Pairwise Sequence Alignment
Database searching with BLAST

Outline

- Why do we sequence ?
  - Genome Annotation
  - Comparative Genomics
  - Expression Profiling
- Pairwise sequence alignment
  - global vs. local
  - scoring statistics
- Searching sequence databases with BLAST
  - heuristic search strategy
  - scoring local alignments
- Running BLAST with Python
  - Issuing system commands (local, remote)
- Parsing BLAST output
  - Structured text: XML
  - BioPython BLAST parser

Why do we sequence?

- Genome Annotation:

  A complete genome sequence provides us with the raw data to construct a "parts list".

- Comparative Genomics:

  Conserved regions in the genome are more likely to play an important role in biology of the species.

- Functional Genomics:

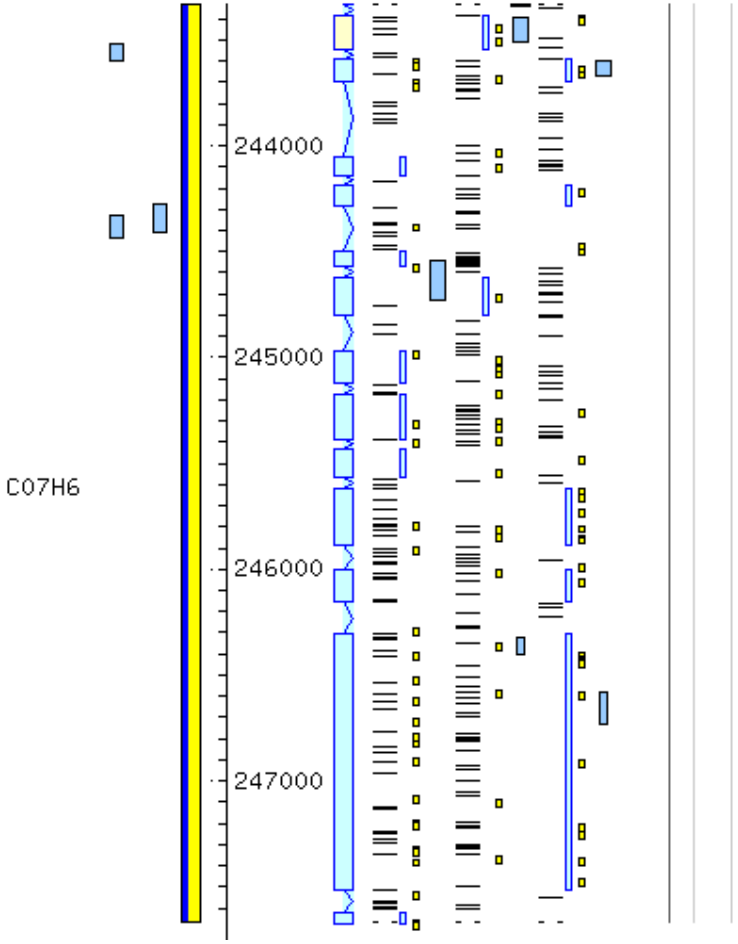
  Sequencing the RNA provides us with an insight into the transcriptionally active regions of the genome.

- Population Genomics:

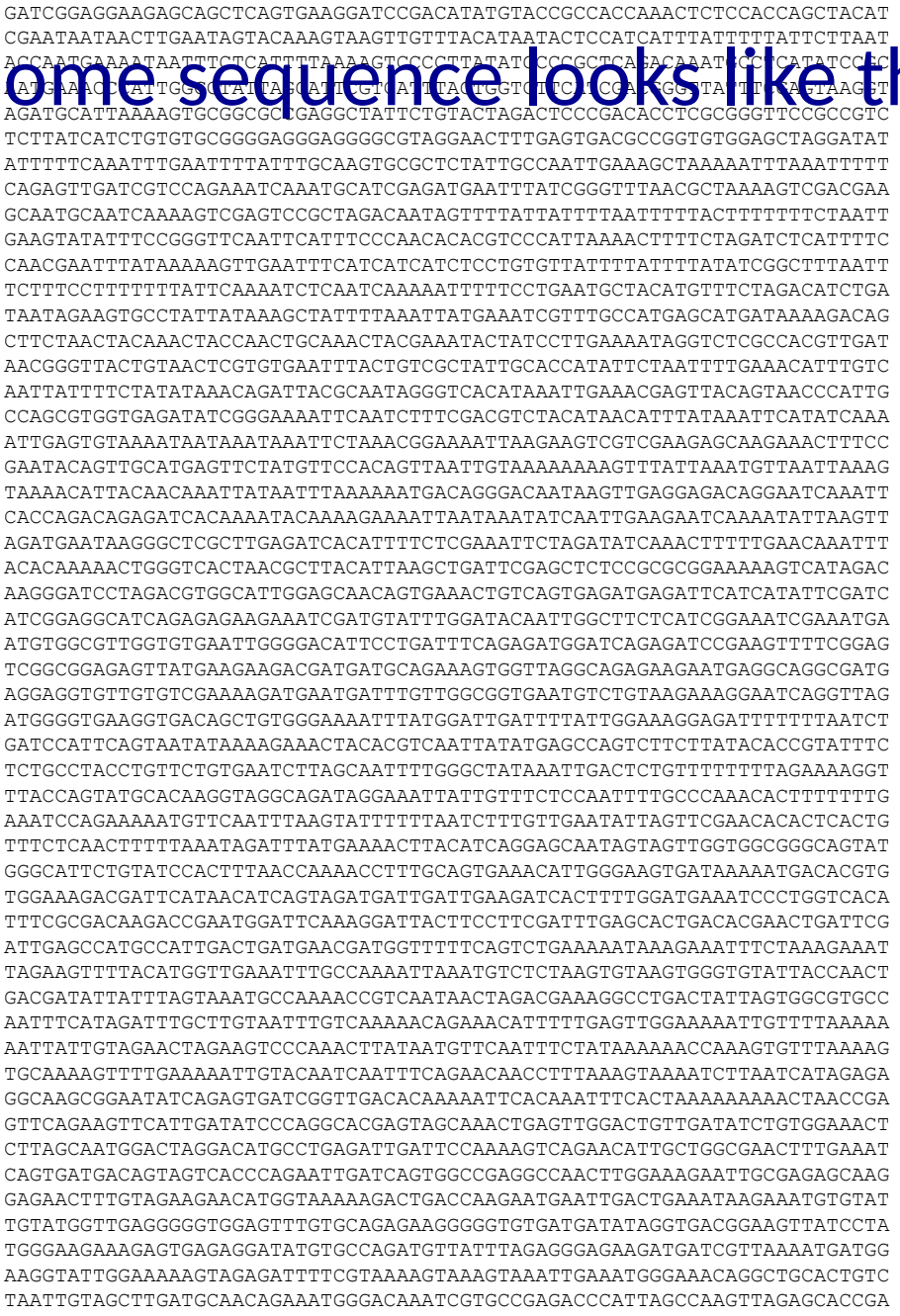
  Genetic structure and diversity reveals history and distribution of phenotypic traits (e.g. disease susceptibility alleles)

Genome Annotation

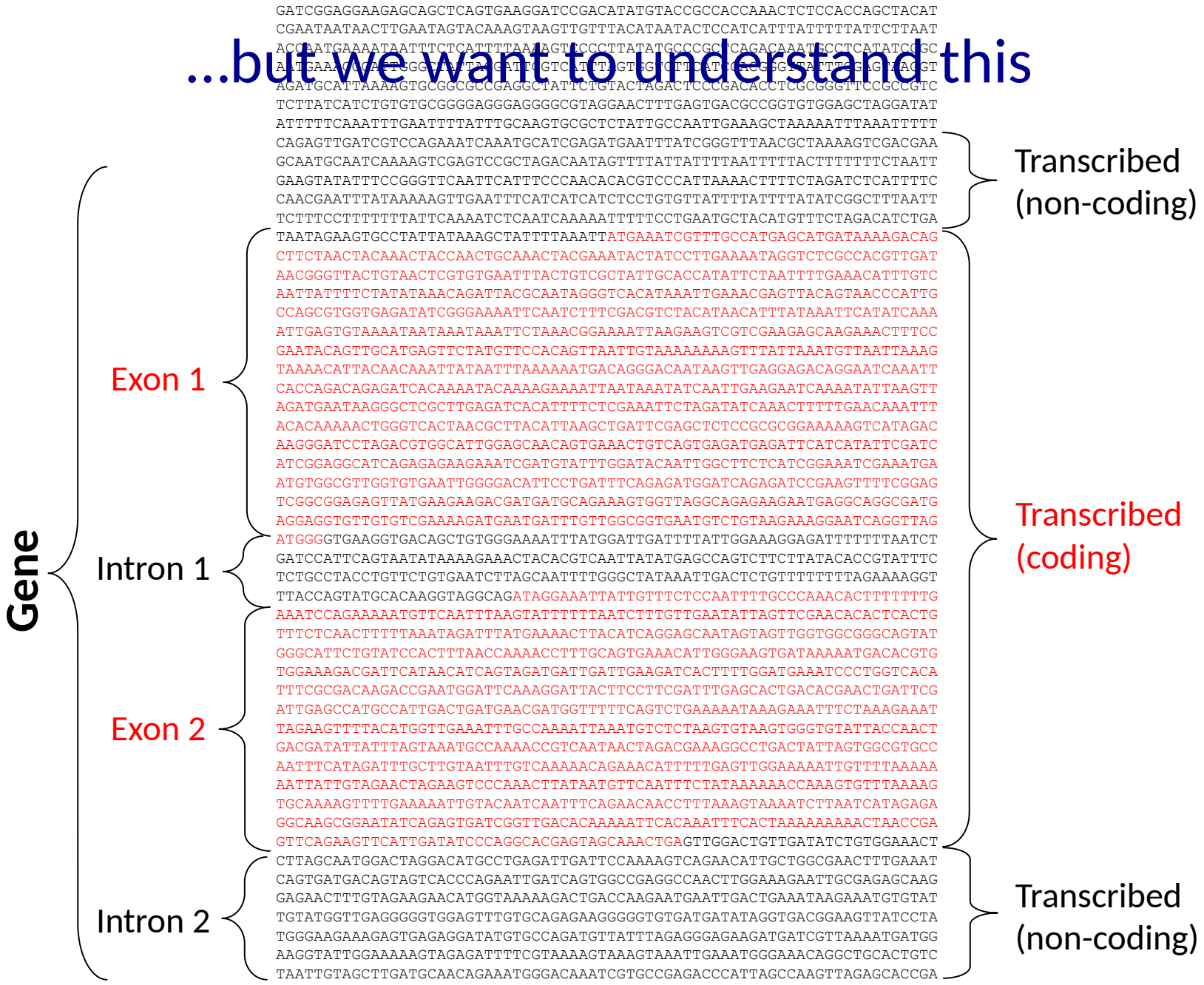
- Annotation performed on completed sequence (or assembled contigs)
- **Computer programs** used to find:
  - Genes
  - Exons and introns
  - Regulatory sequences
  - Repetitive elements
- **Algorithms** use a combination of sequence conservation and known features of different types of elements



Genome sequence looks like this…



...but we want to understand this



Challenges to gene prediction

- Features distinguishing genes are not well understood and our knowledge is constantly expanding.

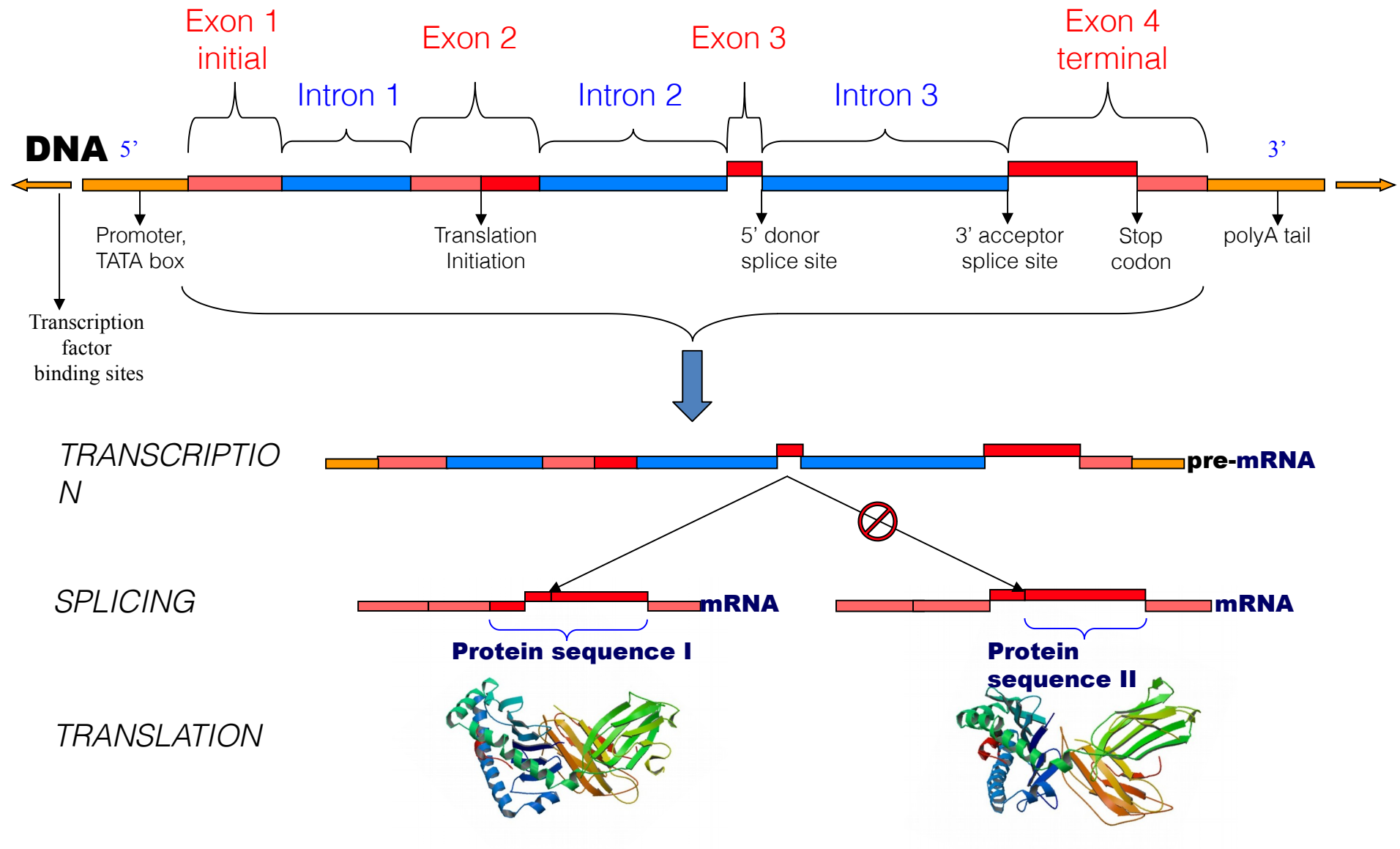
- Some features are fairly well defined, e.g.:
  - splice sites,
  - translation start sites and stop codons,
  - open reading frames (ORFs)

- Others, e.g. structure of promoter regions, much less so.

