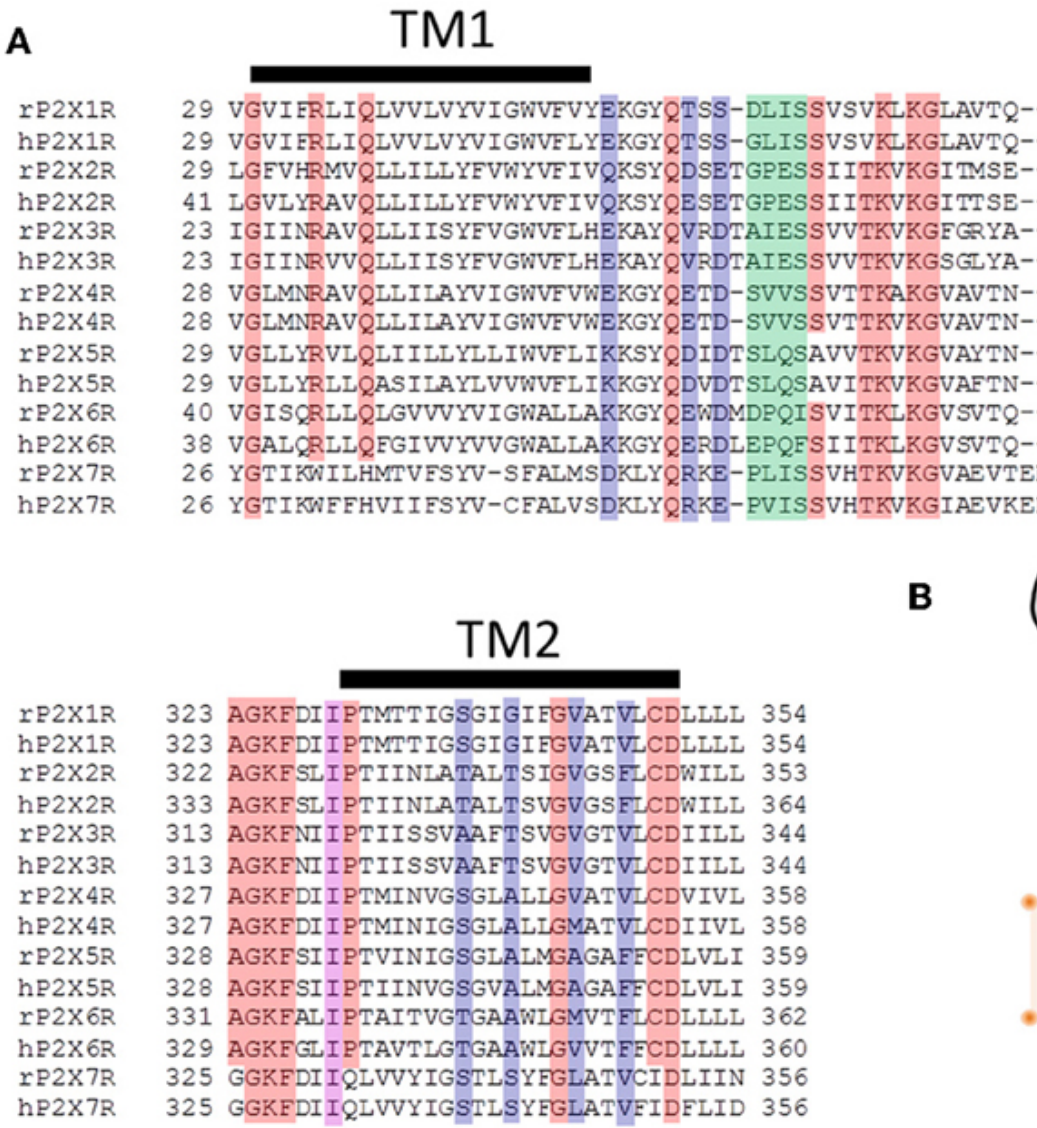


DETERMINING THE SOURCE ORGANISM OF A SEQUENCE

SEQUENCE SIMILARITY

WHAT CAN SEQUENCE SIMILARITY TELL US?

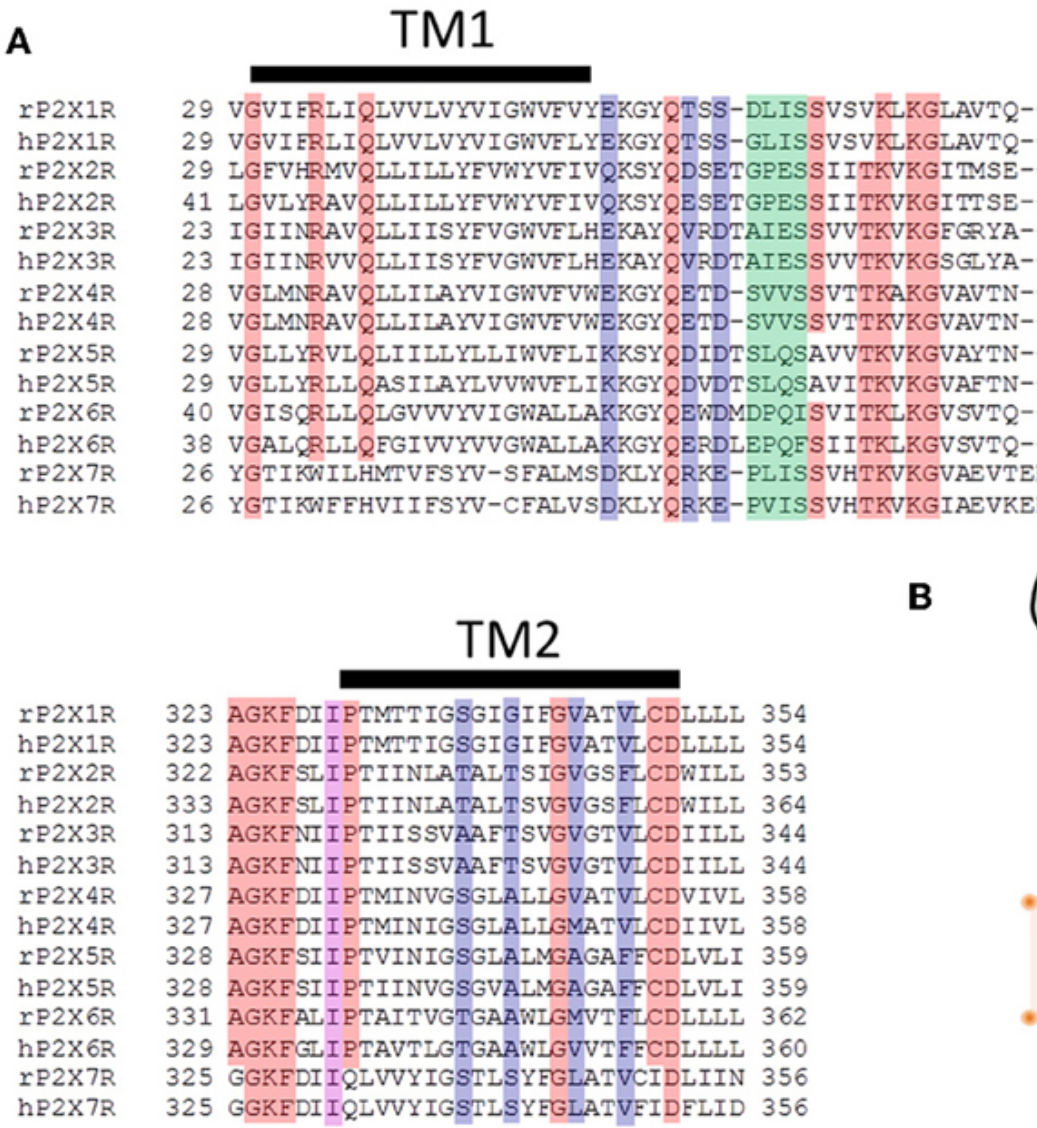
- Assign function to uncharacterized sequences based on characterized sequences
- Determine function of an unknown gene sequence



SEQUENCE SIMILARITY

WHAT CAN SEQUENCE SIMILARITY TELL US?

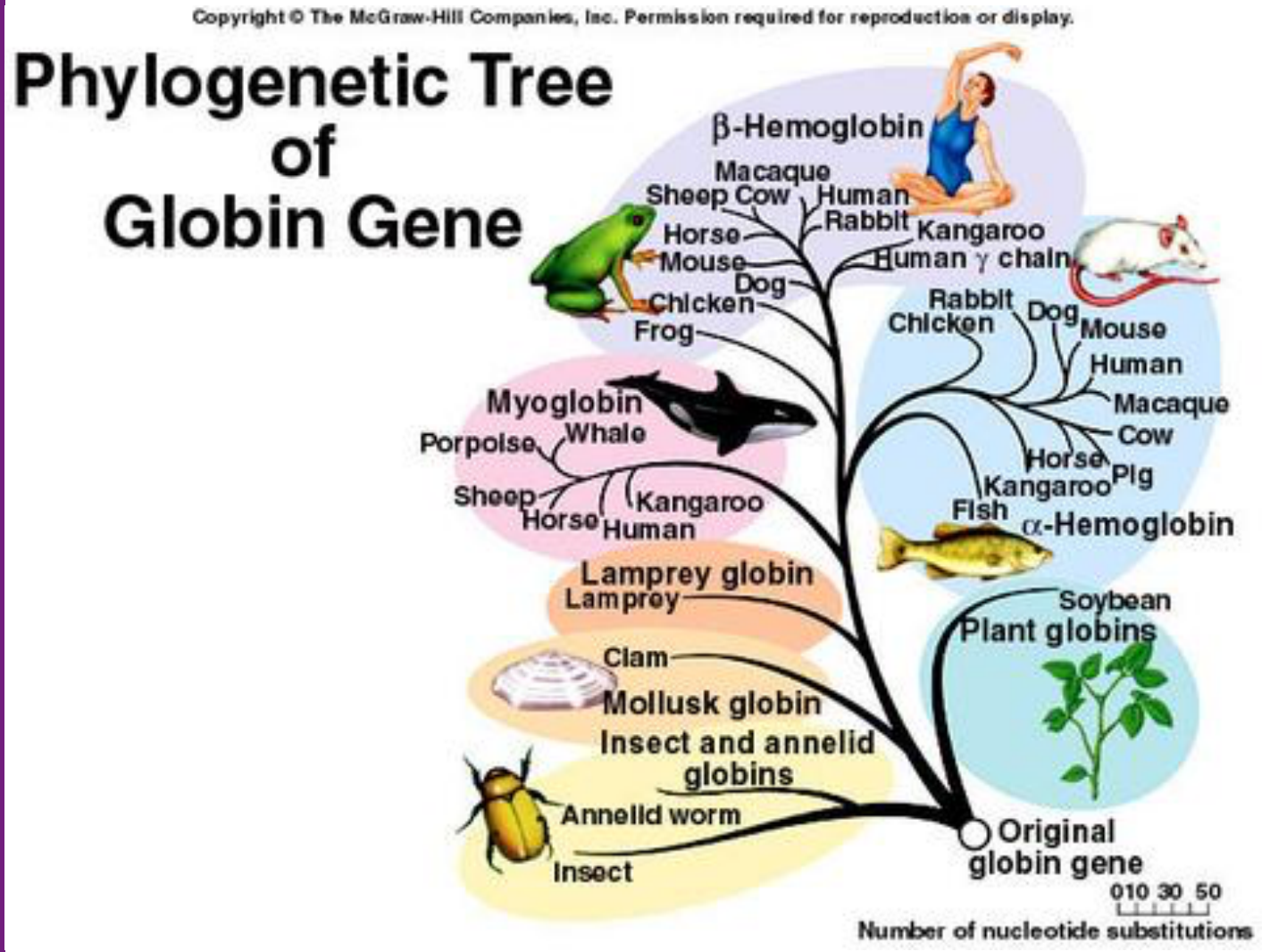
- Developing hypotheses about the relatedness of organisms
  - Grouping sequences from closely related organisms
  - Sequence from different species can be compared to estimate the evolutionary relationships between species



SEQUENCE SIMILARITY

WHAT CAN SEQUENCE SIMILARITY TELL US?

- Alignments reveal homology



SEQUENCE SIMILARITY

- Similarity
  - Degree of match between the two sequences
  - Example: These sequences are 90% identical
- Homology
  - Sequences evolved from a common ancestral sequence
  - Example: These sequences are related

SEQUENCE SIMILARITY

- Sequence similarity does not always imply a common function
- Conserved function does not always imply similarity at the sequence level

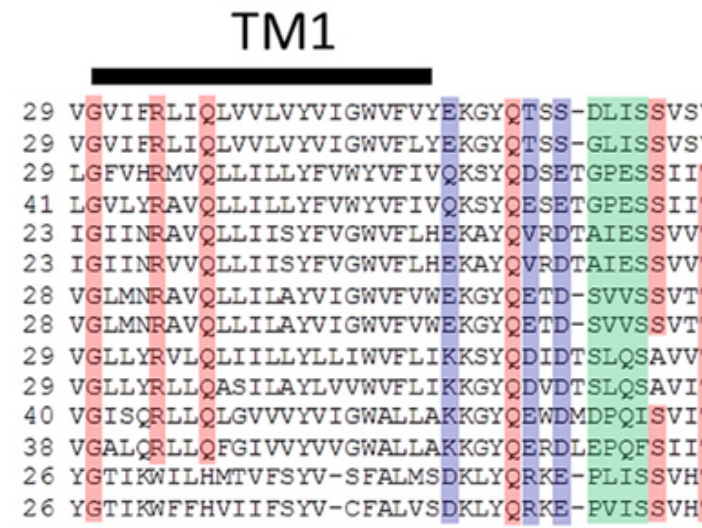
SEQUENCE SIMILARITY

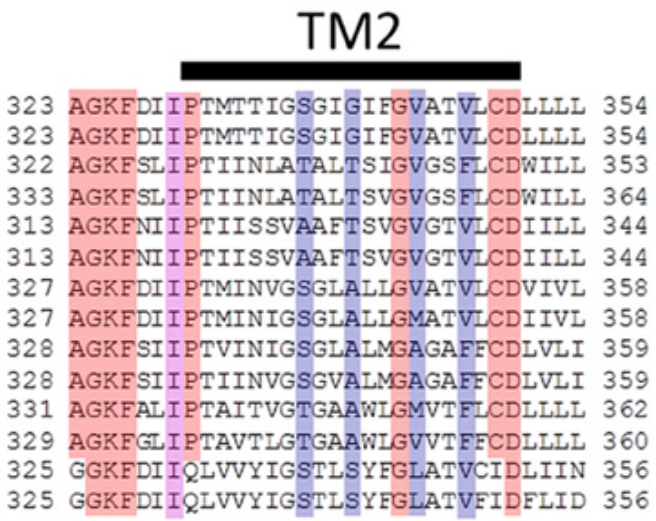
- Convergent evolution
  - Sequences are highly similar, but are not homologous
  - The function evolved through different mechanisms, but converged on the same adaption

SEQUENCE SIMILARITY

DIFFERENT TYPES OF SIMILARITY PROBLEMS

- Similarity between a small number of sequences
  - Pairwise sequence alignment
  - Multiple sequence alignment
- Searching databases for a query sequence
  - Heuristic search using BLAST

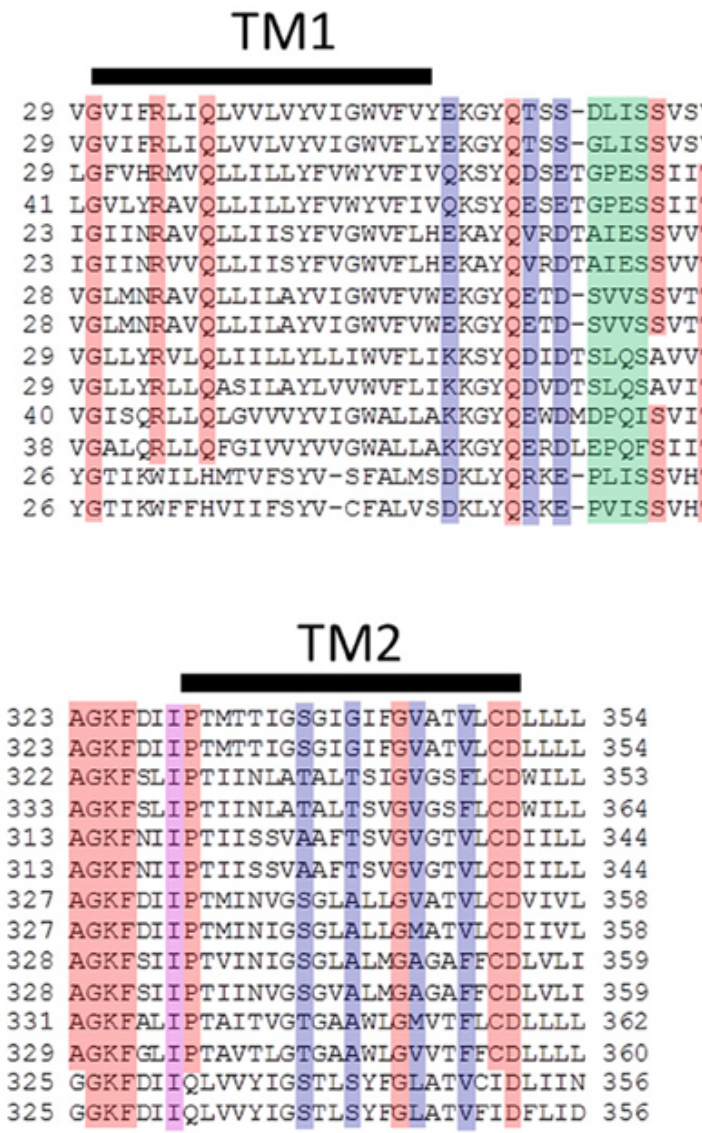




SEQUENCE SIMILARITY

DIFFERENT TYPES OF SIMILARITY PROBLEMS

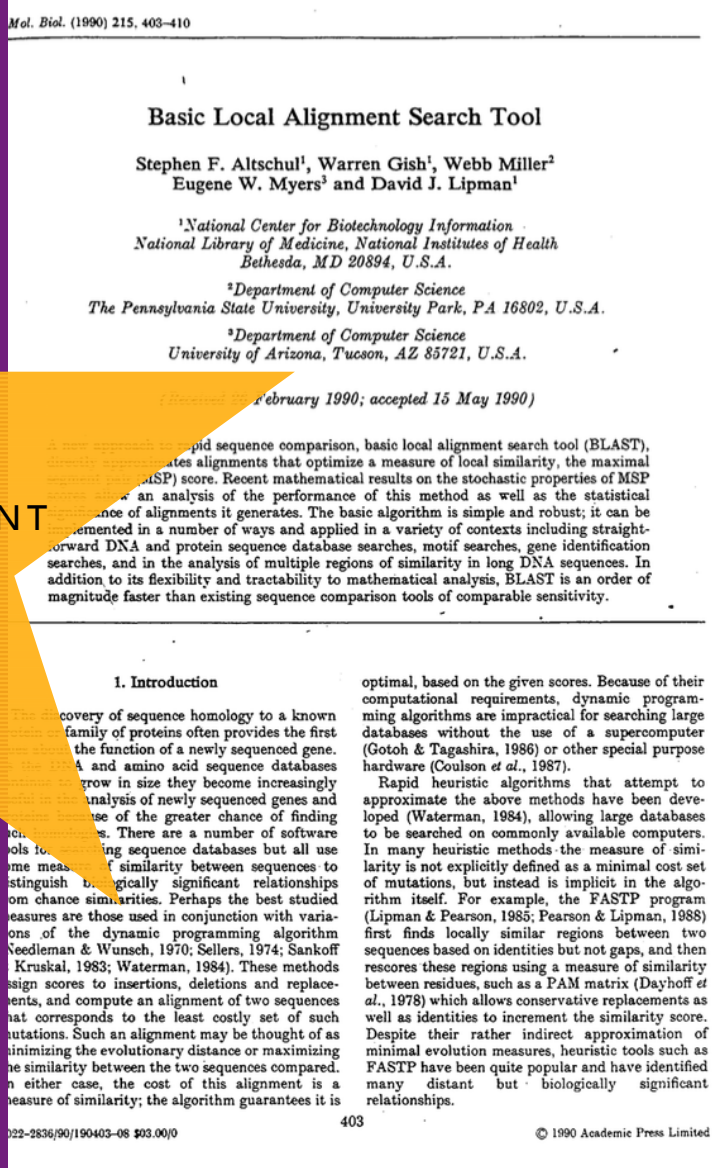
- Protein
- DNA



SEQUENCE SIMILARITY

- Similar techniques can be used for DNA or protein
- Differences
  - Algorithms
  - Parameters
  - Scoring

MOVE/REDUNDANT



Mol. Biol. (1990) 215, 403–410

Basic Local Alignment Search Tool

Stephen F. Altschul[1], Warren Gish[1], Webb Miller[2]
Eugene W. Myers[3] and David J. Lipman[1]

[1]*National Center for Biotechnology Information*
National Library of Medicine, National Institutes of Health
Bethesda, MD 20894, U.S.A.

[2]*Department of Computer Science*
The Pennsylvania State University, University Park, PA 16802, U.S.A.

[3]*Department of Computer Science*
University of Arizona, Tucson, AZ 85721, U.S.A.

ebruary 1990; accepted 15 May 1990)

pid sequence comparison, basic local alignment search tool (BLAST),
ates alignments that optimize a measure of local similarity, the maximal
SP) score. Recent mathematical results on the stochastic properties of MSP
an analysis of the performance of this method as well as the statistical
nce of alignments it generates. The basic algorithm is simple and robust; it can be
emented in a number of ways and applied in a variety of contexts including straight-
orward DNA and protein sequence database searches, motif searches, gene identification
searches, and in the analysis of multiple regions of similarity in long DNA sequences. In
addition to its flexibility and tractability to mathematical analysis, BLAST is an order of
magnitude faster than existing sequence comparison tools of comparable sensitivity.

1. Introduction

covery of sequence homology to a known
family of proteins often provides the first
the function of a newly sequenced gene.
A and amino acid sequence databases
row in size they become increasingly
nalysis of newly sequenced genes and
se of the greater chance of finding
s. There are a number of software
ols fo
ing sequence databases but all use
me meas
f similarity between sequences to
stinguish b
gically significant relationships
om chance simi
arities. Perhaps the best studied
easures are those used in conjunction with varia-
ons of the dynamic programming algorithm
Needleman & Wunsch, 1970; Sellers, 1974; Sankoff
Kruskal, 1983; Waterman, 1984). These methods
ssign scores to insertions, deletions and replace-
ents, and compute an alignment of two sequences
at corresponds to the least costly set of such
utations. Such an alignment may be thought of as
inimizing the evolutionary distance or maximizing
e similarity between the two sequences compared.
n either case, the cost of this alignment is a
easure of similarity; the algorithm guarantees it is
optimal, based on the given scores. Because of their computational requirements, dynamic programming algorithms are impractical for searching large databases without the use of a supercomputer (Gotoh & Tagashira, 1986) or other special purpose hardware (Coulson *et al.*, 1987).