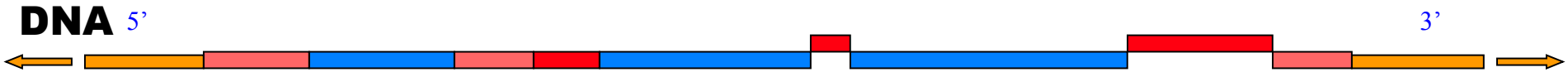
- Even identifying ORFs is not straightforward in eukaryotes and particularly in vertebrate and mammalian genomes:
  - many exons
  - large introns
  - alternative splicing

- Predictions are highly hypothetical, and automatic annotation of eukaryote genomes not entirely reliable (… how many genes are in the human genome?)

genes as complex DNA sequence structures to produce proteins:

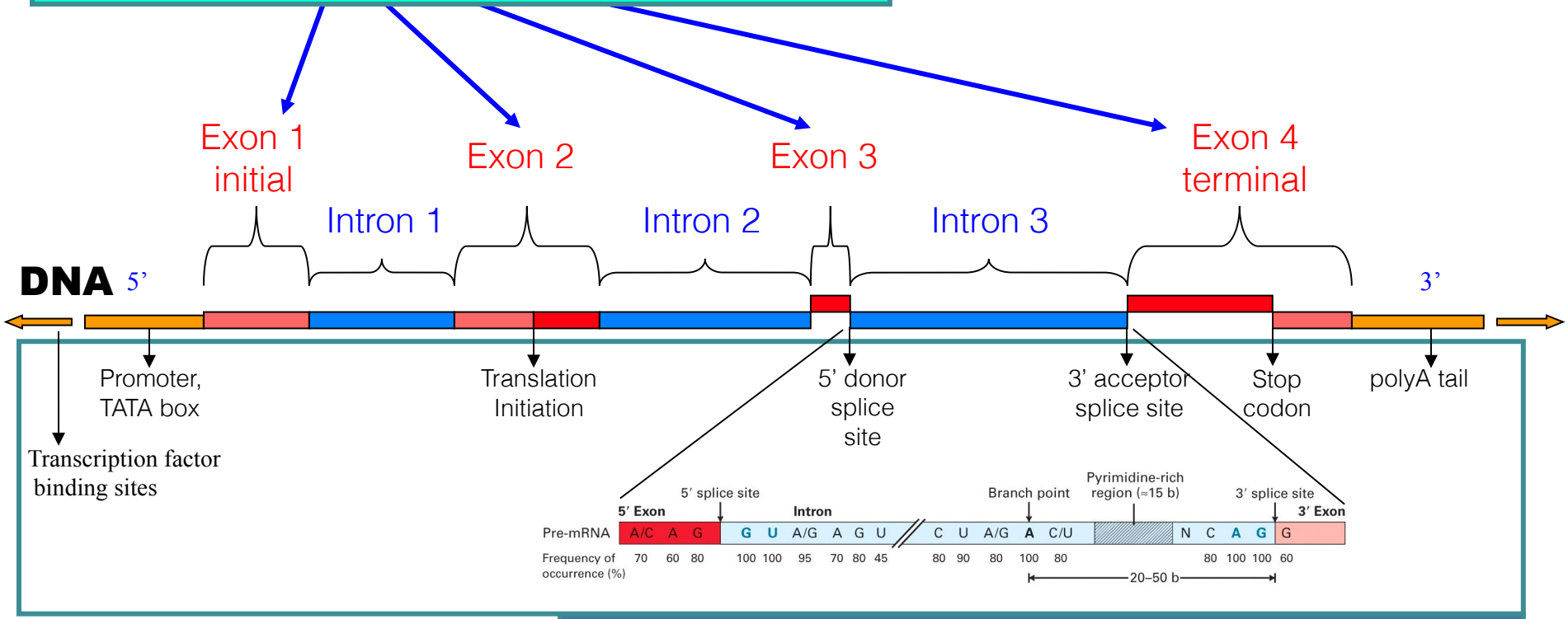


How do we identify genes computationally?



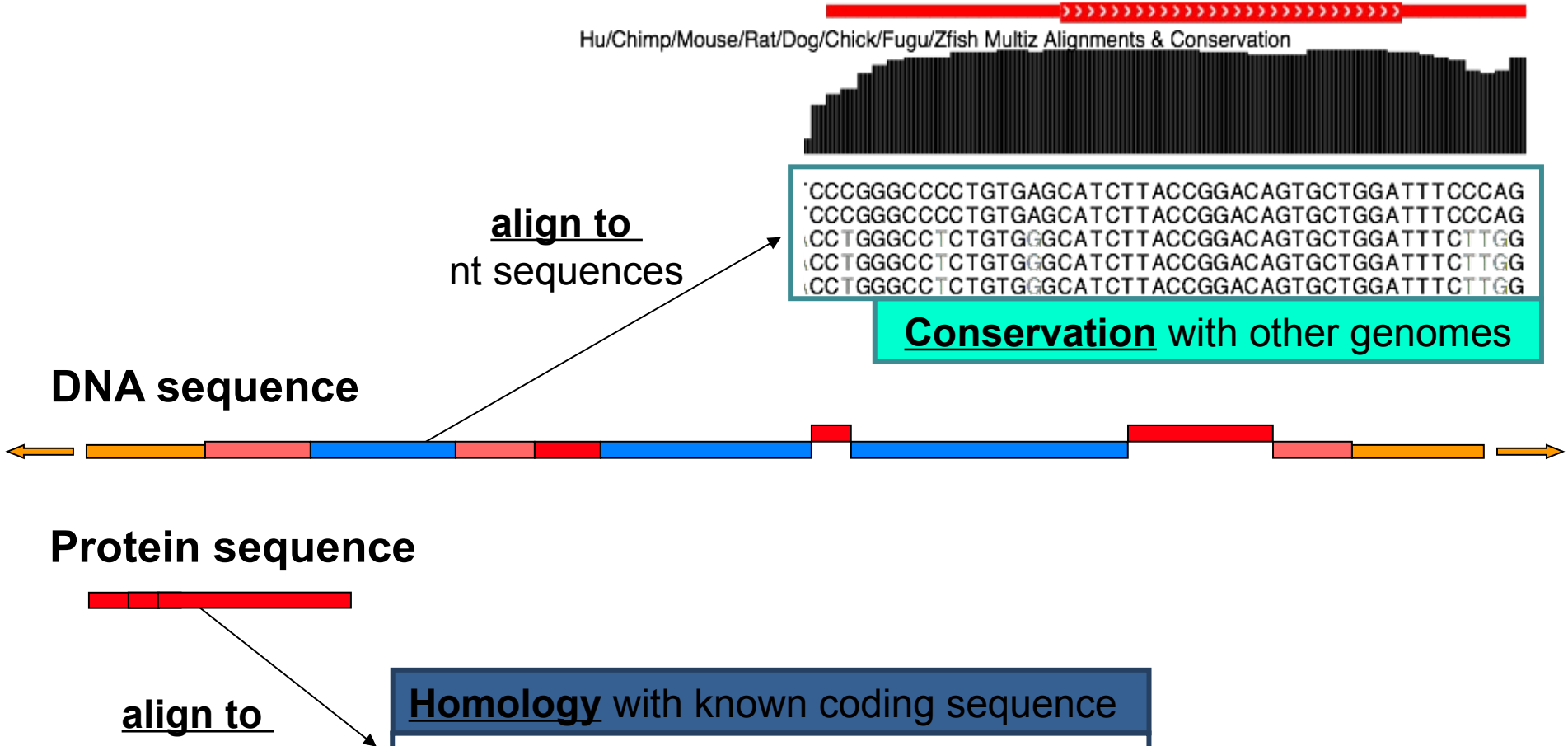
Intrinsic information

Compositional features of coding sequences
- uneven distribution of amino acids in proteins
- uneven distribution of synonymous codons
- 6mer structure, *5th order Markov Models*



Sequence signals: identify putative locations of required sequence features, *position weight matrices*

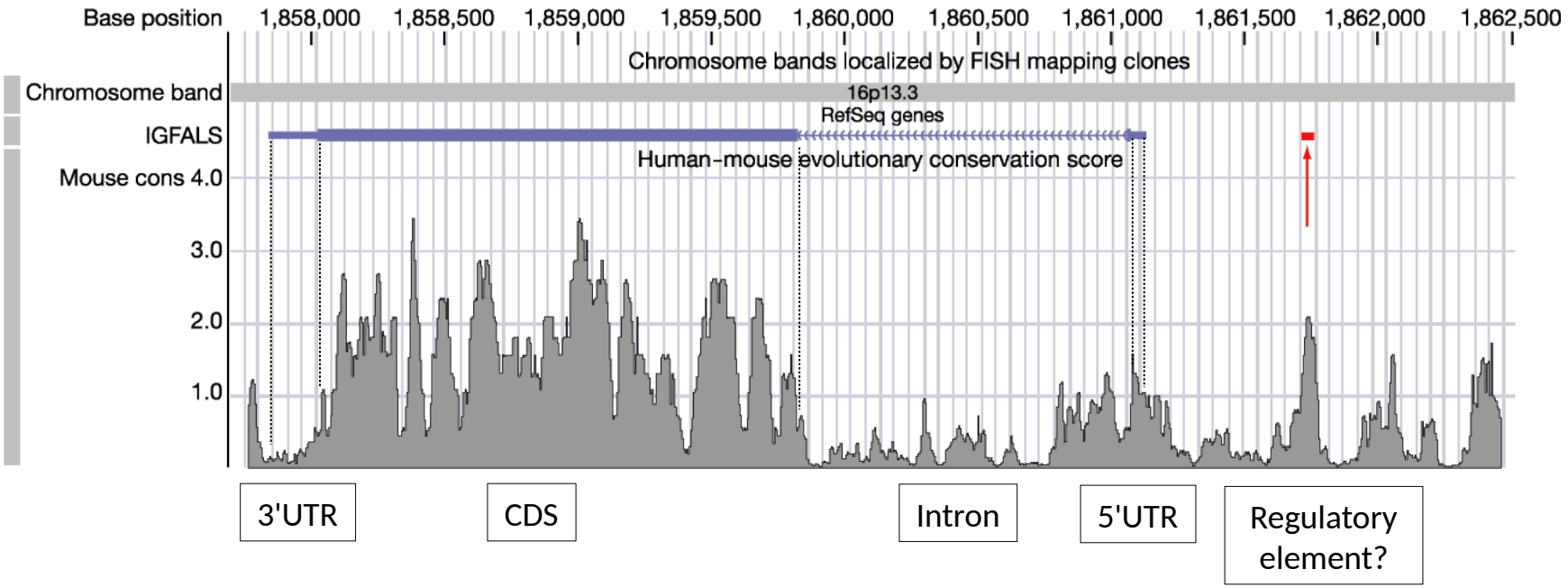
Extrinsic information



Comparative Genomics

Insights gained through comparison of genomes from different species

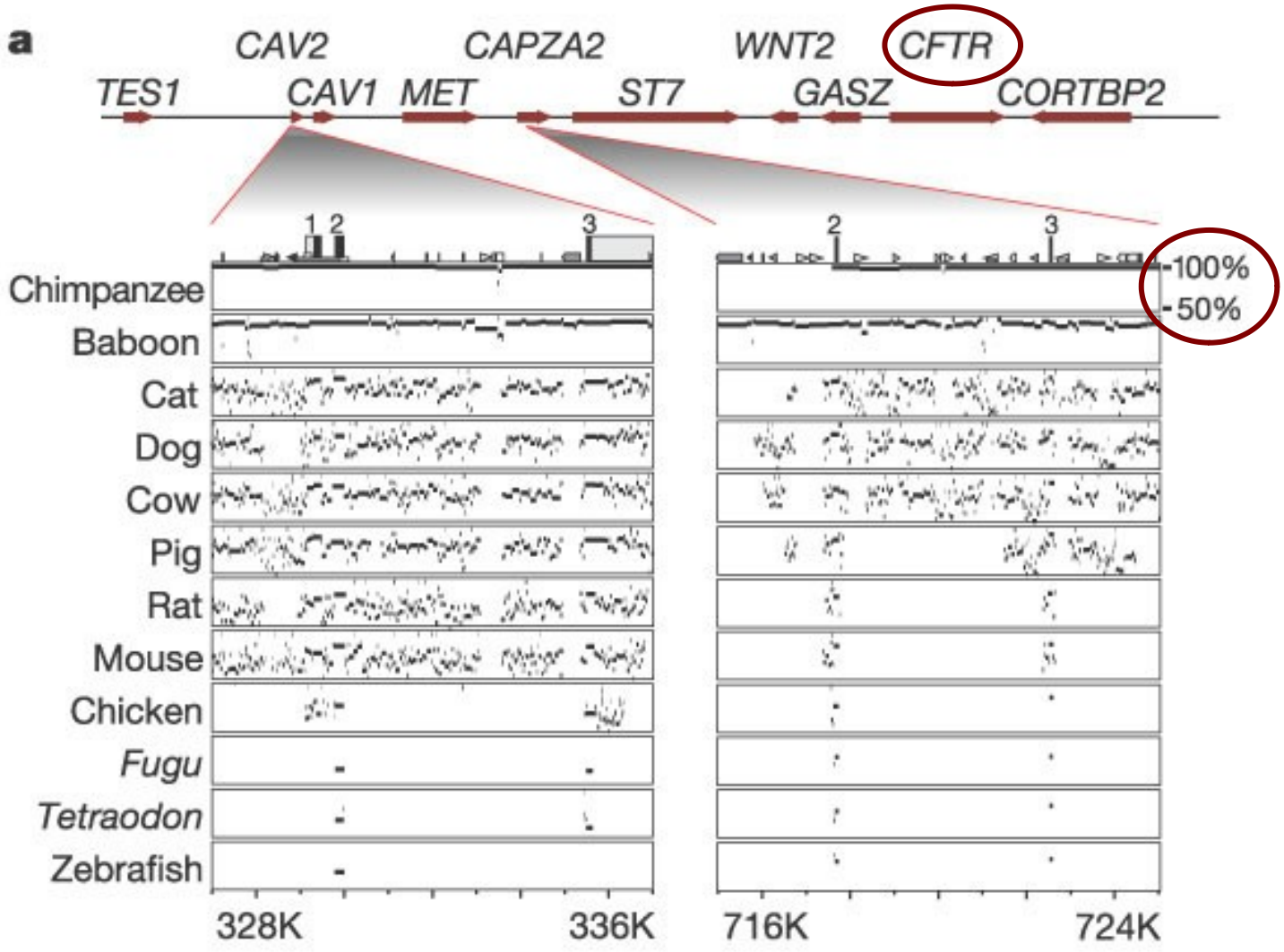
Mouse-human comparison



- Higher sequence similarity indicates functional constraint.
- In general, protein-coding regions (CDS) are the most highly conserved sequence elements.