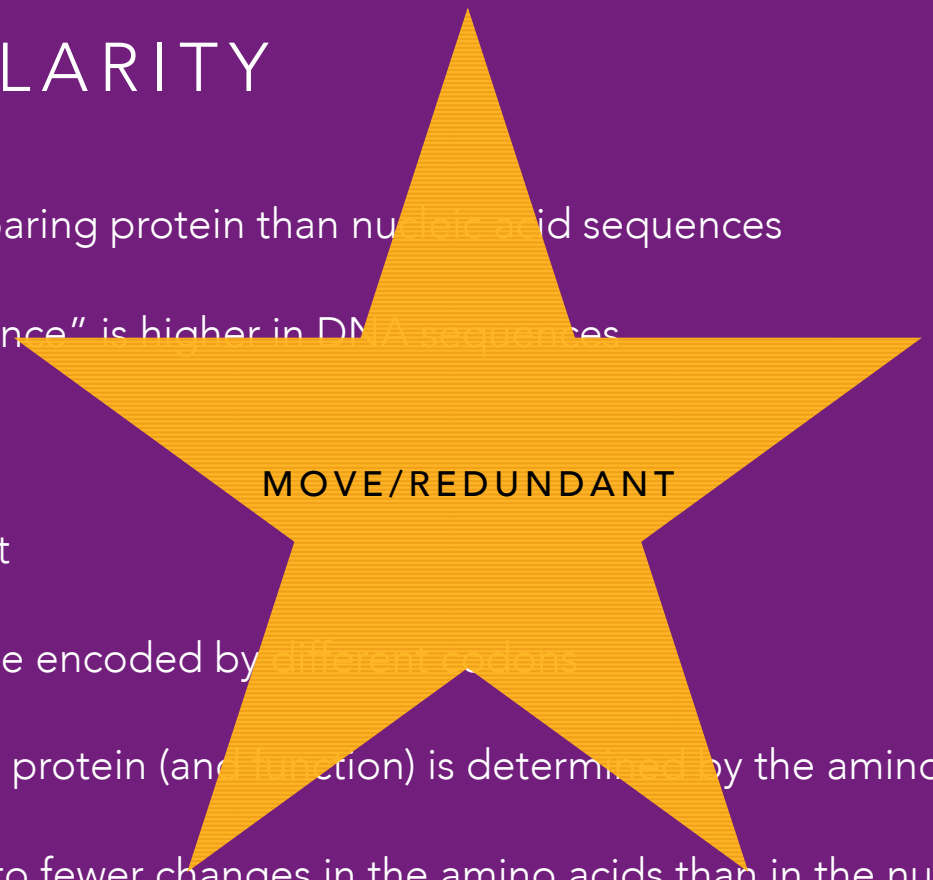
Rapid heuristic algorithms that attempt to approximate the above methods have been developed (Waterman, 1984), allowing large databases to be searched on commonly available computers. In many heuristic methods the measure of similarity is not explicitly defined as a minimal cost set of mutations, but instead is implicit in the algorithm itself. For example, the FASTP program (Lipman & Pearson, 1985; Pearson & Lipman, 1988) first finds locally similar regions between two sequences based on identities but not gaps, and then rescores these regions using a measure of similarity between residues, such as a PAM matrix (Dayhoff *et al.*, 1978) which allows conservative replacements as well as identities to increment the similarity score. Despite their rather indirect approximation of minimal evolution measures, heuristic tools such as FASTP have been quite popular and have identified many distant but biologically significant relationships.

022-2836/90/190403-08 $03.00/0

© 1990 Academic Press Limited

SEQUENCE SIMILARITY

- Homology easier to detect comparing protein than nucleic acid sequences
  - Probability of a "match by chance" is higher in DNA sequences
    - 0.25 vs. 0.05
  - The genetic code is redundant
    - Identical amino acids can be encoded by different codons
  - The complex 3D structure of a protein (and function) is determined by the amino acid sequence
    - Conserving function leads to fewer changes in the amino acids than in the nucleotide sequence

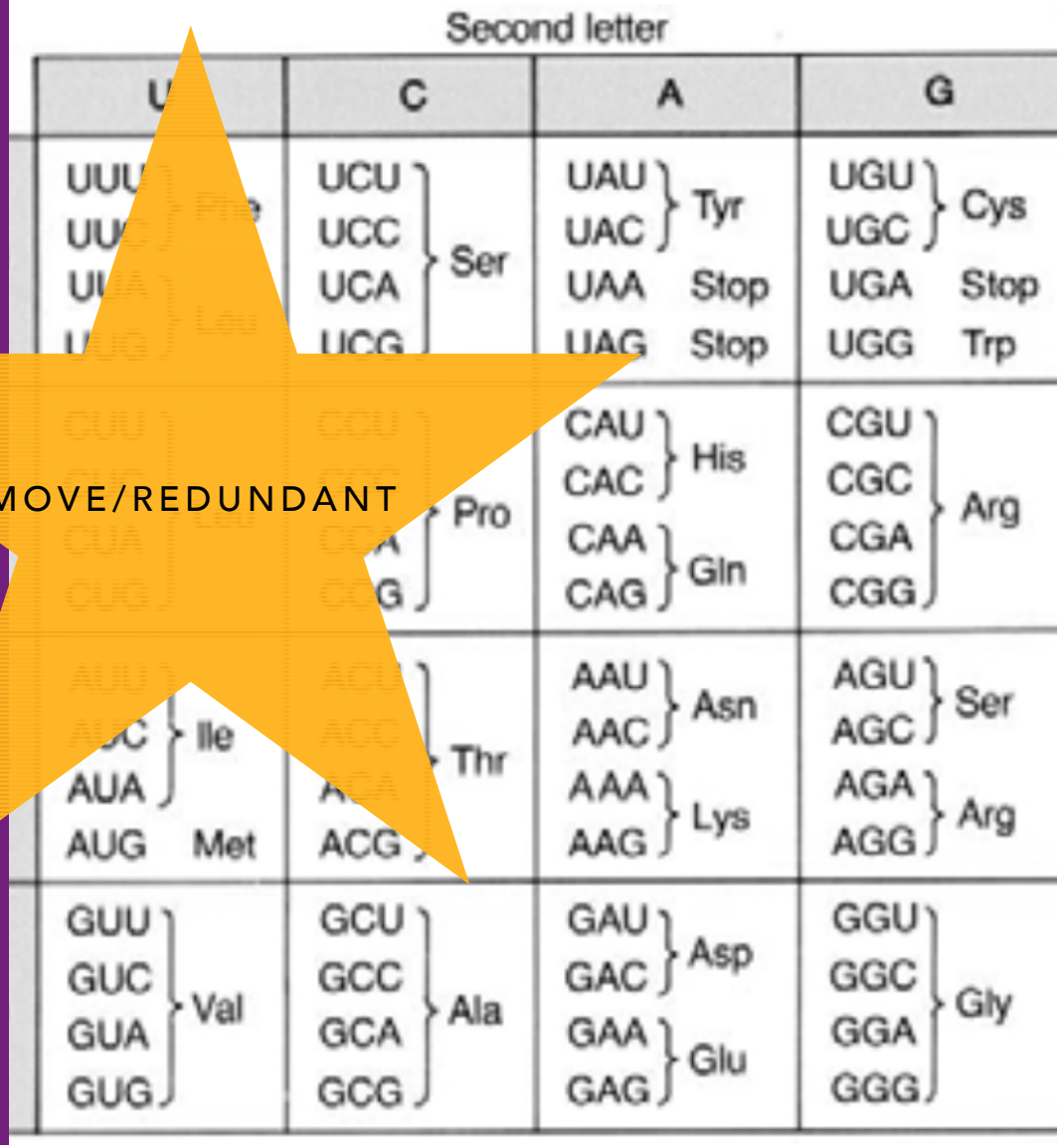


DISCUSSED IN MORE DETAIL LATER IN COURSE

SEQUENCE SIMILARITY

- Protein is more informative
  - 20 vs 4 characters
  - Amino acids share related biophysical properties
  - Codons are degenerate
    - changes in the third position often do not alter the amino acid that is specified
  - Offer a longer "look-back" time
- DNA sequences can be unambiguously translated into protein
  - Use in pairwise alignment

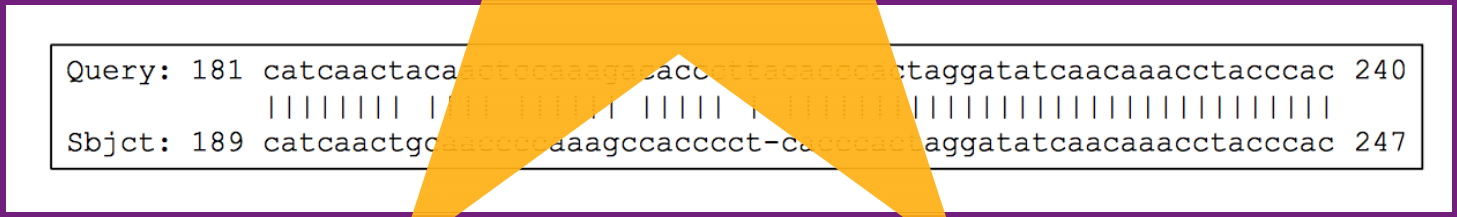
MOVE/REDUNDANT



SEQUENCE SIMILARITY

- Many times, DNA alignments are appropriate
  - To confirm the identity of a cDNA
  - To study noncoding regions of DNA
  - To study DNA
- For example
  - Neanderthal vs modern human DNA

MOVE/REDUNDANT



Issues in Aligning Sequences

ISSUES IN ALIGNING SEQUENCES

- What is sequence alignment?
  - The task of locating equivalent regions of two or more sequences to assess their overall similarity
  - Not necessarily "maximize"
    - Sometimes small segments may be more informative

ISSUES IN ALIGNING SEQUENCES

A Compilation of f(n, m) for $1 \leq n \leq 5, 10$, and $2 \leq m \leq 5$

m	2	3	4	5
$n = 1$	3	13	75	541
2	13	409	23917	2244361
3	63	16081	10681263	14638756721
4	321	699121	5552351121	117629959485121
5	1683	32193253	3147728203035	1.05×10^{18}
10	8097453	9850349744182729	3.32×10^{26}	1.35×10^{38}

$f(m,n)$: The total number of possible alignments between $\vec{a}$ and $\vec{b}$

- Given two sequences, the number of possible alignments is exponential

ISSUES IN ALIGNING SEQUENCES

- Finding the "correct" alignment involves
  - Defining a scoring scheme
  - Finding an alignment with optimal score
- Alignments of related sequences should give good scores compared with alignments of randomly chosen sequences
- The correct alignment of two related sequences should ideally be the one that gives the best score
  - In practice, the correct alignment does not necessarily have the best score, since no "perfect" scoring scheme has been devised

ISSUES IN ALIGNING SEQUENCES

QUESTIONS IN SEQUENCE ALIGNMENT

- What type of alignment?
  - Align the entire sequence or part of it?
  - Two sequences or multiple sequences?

ISSUES IN ALIGNING SEQUENCES

QUESTIONS IN SEQUENCE ALIGNMENT

- How to find the alignment?
  - Search algorithms for alignment

ISSUES IN ALIGNING SEQUENCES

QUESTIONS IN SEQUENCE ALIGNMENT

- How to score an alignment?
  - the sequences we're comparing typically differ in length
  - some characters (nucleotide or amino acid) are more substitutable than others

ISSUES IN ALIGNING SEQUENCES

QUESTIONS IN SEQUENCE ALIGNMENT

- How to tell if the alignment is biologically meaningful?
  - Assessing how likely the alignment could have happened by random chance

ISSUES IN ALIGNING SEQUENCES

THI S SEQUENCE

THATISASEQUENCE

- Sequences of unequal length

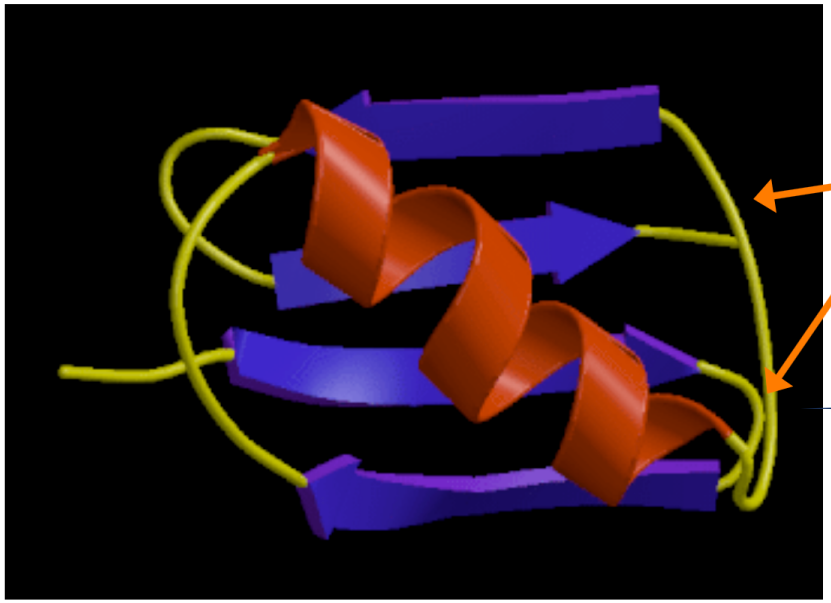
ISSUES IN ALIGNING SEQUENCES

substitutions:	ACGA	AGGA
insertions:	ACGA	ACGGA
deletions:	ACGA	AGA

- Mutations cause changes in sequences that may/may not have effect on biology

INTRODUCES GAPS IN ALIGNMENTS

ISSUES IN ALIGNING SEQUENCES

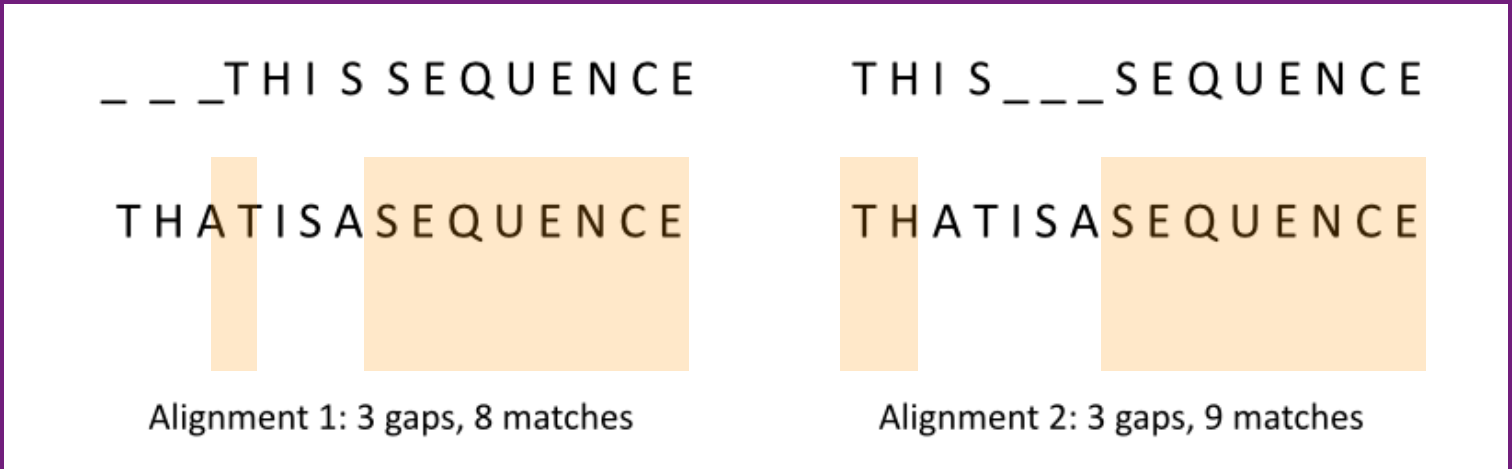


loop structures: insertions/deletions here not so significant

LOOP REGIONS ARE HIGHLY FLEXIBLE; NOT CONSERVED

- Why is it that two "similar" sequences may have large insertions/deletions?
  - Some insertions and deletions may not significantly affect the structure of a protein

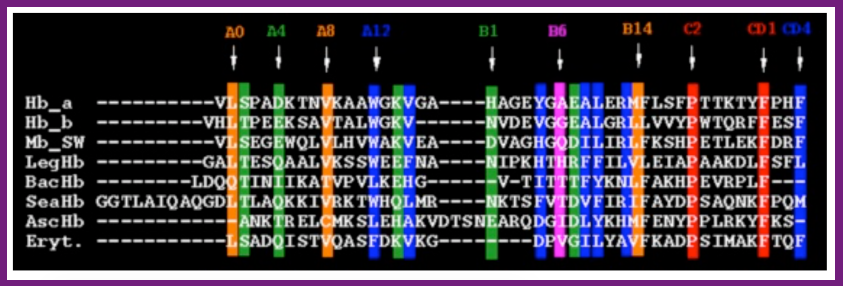
ISSUES IN ALIGNING SEQUENCES

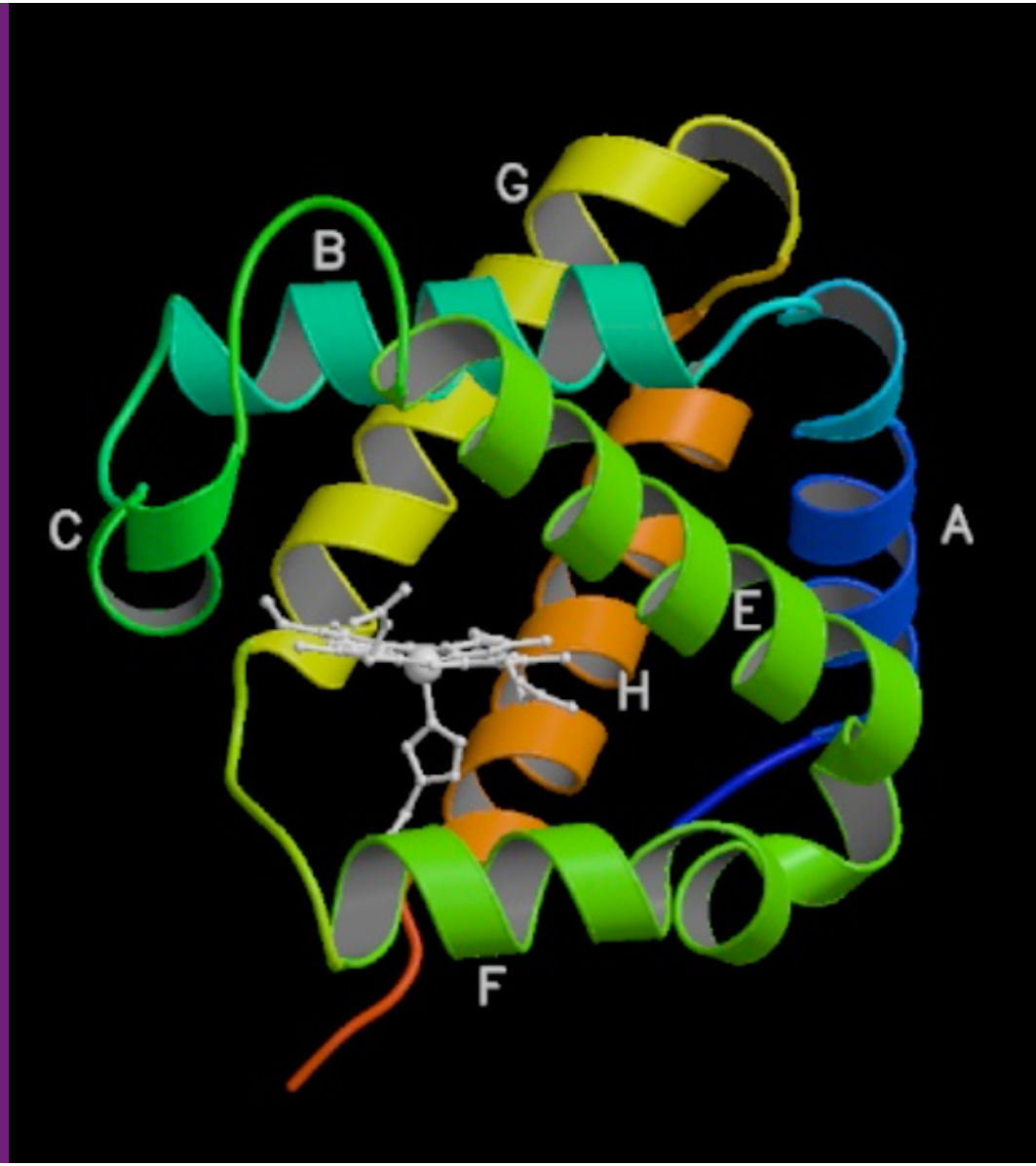


- Incorporating gaps while aligning sequences
  - Which is better?

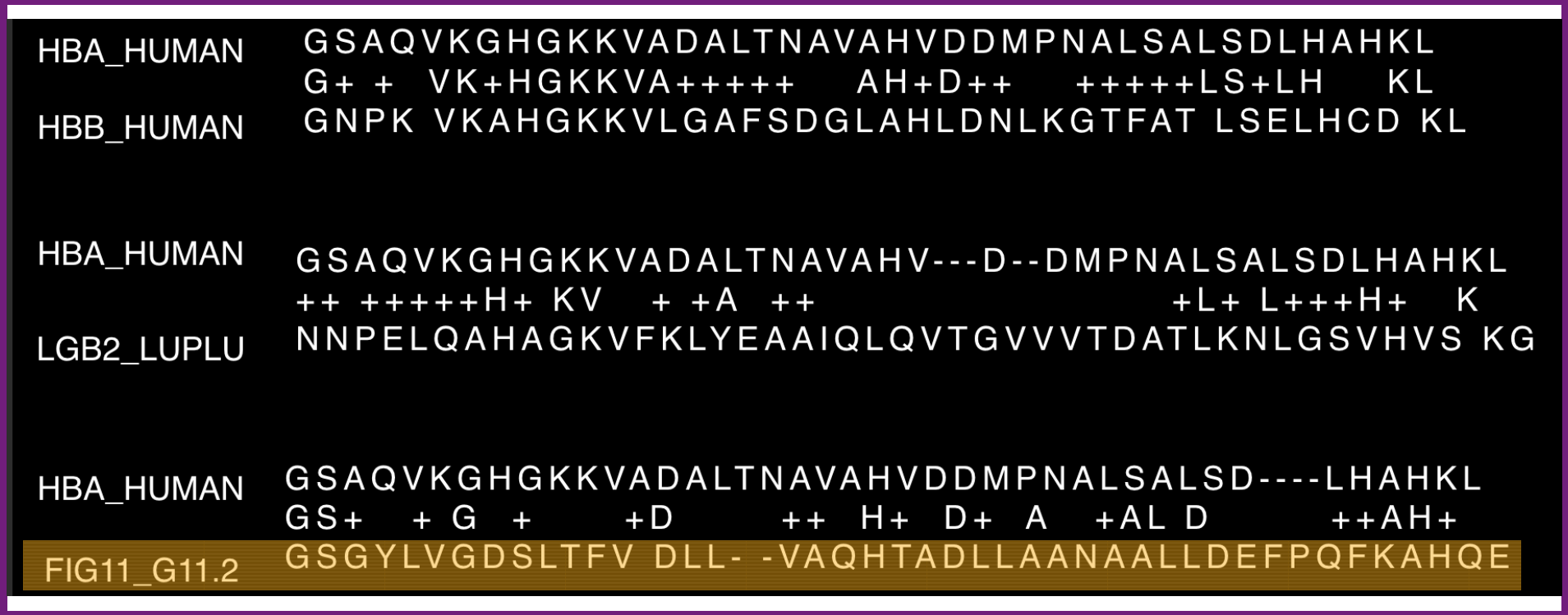
ISSUES IN ALIGNING SEQUENCES

- Aligned sequences (with gaps) shows conservation across evolutionarily related proteins





ISSUES IN ALIGNING SEQUENCES

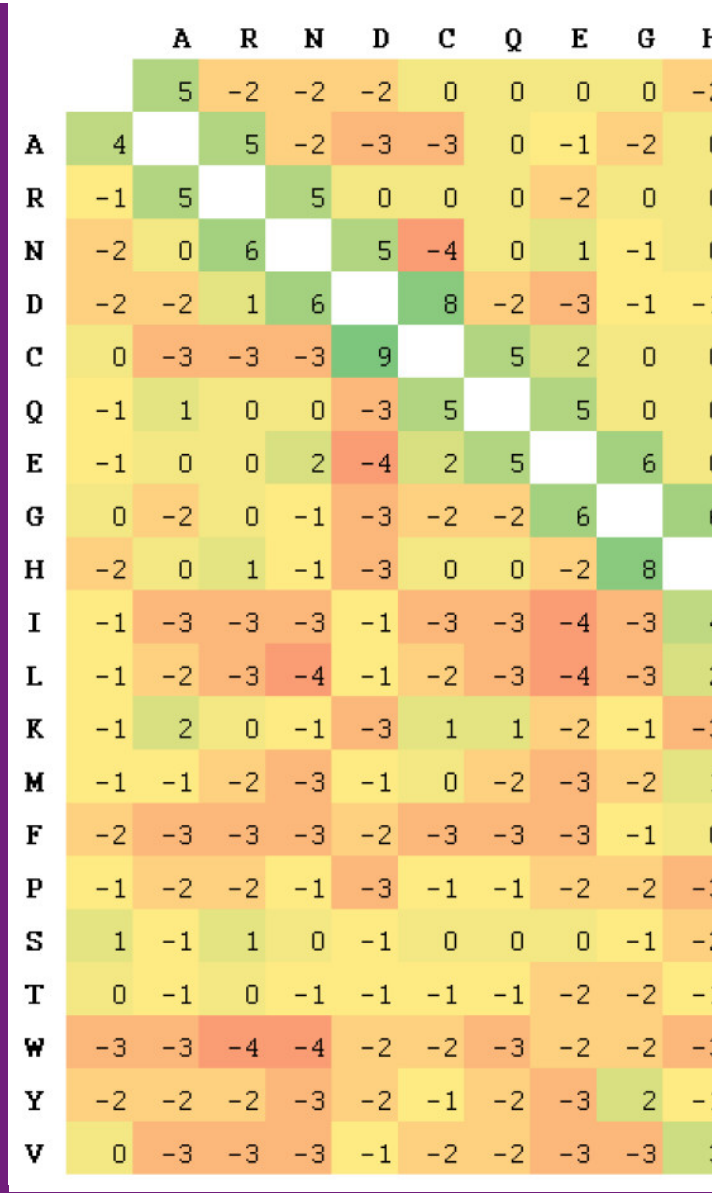


Example from Durbin & Eddy

- Why do we need principled approaches to sequence alignment? Which of these is spurious?