Sequence conservation

in 1.8MB from human CFTR-region



Thomas JW et al., Comparative analyses of multi-species sequences from targeted genomic regions. *Nature 2003; 424:788-793*

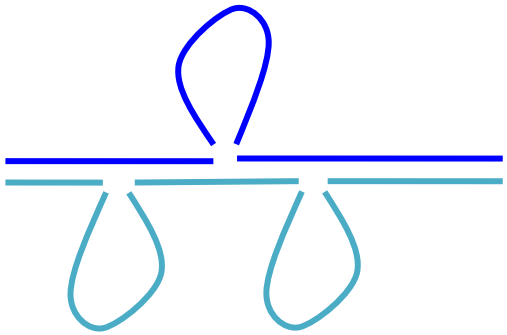
Pair-wise Sequence Alignment

- The basics:
  - global vs. local alignment
  - scoring alignments
  - alignment algorithms
  - sequence database searching with BLAST
  - scoring BLAST hits
  - a quick tour

Global and local approaches to aligning sequences

- Attempt to “match” and assess similarity between two entire sequences; **GLOBAL**
- Find subsequences of high similarity; **LOCAL**

and then try to combine local alignments to obtain an overall comparison of the original sequences.



The second approach is more meaningful

(especially for long sequences, of different lengths … whole genomes)

Two protein or DNA sequences are unlikely to present a straightforward overall “match”, even if they are closely related.

Why?

Substitutions are not the only process by which they diverge; insertions, deletions and rearrangements are common.

How do we decide what a "good" sequence alignment is?

Given a particular alignment, e.g.: `AAGCTAA`

`AA-CCAA`

1) Assign **scores** to "matches", "mismatches", and "indels" at each position

   E.g.: match = 10; mismatch = 1; indel = -10

2) Sum local scores (assuming mutations at different sites occur independently)

   E.g. Score = 5*10 + 1 -10 = 41

Questions we need to address:

- Why should we pick one scoring function over another?
- Given a scoring function, what is **the best ("optimal") score** we could get in aligning two sequences?
- What does the **optimal alignment** look like?
- What is the **significance** of the score ? I.e. how likely is this score when compared to alignments of unrelated ("random") sequences?

PAM Substitution Matrices

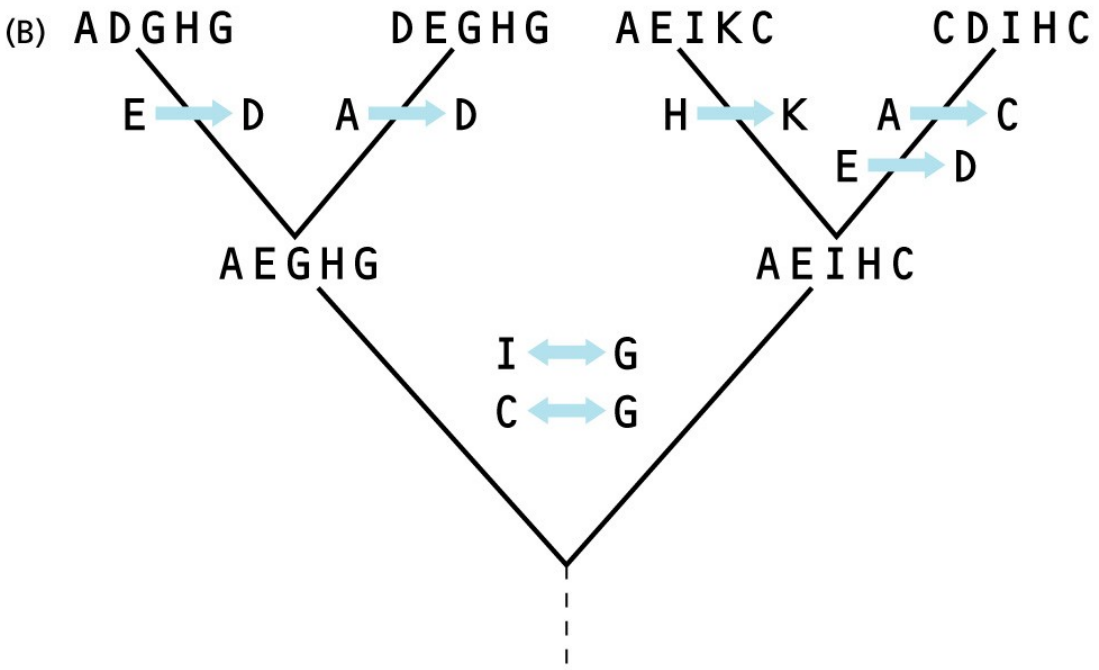
- First developed by Margaret Dayhoff and her coworkers in the 1960s and 1970s
- Matrix is based on real data which models the evolutionary process and does not consider physiochemical similarities of proteins.
- Calculated the probability that any one amino acid would mutate to another over a given period of evolutionary time which is then converted to a score.
- PAM = Point Accepted Mutations – number of mutations in sequence per 100 residues. PAM250 is 250 mutations (some residues have been subjected to more than one mutation)

The identification of accepted point mutations

(A)

DEGHG
ADGHG
CDIHC
AEIKC

(B)



(C)

	A	C	D	E	G	H	I	K
A		1	1					
C	1				1			
D	1			2				
E			2					
G		1					1	
H								1
I					1			
K						1		

BLOSUM substitution matrix

- In the early 1990s, sequences were clustered into group according to level of similarity.
- Substitution frequencies for all possible pairs of amino acids are calculated between the clustered groups which is then used to calculate the score.
- No phylogenetic trees are constructed
- BLOSUM62 is derived using Blocks of peptide sequences that are 62% identity or more.

Choosing a Matrix

- When comparing distant protein sequences PAM 250 or BLOSUM 50 is recommended
- When comparing closely related sequences, PAM120 or BLOSUM 80 may work best.
- Length of the sequence should also be considered
  - Shorter sequences should matrices for closely related sequences
  - Longer sequences (> 100 residues) should use longer evolutionary time scale.

A sequence comparison:

A	D	D	R	Q	C	E	R	A	D
A	Q	E	R	Q	E	C	Q	A	Q
4	0	2	5	5	-4	-4	1	4	0

Total score: 13

$$S_{i,j} = \log\left(\frac{\Pr(i,j)}{\Pr(i)\Pr(j)}\right)$$

probability of *(i,j)* if independent

	A	R	N	D	C	Q	E
A	4	-1	-2	-2	0	-1	-1
R	-1	5	0	-2	-3	1	0
N	-2	0	6	1	-3	0	0
D	-2	-2	1	6	-3	0	2
C	0	-3	-3	-3	9	-3	-4
Q	-1	1	0	0	-3	5	2
E	-1	0	0	2	-4	2	5

subset of the BLOSUM62 matrix

***S*>0**: if i-to-j occurs more often than expected by chance based on their individual frequency

***S*<0**: if it occurs less often

- **Matches** have S>0: magnitude depends on how ***unlikely*** an amino acid is (rarity of occurrence in known sequences).
- **Mismatches**
  - S>0 : conservative amino acid changes
  - S=0 : "neutral" changes
  - S<0 : magnitude depends on how unlikely (infrequent, disruptive) a mismatch is.
- Many assumptions go into creating matrices ("symmetry" of replacements, independence of positions for PAM, etc.)

Gap penalties

- Gaps are needed to create good alignments between sequences
- They let us account for (small) insertions and deletions.
- We want to use them, but to do so sparingly, so they should have score "cost"

A sequence comparison, with a gap:

A	D	D	R	Q	C	E	R	D	D	R	A	D
A	Q	E	R	Q	E	C	-	-	-	Q	A	Q
4	0	2	5	5	-4	-4				1	4	0

Total score: 13

- [G+Ln] = -7

Final score: 6

G = gap opening penalty = 4

L = gap extension penalty = 1

n = number of positions in gap

Pairwise sequence alignment

- **Global: Needleman-Wunsch**
- **Local: Smith-Waterman**

⟹ *Exhaustive search of all possible pairwise alignments is generally infeasible*

Dynamic programming efficiently computes ***optimal*** sequence alignments

- Break the problem into reasonably sized sub-problems
- Use partial results to compute the final answer

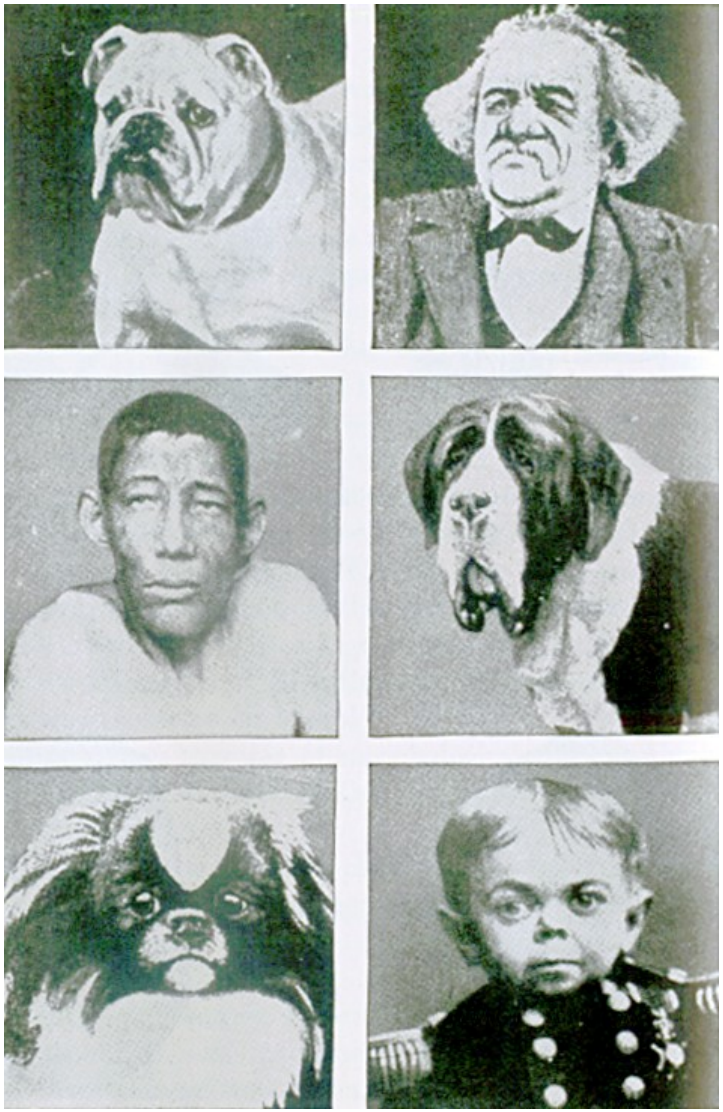


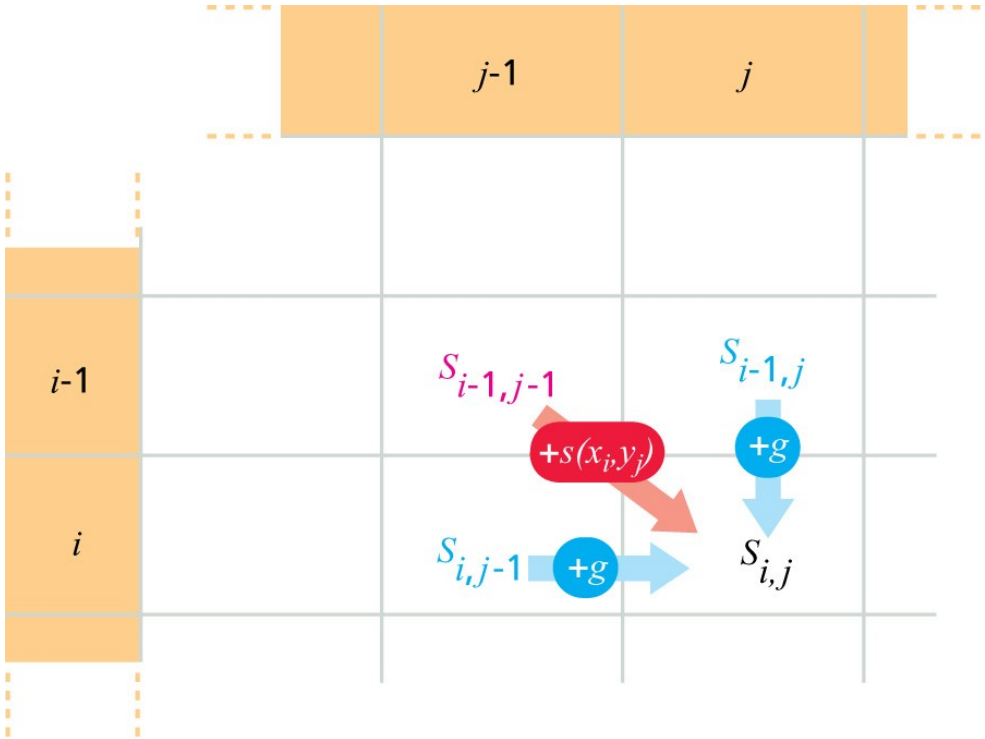
Image from lecture by J. Pevsner

Dynamic programming: Pairwise sequence alignment

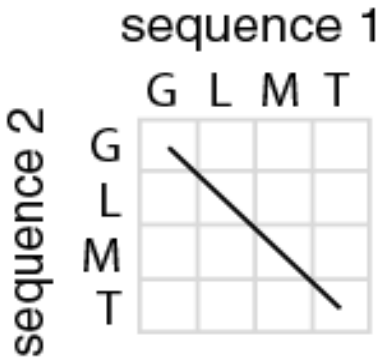
A. GLOBAL alignment

Match score = +1
Mismatch score = 0
Gap penalty = -1

$$\text{Score}(i,j) = \max \begin{cases} \text{(i-1,j-1) + match/mismatch = diagonal move} \\ \text{(i-1,j) - gap penalty = horizontal move} \\ \text{(i,j-1) - gap penalty = vertical move} \end{cases}$$

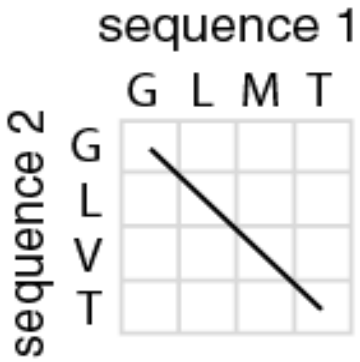


[1] identity (stay along a diagonal)



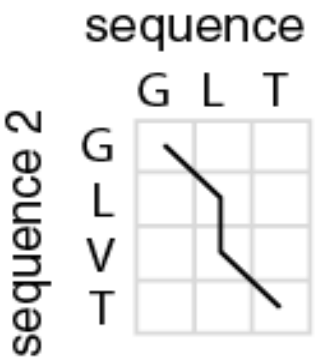
1 GL**M**T
2 GL**M**T

[2] mismatch (stay along a diagonal)



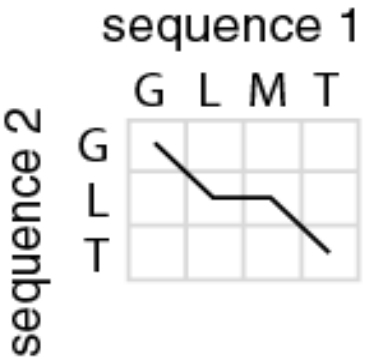
1 GL**M**T
2 GL**V**T

[3] gap in seq1 (move vertically)



1 GL**-**T
2 GL**V**T

[4] gap in seq2 (move horizontally)



1 GL**M**T
2 GL**-**T

Dynamic programming: Pairwise sequence alignment

A. GLOBAL alignment

Example (from *Understanding Bioinformatics*, Ch. 5.2):

What is the optimal alignment for the following two sequences?

```
THIS LINE
IS ALIGNED
```

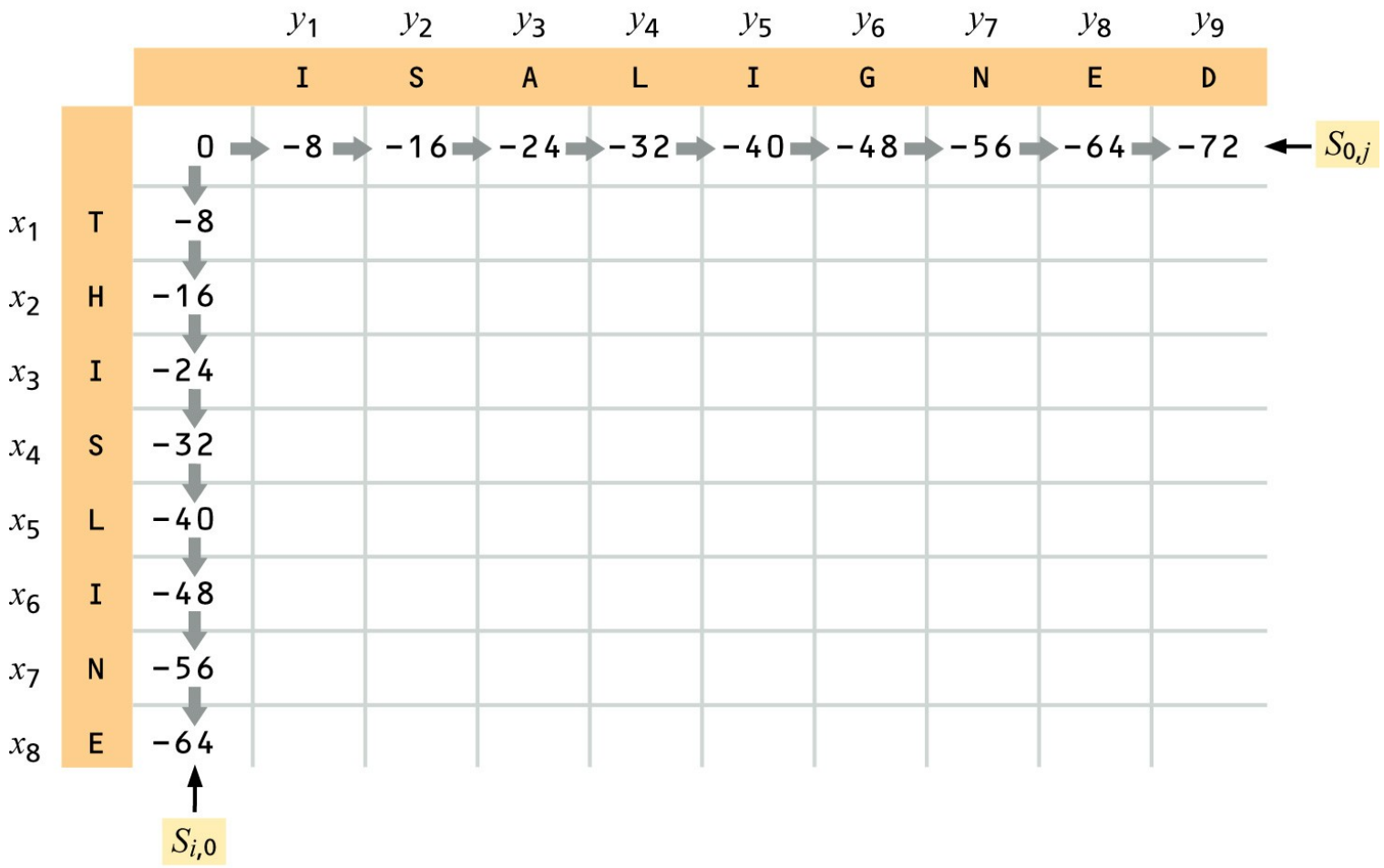
Since this is a short sequence, we can find the optimal alignment by eye:

```
THIS-LI-NE-
  ||  ||  ||
--ISALIGNED
```