ISSUES IN ALIGNING SEQUENCES

- Scoring alignments
  - Percent identity
  - Gap penalty functions
  - Substitution matrices of amino acids
    - Genuine matches may not be identical
    - Specialized matrices



	A	R	N	D	C	Q	E	G
	5	-2	-2	-2	0	0	0	0
A	4	5	-2	-3	-3	0	-1	-2
R	-1	5	5	0	0	0	-2	0
N	-2	0	6	5	-4	0	1	-1
D	-2	-2	1	6	8	-2	-3	-1
C	0	-3	-3	-3	9	5	2	0
Q	-1	1	0	0	-3	5	5	0
E	-1	0	0	2	-4	2	5	6
G	0	-2	0	-1	-3	-2	-2	6
H	-2	0	1	-1	-3	0	0	-2
I	-1	-3	-3	-3	-1	-3	-3	-4
L	-1	-2	-3	-4	-1	-2	-3	-4
K	-1	2	0	-1	-3	1	1	-2
M	-1	-1	-2	-3	-1	0	-2	-3
F	-2	-3	-3	-3	-2	-3	-3	-3
P	-1	-2	-2	-1	-3	-1	-1	-2
S	1	-1	1	0	-1	0	0	0
T	0	-1	0	-1	-1	-1	-1	-2
W	-3	-3	-4	-4	-2	-2	-3	-2
Y	-2	-2	-2	-3	-2	-1	-2	-3
V	0	-3	-3	-3	-1	-2	-2	-3

ISSUES IN ALIGNING SEQUENCES

- What type of algorithms for sequence alignment
  - Visual
  - Dynamic Programming
  - Multiple sequence alignment
  - Database

QUICK VISUAL REPRESENTATION

SLOW, BUT WILL FIND OPTIMAL ALIGNMENT

MULTISTEP, EVOLUTIONARY ANALYSIS

FAST, BUT HEURISTIC


PAIRWISE
SEQUENCE
ALIGNMENT

PAIRWISE SEQUENCE ALIGNMENT

GENERAL APPROACH TO PAIRWISE ALIGNMENT

- Selects an algorithm to generate a score
  - Allow gaps (insertions, deletions)
  - Alignments can be global or local
- Score
- Estimate probability that the alignment occurred by chance

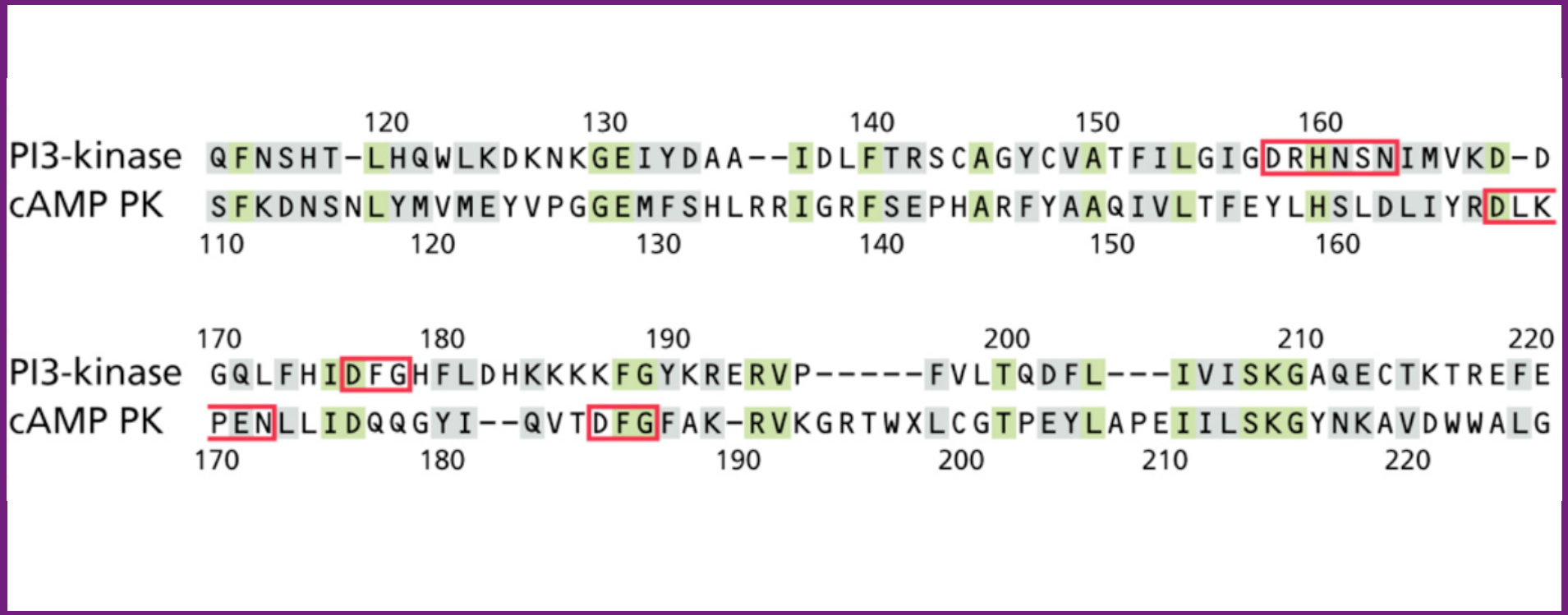
Types of Alignments

TYPES OF ALIGNMENTS

IT IS IMPORTANT TO USE THE "APPROPRIATE" TYPE

- Global alignment
  - Aligning the whole sequences
  - Appropriate when aligning two very closely related sequences
- Local alignment
  - Aligning certain regions in the sequences
  - Appropriate for aligning multi-domain protein sequences

TYPES OF ALIGNMENTS



- Global alignment

TYPES OF ALIGNMENTS

- Local alignment
  - Aligning certain regions in the sequences
  - Appropriate for aligning multi-domain protein sequences

TYPES OF ALIGNMENTS

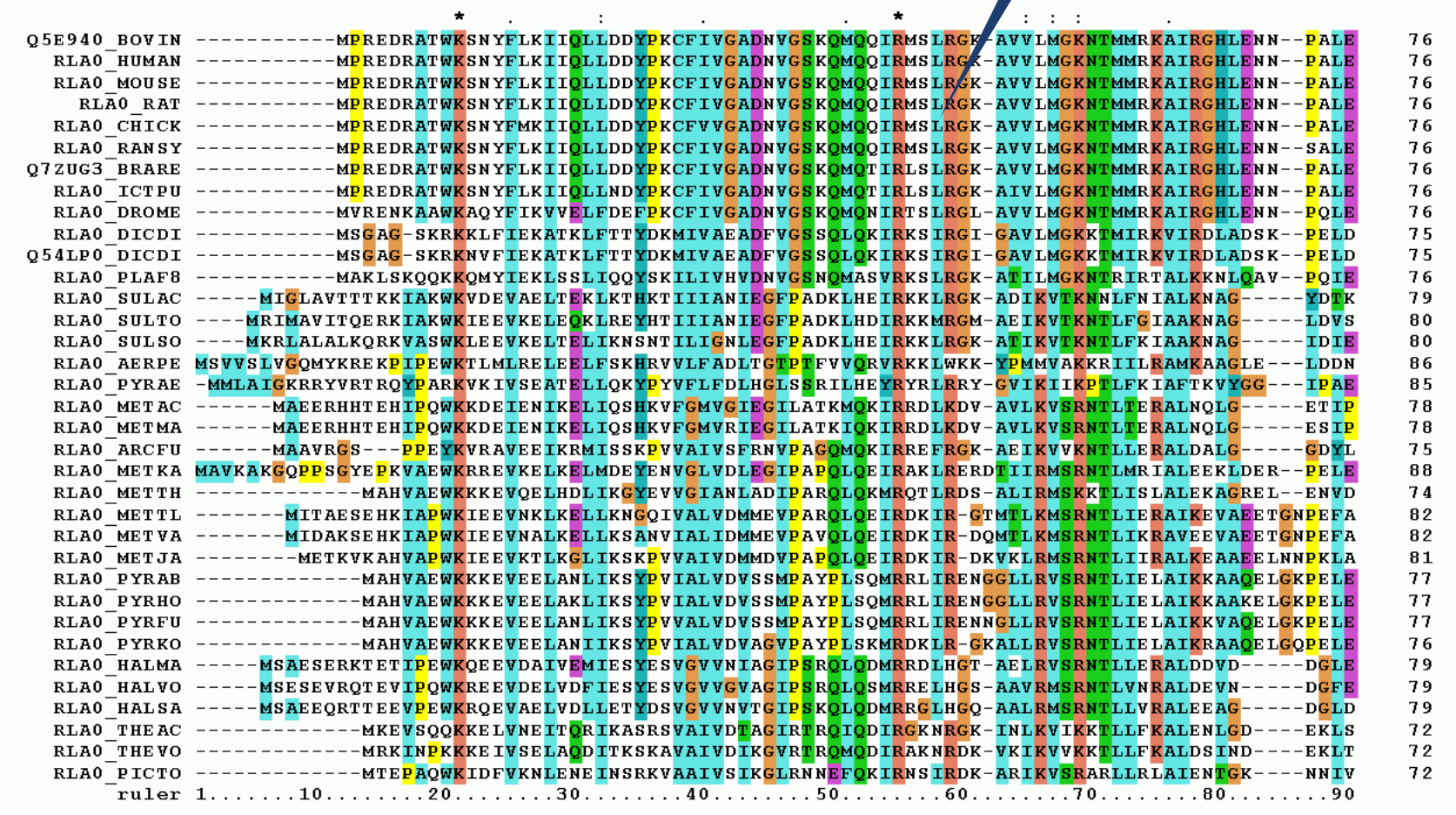
- Pairwise alignment
  - Aligning a pair of sequences
  - Computationally “easy”
- Multiple alignment
  - Aligning more than two sequences
  - Computationally “hard”
  - Useful when sequences of low similarity are being aligned

TYPES OF ALIGNMENTS

- Pairwise alignment (A)
  - Does not align the important active-site residues
- Multiple alignment (B)
  - Does align the important active-site residues

TYPES OF ALIGNMENTS

MULTIPLE SEQUENCE ALIGNMENT



Scoring Alignments

SCORING ALIGNMENTS

- Alignments of related sequences should give good scores compared with alignments of randomly chosen sequences
- The correct alignment of two related sequences should (ideally) be the one that gives the best score
- In practice, the correct alignment does not necessarily have the best score
  - No “perfect” scoring scheme has been devised

SCORING ALIGNMENTS

MEASURES FOR IDENTIFYING SIMILARITY

- Identity
  - The number of identical bases or amino acids matched between two aligned sequences
  - “20 identical residues in alignment”
- Percent identity
  - Dividing this number by the total length of the aligned sequences and multiplying by 100
  - “50% sequence identity”

SCORING ALIGNMENTS

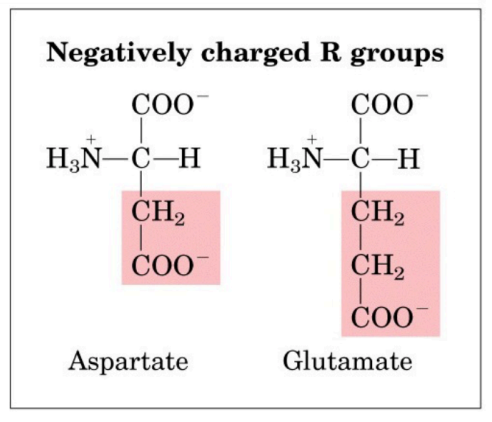
GENUINE MATCHES DO NOT HAVE TO BE IDENTICAL

- Similarity
  - Mutations that replace one amino acid with another conserved (similar) amino acid
  - Likely to have been accepted during evolution
  - Pairs of amino acids with similar properties often represent genuine matches rather than matches occurring randomly
- Percent similarity
  - Percent of amino acids that are identical or similar between aligned sequences

SCORING ALIGNMENTS

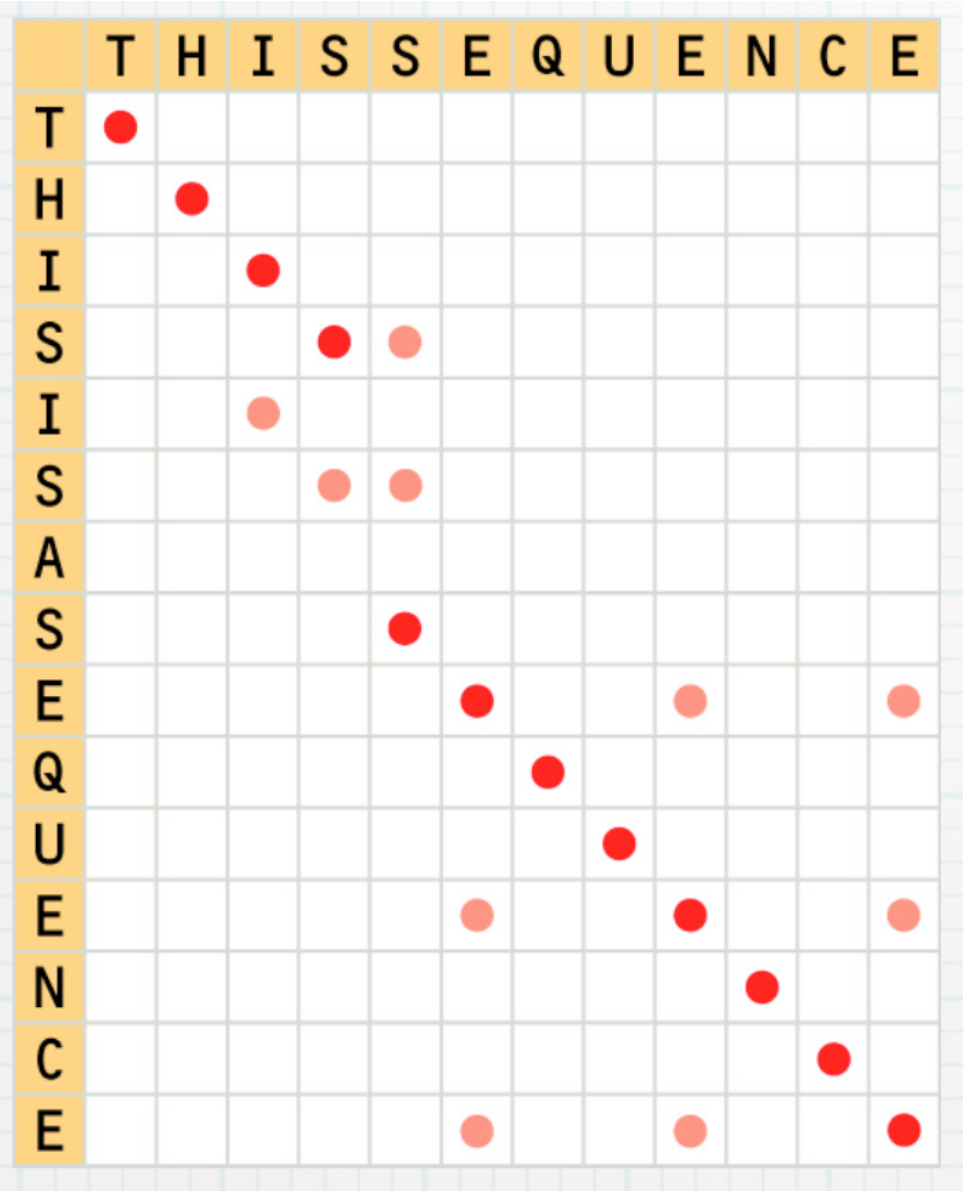
- Identity vs similarity
  - Identity: 5 amino acids
  - Percent identity: 50%
  - Similarity: 10 amino acids
  - Percent similarity: 100%

AADEEEEENY
AADDDDDDNY



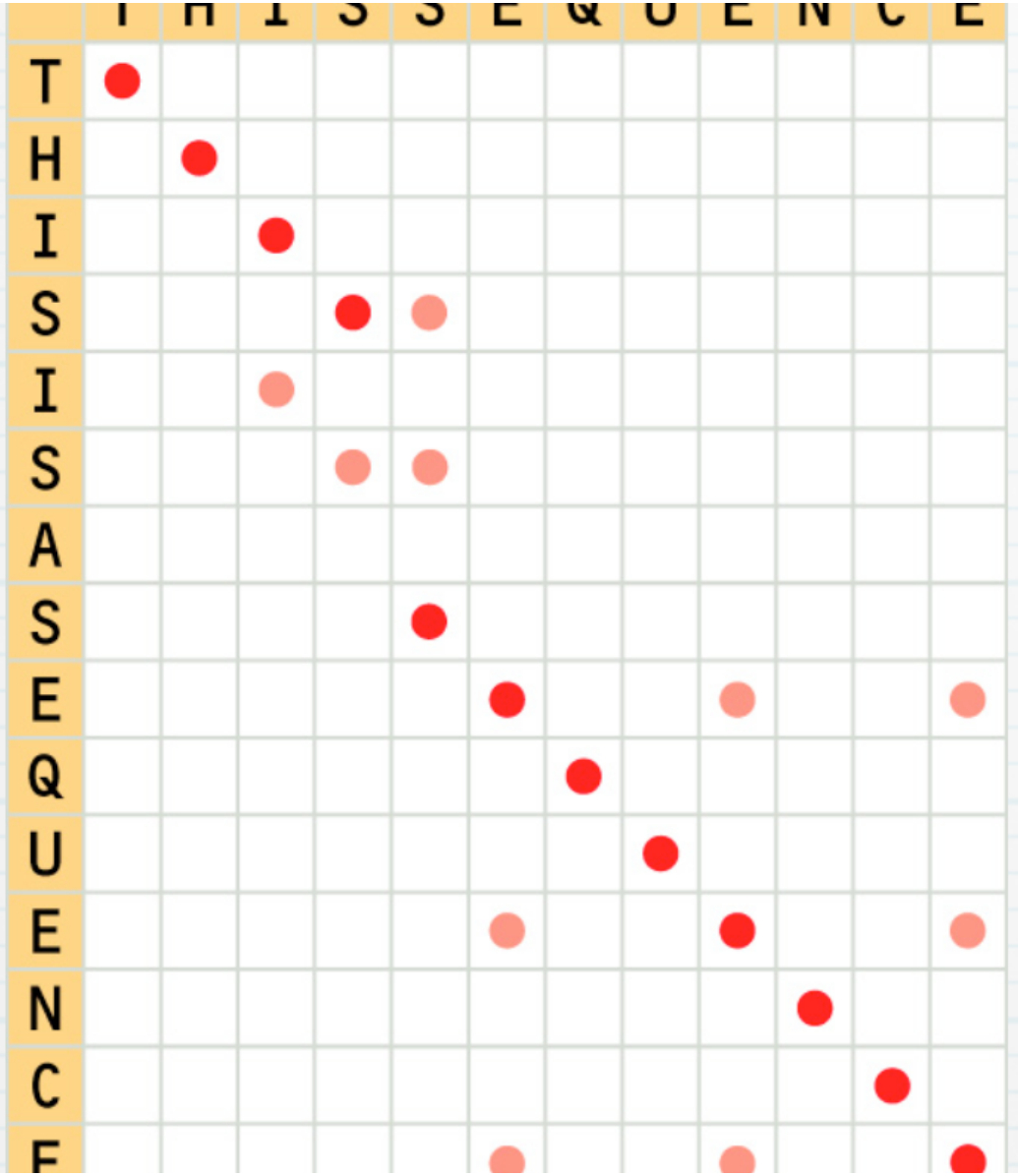
SCORING ALIGNMENTS

- Visualized sequence similarity
  - Dot plot
    - Red dots represent identities that are due to true matching of identical residue-pairs
    - Pink dots represent identities that are due to noise
      - Matching of random identical residue-pairs

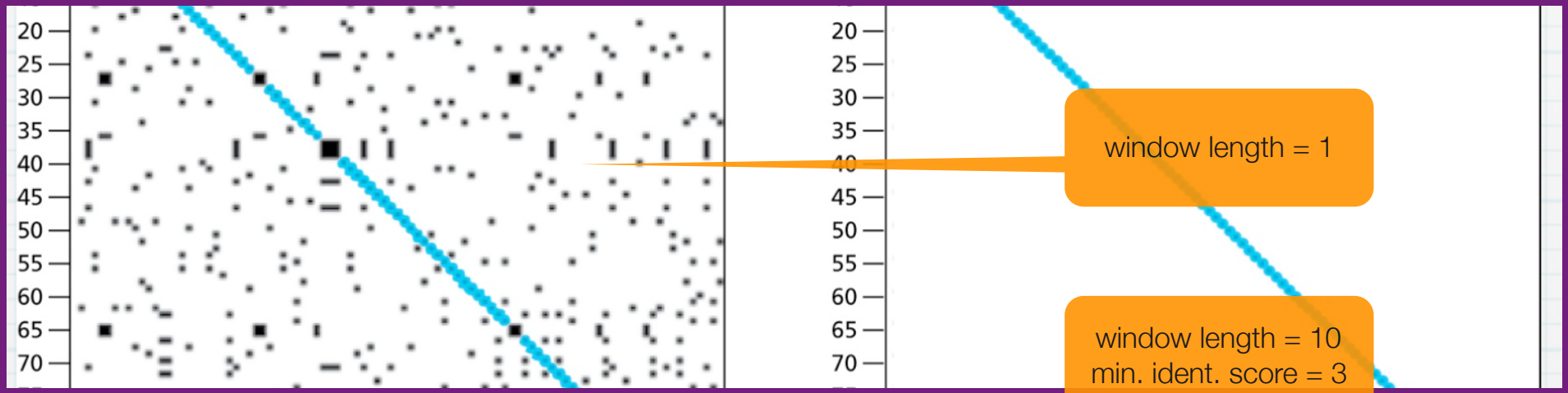


SCORING ALIGNMENTS

- Dot-plots suffer from background noise
- To overcome this problem a filter can be applied



SCORING ALIGNMENTS



- The most-common filtering method
  - Sliding window that requires that the comparison achieves some minimum identity score summed over that window before being considered

SCORING ALIGNMENTS

- There are general (but contested) rules about minimum percent identity that can be accepted as significant
  - >30% are structurally similar proteins
  - 20%-30% identity called “twilight zone”
    - Evolutionary relatedness may exist
    - Not reliably assumed in the absence of other evidence
  - >70% required for functional conservation

SCORING ALIGNMENTS

- Scoring function comprises
  - Substitution matrix s(a,b)
    - Score of aligning a with b
  - Gap penalty function $\gamma(g)$
    - Where g is the gap length

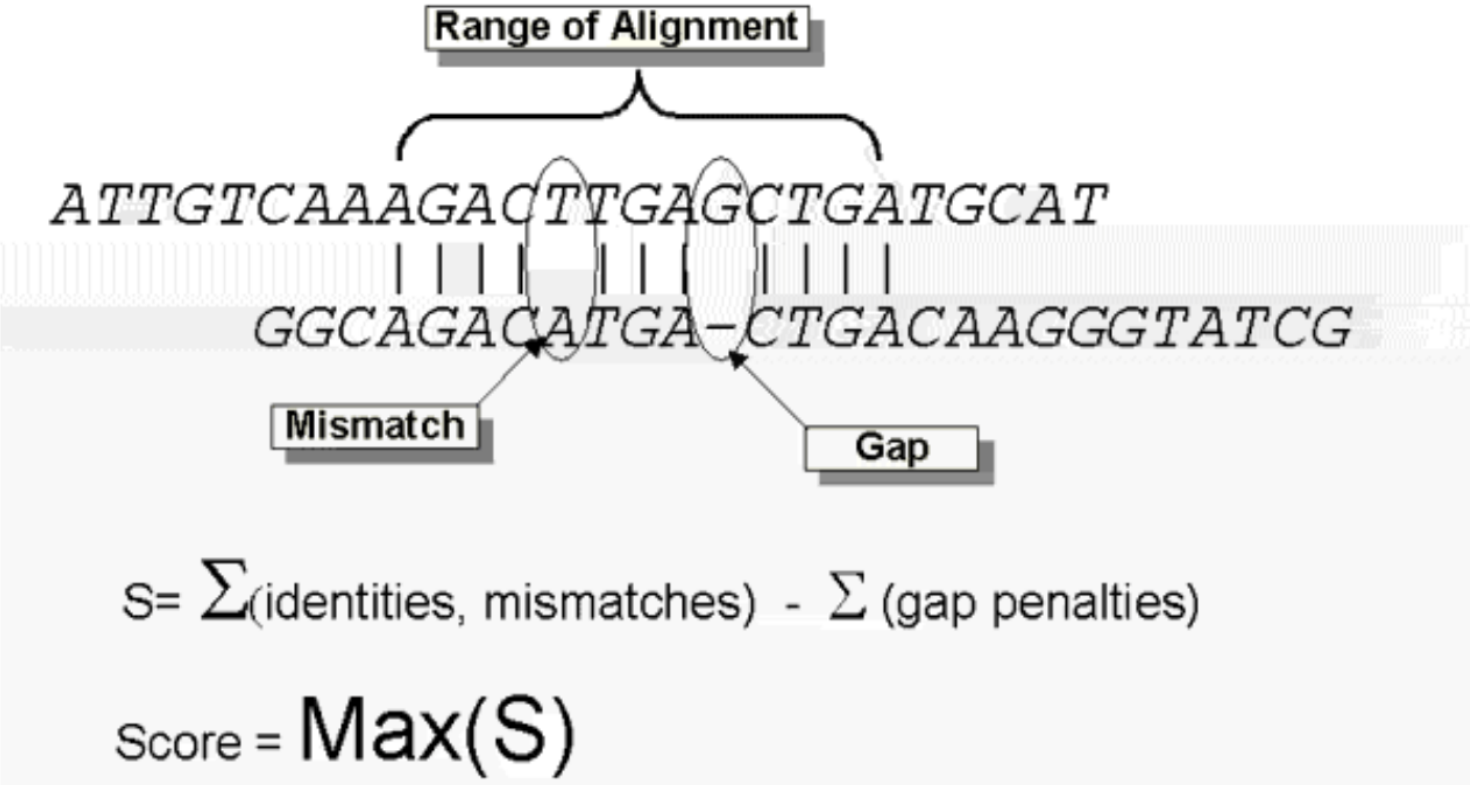
```
VAHV---D--DMPNALSALSDLHAHKL
AIQLQVTGVVVTDATLKNLGSVHVSKG
```

DIFFERENT METHODS USE DIFFERENT MATHEMATICAL NOTATION

SCORING ALIGNMENTS

- Alignment score
  - Score of an alignment is the sum of the scores for pairs of aligned characters plus the scores for gaps
  - Score = s(V,A)+ s(A,I) + s(H,Q) + s(V,L) + 3g + s(D,G) + 2g ...