Dynamic programming: Pairwise sequence alignment

A. GLOBAL alignment

Gap penalty score = -8

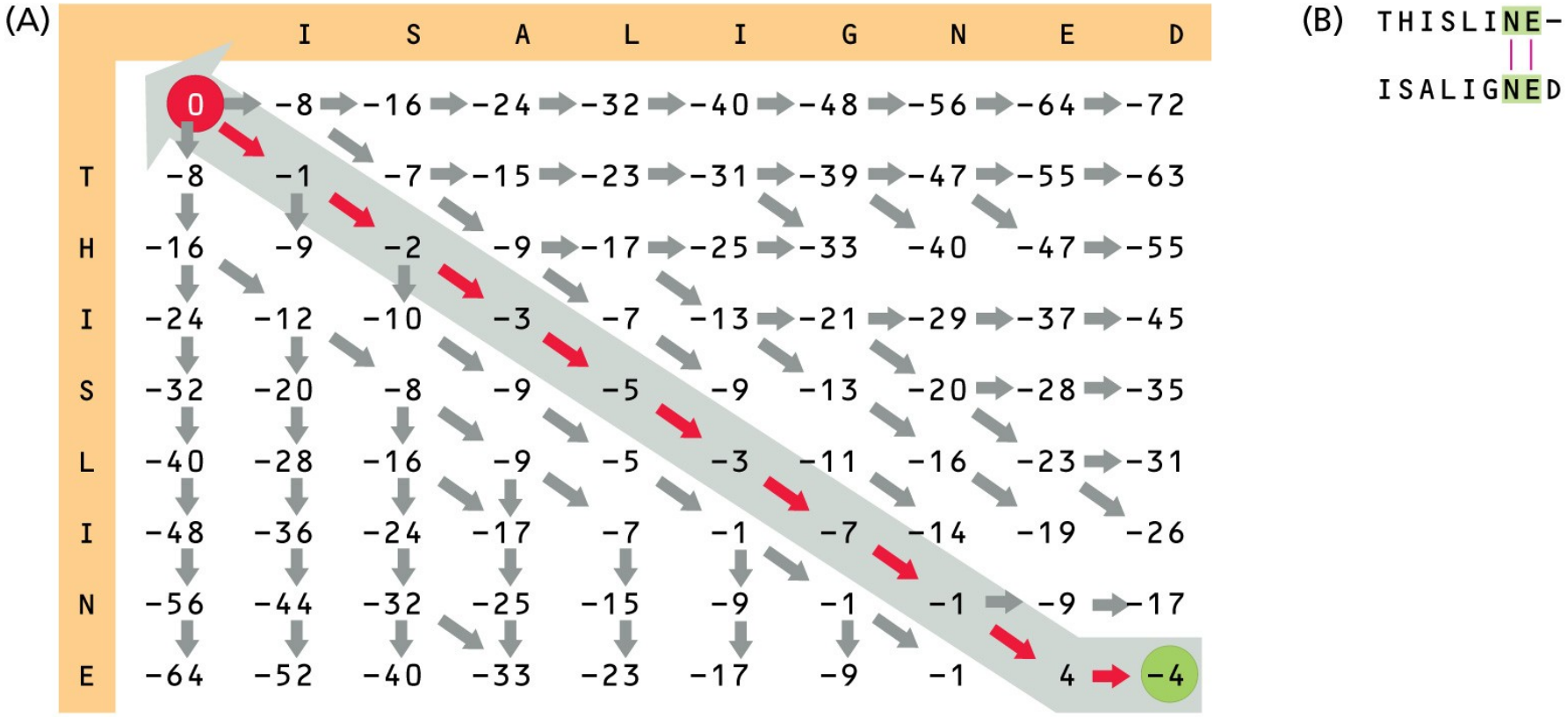


from *Understanding Bioinformatics*, Ch. 5.2

Dynamic programming: Pairwise sequence alignment

A. GLOBAL alignment – Gap penalty matters!!!

Gap penalty score = -8

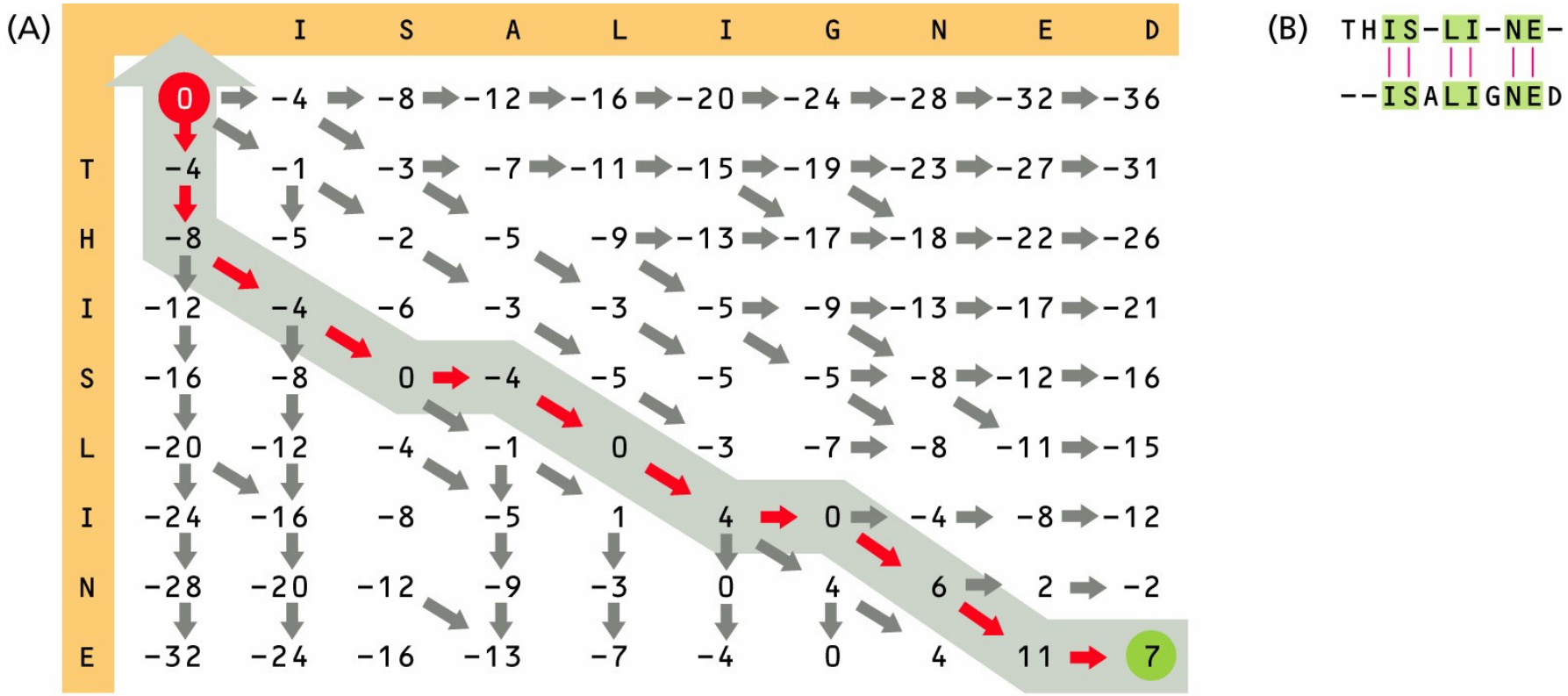


from *Understanding Bioinformatics*, Ch. 5.2

Dynamic programming: Pairwise sequence alignment

A. GLOBAL alignment – Gap penalty matters!!!

Gap penalty score = -4



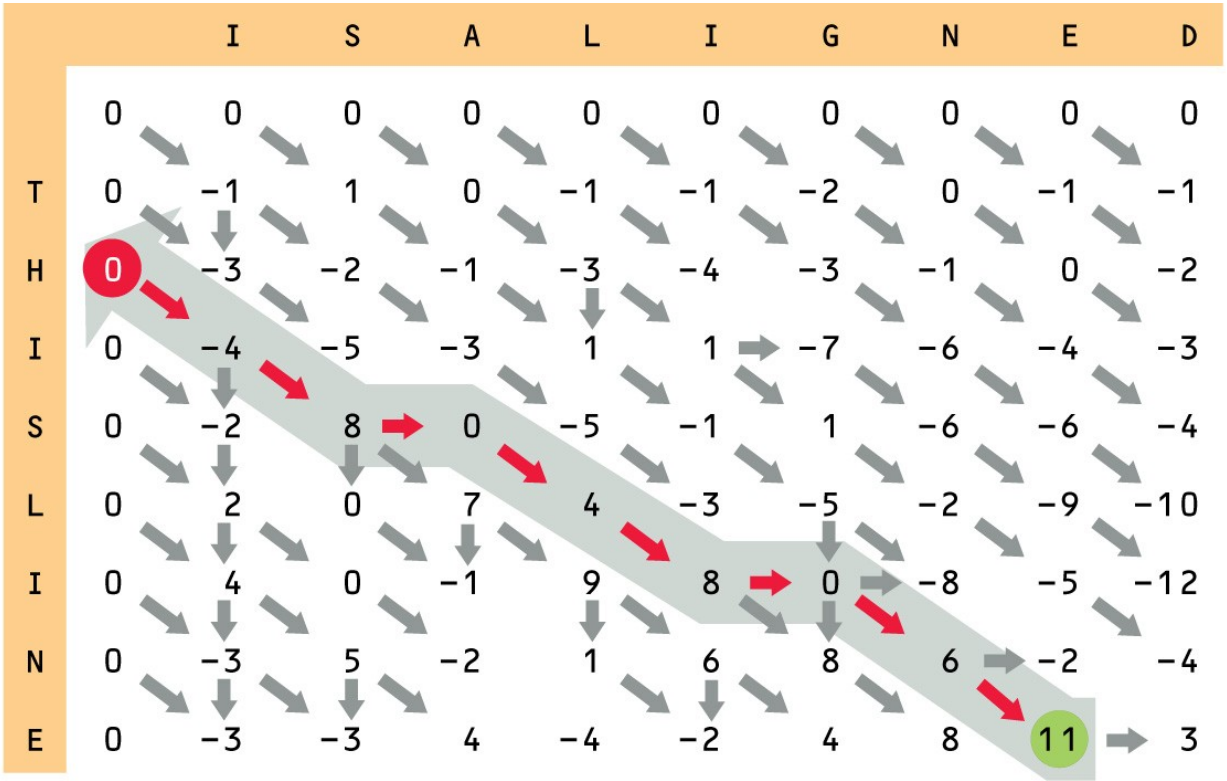
from *Understanding Bioinformatics*, Ch. 5.2

Dynamic programming: Pairwise sequence alignment

B. SEMI-global alignment

- Initialize left column, top row with **zero values**
- Allow free horizontal moves in last row
- Allow free vertical moves in last column

(A)



(B)

```
THIS-LI-NE-
 || || || ||
--ISALIGNED
```


from *Understanding Bioinformatics*, Ch. 5.2

B. LOCAL alignment: Smith-Waterman

- Consider two sequences: AACCTATAGCT and GCGATATA
- Using semi-global alignment, we obtain the following:

```
AAC-CTATAGCT
-GCGATATA---
```

- This doesn't look so great -- BUT …

```
AAC-CTATAGCT
-GCGATATA---
```

… there IS a pretty good **subsequence match** in the middle.

⇒ ***Smith-Waterman*** *finds these and* ***ignores gaps or mismatches outside the aligned region.***

⇒ *This is one of the most* ***fundamental*** *techniques in bioinformatics.*

Dynamic programming: Pairwise sequence alignment

C. LOCAL alignment: Smith-Waterman

- No values in the scoring matrix can be negative! **S ≥ 0**
- The score in each cell is the ***maximum*** of four values:

  [1] s(i-1,j-1) + the new score at [i,j] => a ***match*** or ***mismatch***

  [2] s(i,j-1) – gap penalty => a ***gap*** in sequence at ***left***

  [3] s(i-1,j) – gap penalty => a ***gap*** in sequence at ***top***

  [4] zero

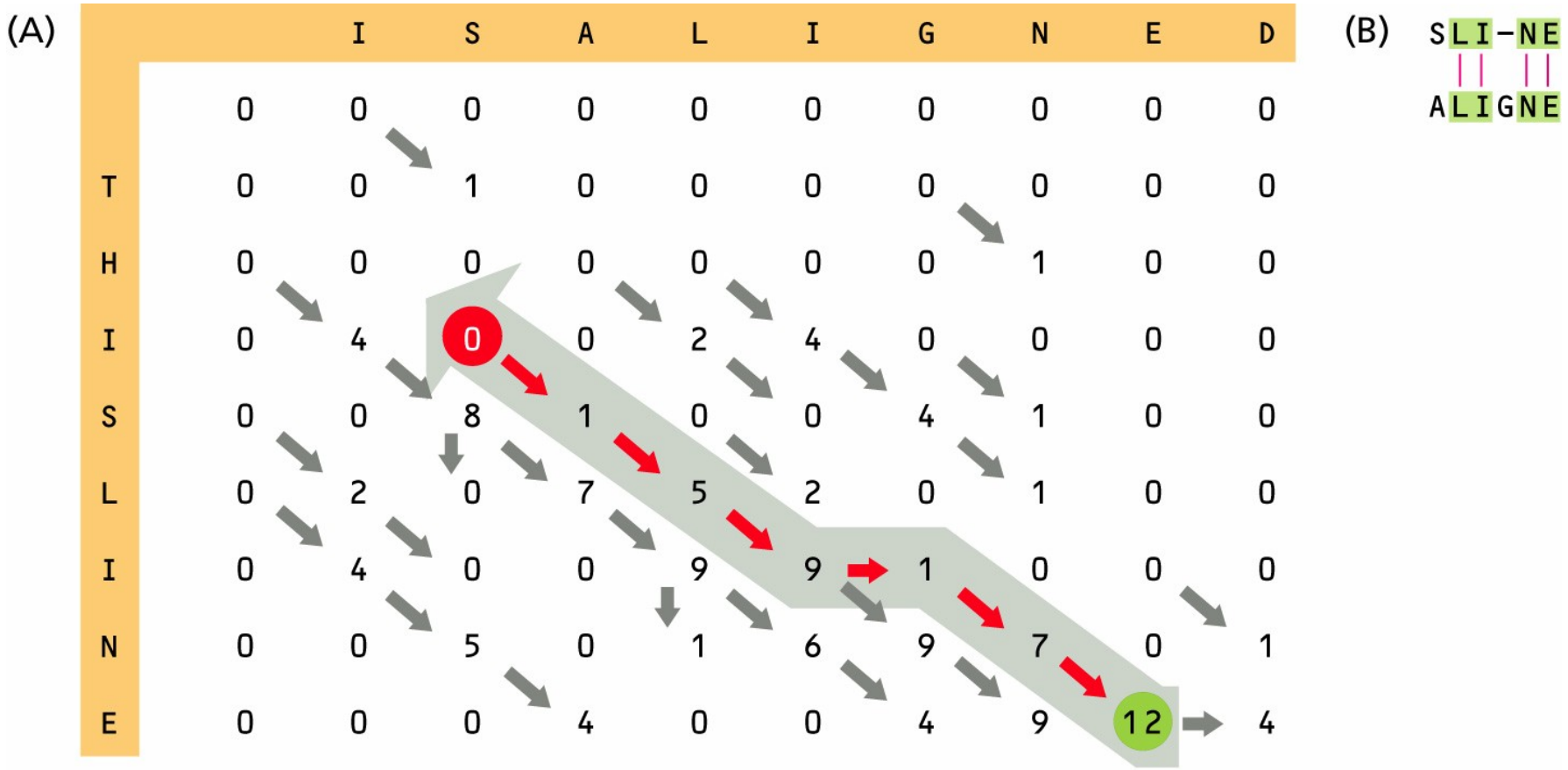
		A	A	C	C	T	A	T	A	G	C	T
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	1	0	0
C	0	0	0	1	1	0	0	0	0	0	2	1
G	0	0	0	0	0	0	0	0	0	1	0	1
A	0	1	1	0	0	0	1	0	1	0	0	0
T	0	0	0	0	0	1	0	2	1	0	0	1
A	0	1	1	0	0	0	2	0	3	2	1	0
T	0	0	0	0	0	1	1	3	2	2	1	2
A	0	1	1	0	0	0	2	2	4	3	2	1

TATA
TATA

Dynamic programming: Pairwise sequence alignment

C. LOCAL alignment: Smith-Waterman – Gap penalty still matters!!!

Gap penalty score = -8

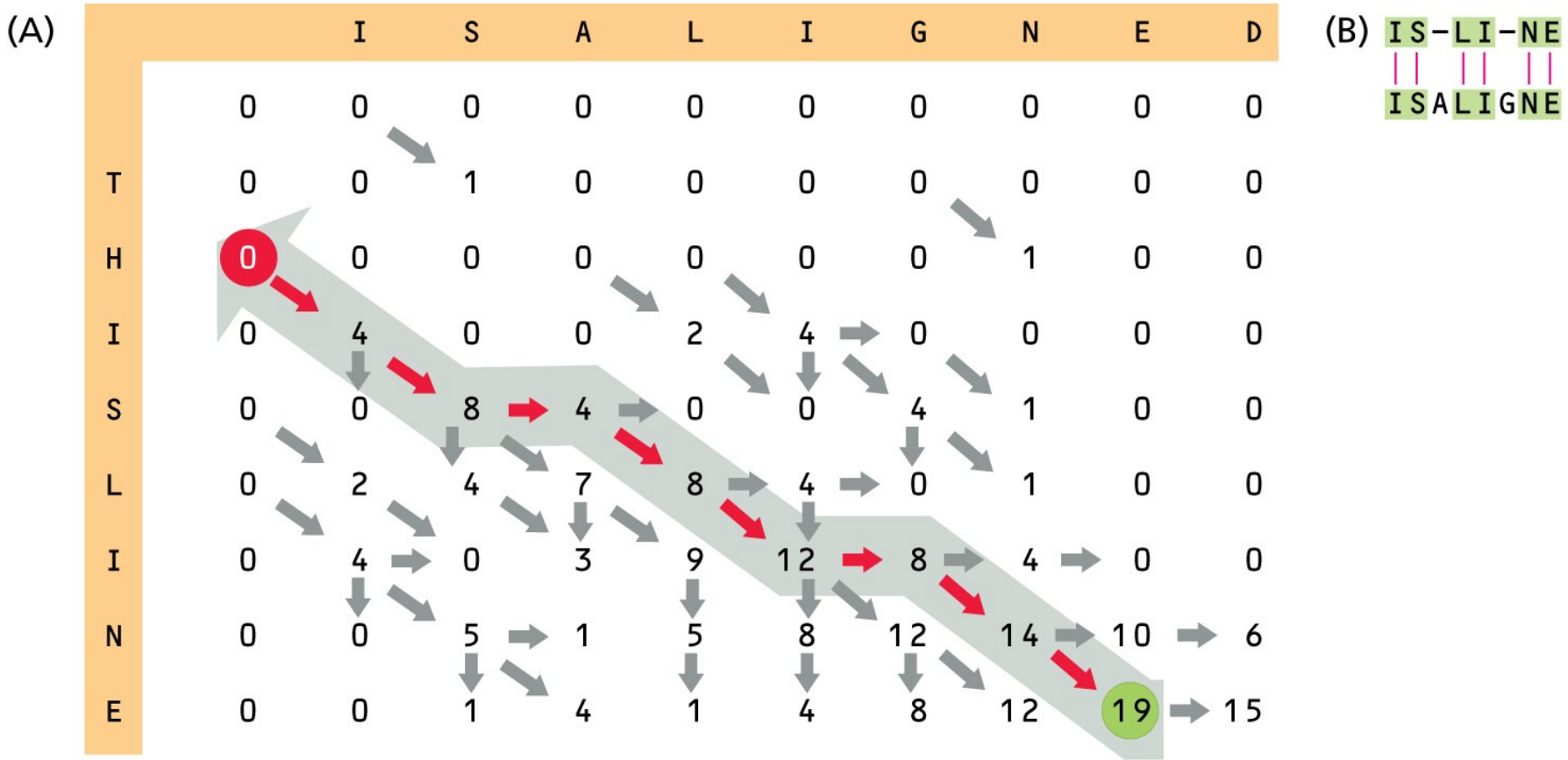


from *Understanding Bioinformatics*, Ch. 5.2

Dynamic programming: Pairwise sequence alignment

C. LOCAL alignment: Smith-Waterman – Gap penalty still matters!!!

Gap penalty score = -4



from *Understanding Bioinformatics*, Ch. 5.2

Scoring matrices for nucleic acid sequences

Matrices derived from analysis of alignments of distince regions of the human and mouse genomes with different G+C content

(A)

	A	C	G	T
A	67	-96	-20	-117
C	-96	100	-79	-20
G	-20	-79	100	-96
T	-117	-20	-96	67

37% G+C
CFTR region

(B)

	A	C	G	T
A	91	-114	-31	-123
C	-114	100	-125	-31
G	-31	-125	100	-114
T	-123	-31	-114	91

47% G+C
HOXD region

(C)

	A	C	G	T
A	100	-123	-28	-109
C	-123	91	-140	-28
G	-28	-140	91	-123
T	-109	-28	-123	100

53% G+C
hum16pter region

Fig. 56. From Chiaromonte et al.

Heuristic (vs. **optimal**) methods for pair-wise alignment: the **BLAST** family of algorithms

Given a scoring system (scoring matrix and gap penalties), alignments can be evaluated quantitatively, and optimal alignments can be sought with

Dynamic Programming algorithms:

- GLOBAL: Needleman-Wunsch-Gotoh algorithm
- LOCAL: Smith-Waterman algorithm

These have **high algorithmic complexity** ($O(N^2)$), and are replaced in most practical applications by ***heuristic procedures***.

Most commonly used: **BLAST family of algorithms** (local alignment).

How likely is it to find a match by chance?

"Given a set of sequences not related to the query sequence (or even random sequences), ***what is the probability of finding a match with alignment score S simply by chance?*** "

Score will depend on:

- Length and composition of the query and target sequence
- Scoring matrix

Statistical significance

P-value: probability of finding one or more sequences of score >=S by random chance

E-value: expected number of sequences of score >=S that would be found by random chance

BLAST E-value = $Kmne^{-\lambda S}$

K,λ are parameters (constants) that depend on the substitution matrix and gap penalties

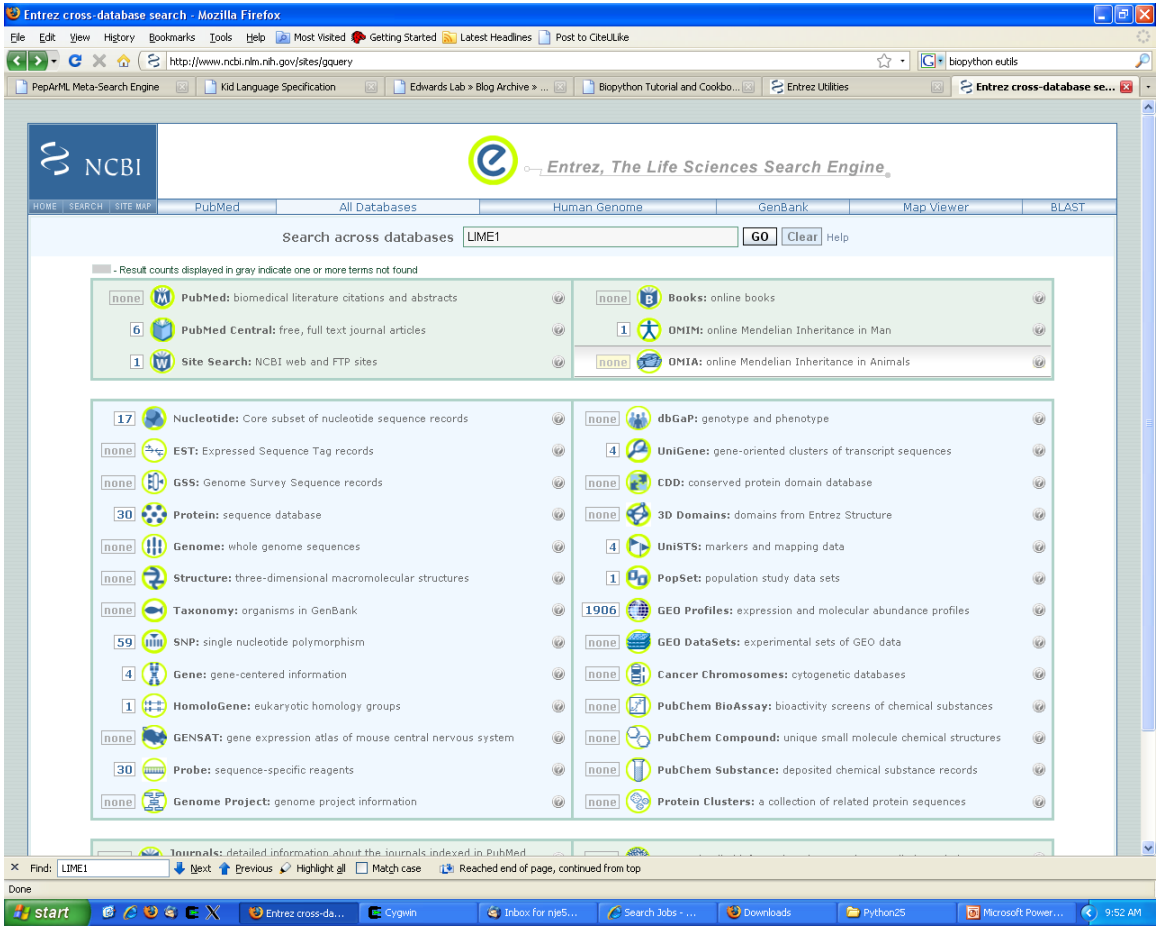
m = sum of lengths of sequences in DB

n = length of query sequence

S = raw score

NCBI Entrez

- Powerful web-portal for NCBI's online databases
  - Nucleotide
  - Protein
  - PubMed
  - Gene
  - Structure
  - Taxonomy
  - OMIM
  - etc…



NCBI/ BLAST Home

BLAST finds regions of similarity between biological sequences. more...

New DELTA-BLAST, a more sensitive protein-protein search Go

BLAST Assembled RefSeq Genomes

Choose a species genome to search, or list all genomic BLAST databases.

- Human
- Mouse
- Rat
- *Arabidopsis thaliana*
- *Oryza sativa*
- *Bos taurus*
- *Danio rerio*
- *Drosophila melanogaster*
- *Gallus gallus*
- *Pan troglodytes*
- *Microbes*
- *Apis mellifera*

Basic BLAST

Choose a BLAST program to run.

nucleotide blast	Search a **nucleotide** database using a **nucleotide** query *Algorithms:* blastn, megablast, discontiguous megablast
protein blast	Search **protein** database using a **protein** query *Algorithms:* blastp, psi-blast, phi-blast, delta-blast
blastx	Search **protein** database using a **translated nucleotide** query
tblastn	Search **translated nucleotide** database using a **protein** query
tblastx	Search **translated nucleotide** database using a **translated nucleotide** query

Specialized BLAST

Choose a type of specialized search (or database name in parentheses.)

- Make specific primers with Primer-BLAST
- Search trace archives
- Find conserved domains in your sequence (cds)
- Find sequences with similar conserved domain architecture (cdart)
- Search sequences that have gene expression profiles (GEO)
- Search immunoglobulins (IgBLAST)
- Search using SNP flanks
- Screen sequence for vector contamination (vecscreen)
- Align two (or more) sequences using BLAST (bl2seq)
- Search protein or nucleotide targets in PubChem BioAssay
- Search SRA transcript and genomic libraries
- Constraint Based Protein Multiple Alignment Tool
- Needleman-Wunsch Global Sequence Alignment Tool
- Search RefSeqGene

News

BLAST 2.2.27+ released

A new version of the stand-alone BLAST applications has been released.
Mon, 10 Sep 2012 14:00:00 EST

More BLAST news...

Tip of the Day

How to save custom search pages.

So you have made a few BLAST searches and after adjusting the database, organism limits and maybe a few Algorithm Parameters you arrive at what you think is a good search strategy.

More tips...

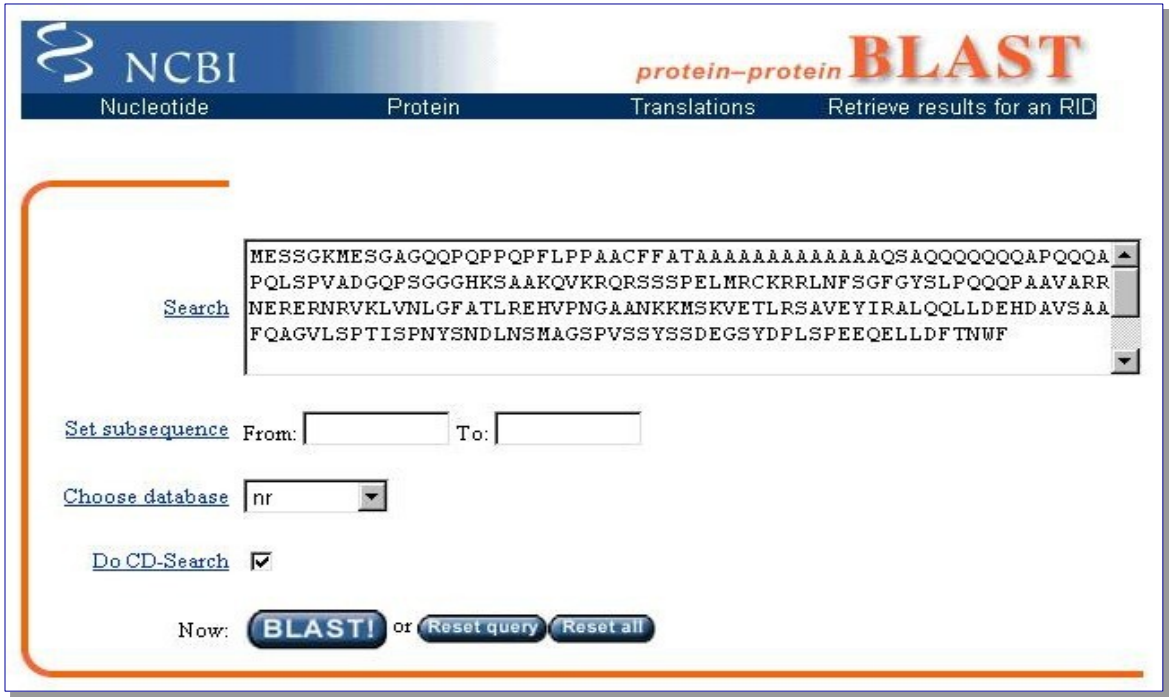
http://blast.ncbi.nlm.nih.gov/

The BLAST suite of tools

- There are many flavors of BLAST:
  - **blastn**: *nucleotide* query vs. *nucleotide* database
  - **blastp**: *protein* query vs. *protein* database
  - **blastx**: *translated nt* query vs. *protein* database
  - **tblastn**: *protein* query vs. *translated nt* database
  - **tblastx**: *translated nt* query vs. *translated nt* database
  - **psi-blast**: position-specific iterative BLAST
  - **megablast**: run large numbers of input sequences at once

BLAST alignments

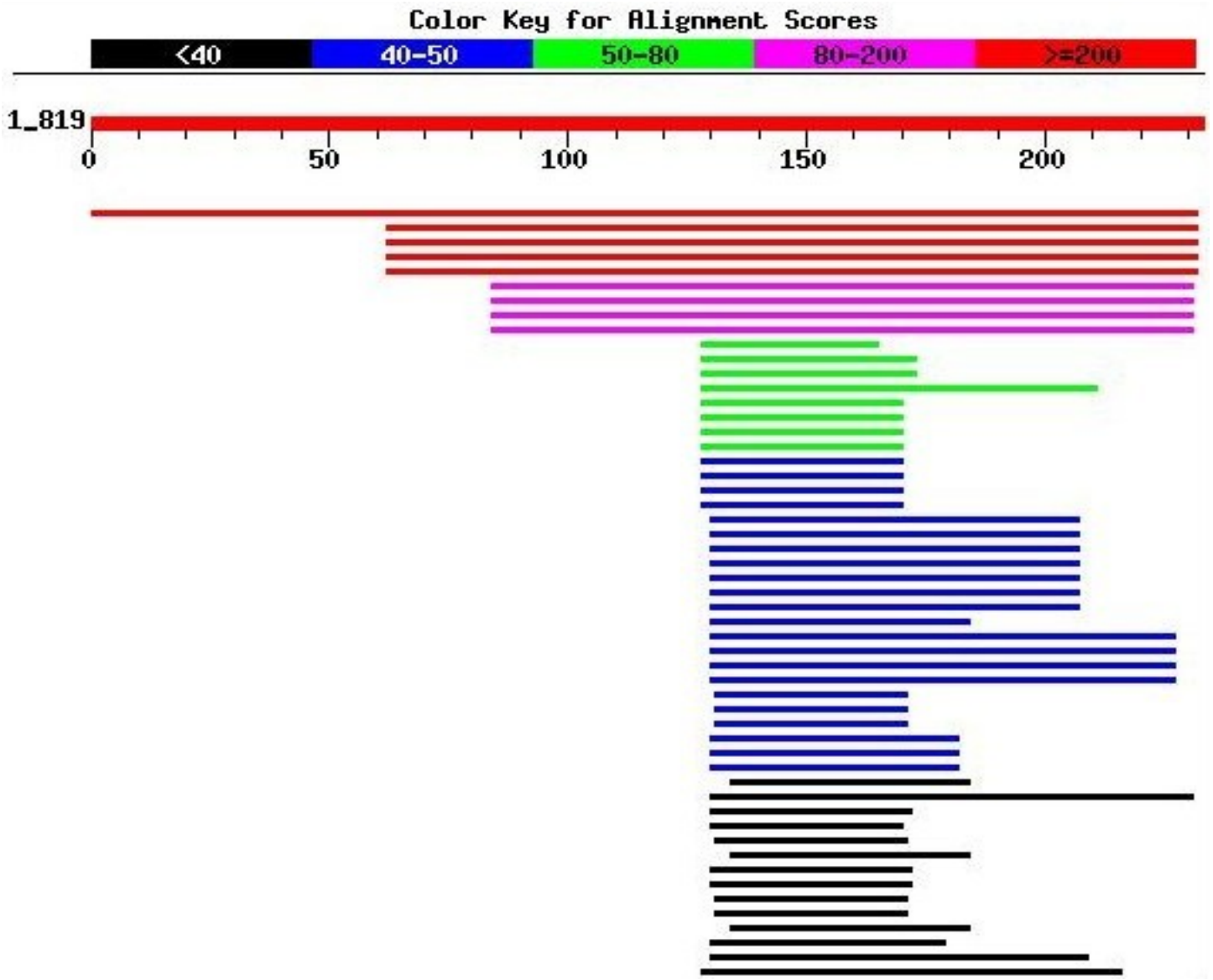
Scan a protein sequence database (targets) for good alignments to a query sequence (e.g. MASH-1, a transcription factor regulating neural development in rats)