PAIRWISE SEQUENCE ALIGNMENT



Source: http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Alignment_Scores2.html

SCORING ALIGNMENTS

- DNA has simpler scoring functions
  - Consider match and mismatch
- Proteins have a more complex scoring function
  - Some amino acid pairs might be more substitutable versus others
  - Scores might capture
    - Similar physical and chemical properties
    - Evolutionary relationships

Ala	4					
Arg	-1	5				
Asn	-2	0	6			
Asp	-2	-2	1	6		
Cys	0	-3	-3	-3	9	
Gln	-1	1	0	0	-3	
Glu	-1	0	0	2	-4	
Gly	0	-2	0	-1	-3	-
His	-2	0	1	-1	-3	
Ile	-1	-3	-3	-3	-1	-
Leu	-1	-2	-3	-4	-1	-
Lys	-1	2	0	-1	-3	
Met	-1	-1	-2	-3	-1	
Phe	-2	-3	-3	-3	-2	-
Pro	-1	-2	-2	-1	-3	-
Ser	1	-1	1	0	-1	
Thr	0	-1	0	-1	-1	-
Trp	-3	-3	-4	-4	-2	-
Tyr	-2	-2	-2	-3	-2	-
Val	0	-3	-3	-3	-1	-
	Ala	**Arg**	**Asn**	**Asp**	**Cys**	**Gl**

Substitution Matrices

SUBSTITUTION MATRICES

- Reflect the degree of similarity of each pair of molecule (base, amino acid)
- Estimate the likelihood that both are derived from the same molecule in the presumed common ancestral sequence

	Ala	Arg	Asn	Asp	Cys	Gl
Ala	4					
Arg	-1	5				
Asn	-2	0	6			
Asp	-2	-2	1	6		
Cys	0	-3	-3	-3	9	
Gln	-1	1	0	0	-3	
Glu	-1	0	0	2	-4	
Gly	0	-2	0	-1	-3	-
His	-2	0	1	-1	-3	
Ile	-1	-3	-3	-3	-1	-
Leu	-1	-2	-3	-4	-1	-
Lys	-1	2	0	-1	-3	
Met	-1	-1	-2	-3	-1	
Phe	-2	-3	-3	-3	-2	-
Pro	-1	-2	-2	-1	-3	-
Ser	1	-1	1	0	-1	
Thr	0	-1	0	-1	-1	-
Trp	-3	-3	-4	-4	-2	-
Tyr	-2	-2	-2	-3	-2	-
Val	0	-3	-3	-3	-1	-

SUBSTITUTION MATRICES

- A substitution matrix
  - Each R for R substitution has an associated score with it
  - Score values differ depending on methodology
- Protein substitution matrices have provide evolutionary distance

Ala	4					
Arg	-1	5				
Asn	-2	0	6			
Asp	-2	-2	1	6		
Cys	0	-3	-3	-3	9	
Gln	-1	1	0	0	-3	
Glu	-1	0	0	2	-4	
Gly	0	-2	0	-1	-3	
His	-2	0	1	-1	-3	
Ile	-1	-3	-3	-3	-1	
Leu	-1	-2	-3	-4	-1	
Lys	-1	2	0	-1	-3	
Met	-1	-1	-2	-3	-1	
Phe	-2	-3	-3	-3	-2	
Pro	-1	-2	-2	-1	-3	
Ser	1	-1	1	0	-1	
Thr	0	-1	0	-1	-1	
Trp	-3	-3	-4	-4	-2	
Tyr	-2	-2	-2	-3	-2	
Val	0	-3	-3	-3	-1	
	Ala	**Arg**	**Asn**	**Asp**	**Cys**	**Gl**

SUBSTITUTION MATRICES

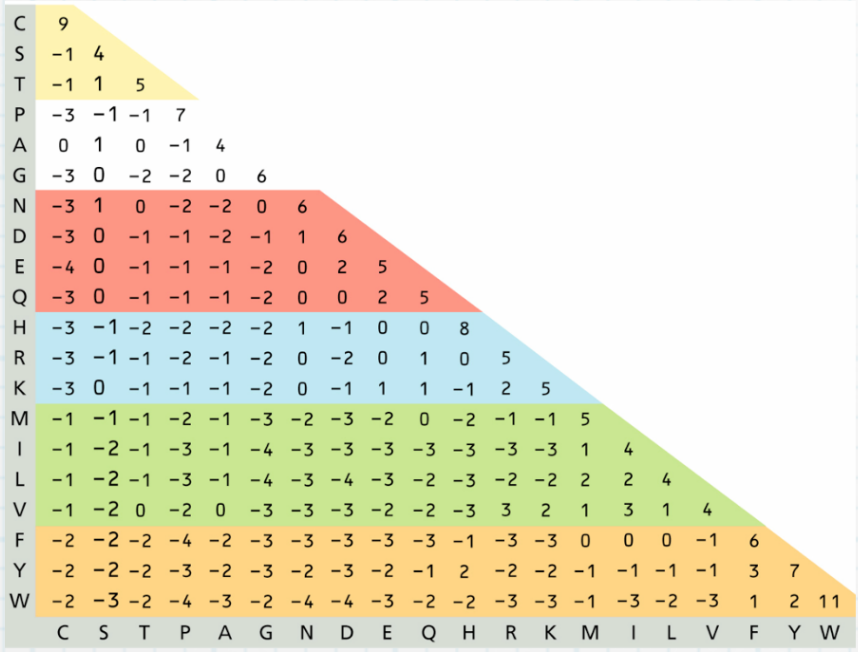
	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
	Ala	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**

SUBSTITUTION MATRICES

COMMONLY USED SUBSTITUTION MATRICES

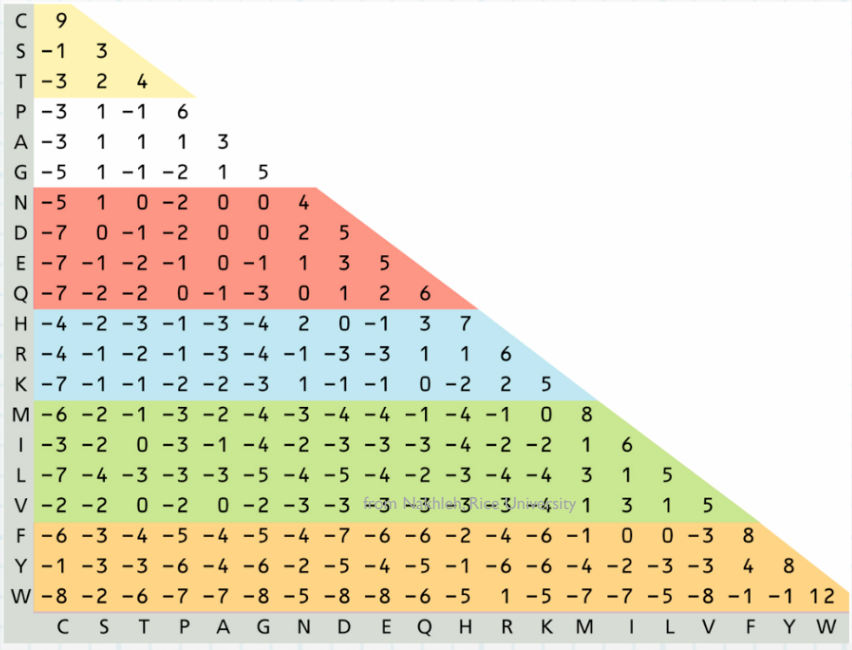
- PAM (Point Accepted Mutation)
  - Derived based on substitution frequencies in sets of closely related protein sequences
- BLOSUM (Blocks of Amino Acid Substitution Matrix)
  - Derived based on mutation data in highly conserved local regions of sequences

SUBSTITUTION MATRICES



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	3	2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

BLOSUM62



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	3																		
T	-3	2	4																	
P	-3	1	-1	6																
A	-3	1	1	1	3															
G	-5	1	-1	-2	1	5														
N	-5	1	0	-2	0	0	4													
D	-7	0	-1	-2	0	0	2	5												
E	-7	-1	-2	-1	0	-1	1	3	5											
Q	-7	-2	-2	0	-1	-3	0	1	2	6										
H	-4	-2	-3	-1	-3	-4	2	0	-1	3	7									
R	-4	-1	-2	-1	-3	-4	-1	-3	-3	1	1	6								
K	-7	-1	-1	-2	-2	-3	1	-1	-1	0	-2	2	5							
M	-6	-2	-1	-3	-2	-4	-3	-4	-4	-1	-4	-1	0	8						
I	-3	-2	0	-3	-1	-4	-2	-3	-3	-3	-4	-2	-2	1	6					
L	-7	-4	-3	-3	-3	-5	-4	-5	-4	-2	-3	-4	-4	3	1	5				
V	-2	-2	0	-2	0	-2	-3	-3	-3	-3	-3	-3	-4	1	3	1	5			
F	-6	-3	-4	-5	-4	-5	-4	-7	-6	-6	-2	-4	-6	-1	0	0	-3	8		
Y	-1	-3	-3	-6	-4	-6	-2	-5	-4	-5	-1	-6	-6	-4	-2	-3	-3	4	8	
W	-8	-2	-6	-7	-7	-8	-5	-8	-8	-6	-5	1	-5	-7	-7	-5	-8	-1	-1	12

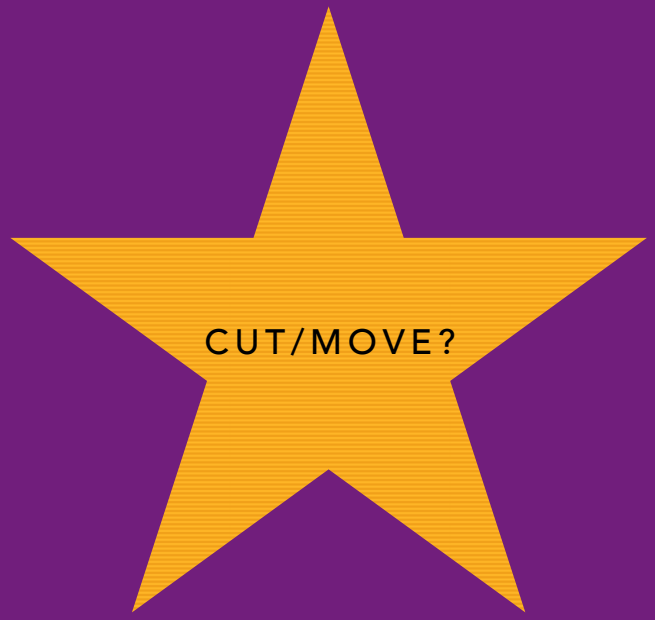
PAM120

SUBSTITUTION MATRICES

- Many other matrices have been developed
  - Specialized for families
  - Evolutionary times scales
- The matrix has profound influence on the outcome of alignment
  - Difficult to determine which matrix to use
  - Part of iterative process of investigation

SUBSTITUTION MATRICES

- General practice rules
  - Aligning distantly related sequences
    - PAM250 and BLOSUM-50 are preferable
  - Aligning closely related sequences
    - PAM120 and BLOSUM-80 may be better



GAPS

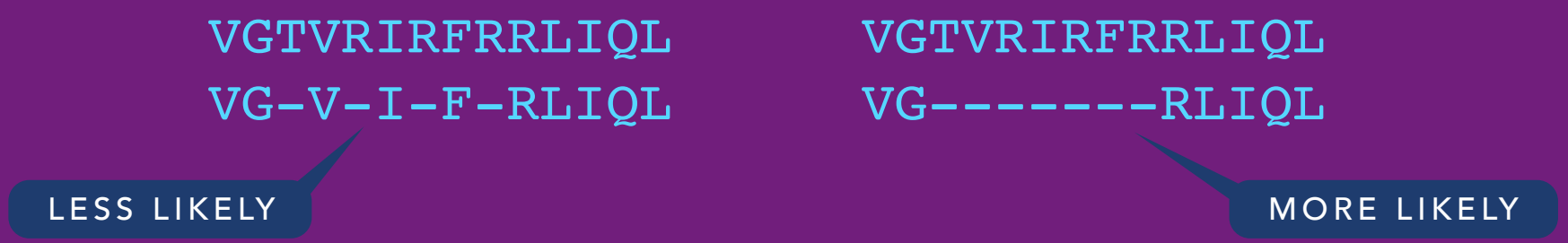
GAPS

- Homologous sequences are often of different lengths as the result of insertions and deletions (indels)
  - Occur as they diverged from the ancestral sequences
- Alignment is generally dealt with by allowing insertion of gaps in the sequences
  - Gaps must be introduced judiciously
- To place limits on the introduction of gaps, alignment programs use a “gap penalty”
  - Each time a gap is introduced, the penalty is subtracted from the score

COMMONLY USED TERM

GAPS

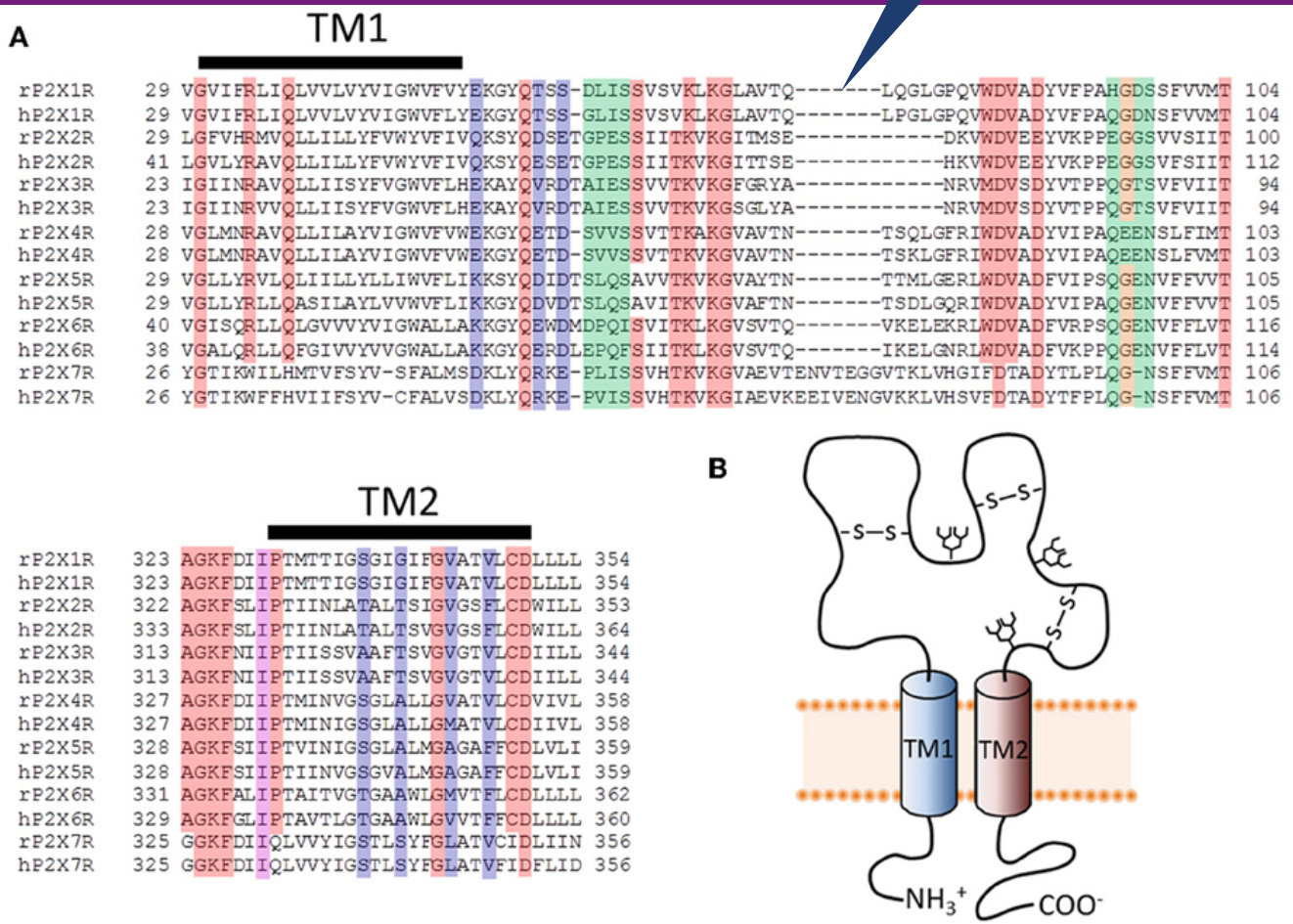
- Structural analysis has shown
  - Fewer indels occur in sequences of structural importance
  - Insertions tend to be several residues long rather than just a single residue long



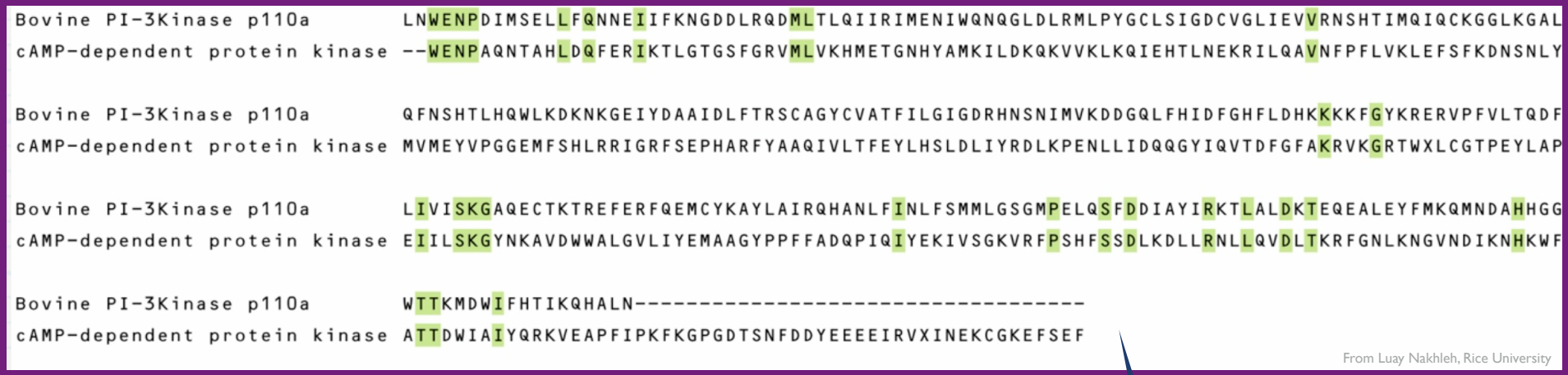
- This informs what gap penalty model to define
  - Different gap penalty models may result in different alignments

GAPS

LARGE GAPS IN NON-FUNCTIONAL REGION



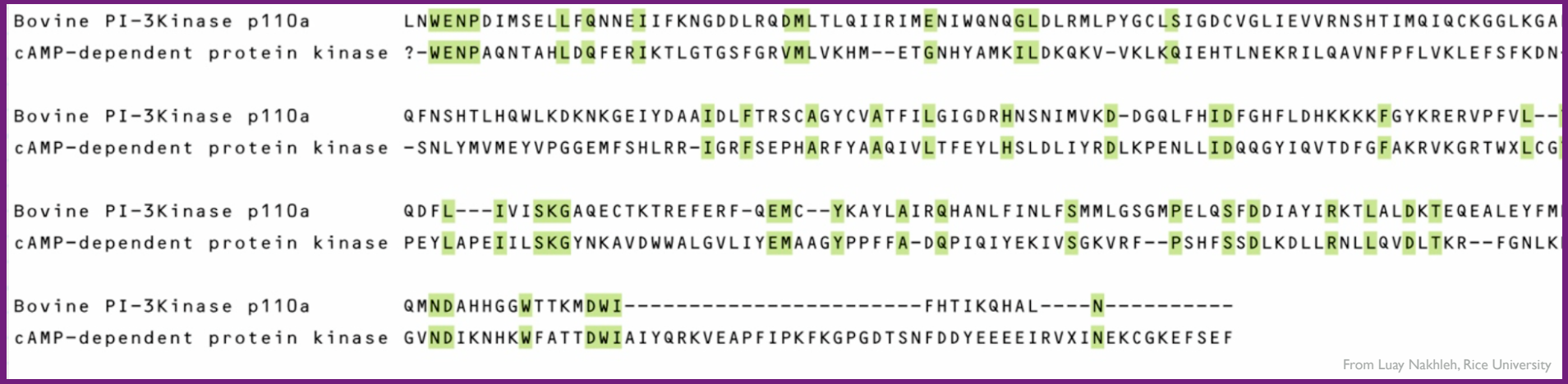
GAPS



- Very high gap penalty results in gaps only at beginning and end
  - 10% sequence identity

COST OF INTRODUCING GAPS IS TOO HIGH

GAPS



From Luay Nakhleh, Rice University

- Very low gap penalty results in many more gaps
  - 18% sequence identity

GAPS

PENALTY FUNCTIONS

- Linear score
  - $\gamma(g)=-dg$, where d is a fixed cost, g is the gap length
  - Longer the gap higher the penalty
- Affine score
  - $\gamma(g)=-(d+(g-1)e)$, d and e are fixed penalties and $e<d$
  - d : gap open penalty
  - e : gap extension penalty

```
VGTVRIRFRRLIQL
VG-------RLIQL
```

LINEAR
7 X -4

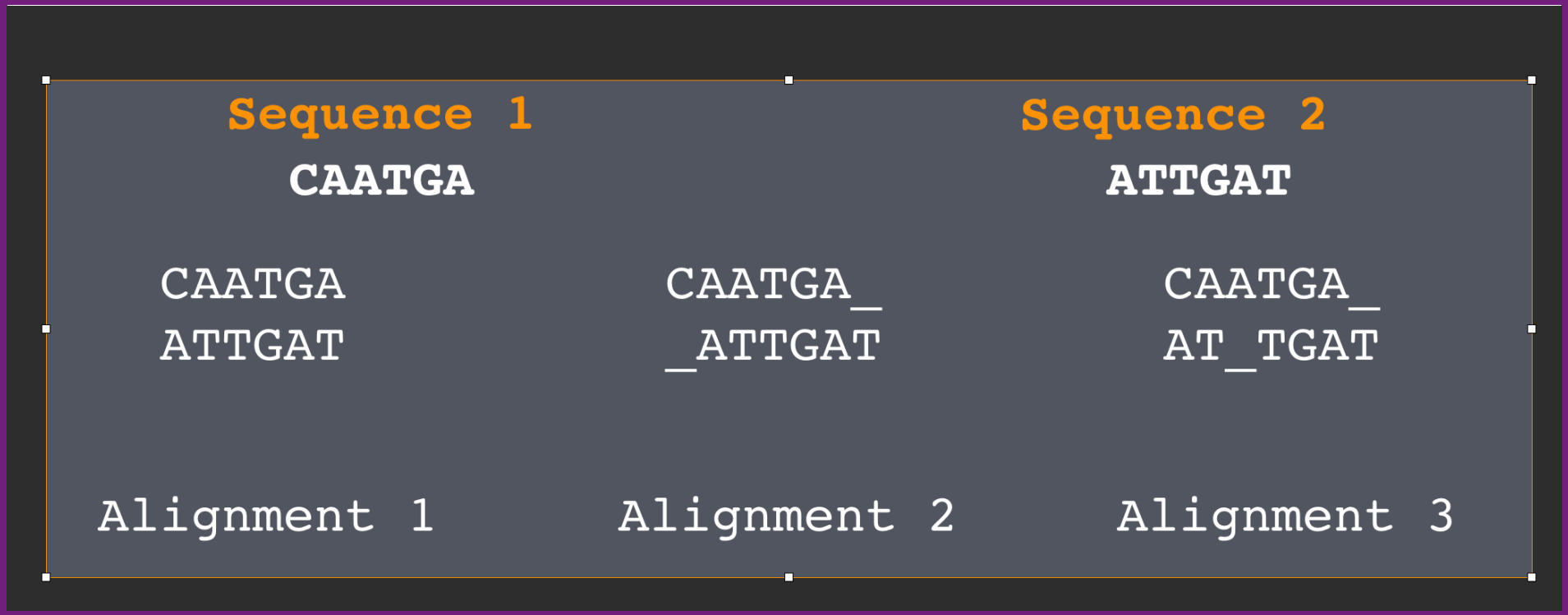
```
VGTVRIRFRRLIQL
VG-------RLIQL
```

AFFINE
(1 X -4) + (6 X -1)


SEQUENCE
ALIGNMENT
ALGORITHMS

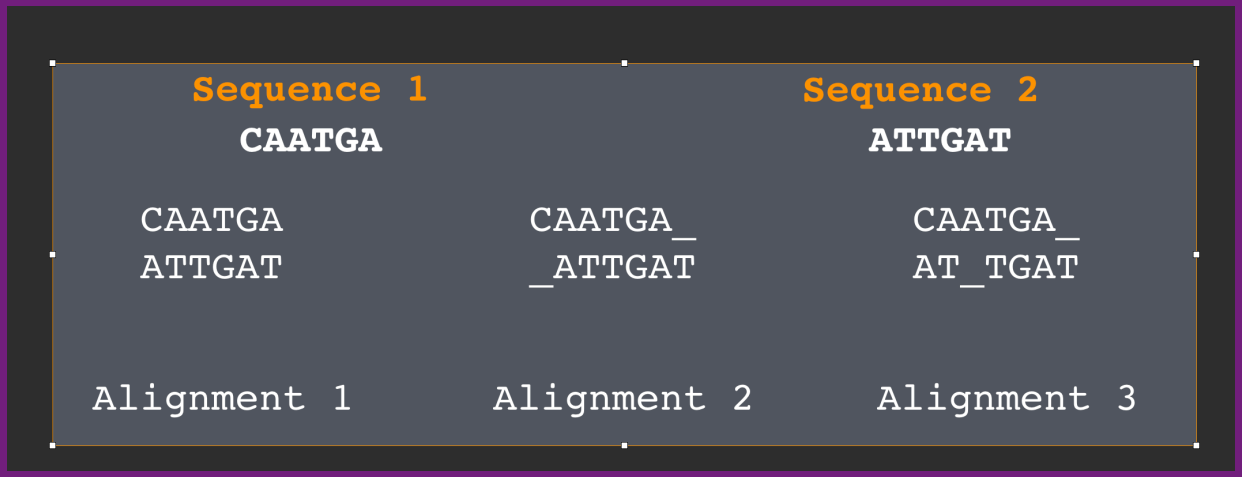
Dynamic Programming

DYNAMIC PROGRAMMING



- Given two sequences u and v and a scoring function find the alignment with the maximal score

DYNAMIC PROGRAMMING



- Number of possible alignments can be huge
- Not necessarily unique

DYNAMIC PROGRAMMING

"N CHOOSE R"; BINOMIAL COEFFICIENT; NO REPEATS ORDER INDEPENDENT

$$\frac{n!}{r!(n-r)!} = \binom{n}{r}$$

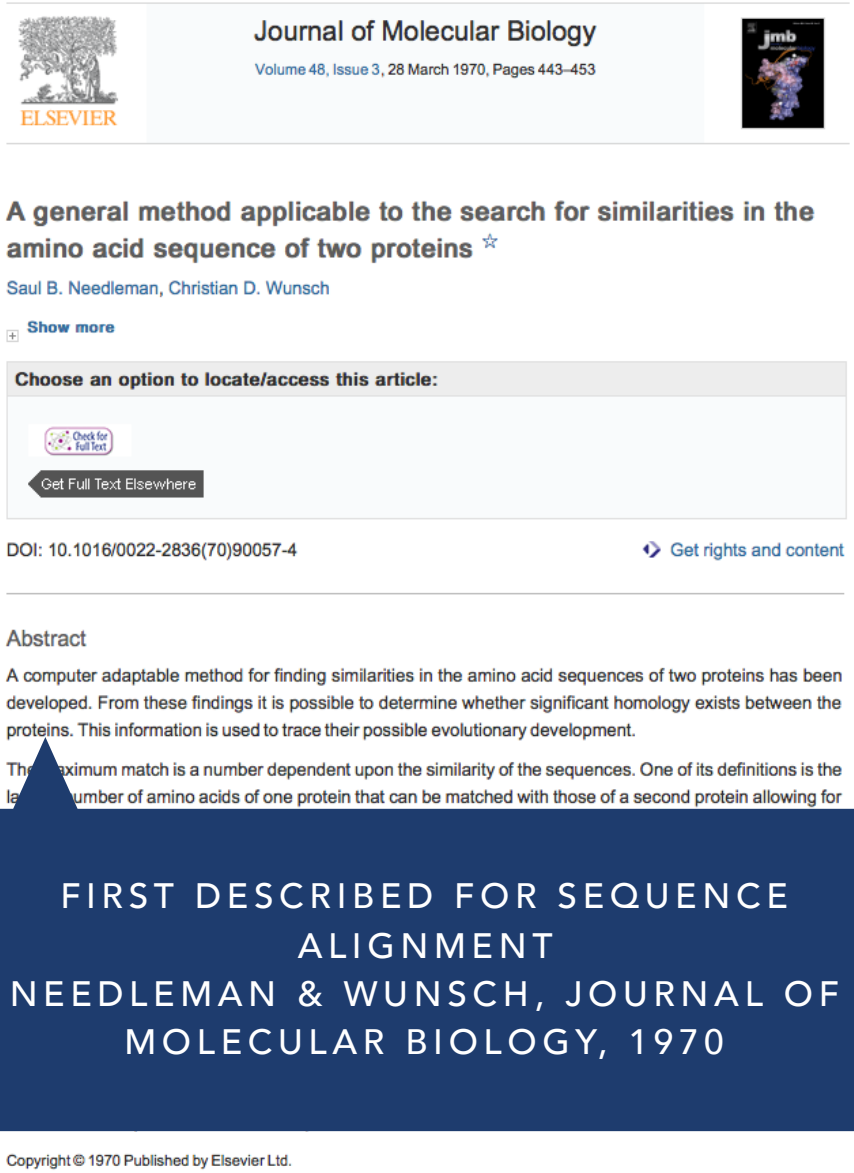
- There are

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

- Possible global alignments for 2 sequences of length n
  - Two sequences of length 100 have ~1077 possible alignments
- Dynamic programming can find an optimal alignment efficiently

DYNAMIC PROGRAMMING

- Divide and conquer
- Reduce problem of best alignment of two sequences to best alignment of all prefixes of the sequences
  - "recurrence relation"
- Avoid recalculating the scores already considered



Journal of Molecular Biology
Volume 48, Issue 3, 28 March 1970, Pages 443–453

A general method applicable to the search for similarities in the amino acid sequence of two proteins ☆

Saul B. Needleman, Christian D. Wunsch

Show more

Choose an option to locate/access this article:

Get Full Text Elsewhere

DOI: 10.1016/0022-2836(70)90057-4

Get rights and content

Abstract

A computer adaptable method for finding similarities in the amino acid sequences of two proteins has been developed. From these findings it is possible to determine whether significant homology exists between the proteins. This information is used to trace their possible evolutionary development.

The maximum match is a number dependent upon the similarity of the sequences. One of its definitions is the largest number of amino acids of one protein that can be matched with those of a second protein allowing for

FIRST DESCRIBED FOR SEQUENCE ALIGNMENT
NEEDLEMAN & WUNSCH, JOURNAL OF MOLECULAR BIOLOGY, 1970

Copyright © 1970 Published by Elsevier Ltd.

DYNAMIC PROGRAMMING

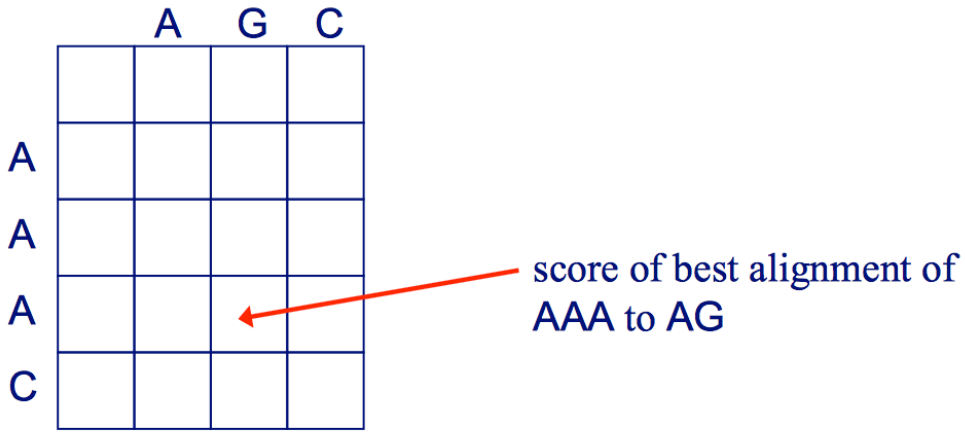
- Two sequences: AAAC, AAG
- x: AAAC
  - x_i: Denotes the ith letter
- y: AAG
  - y_i denotes the j^{th} letter in y
  - Assume cost of gap is *d*
- Assume we have aligned $x_1..x_i$ to $y_1..y_j$
- There are many possibilities

```
AAA  AA_  AAA
AAG  AAG  AA_

 AAC   AAC
 AA_   AAG
```

DYNAMIC PROGRAMMING

- Given 2 sequences: AAAC, AAG
  - x: AAAC
    - x_i: Denotes the ith letter
  - y: AAG
    - y_i denotes the j^{th} letter in y
  - Let x's length be n
  - Let y's length be m
  - Construct a (n+1) × (m+1) matrix, F
  - F (i,j) = score of the best alignment of x1…xi with y1…yj



DYNAMIC PROGRAMMING

CANONICAL ALGORITHMS FOR SEQUENCE ALIGNMENT

- Global alignment (Needleman-Wunsch algorithm)
  - Align the entire sequence
- Local alignment (Smith-Waterman algorithm)
  - Align part of the sequence

Global Alignment

GLOBAL ALIGNMENT

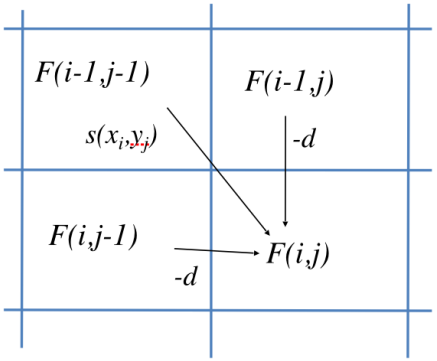
$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

MATCH X_i WITH Y_i

INSERTION IN X

INSERTION IN Y

Score of the best partial alignment between $x_1..x_i$ and $y_1..y_j$

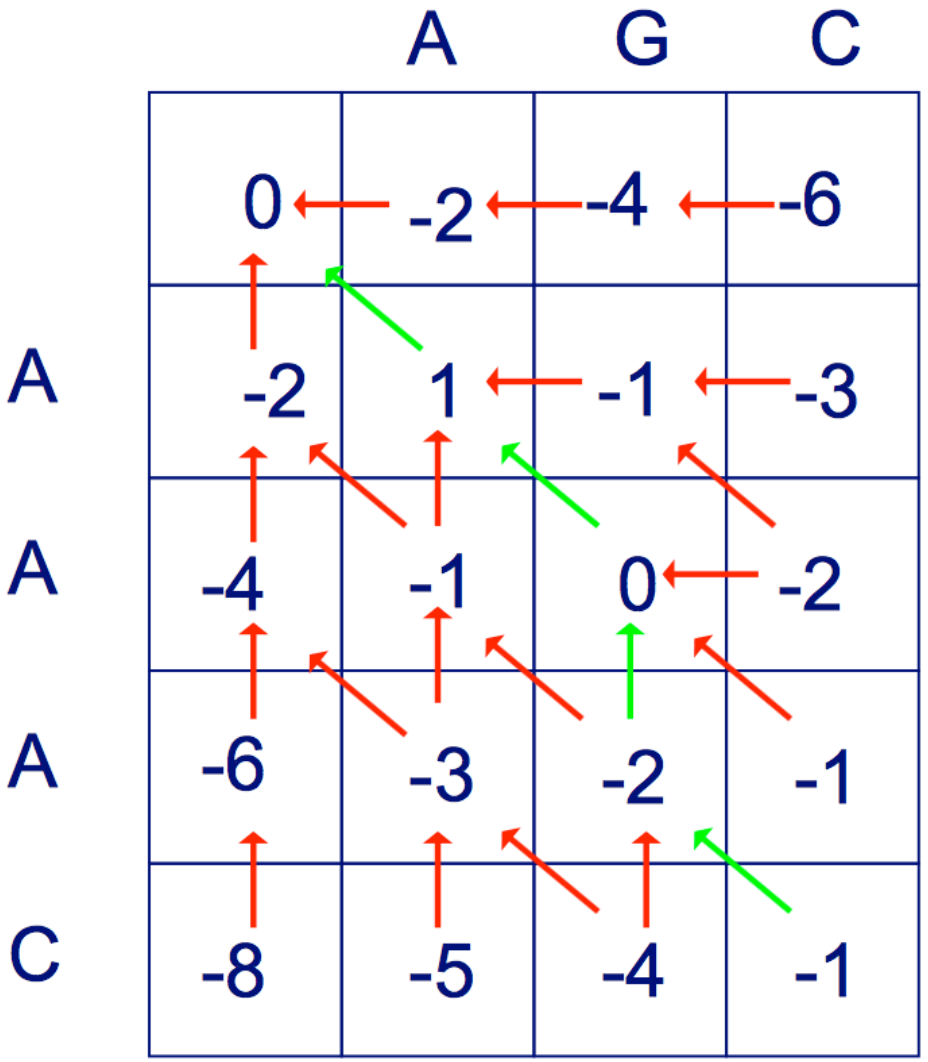


- Needleman-Wunch - dynamic programming for global alignment with linear gap penalty

GLOBAL ALIGNMENT

ALGORITHM

- Initialize first row and column of matrix
- Fill in rest of matrix from top to bottom, left to right
- For each F(i, j), save pointer(s) to cell(s) that resulted in best score
- F(m, n) holds the optimal alignment score
- Trace back from F(m, n) to F(0, 0) to recover alignment



GLOBAL ALIGNMENT

$$s(x_i, y_i) = \begin{cases} +1 \text{ when } x_i = y_i \\ -1 \text{ when } x_i \neq y_i \end{cases}$$

- Simple scoring scheme

GLOBAL ALIGNMENT

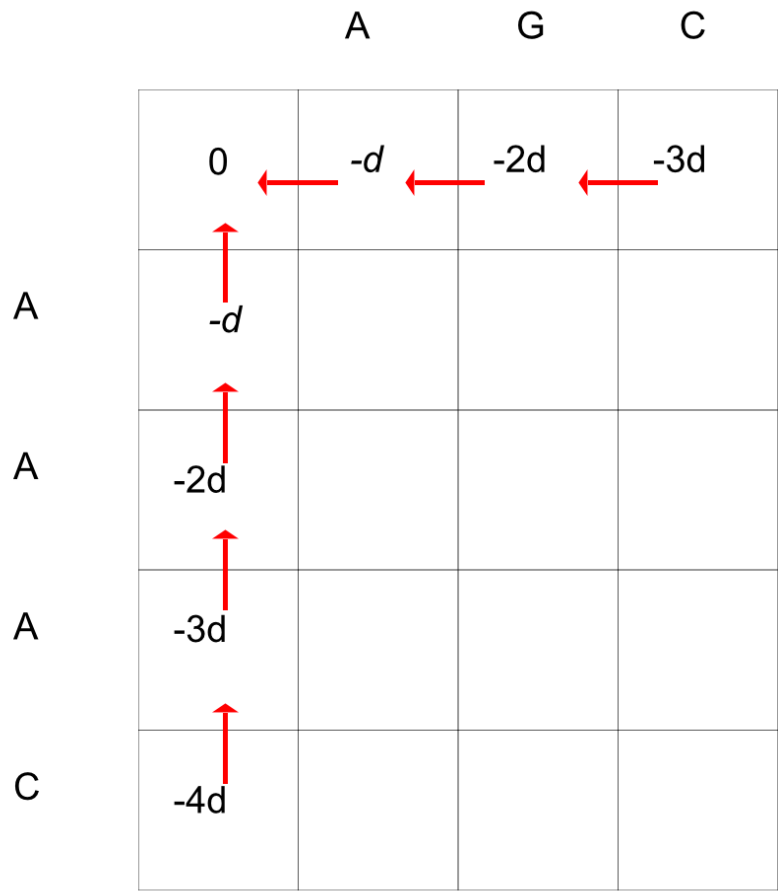
- Initializing the matrix with linear gapped penalty
- Represent complete gapped alignments

```
AGC-----
---AAAC
```

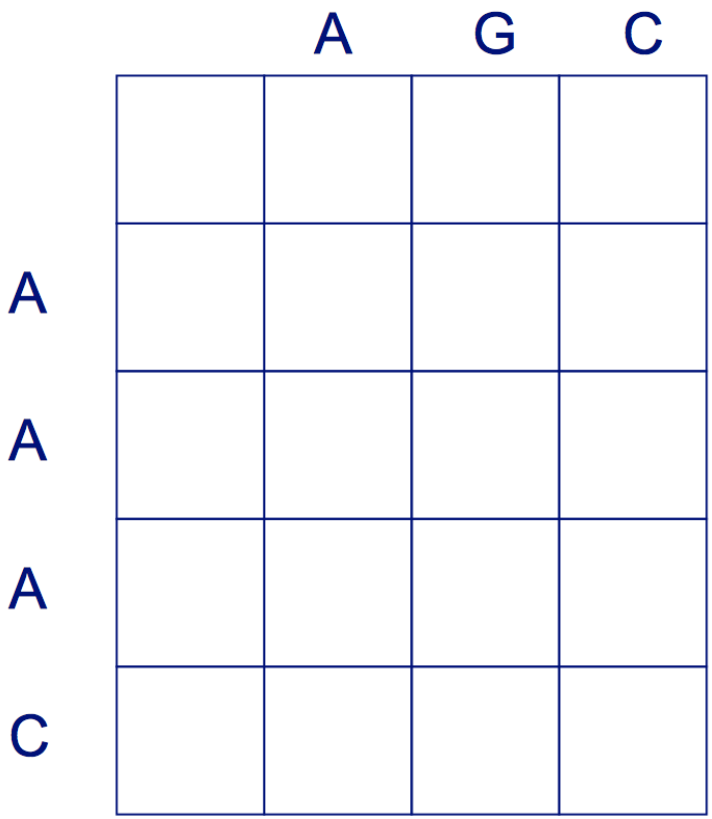
SCORE: -3D

```
-----AGC
AAAC---
```

SCORE: -4D



GLOBAL ALIGNMENT



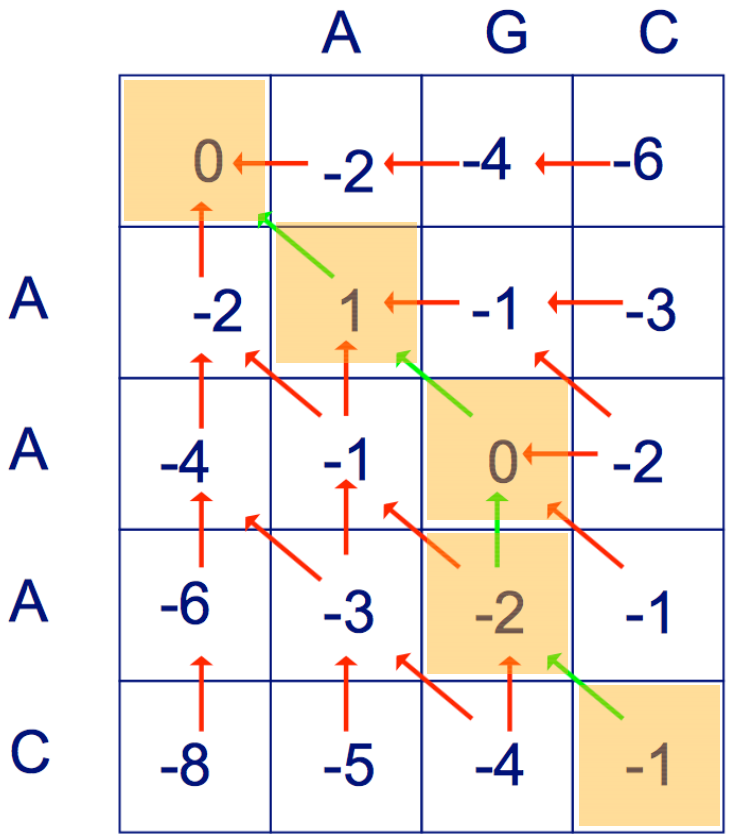
$s(x_i, y_i) =$

$+1$ when $x_i = y_i$

-1 when $x_i \neq y_i$

$g = -2$

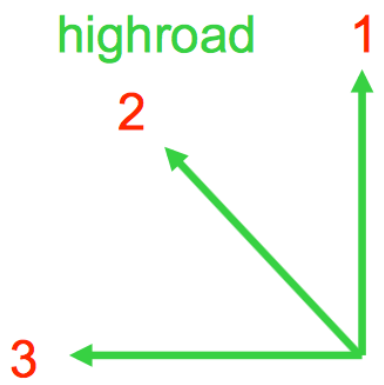
GLOBAL ALIGNMENT

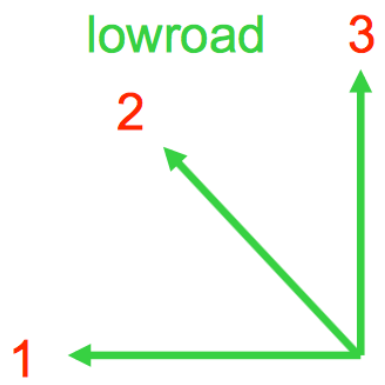


one optimal alignment

x: A A A C
y: A G - C

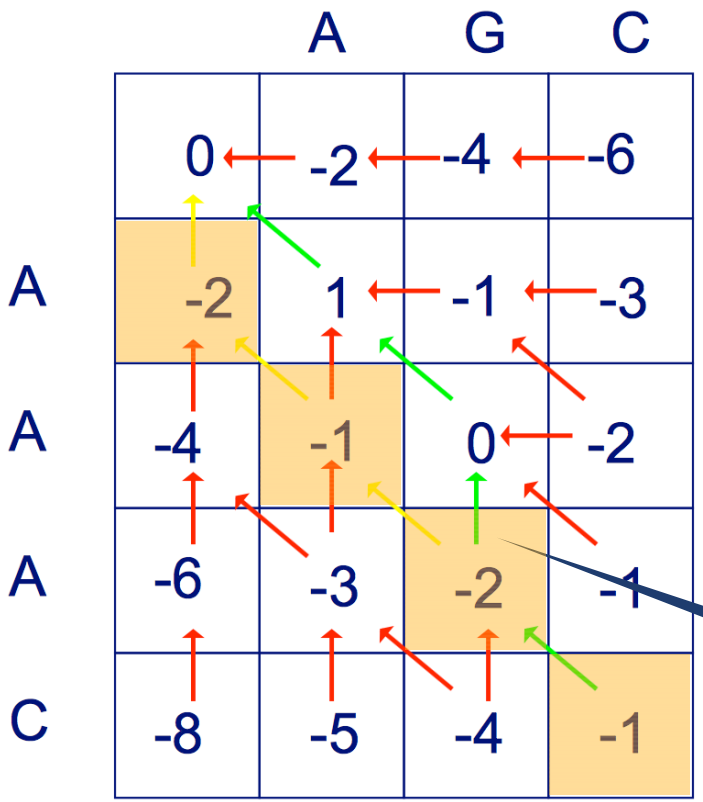
GLOBAL ALIGNMENT





- Equally Optimal Alignments
  - Many optimal alignments may exist for a given pair of sequences
  - Can use preference ordering over paths when doing traceback
- Highroad and lowroad alignments show the two most different optimal alignments

GLOBAL ALIGNMENT



highroad alignment

x: A A A C

y: A G - C

lowroad alignment

x: A A A C

y: - A G C

-2 CAN COME FROM DISTINCT PATHS

-2 CAN COME FROM DISTINCT PATHS

DYNAMIC PROGRAMMING

- Computational complexity of dynamic programming steps
  - Initialization: O(m), O(n) where sequence lengths are m, n
  - Filling in rest of matrix: O(mn)
  - Traceback: O(m + n)

- Since sequences have nearly same length, the computational complexity is $O(n^2)$

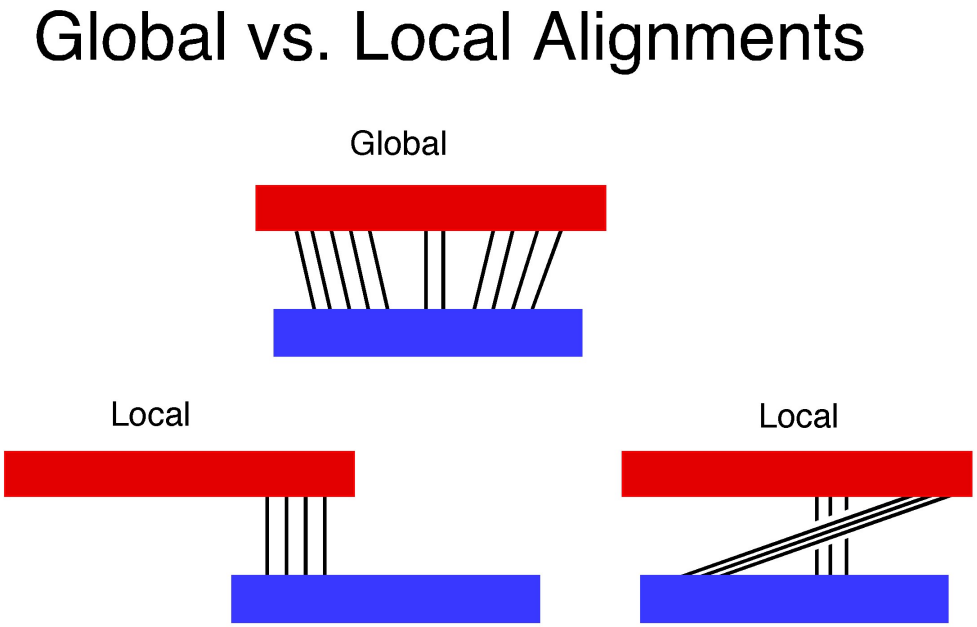
DYNAMIC PROGRAMMING

- Global alignment summary
  - Maximize F(i,j) using F(i-1,j-1), F(i-1,j) or F(i,j-1)
  - Works for either DNA or protein sequences
    - Scoring and substitutions differ
  - Finds an optimal alignment
  - Exact algorithm (and computational complexity) depends on gap penalty function

Local Alignment

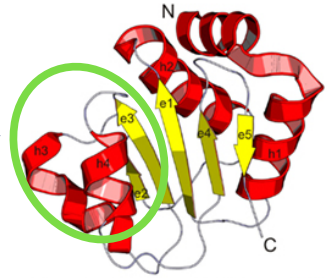
LOCAL ALIGNMENT

- Look for local regions of sequence similarity
- Aligning substrings of x and y
- Try to find the best alignment over all possible substrings

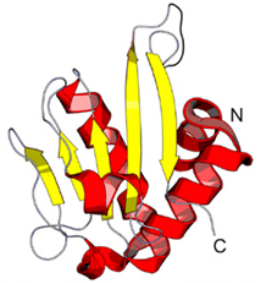


LOCAL ALIGNMENT

EXTRA DOMAIN COULD MASK SIMILARITIES



Typical DENN domain (PDB: 3TW8)



FLCN-DENN domain (PDB: 3V42)

LOCAL ALIGNMENT

- When is it most useful?
  - Proteins
    - Share a common motif (conserved pattern)
  - DNA
    - Share a similar sequence pattern but differ elsewhere
  - Genomic DNA sequences
    - Long stretches of uncharacterized genomic sequence
- More sensitive when comparing highly diverged sequences

Typical DENN domain (PDB: 3TW8)

FLCN-DENN domain (PDB: 3V42)

LOCAL ALIGNMENT

- Original formulation
  - Smith & Waterman, Journal of Molecular Biology, 1981
- Interpretation of matrix values is somewhat different
  - F (i, j) = score of the best alignment of
    - Suffix of x[1…i]
    - Suffix of y[1…j]

Identification of Common Molecular Subsequences

The identification of maximally homologous subsequences among sets of long sequences is an important problem in molecular sequence analysis. The problem is straightforward only if one restricts consideration to contiguous subsequences (segments) containing no internal deletions or insertions. The more general problem has its solution in an extension of sequence metrics (Sellers 1974; Waterman *et al.*, 1976) developed to measure the minimum number of "events" required to convert one sequence into another.

These developments in the modern sequence analysis began with the heuristic homology algorithm of Needleman & Wunsch (1970) which first introduced an iterative matrix method of calculation. Numerous other heuristic algorithms have been suggested including those of Fitch (1966) and Dayhoff (1969). More mathematically rigorous algorithms were suggested by Sankoff (1972), Reichert *et al.* (1973) and Beyer *et al.* (1979), but these were generally not biologically satisfying or interpretable. Success came with Sellers (1974) development of a true metric measure of the distance between sequences. This metric was later generalized by Waterman *et al.* (1976) to include deletions/insertions of arbitrary length. This metric represents the minimum number of "mutational events" required to convert one sequence into another. It is of interest to note that Smith *et al.* (1980) have recently shown that under some conditions the generalized Sellers metric is equivalent to the original homology algorithm of Needleman & Wunsch (1970).

In this letter we extend the above ideas to find a pair of segments, one from each of two long sequences, such that there is no other pair of segments with greater similarity (homology). The similarity measure used here allows for arbitrary length deletions and insertions.

Algorithm

The two molecular sequences will be $\underset{\sim}{A}=a_1a_2\ldots a_n$ and $\underset{\sim}{B}=b_1b_2\ldots b_m$. A similarity $s(a,b)$ is given between sequence elements a and b. Deletions of length k are given weight W_k. To find pairs of segments with high degrees of similarity, we set up a matrix H. First set

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m.$$

Preliminary values of H have the interpretation that H_{ij} is the maximum similarity of two segments *ending* in a_i and b_j, respectively. These values are obtained from the relationship

$$H_{ij}=\max\{H_{i-1,j-1}+s(a_i,b_j), \max_{k\geq 1}\{H_{i-k,j}-W_k\}, \max_{l\geq 1}\{H_{i,j-l}-W_l\}, 0\}, \quad (1)$$

$1\leq i\leq n$ and $1\leq j\leq m$.

0022-2836/80/090195-03 $02.00/0

© 1980 Academic Press Inc. (London) Ltd.

LOCAL ALIGNMENT

- Similar to global alignment with a few exceptions
- Dynamic programming recurrence is different

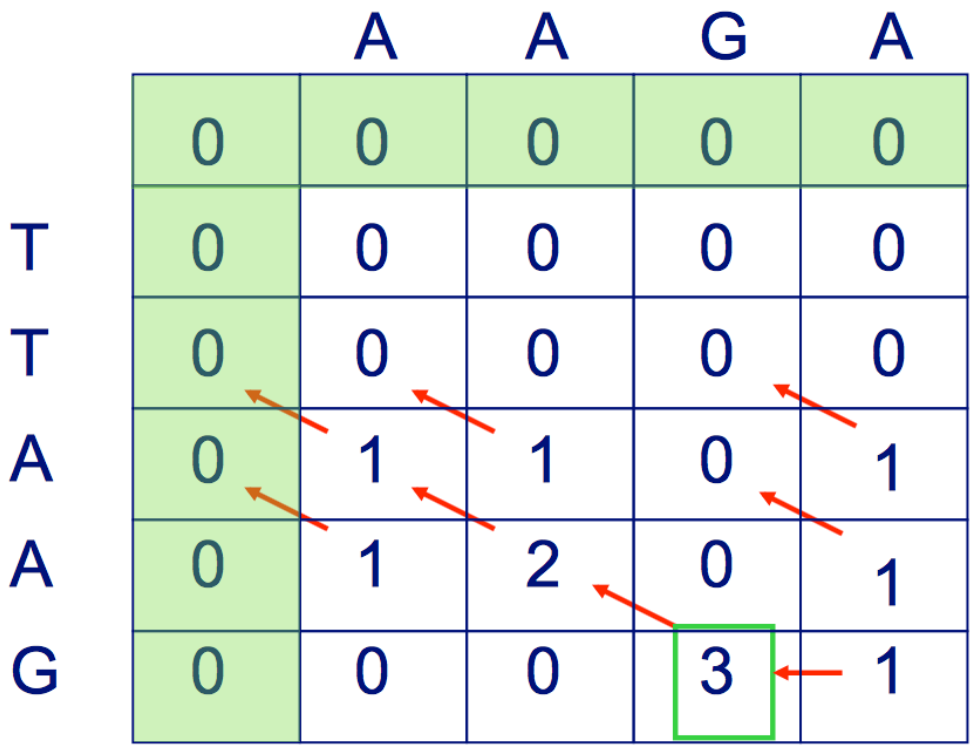
$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \\ 0 \end{cases}$$

New in local: starts a new alignment

- Top and left row are initialized to 0

LOCAL ALIGNMENT

- Initialization:
  - First row and first column initialized with 0's
- Traceback:
  - Start at maximum value of F(i, j)
    - Can be anywhere in matrix
  - Stop when we get to a cell with value 0



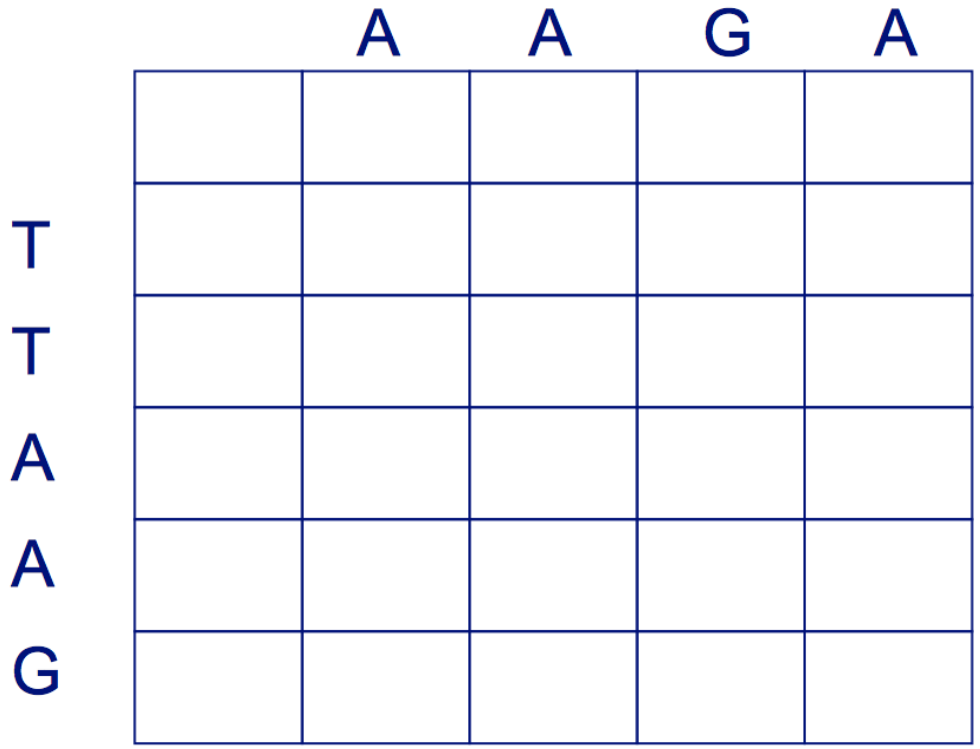
LOCAL ALIGNMENT

$s(x_i, y_i) =$

$+1$ when $x_i = y_i$

-1 when $x_i \neq y_i$

$g = -2$



LOCAL ALIGNMENT

x: A A G
y: A A G

		A	A	G	A
	0	0	0	0	0
T	0	0	0	0	0
T	0	0	0	0	0
A	0	1	1	0	1
A	0	1	2	0	1
G	0	0	0	3	1

TOP SCORE = 3

Gap Penalty Functions

GAP PENALTY FUNCTIONS

```
V-GVIF-RLIQLV-VLV-YVIEGWFVG-VYKQ-R-A-A-A-RQ
VGVIFRLI----------QLVVLVYVIEGWFVGVYKQRAAARQ
```

- A gap of length *k* is more probable than *k* gaps of length 1
  - A gap may be due to a single mutational event
  - Inserted/deleted a stretch of characters
- Separated gaps are probably due to distinct mutational events

GAP PENALTY FUNCTIONS

- A linear gap penalty function treats these cases the same
- Common to use an affine gap penalty function, which involves two terms
  - A initialization penalty, h, associated with starting a gap
  - A smaller penalty, g, for extending the gap

GAP PENALTY FUNCTIONS

LINEAR GAP PENALTY

$$w(k) = gk$$

AFFINE GAP PENALTY

$$w(k) = \begin{cases} h + gk, & k \geq 1 \\ 0, & k = 0 \end{cases}$$

STARTING

EXTENSION

GAP PENALTY FUNCTIONS

- Scoring requires keeping track of 3 matrices
  - Aligned
  - Gap
  - Extension

$M(i, j)$ best score given that $x[i]$ is aligned to $y[j]$

$I_x(i, j)$ best score given that $x[i]$ is aligned to a gap

$I_y(i, j)$ best score given that $y[j]$ is aligned to a gap

GAP PENALTY FUNCTIONS

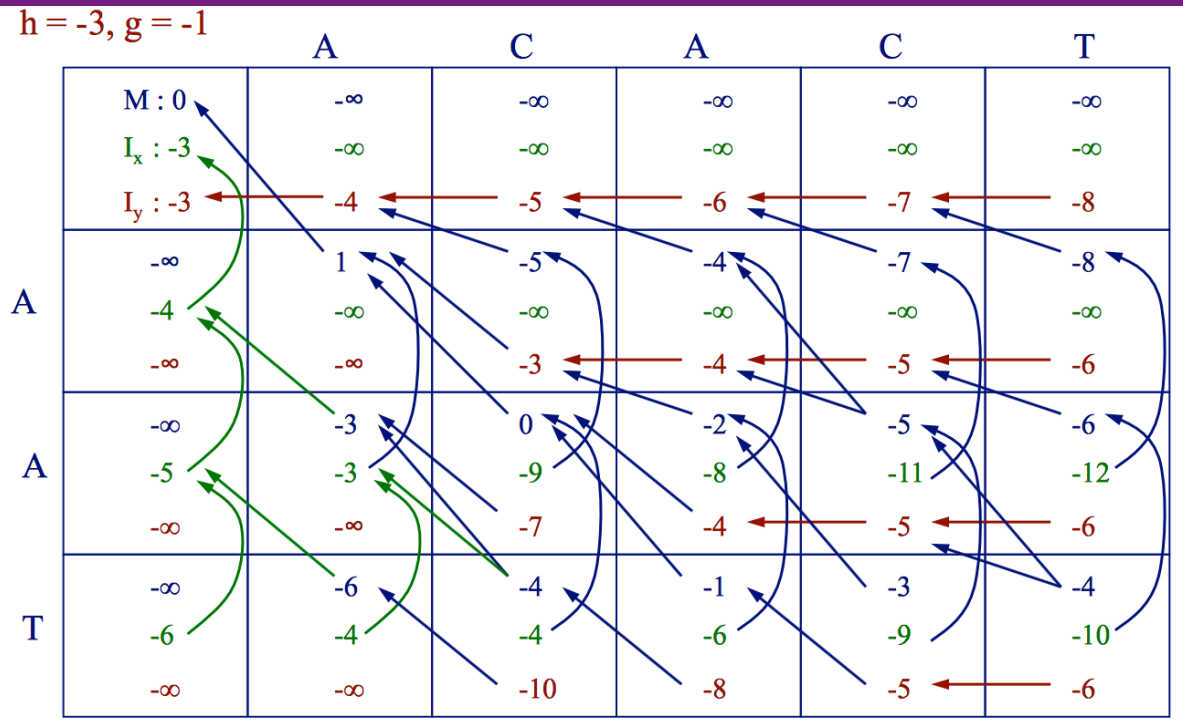
$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) & \text{match } x_i \text{ with } y_j \\ I_x(i-1,j-1) + s(x_i, y_j) & \text{insertion in } x \\ I_y(i-1,j-1) + s(x_i, y_j) & \text{insertion in } y \end{cases}$$

$$I_x(i,j) = \max \begin{cases} M(i-1,j) + h + g & \text{open gap in } x \\ I_x(i-1,j) + g & \text{extend gap in } x \end{cases}$$

$$I_y(i,j) = \max \begin{cases} M(i,j-1) + h + g & \text{open gap in } y \\ I_y(i,j-1) + g & \text{extend gap in } y \end{cases}$$

- Dynamic programming for global alignment with the affine gap penalty

GAP PENALTY FUNCTIONS



- The traceback can traverse all 3 matrices; whichever has the highest score

GAP PENALTY FUNCTIONS

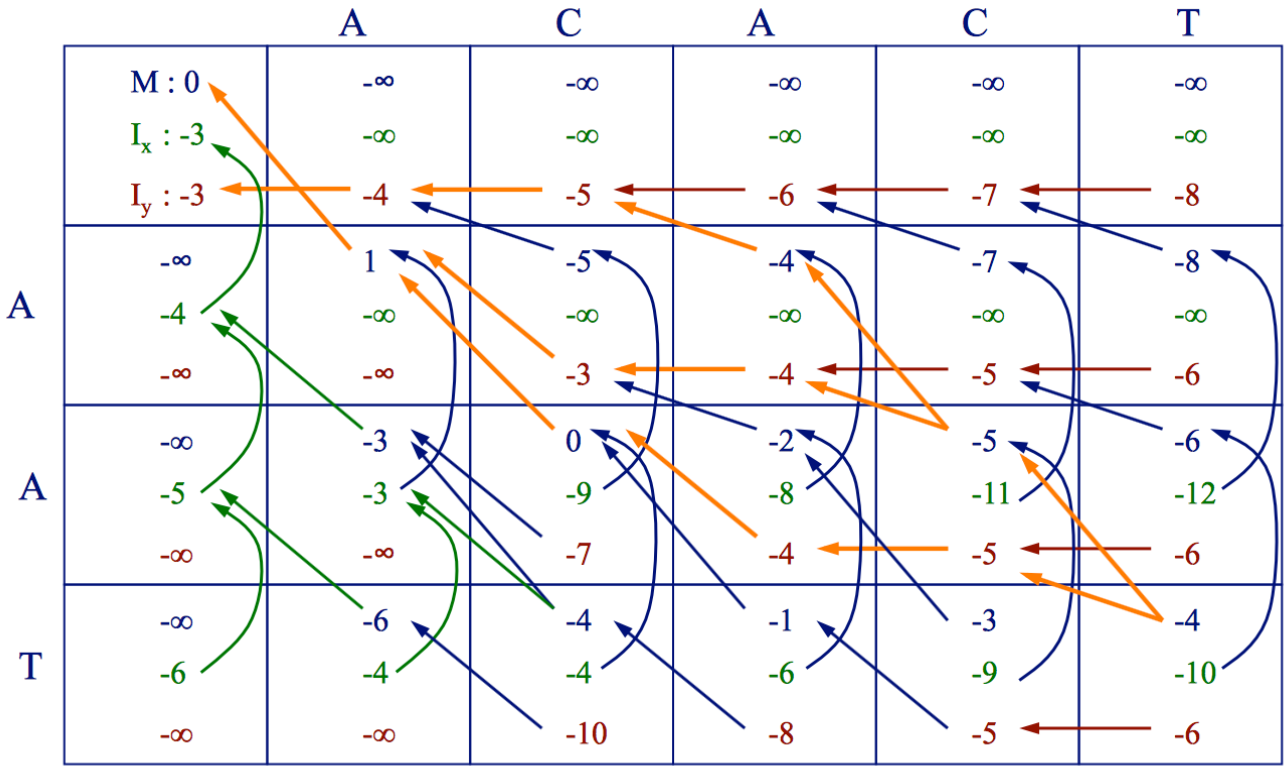
- Global alignment for affine gap penalty
  - Initialization

$$M(0,0) = 0$$
$$I_x(i,0) = h + g \times i$$
$$I_y(0,j) = h + g \times j$$
other cells in top row and leftmost column = $-\infty$

  - Traceback

– start at largest of $M(m,n), I_x(m,n), I_y(m,n)$
– stop at any of $M(0,0), I_x(0,0), I_y(0,0)$
– note that pointers may traverse all three matrices

GAP PENALTY FUNCTIONS



three optimal alignments:

```
ACACT   ACACT   ACACT
AA--T   A--AT   --AAT
```

GAP PENALTY FUNCTIONS

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ I_x(i-1,j-1) + s(x_i, y_j) \\ I_y(i-1,j-1) + s(x_i, y_j) \\ 0 \end{cases}$$

DON'T ALLOW NEGATIVE

$$I_x(i,j) = \max \begin{cases} M(i-1,j) + h + g \\ I_x(i-1,j) + g \end{cases}$$

$$I_y(i,j) = \max \begin{cases} M(i,j-1) + h + g \\ I_y(i,j-1) + g \end{cases}$$

- Dynamic programming for local alignment with affine gap penalty

GAP PENALTY FUNCTIONS

- Local alignment for affine gap penalty
  - Initialization

$$M(0,0) = 0$$
$$M(i,0) = 0$$
$$M(0,j) = 0$$
cells in top row and leftmost column of $I_x, I_y = -\infty$

  - Traceback

– start at largest $M(i,j)$
– stop at $M(i,j) = 0$


SUBSTITUTION
MATRICES

SUBSTITUTION MATRICES

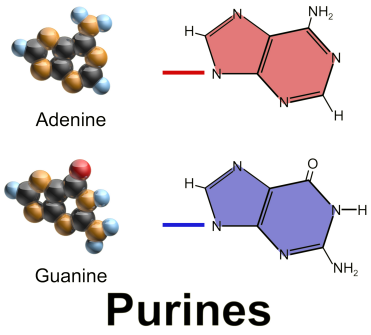
- Gap penalties strategies attempt to mimic evolutionary insertions/deletions
- Simple scoring scheme work only if assumption is that all mutations occur at the same frequency

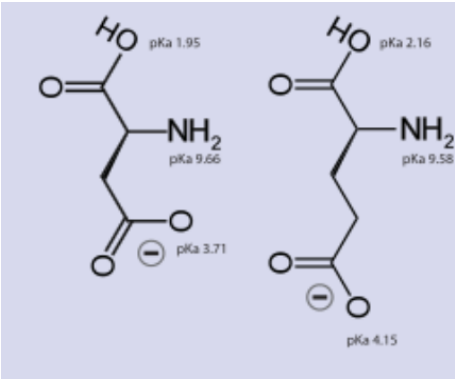
$$s(x_i, y_i) =$$
$$+1 \quad \text{when} \quad x_i = y_i$$
$$-1 \quad \text{when} \quad x_i \neq y_i$$

- Of course, this is not how nature works

SUBSTITUTION MATRICES

- Some substitutions are more common
  - DNA transitions are more common than transversions
    - Transversion (purine to pyrimidine; G<->A)
    - Transition (purine to purine; C<->T)
  - Example scoring matrix:
    - Identical=3, transition=2, transversions=0
  - Protein amino acids have similar properties and can be substituted





SUBSTITUTION MATRICES

DERIVING A SUBSTITUTION MATRIX

- Score attempts to measure the likelihood of a common evolutionary ancestor
- Consider the alignment of two residues from two sequences under two "competing" models
  - A random model, R
  - A match (non-random, evolutionary) model, M

SUBSTITUTION MATRICES

RANDOM MODEL (R)

- All sequences are assumed to be random selections from a given pool of residues
  - Every position in the sequence totally independent of every other
  - Fraction reproduced in the protein amino acid composition

- If the proportion of amino acid type *a* in the pool is p_a, the probability of residue *a* being aligned with residue *b* is simply paps

SUBSTITUTION MATRICES

THE MATCH MODEL (M)

- Sequences are related, due to an evolutionary process; infers a high correlation between aligned residues
- Probability of occurrence of particular residues depends on residue at the equivalent position in the sequence of the common ancestor
- Probability of residue, a, being aligned with residue, b, is $q_{a,b}$
  - Actual values of $q_{a,b}$ depend on the properties of the evolutionary process (and human interpretation)

SUBSTITUTION MATRICES

- The odds ratio
  - Random model
    - $P(a,b|R)=p_a p_b$
  - Matched Model
    - $P(a,b|M)=q_{a,b}$
  - These two models can be compared by taking the odds ratio:
    - Odds ratio = $(q_{a,b}/p_a p_b)$

$$\left(\frac{q_{a,b}}{p_a p_b}\right)$$

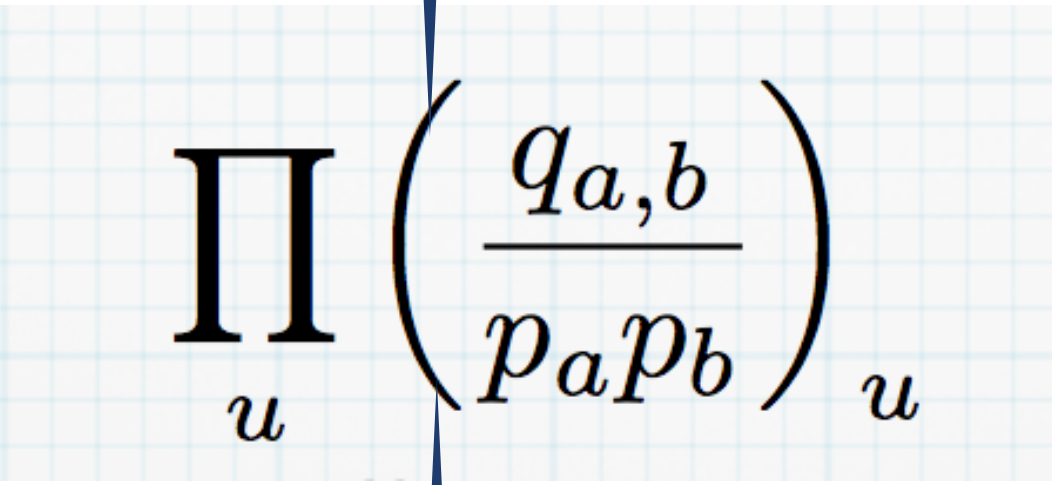
RATIO OF ALIGNED RESIDUE PAIR BEING RELATED BY EVOLUTION OVER RANDOM

SUBSTITUTION MATRICES

- The odds ratio for the entire alignment
  - Product of the odds ratios for the different positions

- Odds ratio > 1, the match model is more likely

ALIGNED RESIDUE PAIR BEING RELATED BY EVOLUTION



ALIGNED RESIDUE PAIR BEING RELATED BY RANDOM

SUBSTITUTION MATRICES

$\log_b(xy) = \log_b(x) + \log_b(y),$
The log of a product is the sum of log of its factors

- The log odds ratio
  - It is frequently more practical to deal with sums rather than products, especially when small numbers are involved
  - This can be achieved by taking logarithms of the odds ratio to give the log-odds ratio
  - This ratio can be summed over all positions of the alignment to give S, the score of the alignment:

$$S = \sum_u \log \left(\frac{q_{a,b}}{p_a p_b} \right)_u = \sum_u (s_{a,b})_u$$

- where $s_{a,b}$ is the substitution matrix element associated with the alignment of residue types a and b

SUBSTITUTION MATRICES

- The log odds ratio

$$S = \sum_u \log \left(\frac{q_{a,b}}{p_a p_b} \right)_u = \sum_u (s_{a,b})_u$$

- S is the relative likelihood of the whole alignment arising due to the match model as compared with the random model
  - A positive value of $s_{a,b}$ means that the probability of match is greater
  - A positive S is not a sufficient test of the alignment's significance

SUBSTITUTION MATRICES

- How do we get the frequencies of the matched model?

PAM Matrices

PAM MATRICES

- PAM (point accepted mutation) developed by Margaret Dayhoff, 1978
- Derived from global alignments of very similar sequences (85% identity)
  - Small likelihood of observed change being result of several mutations
    - Should reflect single one mutation only
- Carefully selected and studied 71 protein families

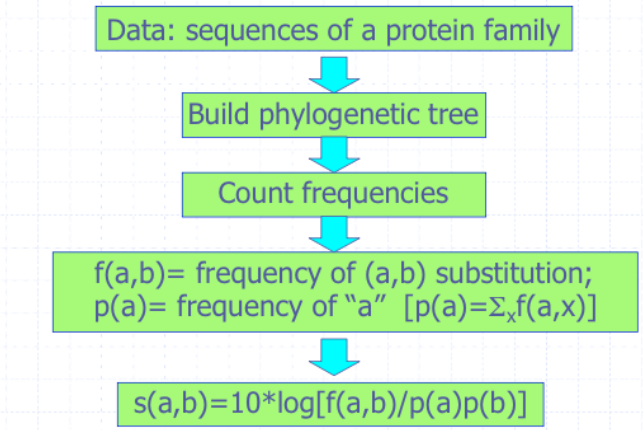
PAM MATRICES

DAYHOFF'S PROCEDURE

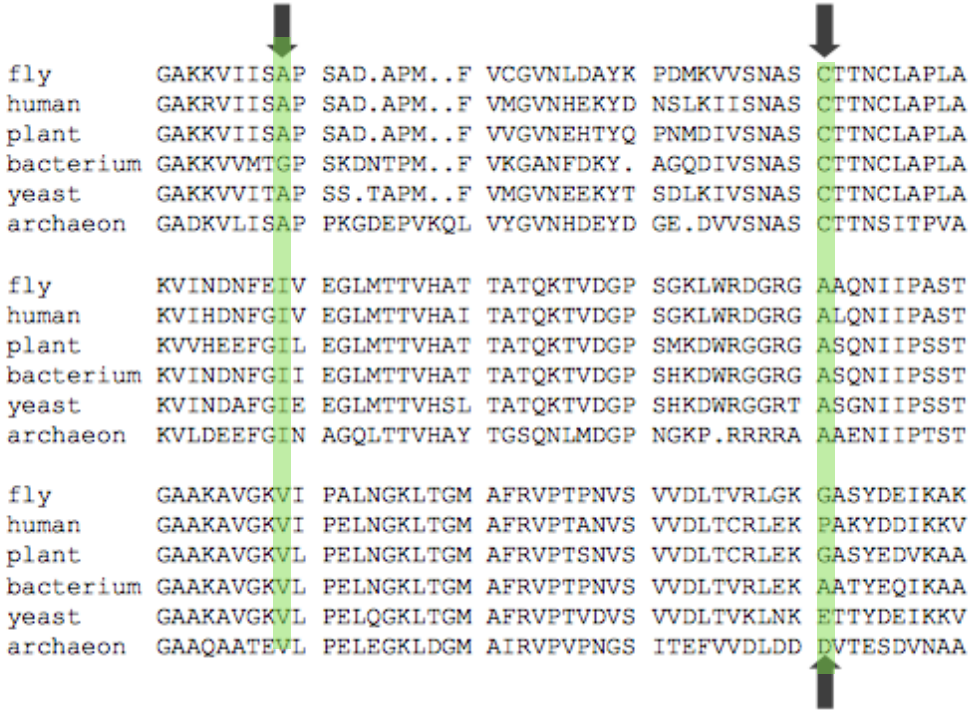
- Divide the set of sequences into groups of similar sequences
  - Construct a multiple alignment of each group
- Construct phylogenetic trees for each group
  - Estimate the mutations on the edges
- Define an evolutionary model to explain the evolution
- Construct substitution matrices
  - For an evolutionary interval, τ, given for each pair (a,b) of residues an estimate for the probability of a to mutate to b in a time interval τ
- Construct scoring matrices from the substitution matrices

PAM MATRICES

- Accepted point mutation (PAM)
- Silent mutation in gene template
  - Same amino acid
- Mutation in protein that retains function
  - Different but conserved amino acid



PAM MATRICES



```
fly        GAKKVIISAP SAD.APM..F VCGVNLDAYK PDMKVVSNAS CTTNCLAPLA
human      GAKRVIISAP SAD.APM..F VMGVNHEKYD NSLKIISNAS CTTNCLAPLA
plant      GAKKVIISAP SAD.APM..F VVGVNEHTYQ PNMDIVSNAS CTTNCLAPLA
bacterium  GAKKVVMTGP SKDNTPM..F VKGANFDKY. AGQDIVSNAS CTTNCLAPLA
yeast      GAKKVVITAP SS.TAPM..F VMGVNEEKYT SDLKIVSNAS CTTNCLAPLA
archaeon   GADKVLISAP PKGDEPVKQL VYGVNHDEYD GE.DVVSNAS CTTNSITPVA

fly        KVINDNFEIV EGLMTTVHAT TATQKTVDGP SGKLWRDGRG AAQNIIPAST
human      KVIHDNFGIV EGLMTTVHAI TATQKTVDGP SGKLWRDGRG ALQNIIPAST
plant      KVVHEEFGIL EGLMTTVHAT TATQKTVDGP SMKDWRGGRG ASQNIIPSST
bacterium  KVINDNFGII EGLMTTVHAT TATQKTVDGP SHKDWRGGRG ASQNIIPSST
yeast      KVINDAFGIE EGLMTTVHSL TATQKTVDGP SHKDWRGGRT ASGNIIPSST
archaeon   KVLDEEFGIN AGQLTTVHAY TGSQNLMDGP NGKP.RRRRA AAENIIPTST

fly        GAAKAVGKVI PALNGKLTGM AFRVPTPNVS VVDLTVRLGK GASYDEIKAK
human      GAAKAVGKVI PELNGKLTGM AFRVPTANVS VVDLTCRLEK PAKYDDIKKV
plant      GAAKAVGKVL PELNGKLTGM AFRVPTSNVS VVDLTCRLEK GASYEDVKAA
bacterium  GAAKAVGKVL PELNGKLTGM AFRVPTPNVS VVDLTVRLEK AATYEQIKAA
yeast      GAAKAVGKVL PELQGKLTGM AFRVPTVDVS VVDLTVKLNK ETTYDEIKKV
archaeon   GAAQAATEVL PELEGKLDGM AIRVPVPNGS ITEFVVDLDD DVTESDVNAA
```

- Alignment of glyceraldehyde 3-phosphatase dehydrogenase

from Bioinformatics and Functional Genomics, Pervner

PAM MATRICES

	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly
A								
R	30							
N	109	17						
D	154	0	532					
C	33	10	0	0				
Q	93	120	50	76	0			
E	266	0	94	831	0	422		
G	579	10	156	162	10	30	112	
H	21	103	226	43	10	243	23	10

- Number of accepted point mutations

from Bioinformatics and Functional Genomics, Pervner

PAM MATRICES

MUTATIONAL PROBABILITY MATRIX

- The probability that an original amino acid (top) will be replaced by another amino acid (side)

	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His
A	9867	2	9	10	3	8	17	21	2
R	1	9913	1	0	1	10	0	0	10
N	4	1	9822	36	0	4	6	6	21
D	6	0	42	9859	0	6	53	6	4
C	1	1	0	0	9973	0	0	0	1
Q	3	9	4	5	0	9876	27	1	23
E	10	0	7	56	0	35	9865	4	2
G	21	1	12	11	1	3	7	9935	1
H	1	8	18	3	1	20	1	0	9912
I	2	2	3	1	2	1	2	0	0

PAM MATRICES

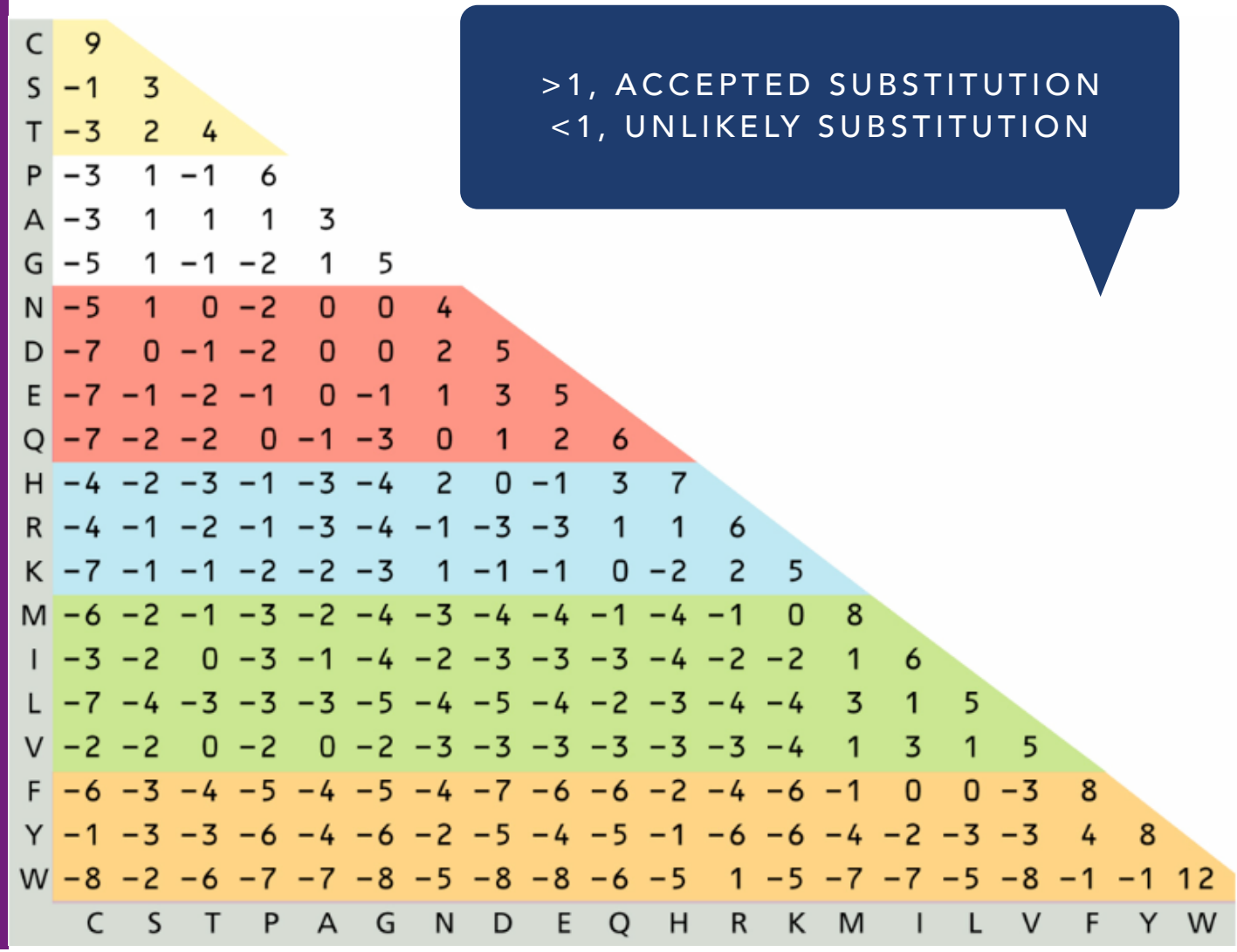
- PAM scoring matrix from probability matrix
  - Using the log-odds ratio
    - Divide the probability under the match model (given by the substitution matrix) by the probability under the random model

LOG ODDS/ RATIO

$$S_{ab} = \log \frac{M_{ab}}{p_b}$$

PAM MATRICES

- PAM 120



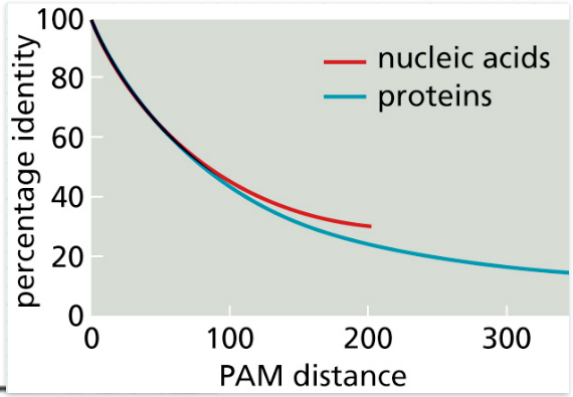
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	−1	3																		
T	−3	2	4																	
P	−3	1	−1	6																
A	−3	1	1	1	3															
G	−5	1	−1	−2	1	5														
N	−5	1	0	−2	0	0	4													
D	−7	0	−1	−2	0	0	2	5												
E	−7	−1	−2	−1	0	−1	1	3	5											
Q	−7	−2	−2	0	−1	−3	0	1	2	6										
H	−4	−2	−3	−1	−3	−4	2	0	−1	3	7									
R	−4	−1	−2	−1	−3	−4	−1	−3	−3	1	1	6								
K	−7	−1	−1	−2	−2	−3	1	−1	−1	0	−2	2	5							
M	−6	−2	−1	−3	−2	−4	−3	−4	−4	−1	−4	−1	0	8						
I	−3	−2	0	−3	−1	−4	−2	−3	−3	−3	−4	−2	−2	1	6					
L	−7	−4	−3	−3	−3	−5	−4	−5	−4	−2	−3	−4	−4	3	1	5				
V	−2	−2	0	−2	0	−2	−3	−3	−3	−3	−3	−3	−4	1	3	1	5			
F	−6	−3	−4	−5	−4	−5	−4	−7	−6	−6	−2	−4	−6	−1	0	0	−3	8		
Y	−1	−3	−3	−6	−4	−6	−2	−5	−4	−5	−1	−6	−6	−4	−2	−3	−3	4	8	
W	−8	−2	−6	−7	−7	−8	−5	−8	−8	−6	−5	1	−5	−7	−7	−5	−8	−1	−1	12

PAM MATRICES

- PAM1 matrix is calculated from comparisons of sequences with 1% divergence
  - 1 change over a length of 100 amino acids
- Other PAM matrices are extrapolated from PAM1
  - PAM120 - 120 changes occurred over length of 100 amino acids
  - PAM250 - 250 changes occurred over length of 100 amino acids

Correspondence between Observed Differences and the Evolutionary Distance

Observed Percent Difference	Evolutionary Distance in PAMs
1	1
5	5
10	11
15	17
20	23
25	30
30	38
35	47
40	56
45	67
50	
55	
60	
65	
70	
75	
80	
85	



BLOSUM MATRIX

BLOSUM MATRICES

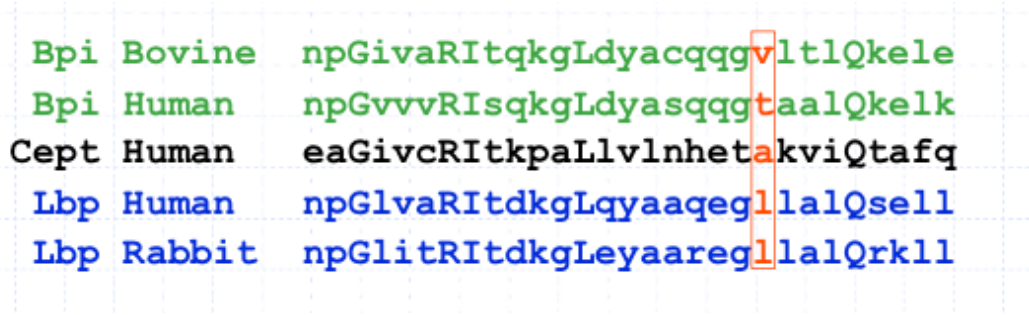
- BLOSUM (blocks substitution matrix) created by Henikoff & Henikoff
  - Based on local alignments of more distantly related proteins
  - Based on observed alignments
    - Not extrapolated from comparisons of closely related proteins
  - Underlying assumption
    - Families would have patterns that are conserved

BLOSUM MATRICES

- Multiple alignments of short regions (without gaps) or related sequences were gathered into BLOCKS database
  - Thousands of groups (as opposed to PAMs 71)
  - Similar sequences were clustered at thresholds of similarity
    - BLOSUM62 is a matrix calculated from comparisons of sequences with no less than 62% divergence.

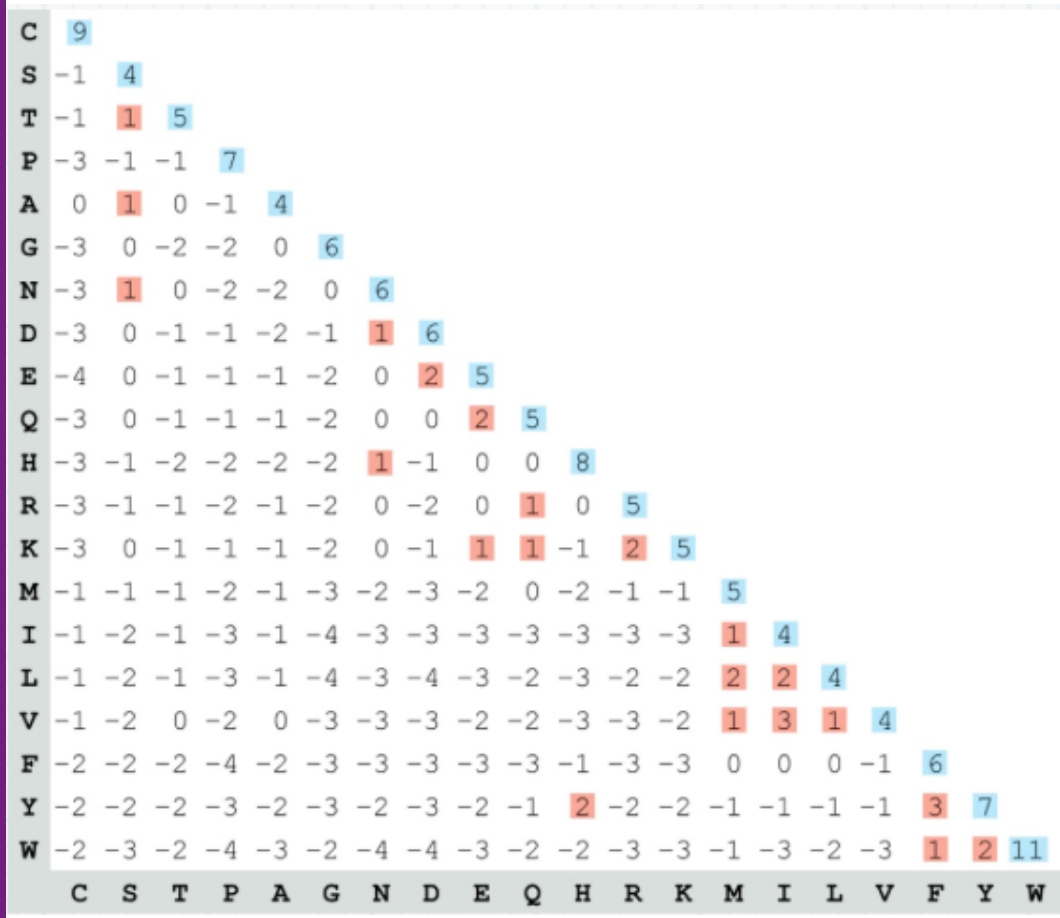
BLOSUM MATRICES

- Procedure:
  - Note the occurrences of amino acid pair, h_{ab}
  - Calculate the probability they are aligned by chance
    - Observed frequency >1 - biological relationship
    - Observed frequency <1 - no biological relationship (bad substitution)
    - Observed frequency =1 - neutral
  - Compute log odds matrix of probabilities of amino acid pairs



BLOSUM MATRICES

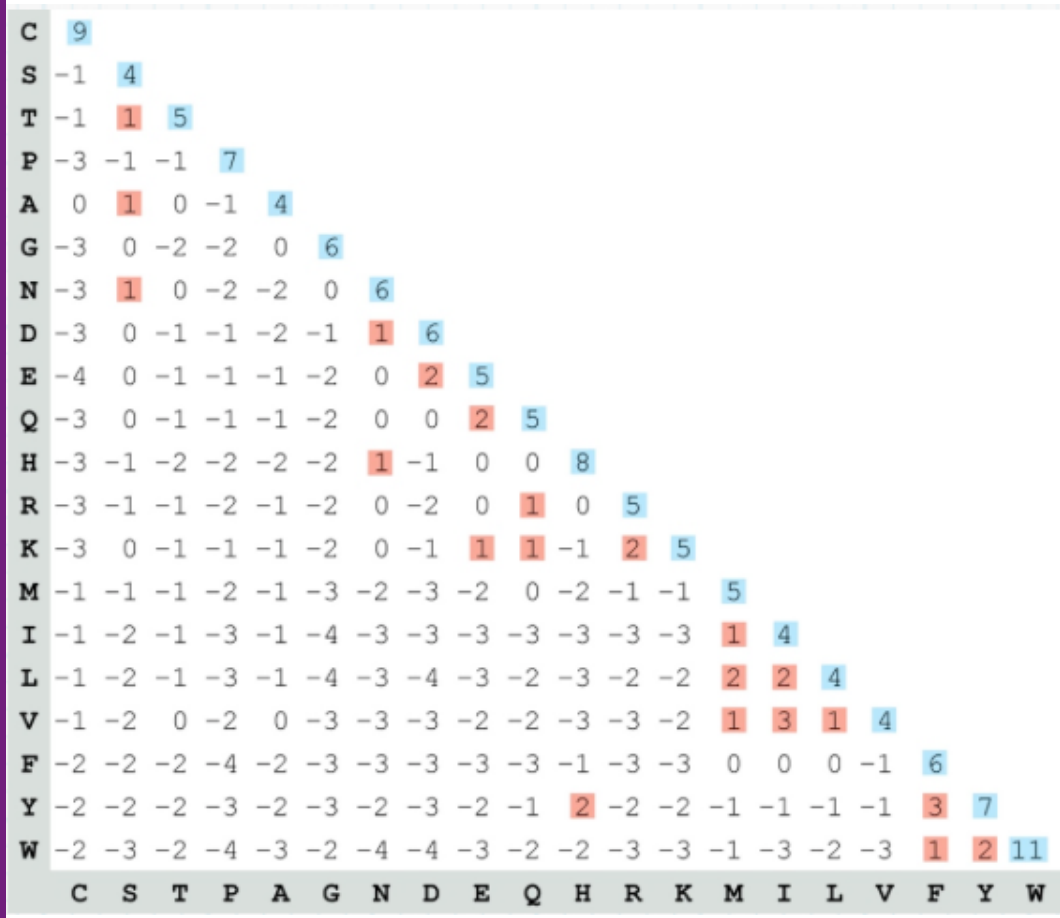
- BLOSUM-62 matrix is calculated from comparisons of sequences with no less than 62% divergence



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

BLOSUM MATRICES

- BLOSUM variants
  - Constrain sequences to percent identity
    - 62% = BLOSUM62
    - 80% = BLOSUM80
    - 25% = BLOSUM25



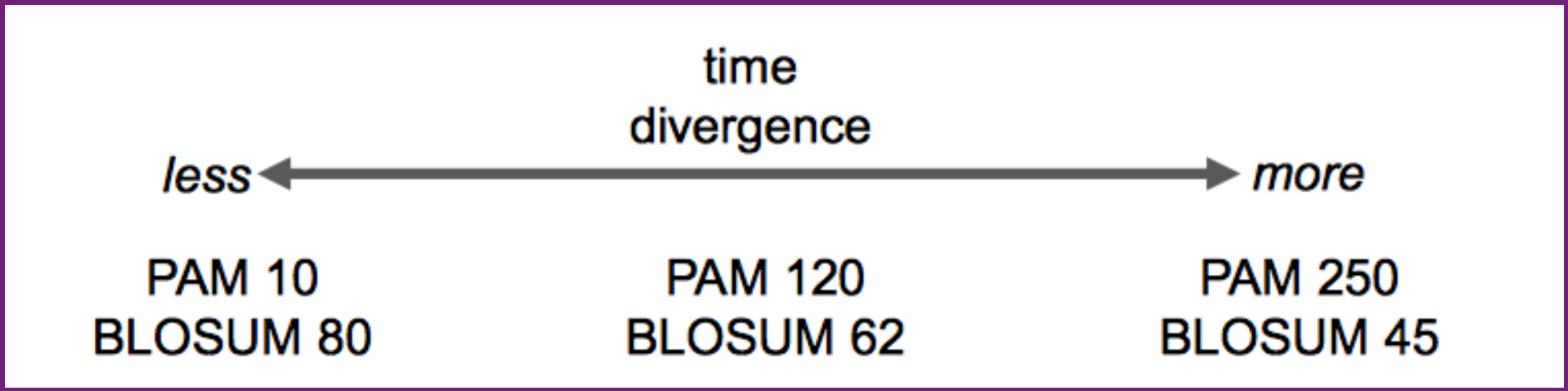
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

PAM VERSUS BLOSUM

PAM VERSUS BLOSUM

- The basis for constructing the two sets of matrices is different
- BLOSUM matrices with a low percentage correspond to PAM matrices for large evolutionary distances
- Rough correspondence
  - PAM250 corresponds to BLOSUM-45
  - PAM160 corresponds to BLOSUM-62
  - PAM120 corresponds to BLOSUM-80

PAM VERSUS BLOSUM



- Different studies have concluded that for the PAM matrices it is generally best to try PAM40, PAM120, and PAM250
- When used for local alignments
  - Lower PAM matrices find short local alignments
  - Higher PAM matrices find longer but weaker local alignments

PAM VERSUS BLOSUM

- Of course, the answer depends
  - Several different matrices should be tested
  - The alignment that is judged to be evolutionarily the most accurate should be chosen
  - What if you don’t know what you are looking for?


SIGNIFICANT OF
ALIGNMENTS

SIGNIFICANCE OF ALIGNMENTS

- We have a score, how likely are we to get this by chance?
- Hypothesis testing for sequence homology
  - When a best local alignment is found, the next task is to assess its biological relevance
  - This is most often done based on hypothesis testing

SIGNIFICANCE OF ALIGNMENTS

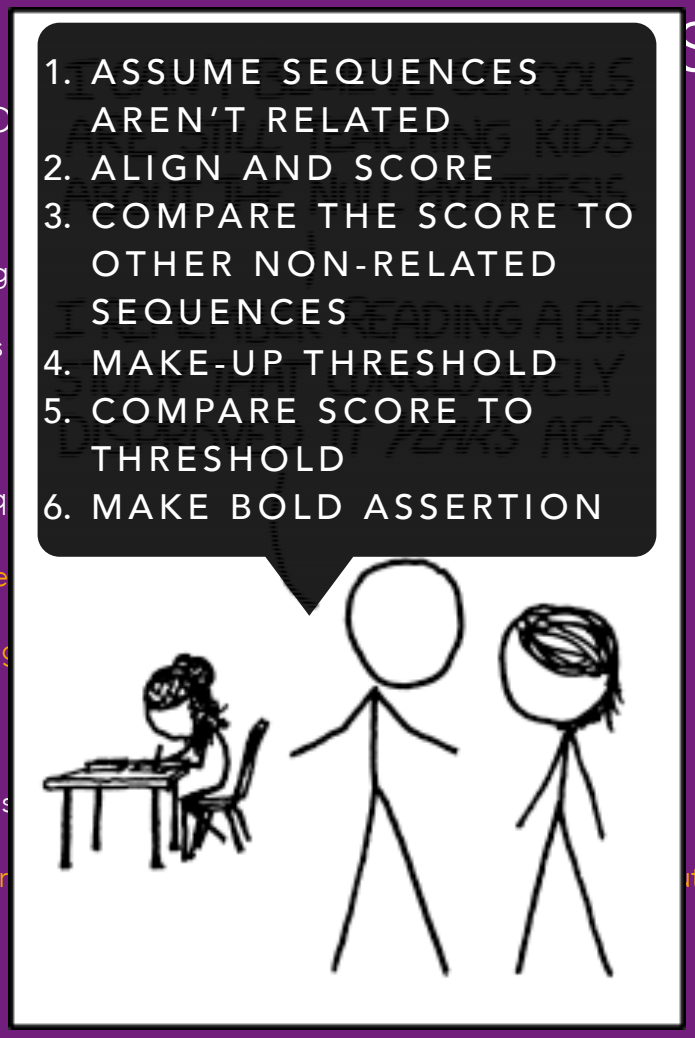
TESTING FOR SEQUENCE HOMOLOGY

- Formulate your hypothesis
  - H_0: the two sequences are not homologous (null hypothesis)
  - H_1: the two sequences are homologous
- Determine the experiment
  - Find the segment pair from the two sequences with the highest score
- Determine the probability of the result, given H_0
- Determine the rejection threshold for H_0 (e.g., .001)
- Perform the experiment chosen
  - Find the segment pair with the highest score and record the result
- Determine the probability of achieving the result or higher, given H_0 (use the probability distribution found above)
- Compare with the rejection level for H_0

SIGNIFICANCE S

TESTING FOR SEQUENCE HO

- Formulate your hypothesis
  - H_0: the two sequences are not homolog
  - H_1: the two sequences are homologous
- Determine the experiment
  - Find the segment pair from the two seq
- Determine the probability of the result, give
- Determine the rejection threshold for H_0 (e.g
- Perform the experiment chosen
  - Find the segment pair with the highest s
- Determine the probability of achieving the r ition found above)
- Compare with the rejection level for H_0



SIGNIFICANCE OF ALIGNMENTS

TESTING FOR SEQUENCE HOMOLOGY

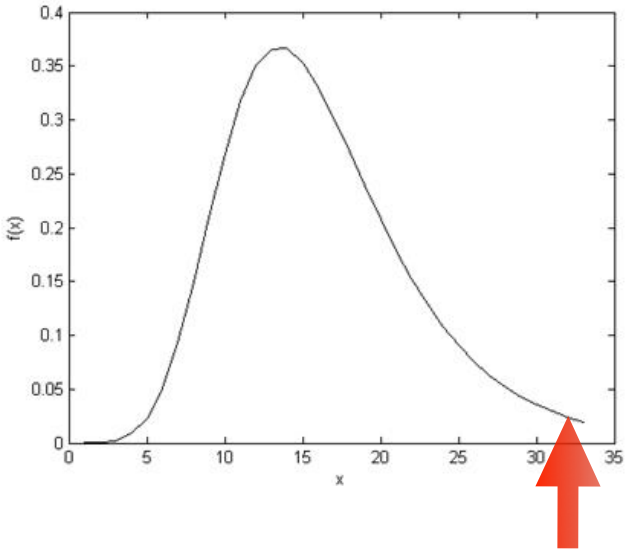
- Approach 1
  - Compare scores to other non-homologous sequences
- Approach 2
  - Compare to a bunch of randomly generated sequences
- Approach 3
  - Shuffle one of the sequences and do many alignments to observe score distributions

APPROACH WILL DEPEND ON RESEARCH GOAL

SIGNIFICANCE OF ALIGNMENTS

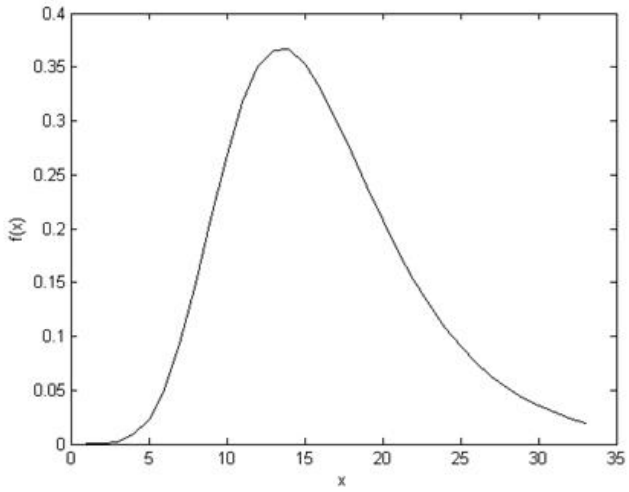
- Determined the probability of the results, given H_0
  - Find the probability distribution for the highest-scoring segment pairs in randomly generated sequences
- How to find the probability distribution for alignments
  - Generate a large number of sequences and align
  - Scores are the basis for the probability distribution

DISTRIBUTION OF SCORES FOR RANDOM SEQUENCES



SIGNIFICANCE OF ALIGNMENTS

- Generating random sequences
  - Match frequency distribution (shuffle sequence)
  - Perform alignment (with same parameters)



MORE COMPLEX STRATEGIES EXIST

SIGNIFICANCE OF ALIGNMENTS

Assume an alphabet of four symbols {A, C, D, E} and the sequences

$$q = \text{ACADAEA} \quad \text{and} \quad d = \text{ECAEDACECE},$$

and we find the best local alignment to be

```
CA-DA
CAEDA
```

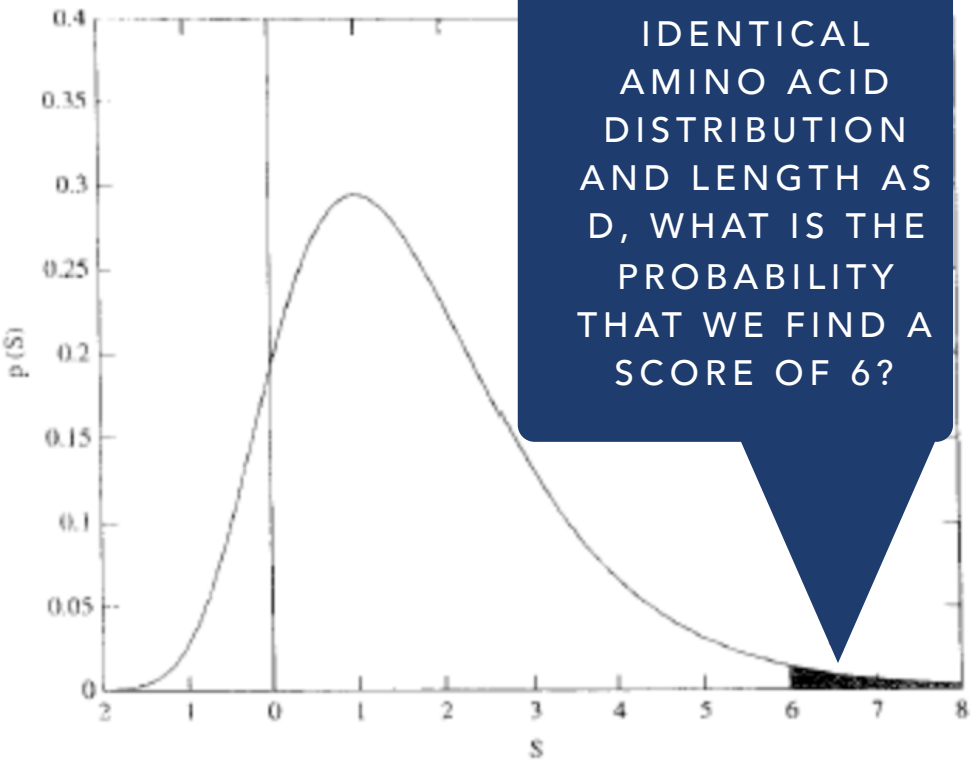
with score $S = 6$.

We then make random sequences, with the same amino acid distribution and length as d. The frequency distribution to use when generating random sequences is then $\{f_A = 0.2, f_C = 0.3, f_D = 0.1, f_E = 0.4\}$. E will be drawn with a probability twice the probability of drawing A.

- Example of randomly generating sequences

SIGNIFICANCE OF ALIGNMENTS

- Calculating significance
  - Generate 1,000 shuffled sequences
  - Align and collect the distribution of the scores
  - Find probability of finding a local alignment with S>=6



GIVEN THAT Q IS ALIGNED WITH A RANDOM SEQUENCE WITH IDENTICAL AMINO ACID DISTRIBUTION AND LENGTH AS D, WHAT IS THE PROBABILITY THAT WE FIND A SCORE OF 6?

SIGNIFICANCE OF ALIGNMENTS

- Z-score indicates how many standard deviations a element is from mean
  - Standardized score

1. Compare the two sequences, and record S', the score of the highest-scoring segment pair.
2. Make a distribution of the scores for random sequences, as explained above, or use a known distribution. Let μ be the mean of the random scores, and σ the standard deviation. The standard deviation is a measure of variance, and for discrete values defined as

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \mu)^2},$$

where y_i are the scores obtained for randomized sequences.

3. Find the Z value of the highest-scoring segment pair, with scoring S'. The Z value is the number of standard deviations that the score S' is above the mean value, calculated as

$$Z(S') = \frac{S' - \mu}{\sigma}.$$

SIGNIFICANCE OF ALIGNMENTS

- It is possible to convert a Z-score to p-value
  - Only if normally distributed
  - Pairwise sequence alignments are generally not
- If score is best out of 100
  - "Probability it occurred by chance is (p-value) is < .01"

STANDARD NORMAL DISTRIBUTION: Table Values Represent AREA to the LEFT of the Z score.

Z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
-3.9	.00005	.00005	.00004	.00004	.00004	.00004	.00004	.00004	.00003	.00003
-3.8	.00007	.00007	.00007	.00006	.00006	.00006	.00006	.00005	.00005	.00005
-3.7	.00011	.00010	.00010	.00010	.00009	.00009	.00008	.00008	.00008	.00008
-3.6	.00016	.00015	.00015	.00014	.00014	.00013	.00013	.00012	.00012	.00011
-3.5	.00023	.00022	.00022	.00021	.00020	.00019	.00019	.00018	.00017	.00017
-3.4	.00034	.00032	.00031	.00030	.00029	.00028	.00027	.00026	.00025	.00024
-3.3	.00048	.00047	.00045	.00043	.00042	.00040	.00039	.00038	.00036	.00035
-3.2	.00069	.00066	.00064	.00062	.00060	.00058	.00056	.00054	.00052	.00050
-3.1	.00097	.00094	.00090	.00087	.00084	.00082	.00079	.00076	.00074	.00071
-3.0	.00135	.00131	.00126	.00122	.00118	.00114	.00111	.00107	.00104	.00100
-2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139
-2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
-2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.00272	.00264
-2.6	.00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
-2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
-2.4	.00820	.00798	.00776	.00755	.00734	.00714	.00695	.00676	.00657	.00639
-2.3	.01072	.01044	.01017	.00990	.00964	.00939	.00914	.00889	.00866	.00842
-2.2	.01390	.01355	.01321	.01287	.01255	.01222	.01191	.01160	.01130	.01101
-2.1	.01786	.01743	.01700	.01659	.01618	.01578	.01539	.01500	.01463	.01426
-2.0	.02275	.02222	.02169	.02118	.02068	.02018	.01970	.01923	.01876	.01831
-1.9	.02872	.02807	.02743	.02680	.02619	.02559	.02500	.02442	.02385	.02330

SIGNIFICANCE OF ALIGNMENTS

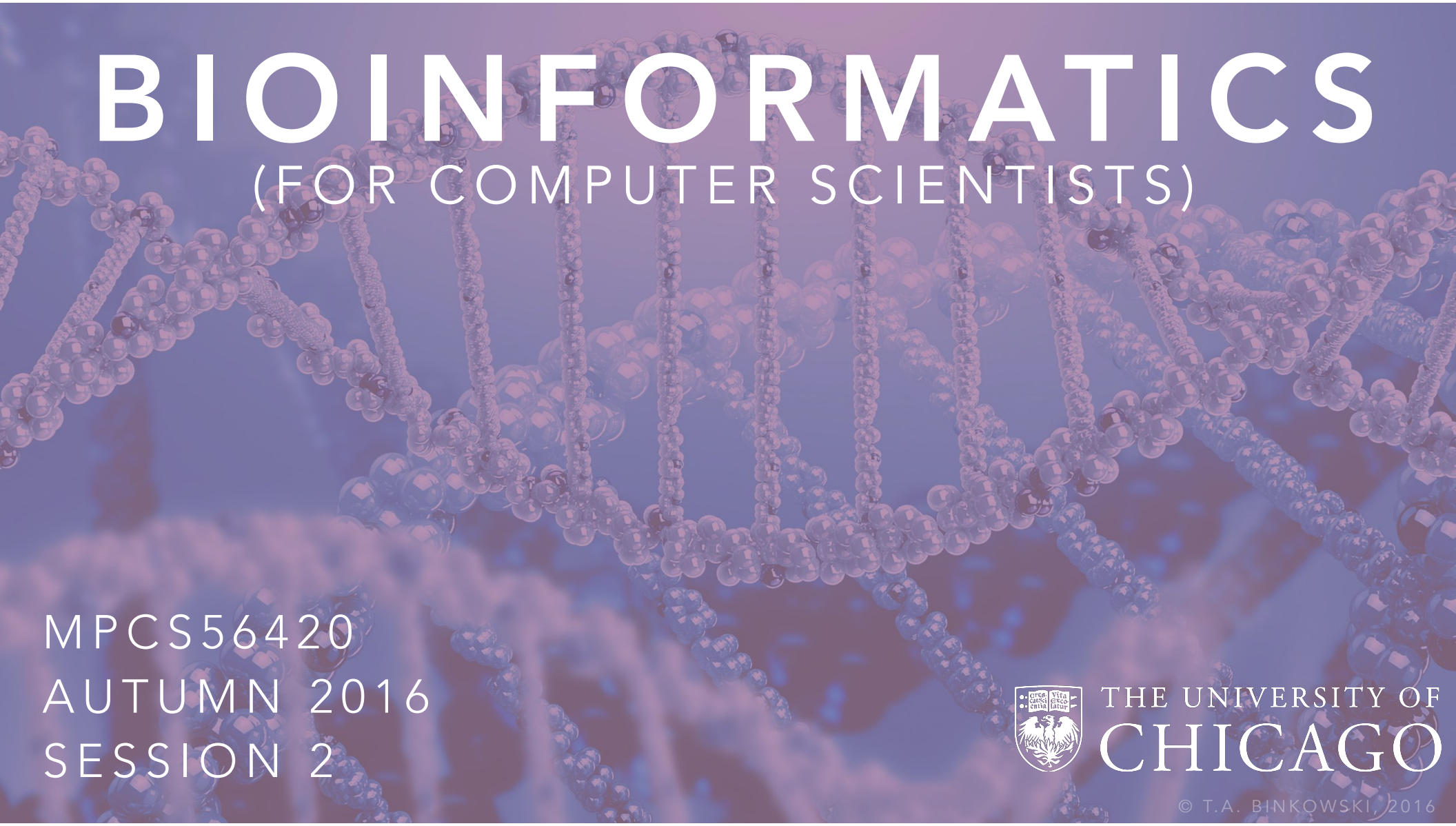
- Historical evidence suggests that Z=7 indicates biologically related proteins

SIGNIFICANCE OF ALIGNMENTS

- Assume we have found a score of S′=17.2. Calculate the Z-score for S′. Let the probability distribution for the scores from random sequences have the mean and standard deviation as μ=4.2, σ=3.4.
- Does the alignment suggest homology between the sequences?

```
Z-score    = (17.2-4.2)/3.4
           = 3.8
```


HOMEWORK



BIOINFORMATICS

(FOR COMPUTER SCIENTISTS)

MPCS56420
AUTUMN 2016
SESSION 2

THE UNIVERSITY OF CHICAGO

© T.A. BINKOWSKI, 2016