- Many other BLAST options available through the homepage
- Algorithm parameters have defaults, but can be changed
- Also, masking filters can be specified

http://blast.ncbi.nlm.nih.gov/

Distribution of BLAST hits on the query sequence



BLAST "hit list" (with more details)

Sequences producing significant alignments:	Score (bits)	E Value
gi|112189|pir||S11563 probable MASH-2 protein - rat >gi|227...	291	8e-79
gi|440957|gb|AAB28830.1| Achaete-Scute homolog Mash-1 gene ...	283	3e-76
gi|2134688|pir||A48279 achaete scute protein - human >gi|30...	283	3e-76
gi|20455478|sp|P50553|ASH1_HUMAN Achaete-scute homolog 1 (H...	283	3e-76
gi|6678806|ref|NP_032579.1| achaete-scute complex homolog-l...	278	7e-75
gi|2642465|gb|AAB86993.1| Achaete-Scute homologue 2 [Homo s...	105	2e-22
gi|112188|pir||S11562 probable MASH-1 protein - rat >gi|566...	92	2e-18
gi|17432908|sp|O35885|ASH2_MOUSE Achaete-scute homolog 2 (M...	90	5e-18
gi|8574075|emb|CAB94773.1| Mash2 protein [Mus musculus] >gi...	89	1e-17
gi|1754729|gb|AAB39362.1| ASCL2 [Homo sapiens]	65	3e-10
gi|17456298|ref|XP_062690.1| similar to putative bHLH trans...	55	2e-07
gi|20863265|ref|XP_137216.1| similar to transcription facto...	53	1e-06
gi|27717809|ref|XP_235013.1| similar to Achaete-scute homol...	52	1e-06
gi|27679426|ref|XP_215039.1| similar to putative bHLH trans...	52	2e-06
gi|18249653|dbj|BAB83912.1| putative bHLH transcription fac...	51	3e-06
gi|28273166|tpg|DAA00301.1| TPA: class II basic helix-loop-...	51	3e-06
gi|20910395|ref|XP_136181.1| similar to putative bHLH trans...	50	4e-06
gi|13928056|emb|CAC37689.1| MASH5 protein [Mus musculus] >g...	50	7e-06
gi|18249655|dbj|BAB83913.1| putative bHLH transcription fac...	49	2e-05
gi|10190680|ref|NP_065697.1| ASCL3 [Homo sapiens] >gi|80522...	49	2e-05
gi|20454833|sp|Q9NQ33|ASH3_HUMAN Achaete-scute homolog 3 (b...	49	2e-05
gi|8648972|emb|CAB94840.1| dHAND basic helix-loop-helix tra...	48	2e-05
gi|12054812|emb|CAC20671.1| dHand protein [Mus musculus]	48	2e-05

A pairwise alignment with MASH-1

HASH-2, a human homolog of MASH-1

- "**+**" indicates conservative amino acid substitution
- "**–**" indicates gap/insertion
- XXXX… indicates areas of low complexity

```
 Score =  105 bits (261), Expect = 2e-22
 Identities = 73/170 (42%), Positives = 92/170 (54%), Gaps = 25/170 (14%)

Query: 85   RQRSSSPELMRCKRRLNFSGFGYSLPQQQPXXXXXXXXXXXXXXKLVNLGFATLREHVPN 144
            R+R +SPEL+RC RR    +    +                    KLVNLGF  LR+HVP+
Sbjct: 23   RRRPASPELLRCSRRRRPAT---AETGGGAAAVARRNERERNRVKLVNLGFQALRQHVPH 79

Query: 145 GAANKKMSKVETLRSAVEYIRALQQLLDEHDAVSAAFQAGVLSPTISPN---------- 193
            G A+KK+SKVETLRSAVEYIRALQ+LL EHDAV  A   G+    + P+
Sbjct: 80   GGASKKLSKVETLRSAVEYIRALQRLLAEHDAVRNALAGGLRPQAVRPSAPRGPPGTTPV 139

Query: 194 ----------YSNDLNSMAGSPVSSYSSDE-GSYDPLSPEEQELLDFTNW 232
                      +S  GSP S+YSSD+ G    LSP E+ELLDF++W
Sbjct: 140 AASPSRASSSPGRGGSSEPGSPRSAYSSDDSGCEGALSPAERELLDFSSW 189
```

BLAST+: an improved BLAST implementation

BMC Bioinformatics

BioMed Central

Software

Open Access

BLAST+: architecture and applications

Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer and Thomas L Madden*

Address: National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Building 38A, 8600 Rockville Pike, Bethesda, MD 20894, USA

Email: Christiam Camacho - camacho@ncbi.nlm.nih.gov; George Coulouris - coulouri@ncbi.nlm.nih.gov; Vahram Avagyan - avagyanv@ncbi.nlm.nih.gov; Ning Ma - maning@ncbi.nlm.nih.gov; Jason Papadopoulos - jasonp@boo.net; Kevin Bealer - kevinbealer@gmail.com; Thomas L Madden* - madden@ncbi.nlm.nih.gov

* Corresponding author

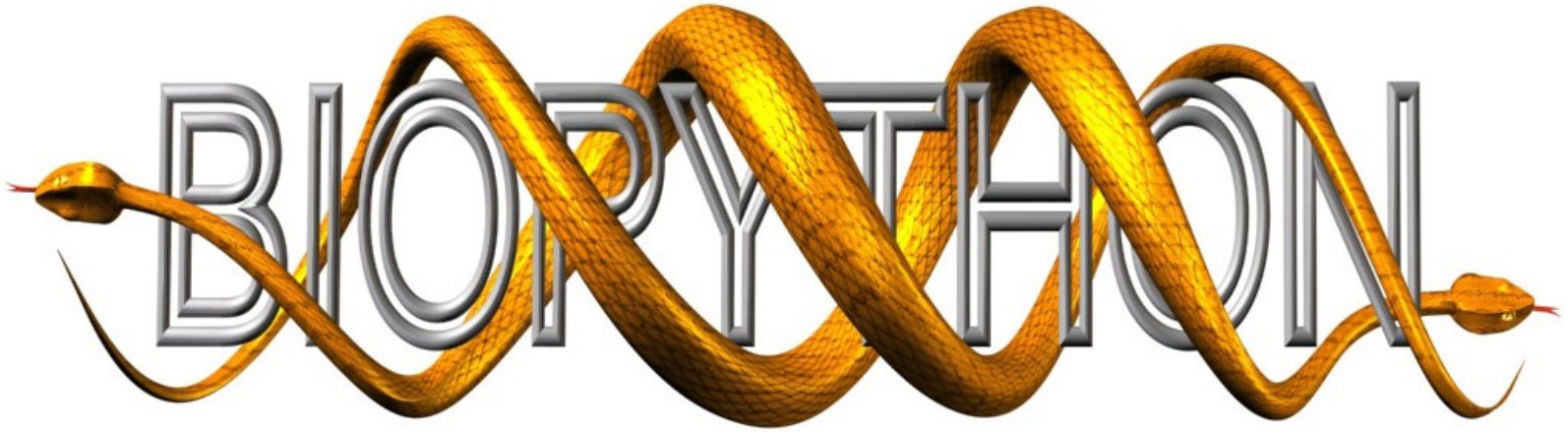
Published: 15 December 2009

BMC Bioinformatics 2009, **10**:421 doi:10.1186/1471-2105-10-421

Received: 28 July 2009
Accepted: 15 December 2009

This article is available from: http://www.biomedcentral.com/1471-2105/10/421

http://www.biomedcentral.com/content/pdf/1471-2105-10-421.pdf



Biopython Tutorial and Cookbook

**Jeff Chang, Brad Chapman, Iddo Friedberg, Thomas Hamelryck,
Michiel de Hoon, Peter Cock, Tiago Antao, Eric Talevich, Bartek Wilczyński**

Last Update – 25 June 2012 (Biopython 1.60)

Contents

http://biopython.org/DIST/docs/tutorial/Tutorial.html

Programmatic access to BLAST programs

- You can perform BLAST searches automatically from a local or remote server using an **API (application program interface).**
- **BioPython** contains modules that specifically deal with handling sequences and BLAST program data.
- To understand the context of these, we need to learn a little bit about:
  - Executing commandline commands using ***subprocess***
  - Structured text (for parsing BLAST output)
  - Objects (complex data structures that encapsulate **information** about sequences, or HSPs, etc. along with **methods** that operate on them).

Python **subprocess** module

- The ***subprocess*** modules allows you to execute system commands as if they had been typed at the commandline.
- You can specify program names to run, parameter lists, and control the destination for input, output and error streams (STDIN, STDOUT, STDERR).
- This module contains different functions that allow you to check, issue, and get output from system commandline calls.
- Simple invocation uses the ***call*** function:

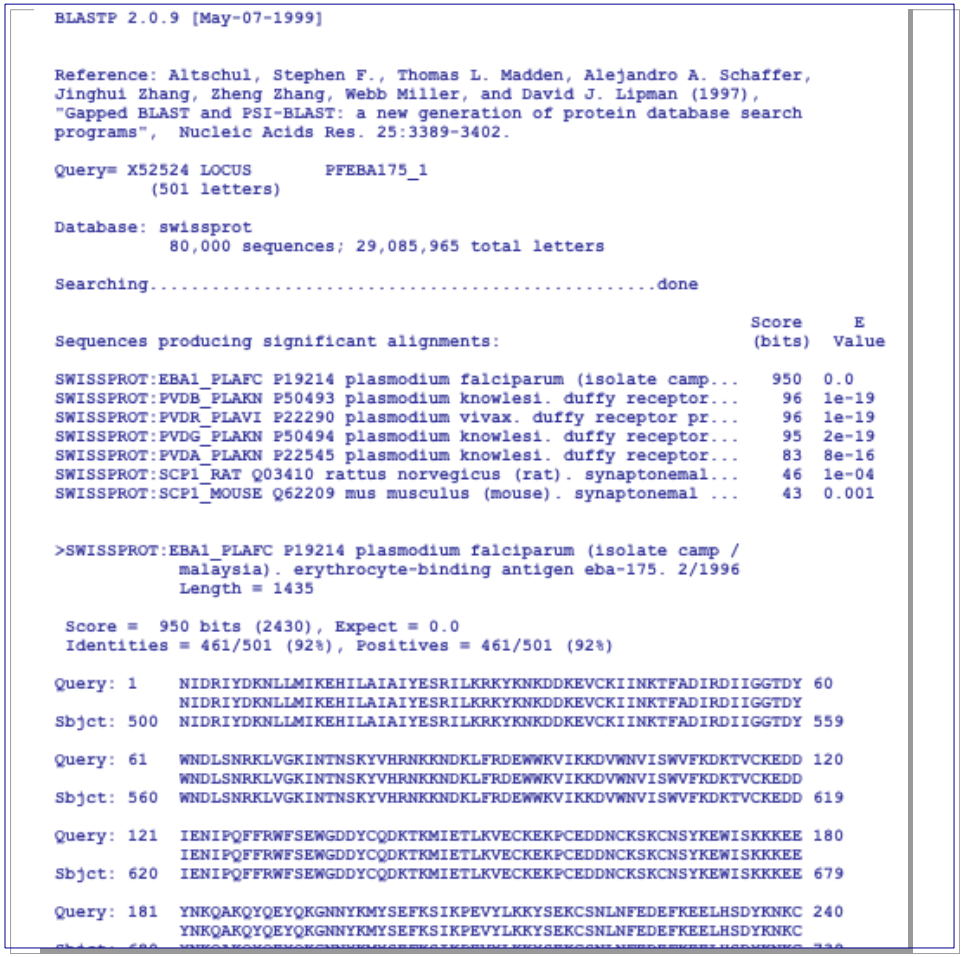
  `subprocess.call([`*`program_name,parameter_list`*`])`

- It will return the "return code" resulting from issuing the commandline call (usually, the return code is 0 if everything is ok).
- *Examples*:

```
subprocess.call(['ls','-l'])
subprocess.call(['blastp','-query','1UBQ.fa','-db','nr',
'-outfmt','5','-outfile','1UBQ.blast.xml'])
```

Flat File Formats

- So far, we've been dealing with "flat files", e.g. regular text or tab-delimited text files. An example for BLAST output is shown below.



```
BLASTP 2.0.9 [May-07-1999]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= X52524 LOCUS       PFEBA175_1
         (501 letters)

Database: swissprot
           80,000 sequences; 29,085,965 total letters

Searching.................................................done

                                                                  Score     E
Sequences producing significant alignments:                       (bits)  Value

SWISSPROT:EBA1_PLAFC P19214 plasmodium falciparum (isolate camp...   950  0.0
SWISSPROT:PVDB_PLAKN P50493 plasmodium knowlesi. duffy receptor...    96  1e-19
SWISSPROT:PVDR_PLAVI P22290 plasmodium vivax. duffy receptor pr...    96  1e-19
SWISSPROT:PVDG_PLAKN P50494 plasmodium knowlesi. duffy receptor...    95  2e-19
SWISSPROT:PVDA_PLAKN P22545 plasmodium knowlesi. duffy receptor...    83  8e-16
SWISSPROT:SCP1_RAT Q03410 rattus norvegicus (rat). synaptonemal...    46  1e-04
SWISSPROT:SCP1_MOUSE Q62209 mus musculus (mouse). synaptonemal ...    43  0.001


>SWISSPROT:EBA1_PLAFC P19214 plasmodium falciparum (isolate camp /
            malaysia). erythrocyte-binding antigen eba-175. 2/1996
            Length = 1435

 Score =  950 bits (2430), Expect = 0.0
 Identities = 461/501 (92%), Positives = 461/501 (92%)

Query: 1    NIDRIYDKNLLMIKEHILAIAIYESRILKRKYKNKDDKEVCKIINKTFADIRDIIGGTDY 60
            NIDRIYDKNLLMIKEHILAIAIYESRILKRKYKNKDDKEVCKIINKTFADIRDIIGGTDY
Sbjct: 500  NIDRIYDKNLLMIKEHILAIAIYESRILKRKYKNKDDKEVCKIINKTFADIRDIIGGTDY 559

Query: 61   WNDLSNRKLVGKINTNSKYVHRNKKNDKLFRDEWWKVIKKDVWNVISWVFKDKTVCKEDD 120
            WNDLSNRKLVGKINTNSKYVHRNKKNDKLFRDEWWKVIKKDVWNVISWVFKDKTVCKEDD
Sbjct: 560  WNDLSNRKLVGKINTNSKYVHRNKKNDKLFRDEWWKVIKKDVWNVISWVFKDKTVCKEDD 619

Query: 121  IENIPQFFRWFSEWGDDYCQDKTKMIETLKVECKEKPCEDDNCKSKCNSYKEWISKKKEE 180
            IENIPQFFRWFSEWGDDYCQDKTKMIETLKVECKEKPCEDDNCKSKCNSYKEWISKKKEE
Sbjct: 620  IENIPQFFRWFSEWGDDYCQDKTKMIETLKVECKEKPCEDDNCKSKCNSYKEWISKKKEE 679

Query: 181  YNKQAKQYQEYQKGNNYKMYSEFKSIKPEVYLKKYSEKCSNLNFEDEFKEELHSDYKNKC 240
            YNKQAKQYQEYQKGNNYKMYSEFKSIKPEVYLKKYSEKCSNLNFEDEFKEELHSDYKNKC
```

Structured Text

- Alternative file formats such as HTML and XML use **structured text**.
- The idea is to encapsulate **metadata**, or "data about data", in the file structure.
- HTML deals mostly with how to format web pages, e.g.:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<title>Hello World!</title>
</head>

<body>
     ... Stuff about the world here ...
</body>
```

Parsing BLAST XML output

- XML, or "extensible markup language", offers the possibility to use metadata tags in a structured hierarchy to describe different aspects of a complex data type's components.
- For example, a BLAST record has different elements that each reside in their own containers and describe different information about the result.
- Different bits of information have their own special **tags**:

```
<?xml version="1.0"?>
<!DOCTYPE BlastOutput PUBLIC "-//NCBI//NCBI BlastOutput/EN" "NCBI_BlastOutput.dtd">
<BlastOutput>
  <BlastOutput_program>blastn</BlastOutput_program>
  <BlastOutput_version>blastn 2.2.3 [May-13-2002]</BlastOutput_version>
  <BlastOutput_reference>~Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, ~Jinghui Zhang, Zheng Zhang, Web
b Miller, and David J. Lipman (1997), ~&amp;quot;Gapped BLAST and PSI-BLAST: a new generation of protein database search~programs&amp;quot;,
  Nucleic Acids Res. 25:3389-3402.</BlastOutput_reference>
  <BlastOutput_db>embl</BlastOutput_db>
  <BlastOutput_query-ID>lcl|QUERY</BlastOutput_query-ID>
  <BlastOutput_query-def>AF178033</BlastOutput_query-def>
  <BlastOutput_query-len>811</BlastOutput_query-len>
  <BlastOutput_param>
    <Parameters>
      <Parameters_expect>10</Parameters_expect>
      <Parameters_include>0</Parameters_include>
      <Parameters_sc-match>1</Parameters_sc-match>
      <Parameters_sc-mismatch>-3</Parameters_sc-mismatch>
      <Parameters_gap-open>5</Parameters_gap-open>
      <Parameters_gap-extend>2</Parameters_gap-extend>
      <Parameters_filter>D</Parameters_filter>
    </Parameters>
  </BlastOutput_param>
  <BlastOutput_iterations>
    <Iteration>
      <Iteration_iter-num>1</Iteration_iter-num>
      <Iteration_hits>
        <Hit>
```

BLAST XML output

- Each match hit gets its own multi-part container, which gives information about individual HSPs on the target sequence, including scores, alignments, etc.:


```
        <Hit>
          <Hit_num>1</Hit_num>
          <Hit_id>gnl|BL_ORD_ID|142512</Hit_id>
          <Hit_def>EMORG:AF178033 Af178033 Poecilia reticulata from Trinidad and Tobago NADH dehydrogenase subunit 2 (NADH2) gene, p
artial cds; mitochondrial gene for mitochondrial product. 3/2002</Hit_def>
          <Hit_accession>142512</Hit_accession>
          <Hit_len>811</Hit_len>
          <Hit_hsps>
            <Hsp>
              <Hsp_num>1</Hsp_num>
              <Hsp_bit-score>1566.56</Hsp_bit-score>
              <Hsp_score>790</Hsp_score>
              <Hsp_evalue>0</Hsp_evalue>
              <Hsp_query-from>1</Hsp_query-from>
              <Hsp_query-to>811</Hsp_query-to>
              <Hsp_hit-from>1</Hsp_hit-from>
              <Hsp_hit-to>811</Hsp_hit-to>
              <Hsp_pattern-from>0</Hsp_pattern-from>
              <Hsp_pattern-to>0</Hsp_pattern-to>
              <Hsp_query-frame>1</Hsp_query-frame>
              <Hsp_hit-frame>1</Hsp_hit-frame>
              <Hsp_identity>811</Hsp_identity>
              <Hsp_positive>811</Hsp_positive>
              <Hsp_gaps>0</Hsp_gaps>
              <Hsp_align-len>811</Hsp_align-len>
              <Hsp_density>0</Hsp_density>
              <Hsp_qseq>AGCACCCACTGGTATCTTGCCTGAATAGGAATTGAAATTAACACATTAGCCATTATCCCCCTAATATCACAAAACCACACCCCACGAGCAACTGAGGCCACCACTAAA
GCCATAAAAATTGGACTTGCCCCCCTTCACAGCTGAATACCAGAAGTAATACAAGGCTTAAGCCTACTTAATGGATTAATTCTATCCACTTGACAAAAACTTGCCCCCCTTTACCTCATCTACCAAATTCAA
CCAACCAACTCCAACATTTTTATTACCCTAGGACTTCTATCCATTATTGTAGGGGGGTGAGGGGGATTTAACCAAGTACAACTCCGAAAAATCCTAGCATACTCATCAATTGCCCACTTAGGGTGAATAATT
TTAATTCTTTCATTCTCACCTCCACTAGCCCTACTCACAATCCTAATTTATCTCCTAATAACCTTCTCACTATTTTCTTCTTTTATATTAACCCGAACCACACACATCAACTCTCTAGCCACTACATGGGCC
AAAATCCCAATTCTAACCATCTCAGCCCCCCTAGTCCTATTATCCCTAGGAGGATTGCCCCCTCTTACAGGATTTATACCAAAATGACTTATTCTCCAAGAATTAACAAAGCAAGACCTAGCCCCAATTGCC
ACTCTAGCCGCACTTTCATCCCTATTCAGCCTATATTTTTATC</Hsp_qseq>
              <Hsp_hseq>AGCACCCACTGGTATCTTGCCTGAATAGGAATTGAAATTAACACATTAGCCATTATCCCCCTAATATCACAAAACCACACCCCACGAGCAACTGAGGCCACCACTAAA
GCCATAAAAATTGGACTTGCCCCCCTTCACAGCTGAATACCAGAAGTAATACAAGGCTTAAGCCTACTTAATGGATTAATTCTATCCACTTGACAAAAACTTGCCCCCCTTTACCTCATCTACCAAATTCAA
CCAACCAACTCCAACATTTTTATTACCCTAGGACTTCTATCCATTATTGTAGGGGGGTGAGGGGGATTTAACCAAGTACAACTCCGAAAAATCCTAGCATACTCATCAATTGCCCACTTAGGGTGAATAATT
TTAATTCTTTCATTCTCACCTCCACTAGCCCTACTCACAATCCTAATTTATCTCCTAATAACCTTCTCACTATTTTCTTCTTTTATATTAACCCGAACCACACACATCAACTCTCTAGCCACTACATGGGCC
AAAATCCCAATTCTAACCATCTCAGCCCCCCTAGTCCTATTATCCCTAGGAGGATTGCCCCCTCTTACAGGATTTATACCAAAATGACTTATTCTCCAAGAATTAACAAAGCAAGACCTAGCCCCAATTGCC
ACTCTAGCCGCACTTTCATCCCTATTCAGCCTATATTTTTATC</Hsp_hseq>
              <Hsp_midline>||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
||||||||||||||||||||||||||||||||||||||||||||||</Hsp_midline>
            </Hsp>
          </Hit_hsps>
        </Hit>
```

Biopython and NCBI Blast

- You can use BioPython to run BLAST either **remotely** or **locally**.
- To run it **locally**, the BLAST suite of tools must be installed on your local server.
- This will usually be faster than running the search remotely.
- **BLAST is installed on prince**.

- There are LOTS of parameters…
- **You need to know how to use BLAST first!**

- **You will see the nuts and bolts during lab today.**

Example: BLAST using a remote server

```
In [1]: from Bio.Blast import NCBIWWW

In [2]: help(NCBIWWW.qblast)

Help on function qblast in module Bio.Blast.NCBIWWW:

qblast(program, database, sequence, auto_format=None, composition_based_statistics=None, db_genetic_code=None, endpoints=None,
entrez_query='(none)', expect=10.0, filter=None, gapcosts=None, genetic_code=None, hitlist_size=50, i_thresh=None, layout=None,
lcase_mask=None, matrix_name=None, nucl_penalty=None, nucl_reward=None, other_advanced=None, perc_ident=None, phi_pattern=None,
query_file=None, query_believe_defline=None, query_from=None, query_to=None, searchsp_eff=None, service=None, threshold=None,
ungapped_alignment=None, word_size=None, alignments=500, alignment_view=None, descriptions=500, entrez_links_new_window=None,
expect_low=None, expect_high=None, format_entrez_query=None, format_object=None, format_type='XML', ncbi_gi=None, results_file=None,
show_overview=None, megablast=None)
    Do a BLAST search using the QBLAST server at NCBI.

    Supports all parameters of the qblast API for Put and Get.
    Some useful parameters:
    program        blastn, blastp, blastx, tblastn, or tblastx (lower case)
    database       Which database to search against (e.g. "nr").
    sequence       The sequence to search.
    ncbi_gi        TRUE/FALSE whether to give 'gi' identifier.
    descriptions   Number of descriptions to show.  Def 500.
    alignments     Number of alignments to show.  Def 500.
    expect         An expect value cutoff.  Def 10.0.
    matrix_name    Specify an alt. matrix (PAM30, PAM70, BLOSUM80, BLOSUM45).
    filter         "none" turns off filtering.  Default no filtering
    format_type    "HTML", "Text", "ASN.1", or "XML".  Def. "XML".
    entrez_query   Entrez query to limit Blast search
    hitlist_size   Number of hits to return. Default 50
    megablast      TRUE/FALSE whether to use MEga BLAST algorithm (blastn only)
    service        plain, psi, phi, rpsblast, megablast (lower case)

    This function does no checking of the validity of the parameters
    and passes the values to the server as is.  More help is available at:
    http://www.ncbi.nlm.nih.gov/BLAST/blast_overview.html
(END)
```

http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc83

Example: BLAST using a remote server

- Required parameters:
  - Blast program, Blast database, Sequence
  - Returns **XML formatted** results, by default.
- Save results to a file, for parsing…

```
In [3]: result_handle = NCBIWWW.qblast("blastn","nr","8332116")

In [4]: blast_results = result_handle.read()

In [5]: result_handle.close()

In [6]: save_file = open("blastn-nr-8332166.xml","w")

In [7]: save_file.write(blast_results)

In [8]: save_file.close()
```

Example: BLAST using a **local** server

A typical blast commandline looks like this:

```
blastx -query opuntia.fasta -db nr -out opuntia.xml -evalue 0.001 -outfmt 5
```

- This command will run BLASTX against the non-redundant (NR) database, using an e-value cutoff of 0.001, and output the results to an output file in XML format.

From within Biopython we can use the NCBI BLASTX wrapper from the `Bio.Blast.Applications` module to build the command line string, and run it:

```
>>> from Bio.Blast.Applications import NcbiblastxCommandline
>>> help(NcbiblastxCommandline)
...
>>> blastx_cline = NcbiblastxCommandline(query="opuntia.fasta", db="nr", evalue=0.001,
...                                      outfmt=5, out="opuntia.xml")
>>> blastx_cline
NcbiblastxCommandline(cmd='blastx', out='opuntia.xml', outfmt=5, query='opuntia.fasta',
db='nr', evalue=0.001)
>>> print blastx_cline
blastx -out opuntia.xml -outfmt 5 -query opuntia.fasta -db nr -evalue 0.001
>>> stdout, stderr = blastx_cline()
```

In this example there shouldn't be any output from BLASTX to the terminal, so stdout and stderr should be empty. You may want to check the output file `opuntia.xml` has been created.

- You can then use `Bio.Blast.NCBIXML.parse()` to parse the BLAST XML output.
- **You will do this in lab today.**

http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc83