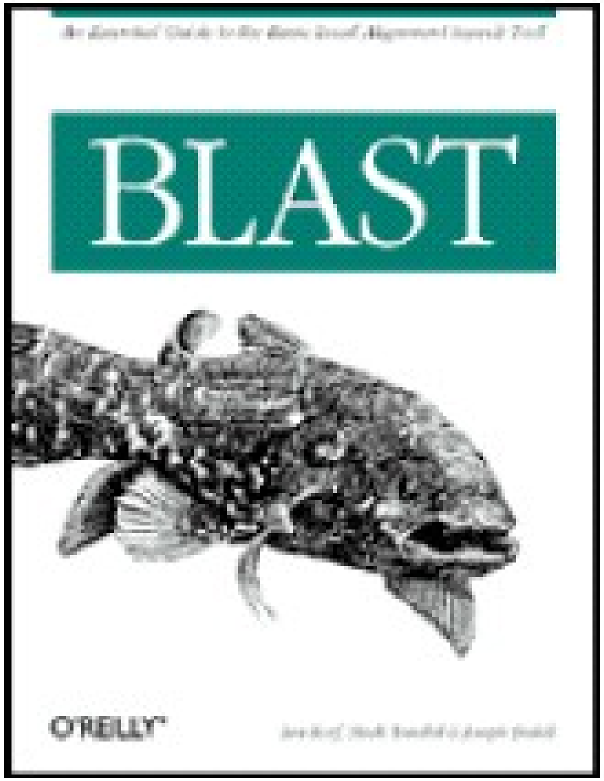


An Essential Guide to the Basic Local Alignment Search Tool

BLAST

O'REILLY®

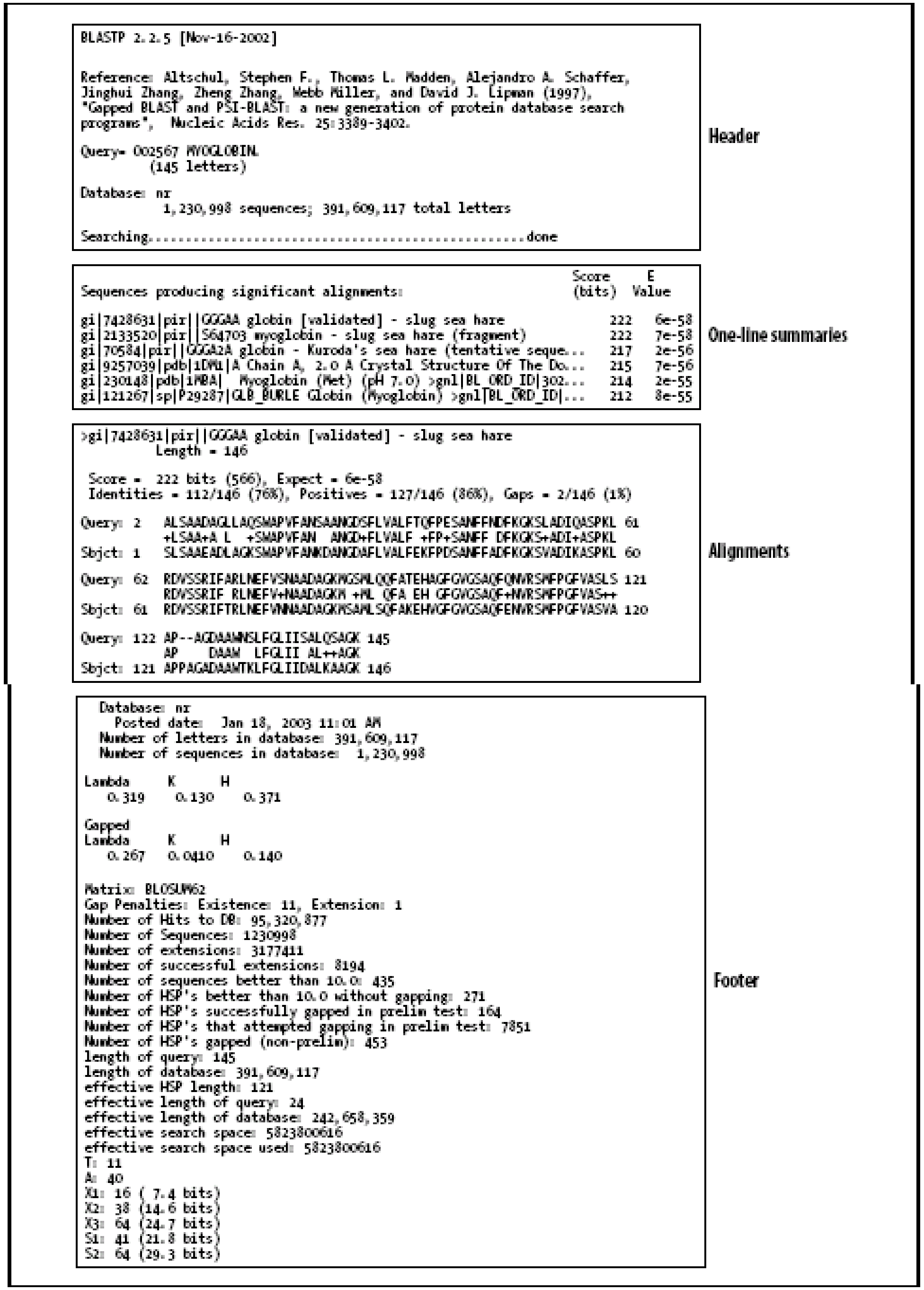
Ian Korf, Mark Yandell & Joseph Bedell

The 5 Standard BLAST Programs

Program	Database	Query	Typical Uses
BLASTN	Nucleotide	Nucleotide	Mapping oligonucleotides, amplimers, ESTs, and repeats to a genome. Identifying related transcripts.
BLASTP	Protein	Protein	Identifying common regions between proteins. Collecting related proteins for phylogenetic analysis.
BLASTX	Protein	Nucleotide	Finding protein-coding genes in genomic DNA.
TBLASTN	Nucleotide	Protein	Identifying transcripts similar to a known protein (finding proteins not yet in GenBank). Mapping a protein to genomic DNA.
TBLASTX	Nucleotide	Nucleotide	Cross-species gene prediction. Searching for genes missed by traditional methods.

WU-BLAST vs. NCBI-BLAST

- faster (except for BLASTN)
- word size unlimited
- nucleotide matrices
- gapped lambda for BLASTN
- links, topcomboN, kap
- altscore
- no additional output formats
- no PSI-BLAST, PHI-BLAST, MegaBLAST

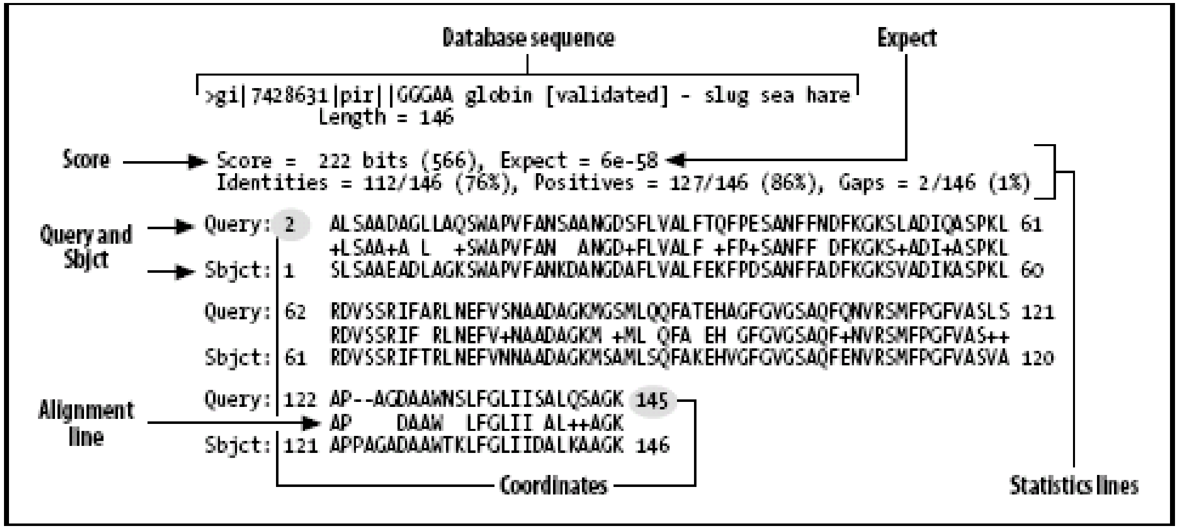


```
BLASTP 2.2.5 [Nov-16-2002]


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= 002567 MYOGLOBIN.
         (145 letters)

Database: nr
           1,230,998 sequences; 391,609,117 total letters

Searching..................................................done
```

```
                                                                 Score    E
Sequences producing significant alignments:                      (bits)  Value

gi|7428631|pir||GGGAA globin [validated] - slug sea hare             222   6e-58
gi|2133520|pir||S64703 myoglobin - slug sea hare (fragment)          222   7e-58
gi|70584|pir||GGGA2A globin - Kuroda's sea hare (tentative seque...  217   2e-56
gi|9257039|pdb|1DM1|A Chain A, 2.0 A Crystal Structure Of The Do...  215   7e-56
gi|230148|pdb|1MBA|  Myoglobin (Met) (pH 7.0) >gnl|BL_ORD_ID|302...  214   2e-55
gi|121267|sp|P29287|GLB_BURLE Globin (Myoglobin) >gnl|BL_ORD_ID|...  212   8e-55
```

```
>gi|7428631|pir||GGGAA globin [validated] - slug sea hare
          Length = 146

 Score =  222 bits (566), Expect = 6e-58
 Identities = 112/146 (76%), Positives = 127/146 (86%), Gaps = 2/146 (1%)

Query: 2   ALSAADAGLLAQSWAPVFANSAANGDSFLVALFTQFPESANFFNDFKGKSLADIQASPKL 61
           +LSAA+A L  +SWAPVFAN  ANGD+FLVALF +FP+SANFF DFKGKS+ADI+ASPKL
Sbjct: 1   SLSAAEADLAGKSWAPVFANKDANGDAFLVALFEKFPDSANFFADFKGKSVADIKASPKL 60

Query: 62  RDVSSRIFARLNEFVSNAADAGKMGSMLQQFATEHAGFGVGSAQFQNVRSMFPGFVASLS 121
           RDVSSRIF RLNEFV+NAADAGKM +ML QFA EH GFGVGSAQF+NVRSMFPGFVAS++
Sbjct: 61  RDVSSRIFTRLNEFVNNAADAGKMSAMLSQFAKEHVGFGVGSAQFENVRSMFPGFVASVA 120

Query: 122 AP--AGDAAWNSLFGLIISALQSAGK 145
           AP    DAAW  LFGLII AL++AGK
Sbjct: 121 APPAGADAAWTKLFGLIIDALKAAGK 146
```

```
  Database: nr
    Posted date:  Jan 18, 2003 11:01 AM
  Number of letters in database: 391,609,117
  Number of sequences in database:  1,230,998

Lambda     K      H
   0.319    0.130    0.371

Gapped
Lambda     K      H
   0.267   0.0410    0.140


Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Number of Hits to DB: 95,320,877
Number of Sequences: 1230998
Number of extensions: 3177411
Number of successful extensions: 8194
Number of sequences better than 10.0: 435
Number of HSP's better than 10.0 without gapping: 271
Number of HSP's successfully gapped in prelim test: 164
Number of HSP's that attempted gapping in prelim test: 7851
Number of HSP's gapped (non-prelim): 453
length of query: 145
length of database: 391,609,117
effective HSP length: 121
effective length of query: 24
effective length of database: 242,658,359
effective search space: 5823800616
effective search space used: 5823800616
T: 11
A: 40
X1: 16 ( 7.4 bits)
X2: 38 (14.6 bits)
X3: 64 (24.7 bits)
S1: 41 (21.8 bits)
S2: 64 (29.3 bits)
```



```
>gi|7428631|pir||GGGAA globin [validated] - slug sea hare
          Length = 146

 Score =  222 bits (566), Expect = 6e-58
 Identities = 112/146 (76%), Positives = 127/146 (86%), Gaps = 2/146 (1%)

Query: 2   ALSAADAGLLAQSWAPVFANSAANGDSFLVALFTQFPESANFFNDFKGKSLADIQASPKL 61
            +LSAA+A L  +SWAPVFAN  ANGD+FLVALF +FP+SANFF DFKGKS+ADI+ASPKL
Sbjct: 1   SLSAAEADLAGKSWAPVFANKDANGDAFLVALFEKFPDSANFFADFKGKSVADIKASPKL 60

Query: 62  RDVSSRIFARLNEFVSNAADAGKMGSMLQQFATEHAGFGVGSAQFQNVRSMFPGFVASLS 121
            RDVSSRIF RLNEFV+NAADAGKM +ML QFA EH GFGVGSAQF+NVRSMFPGFVAS++
Sbjct: 61  RDVSSRIFTRLNEFVNNAADAGKMSAMLSQFAKEHVGFGVGSAQFENVRSMFPGFVASVA 120

Query: 122 AP--AGDAAWNSLFGLIISALQSAGK 145
            AP    DAAW  LFGLII AL++AGK
Sbjct: 121 APPAGADAAWTKLFGLIIDALKAAGK 146
```

Figure 6-2. A BLASTP alignment

```
>gi|23098447|ref|NP_691913.1| (NC_004193) 3-oxoacyl-(acyl carrier
          protein) reductase [Oceanobacillus iheyensis]
         Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146 VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI
4027
           VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10   VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```

TWO ASPECTS OF BLAST

BLAST ALGORITHM

Word Hit Heuristic

Extension Heuristic

BLAST STATISTCS

Karlin-Altschul statistics:
a general theory of alignment statistics
Applicability goes well beyond BLAST

BLAST uses Karlin-Altschul Statistics to determine
the statistical significance of the alignments it produces.

TWO ASPECTS OF BLAST

BLAST ALGORITHM

Word Hit Heuristic

Extension Heuristic

BLAST STATISTCS

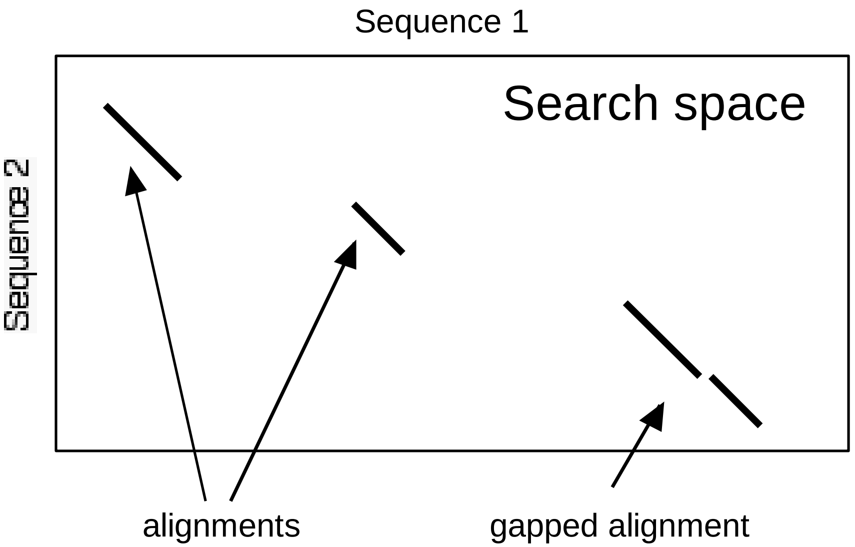
Karlin-Altschul statistics:
a general theory of alignment statistics
Applicability goes well beyond BLAST

BLAST uses Karlin-Altschul Statistics to determine
the statistical significance of the alignments it produces.

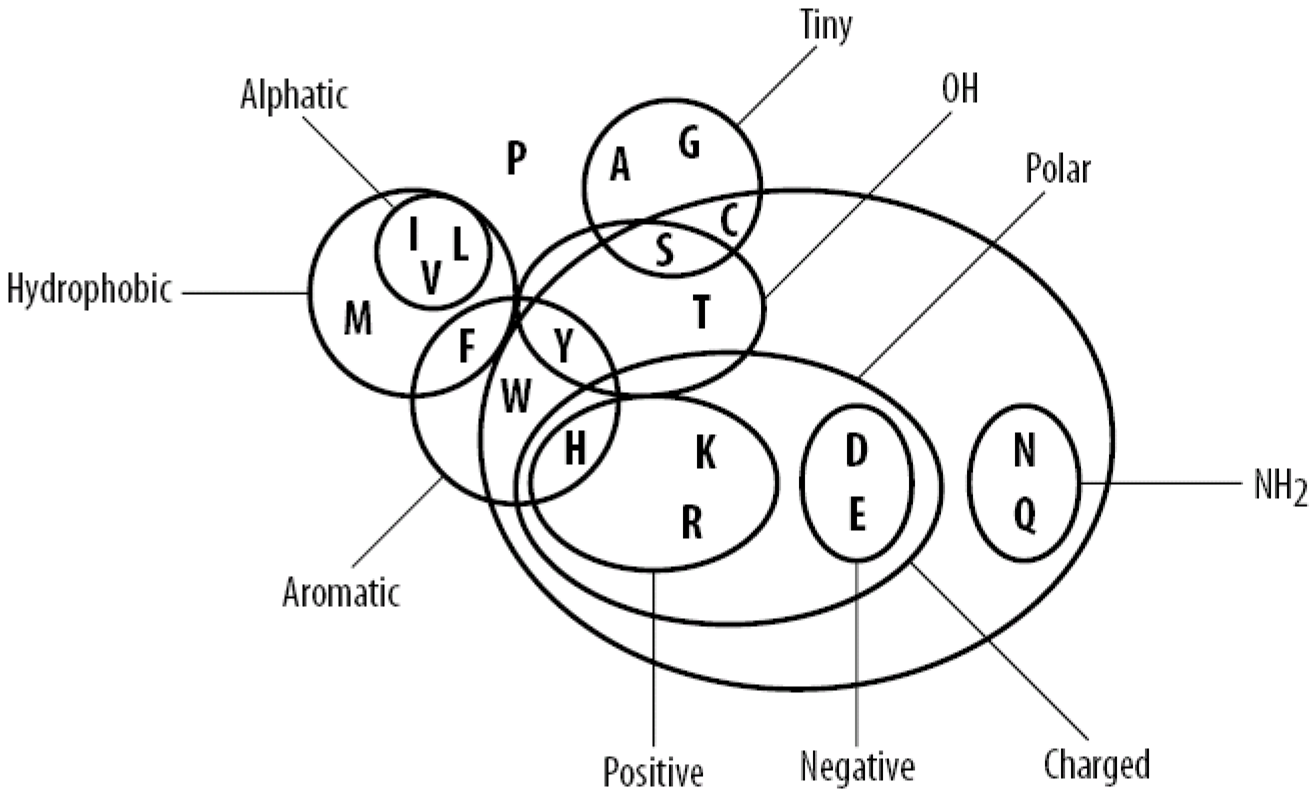
```
>gi|23098447|ref|NP_691913.1| (NC_004193) 3-oxoacyl-(acyl carrier
          protein) reductase [Oceanobacillus iheyensis]
         Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146
VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
            VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10
VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```

Alignment Overview

Sequence alignment takes place in a 2-dimensional space where diagonal lines represent regions of similarity. Gaps in an alignment appear as broken diagonals. The search space is sometimes considered as 2 sequences and somtimes as query x database.



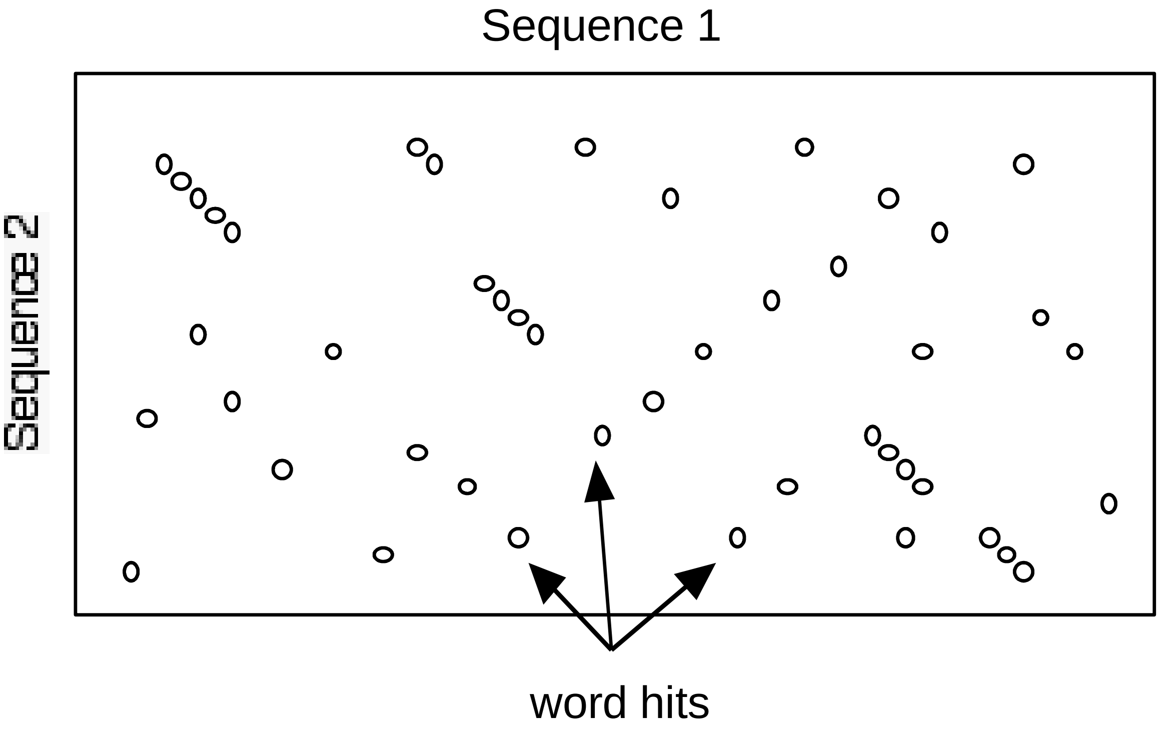
- Global alignment vs. local alignment
  - BLAST is local
- Maximum scoring pair (MSP) vs. High-scoring pair (HSP)
  - BLAST finds HSPs (usually the MSP too)
- Gapped vs. ungapped
  - BLAST can do both



```
PGNPFATPLEILPEWYLYPVFQILRVLPNKLLGIACQGAIPLGLMMVPFIE
PANPFATPLEILPEWYFYPVFQILRTVPNKLLGVLAMAAVPVGLLTVPFIE
PANPMSTPAHIVPEWYFLPVYAILRSIPNKLGGVAAIGLVFVSLLALPFIN
PANPLVTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLFSILMLLLVPFLH
PANPLSTPAHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILVLIFIPMLQ
PANPLSTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILILIFIPMLQ
IANPMNTPTHIKPEWYFLFAYSILRAIPNKLGGVIGLVMSILIL..YIMIF
ESDPMMSPVHIVPEWYFLFAYAILRAIPNKVLGVVSLFASILVL..VVFVL
IVDTLKTSDKILPEWFFLYLFGFLKAIPDKFMGLFLMVILLFSL..FLFIL
```

The BLAST Algorithm: Seeding (W and T)

- Speed gained by minimizing search space
- Alignments require word hits
- Neighborhood words
- W and T modulate speed and sensitivity

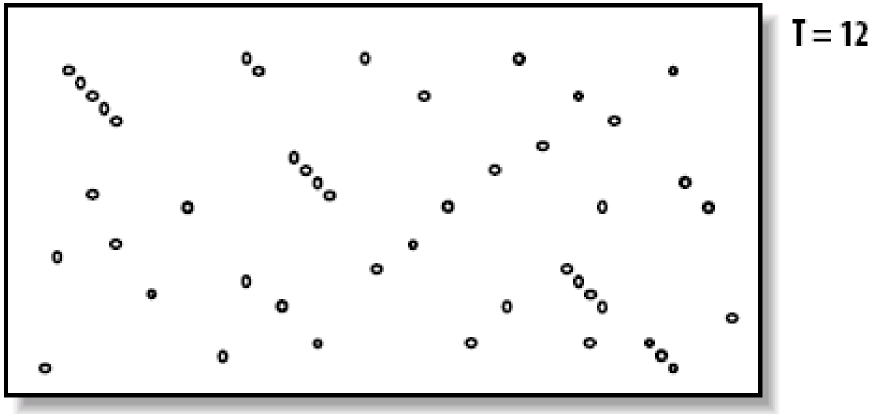


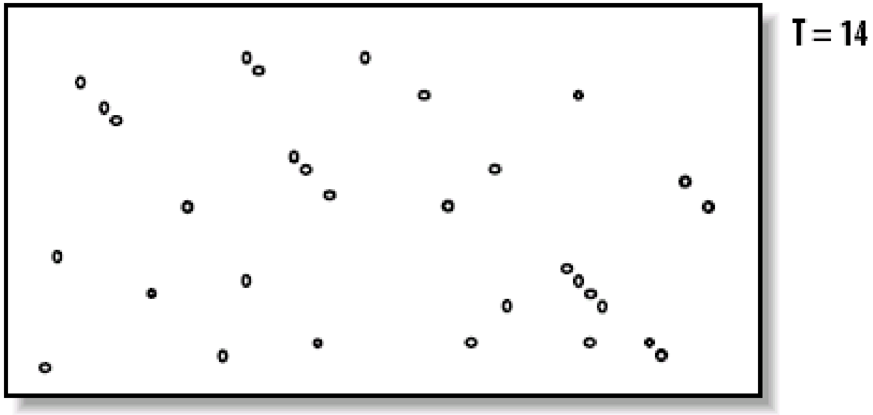
BLOSUM62 neighborhood of RGD

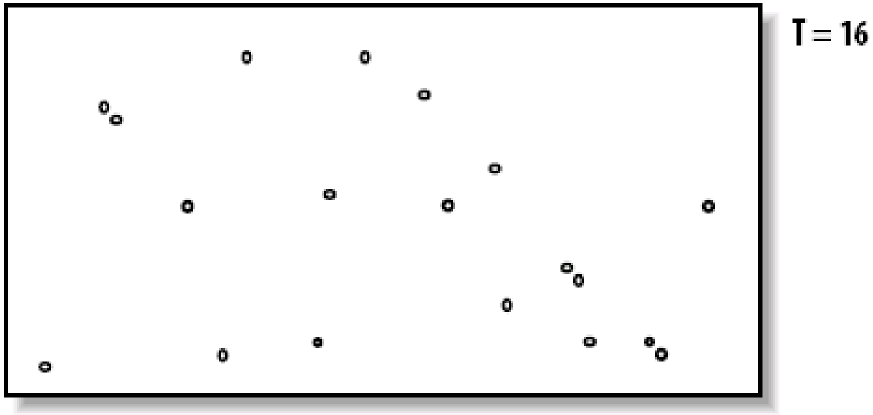
RGD	17
KGD	14
QGD	13
RGE	13
EGD	12
HGD	12
NGD	12
RGN	12

T=12

AGD	11
MGD	11
RAD	11
RGQ	11
RGS	11
RND	11
RSD	11
SGD	11
TGD	11

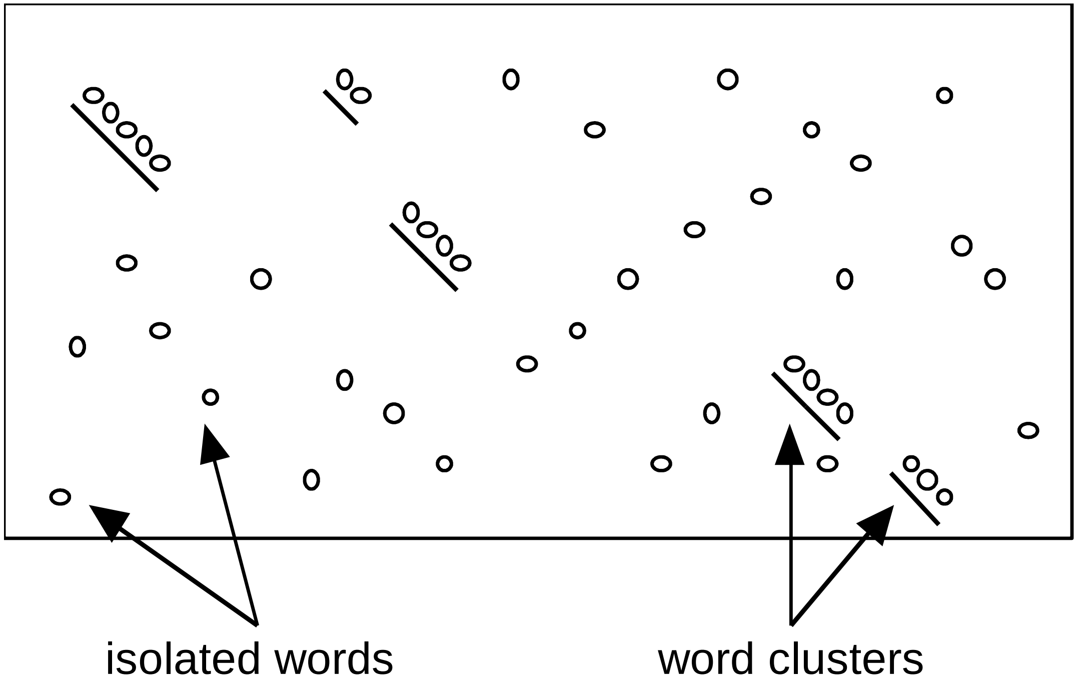

T = 12


T = 14


T = 16

The BLAST Algorithm: 2-hit Seeding

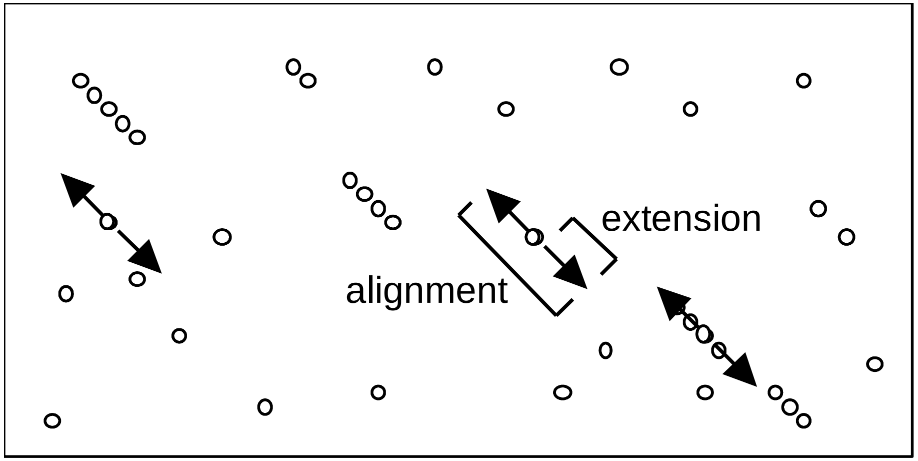
- Alignments tend to have multiple word hits.
- Isolated word hits are frequently false leads.
- Most alignments have large ungapped regions.



- Requiring 2 word hits on the same diagonal (of 40 aa for example), greatly increases speed at a slight cost in sensitivity.

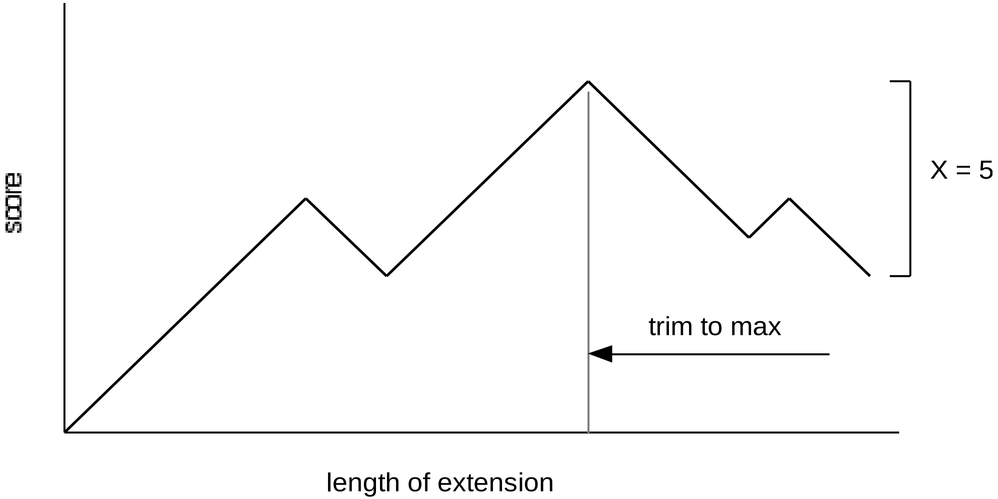
The BLAST Algorithm: Extension

- Alignments are extended from seeds in each direction.
- Extension is terminated when the maximum score drops below X.



Text example
match +1
mismatch -1
no gaps

```
The quick brown fox jumps over the lazy dog.
The quiet brown cat purrs when she sees him.
```



```
>gi|23098447|ref|NP_691913.1| (NC_004193) 3-oxoacyl-(acyl carrier
            protein) reductase [Oceanobacillus iheyensis]
          Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146 VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI
4027
            VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10   VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```

TWO ASPECTS OF BLAST

BLAST ALGORITHM

Word Hit Heuristic

Extension Heuristic

BLAST STATISTCS

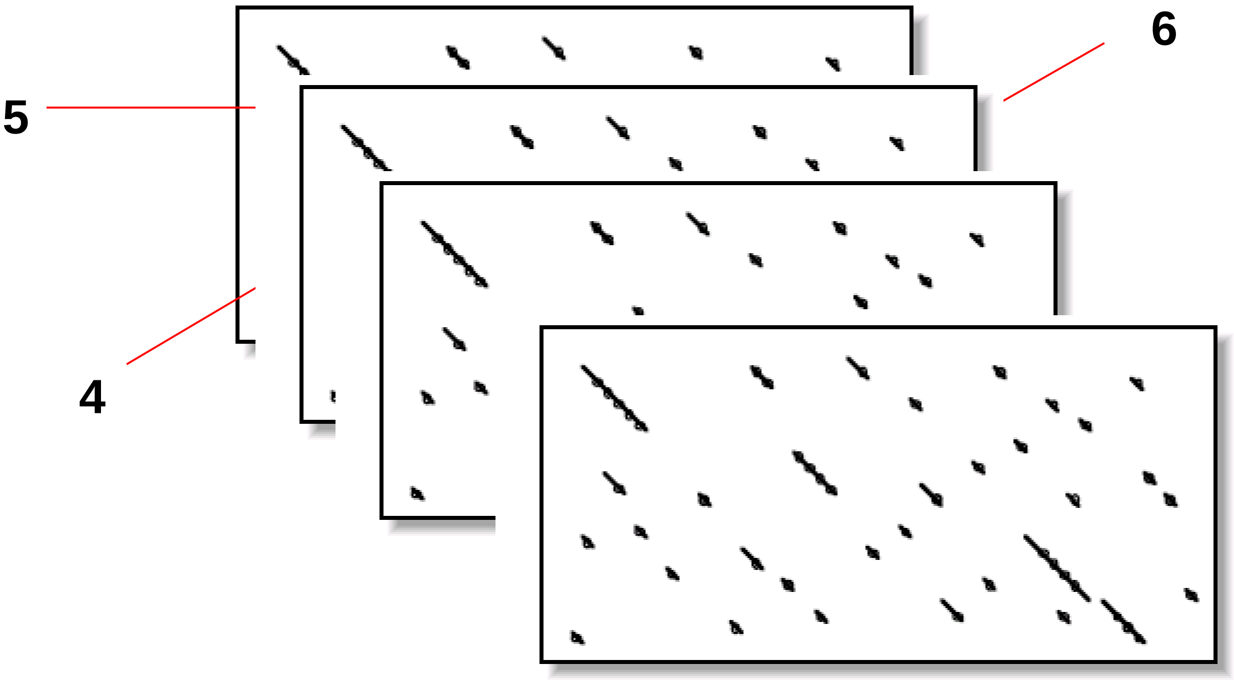
Karlin-Altschul statistics:
a general theory of alignment statistics
Applicability goes well beyond BLAST

BLAST uses Karlin-Altschul Statistics to determine
the statistical significance of the alignments it produces.

BLAST STATISTCS

Karlin-Altschul statistics:
a general theory of alignment statistics; applicability goes well beyond BLAST

Notational issues
Information theory: nats & bits
How alignments are scored
Hw scoring schemes are created
λ , E & H



How many runs with a score of X do we expect to find?

Understanding Gaussian sum notation

$$total = \sum_{i=1}^{n} p_i$$

```
 my %frequences;

 $frequencies{‘A’} = 0.25;
 $frequencies{‘T’} = 0.25;
 $frequencies{‘G’} = 0.25;
 $frequencies{‘C’} = 0.25;

my $total = 0;
foreach my $k (keys %frequencies){
            $total += $frequencies{$k};
}
```

A little information theory

$$-\log_2(0.5) = 1$$

tththttt

$$H = -\sum_{i}^{n} p_i \log_2 p_i$$

- ((0.5)(-1) + (0.5)(-1)) = 1 bit

- ((0.75)(-0.415) + (0.25)(-2)) = 0.81 bits

G=A=T=C=0.25

$$- (\ (0.25)(-2) + (0.25)(-2) + (0.25)(-2) + (0.25)(-2)\ ) = 2 \text{ bits}$$

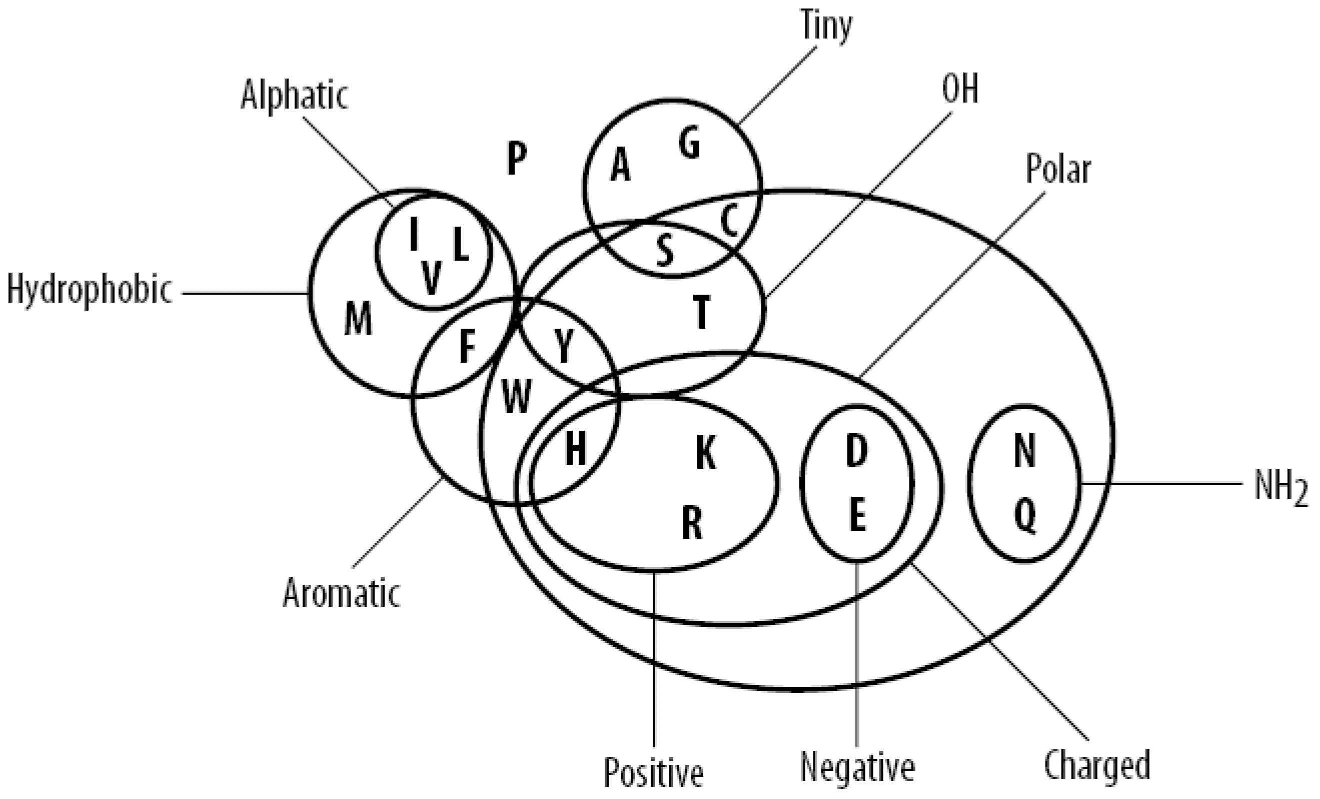
A=T=0.45; G=C=0.05

$$- (\ 2(0.45)(-1.15) + 2(0.05)(-4.32)\ ) = 1.47 \text{ bits}$$

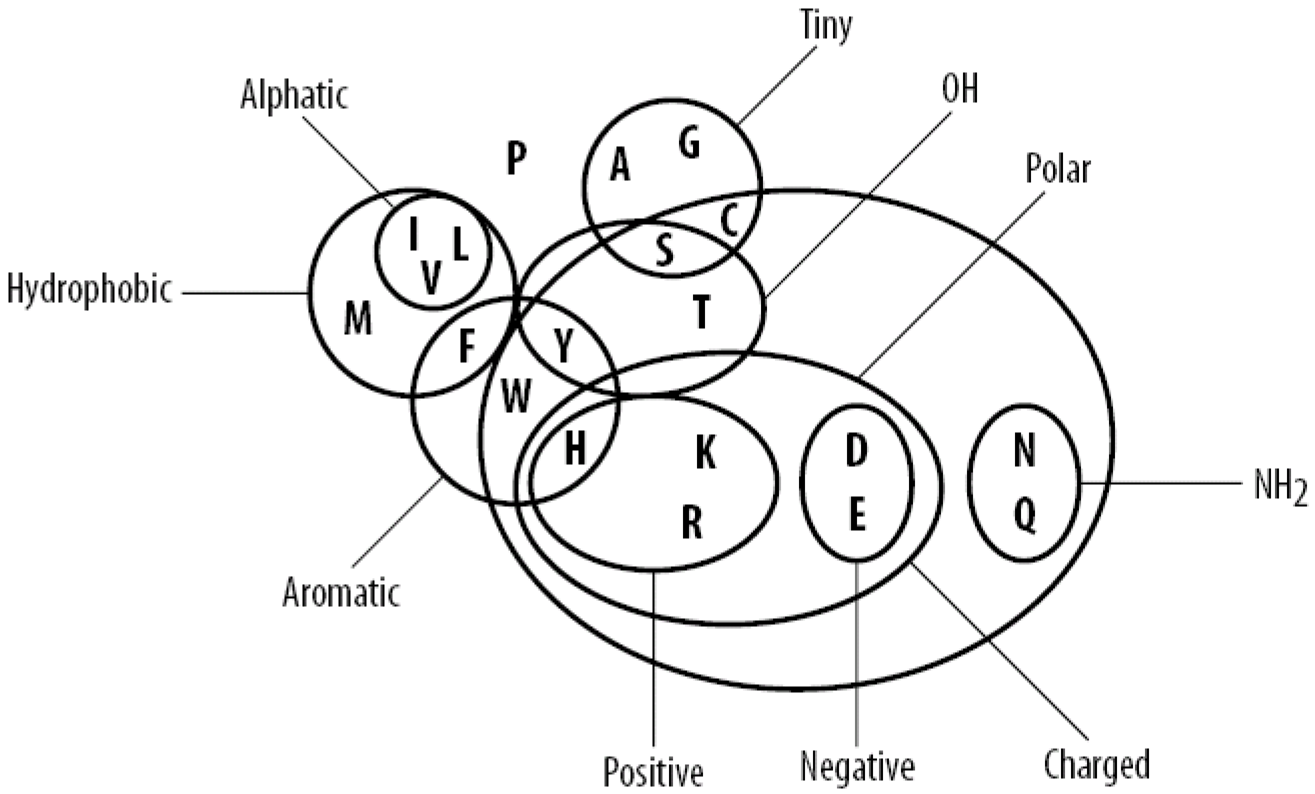
bits vs. nats

$$bits = \log_2(n)$$
$$nats = \log_e(n)$$

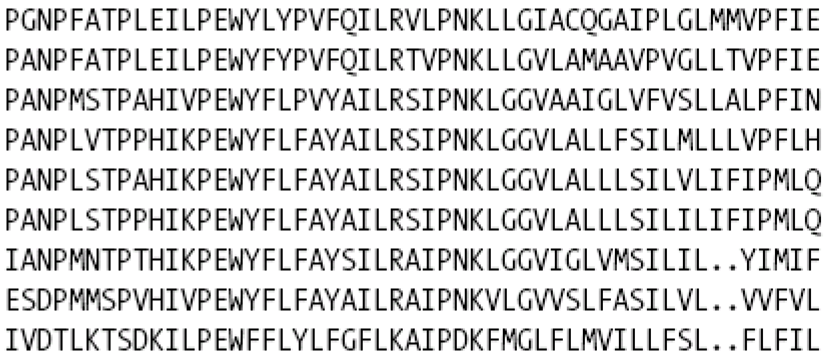
$$\log_2(n) = \log_e(n)/\log_e(2)$$



```
PGNPFATPLEILPEWYLYPVFQILRVLPNKLLGIACQGAIPLGLMMVPFIE
PANPFATPLEILPEWYFYPVFQILRTVPNKLLGVLAMAAVPVGLLTVPFIE
PANPMSTPAHIVPEWYFLPVYAILRSIPNKLGGVAAIGLVFVSLLALPFIN
PANPLVTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLFSILMLLLVPFLH
PANPLSTPAHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILVLIFIPMLQ
PANPLSTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILILIFIPMLQ
IANPMNTPTHIKPEWYFLFAYSILRAIPNKLGGVIGLVMSILIL..YIMIF
ESDPMMSPVHIVPEWYFLFAYAILRAIPNKVLGVVSLFASILVL..VVFVL
IVDTLKTSDKILPEWFFLYLFGFLKAIPDKFMGLFLMVILLFSL..FLFIL
```



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



```
PGNPFATPLEILPEWYLYPVFQILRVLPNKLLGIACQGAIPLGLMMVPFIE
PANPFATPLEILPEWYFYPVFQILRTVPNKLLGVLAMAAVPVGLLTVPFIE
PANPMSTPAHIVPEWYFLPVYAILRSIPNKLGGVAAIGLVFVSLLALPFIN
PANPLVTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLFSILMLLLVPFLH
PANPLSTPAHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILVLIFIPMLQ
PANPLSTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILILIFIPMLQ
IANPMNTPTHIKPEWYFLFAYSILRAIPNKLGGVIGLVMSILIL..YIMIF
ESDPMMSPVHIVPEWYFLFAYAILRAIPNKVLGVVSLFASILVL..VVFVL
IVDTLKTSDKILPEWFFLYLFGFLKAIPDKFMGLFLMVILLFSL..FLFIL
```

$$S_{ij} = \log(q_{ij}/p_i p_j)$$

p_M=0.01

p_I =0.1

q_{MI}=0.002

S_{MI}=log$_2$(.002/0.01*0.1) = +1 bits

S_{MI}=log$_e$(.002/0.01*0.1) = +.693 nats

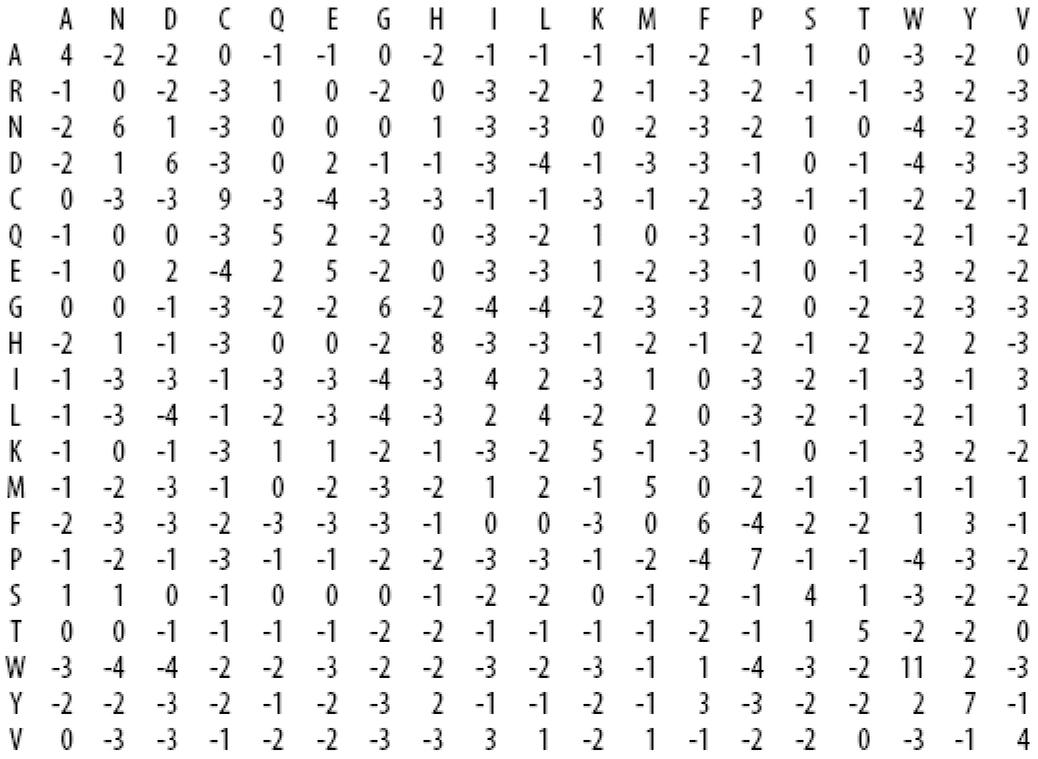
p_R=0.1

p_L =0.1

q_{RL}=0.002

S_{RL}=log$_2$(.002/0.1*0.1) = -2.322 bits

S_{RL}=log$_e$(.002/0.01*0.1) = -1.609 nats



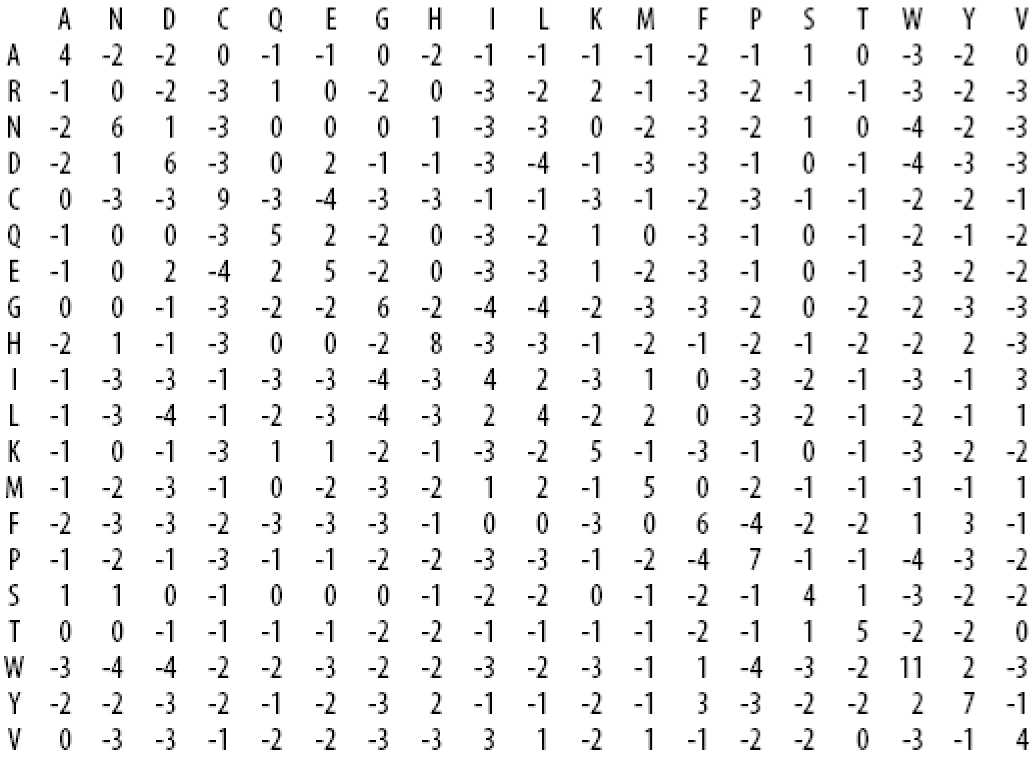
	A	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2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

The BLOSUM MATRICES are int($\log_2$ *3)

‘munge’ factor

The BLOSUM MATRICES are int(log$_2$ *3)

‘munge’ factor



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

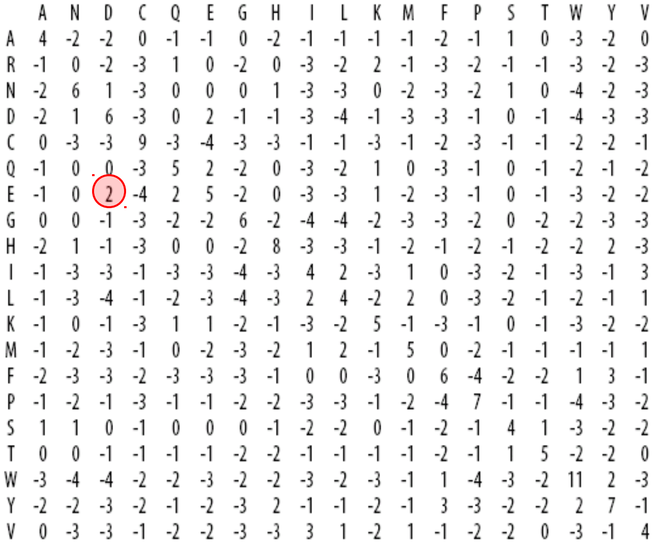
Why do this?

Recall that :

$$\text{Int}(3* S_{ij}) = \log_2(q_{ij}/p_ip_j)$$

λ is the number that will convert the 'munged' ***S******* back into its 'original' ***q*** for purposes of further calculation.

$$q_{ij} = p_ip_je^{\lambda s_{ij}}$$



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

$$\mathrm{Int}(3*S_{ij}) = \log_2(q_{ij}/p_i p_j)$$

$$q_{ij} = p_i p_j e^{S_{ij}}$$

$$\cancel{\sum_{i=1}^{n}\sum_{j=1}^{i}} q_{ij} = 1$$

$$\lambda S_{ij} = \log_e(q_{ij}/p_i p_j)$$

$$q_{ij} = p_i p_j e^{\lambda s_{ij}}$$

λ allows us to recover that original $\boldsymbol{q_{ij}}$ for purposes of further calculation

$$\sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij} = 1 \qquad q_{ij} = p_i p_j e^{\lambda s_{ij}}$$

λ is found by successive approximation using the Identity below

$$\sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{i} p_i p_j e^{\lambda s_{ij}} = 1$$

Further calculations you can do once you know lambda

Expected score
Relative entropy
Target frequencies
Convert a raw score to a nat/bit score

Expected score of the matrix

$$E = \sum_{i=1}^{20} \sum_{j=1}^{i} p_i p_j \lambda S_{ij}$$

Note must be negative for K-A stats to apply

What is the expected score of a +1/-3 scoring scheme?

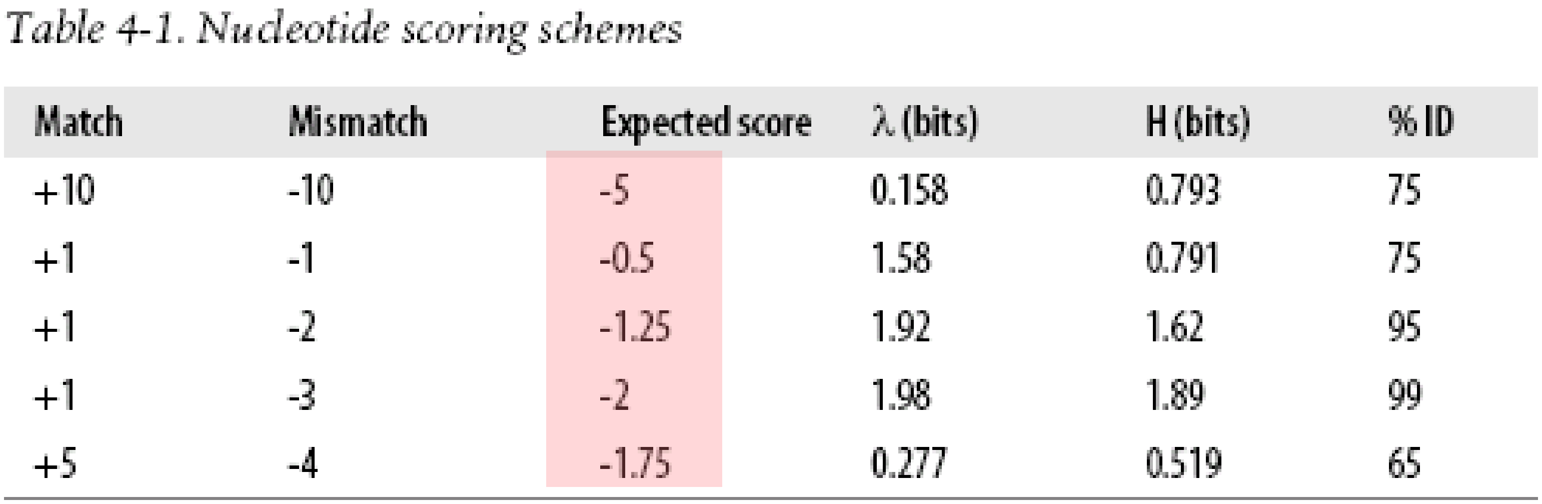


Table 4-1. Nucleotide scoring schemes

Match	Mismatch	Expected score	λ (bits)	H (bits)	% ID
+10	-10	-5	0.158	0.793	75
+1	-1	-0.5	1.58	0.791	75
+1	-2	-1.25	1.92	1.62	95
+1	-3	-2	1.98	1.89	99
+5	-4	-1.75	0.277	0.519	65

Relative Entropy of the matrix

$$H = -\sum_{i=1}^{20} \sum_{j=1}^{i} q_{ij} \lambda S_{ij}$$

BLOSUM 42 < BLOSUM 62 < BLOSUM 80

‘Think of Entropy in terms of degeneracy and promiscuity’

H ↑ = far from equilibrium

H ↓ = near equilibrium, alignments contain little information

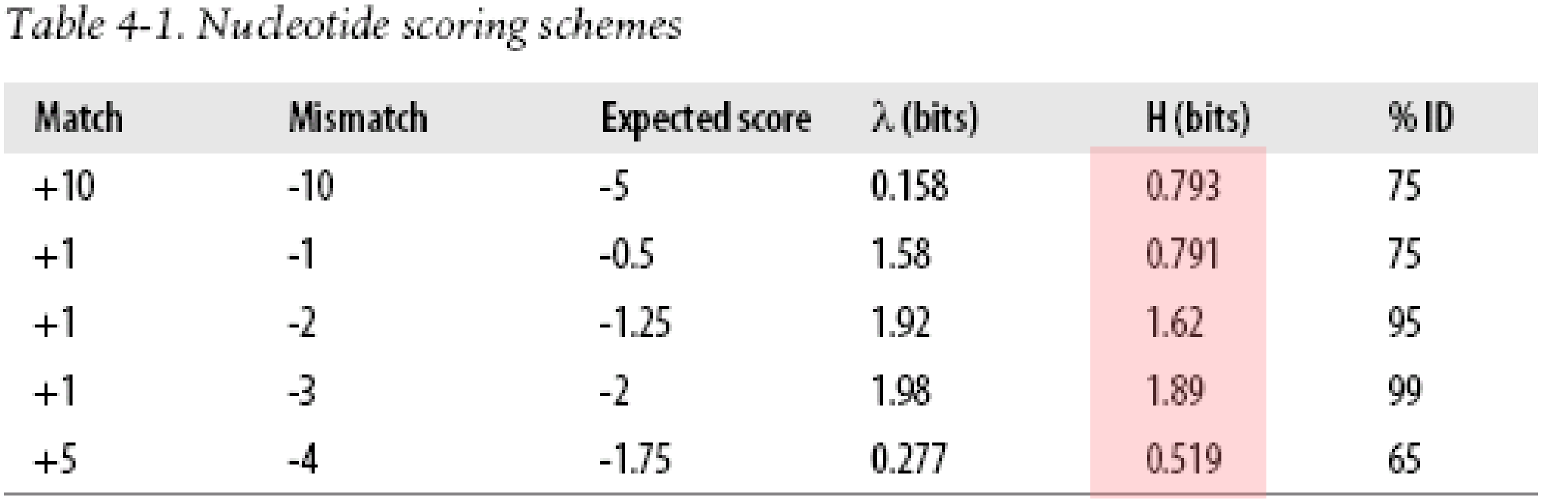


Table 4-1. Nucleotide scoring schemes

Match	Mismatch	Expected score	λ (bits)	H (bits)	% ID
+10	-10	-5	0.158	0.793	75
+1	-1	-0.5	1.58	0.791	75
+1	-2	-1.25	1.92	1.62	95
+1	-3	-2	1.98	1.89	99
+5	-4	-1.75	0.277	0.519	65

Target Frequencies

Every scoring scheme is implicitly an log-odds scoring scheme.
Every scoring scheme has a set of target frequencies

$$\lambda S_{ij} = \log_e(q_{ij}/p_i p_j)$$

$$q_{ij} = p_i p_j e^{\lambda s_{ij}}$$

In other words, even a simple +1/-3 scoring scheme is implictly a log odds scheme.

What data justify this scheme; what imaginary data
Does the scheme imply?

Further calculations you can do once you know lambda

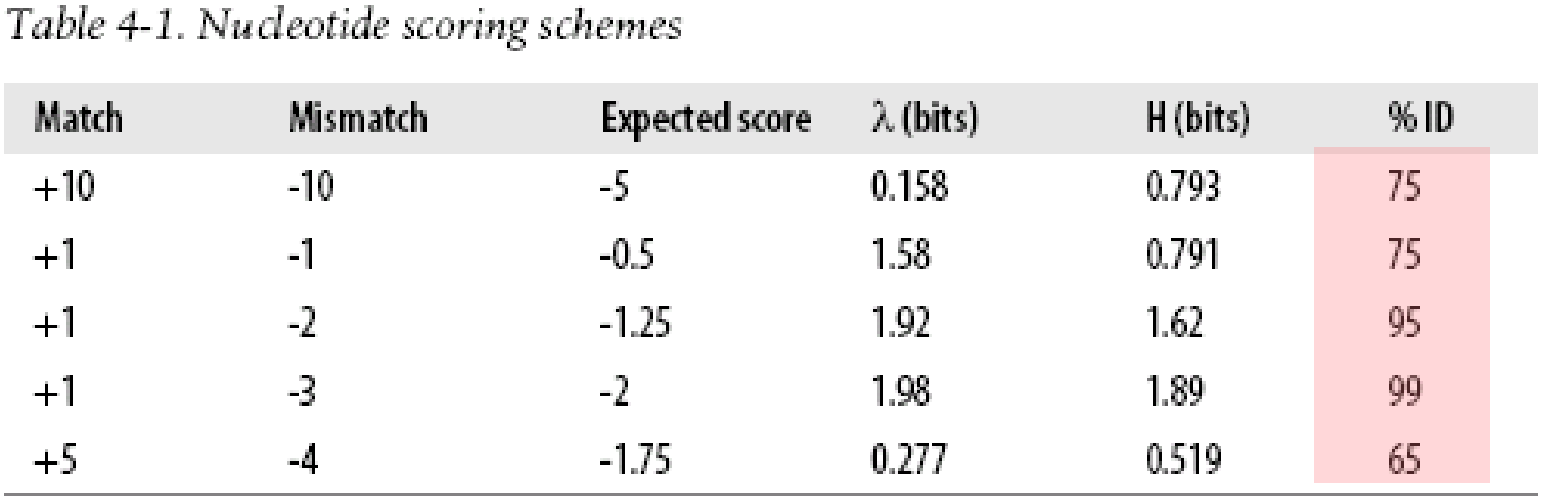
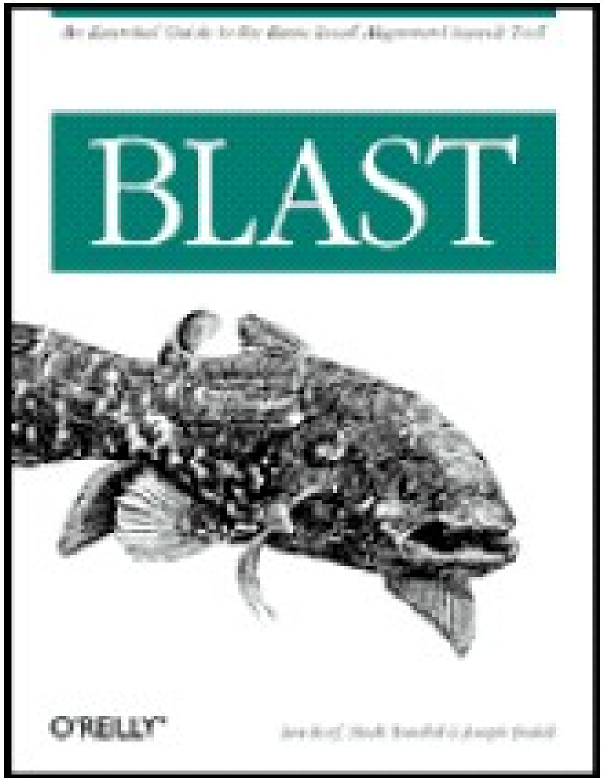


Table 4-1. Nucleotide scoring schemes

Match	Mismatch	Expected score	λ (bits)	H (bits)	% ID
+10	-10	-5	0.158	0.793	75
+1	-1	-0.5	1.58	0.791	75
+1	-2	-1.25	1.92	1.62	95
+1	-3	-2	1.98	1.89	99
+5	-4	-1.75	0.277	0.519	65

Every scoring scheme is implicitly a log odds scoring matrix;
Every log odds matrix has an implicit set of target frequencies.
This is quite profound insight.

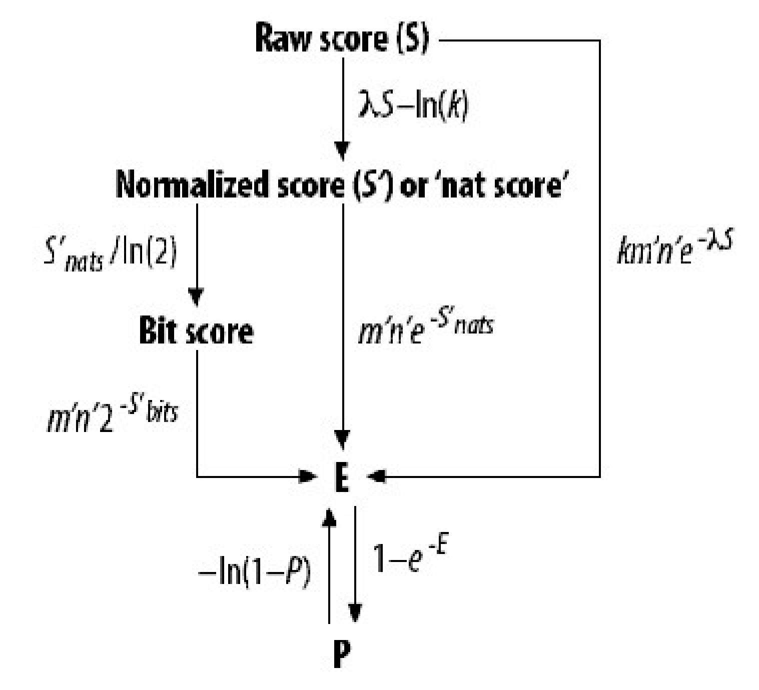
Commercial break!


An Essential Guide to the Basic Local Alignment Search Tool
BLAST
O'REILLY
Ian Korf, Mark Yandell & Joseph Bedell

BLAST STATISTCS

The basic operations:

Actual *vs.* Effective lengths,
Raw scores,
Normalized scores e.g. nat and bit scores
E & P

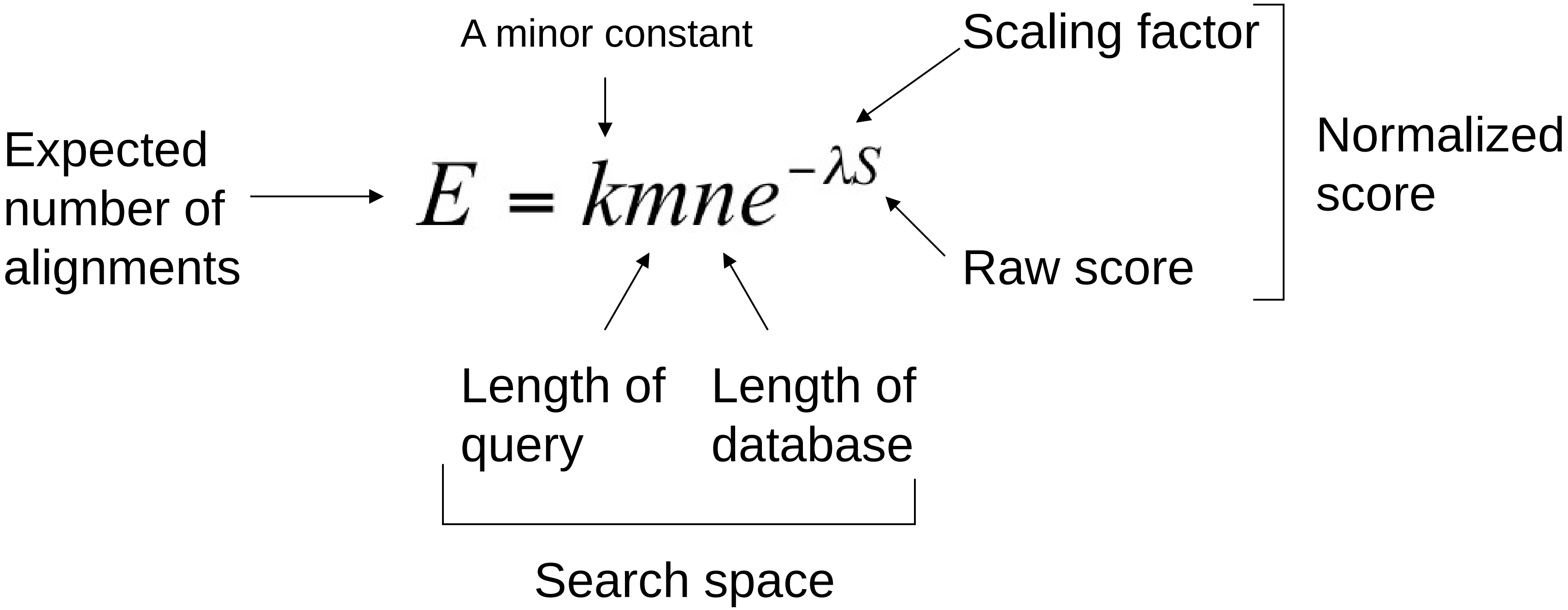


```
>gi|23098447|ref|NP_691913.1| (NC_004193) Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146
VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
           VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10
VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```

Table 7-1. The parameters and their values required for Karlin-Altschul statistical calculations

Parameter	Value
λ	0.267 nats (gapped)
k	0.0410 nats (gapped)
H	0.140 nats/aligned residue
m	321 (length of the query sequence)
n	9418064 (number of letters in the database searched)
Effective HSP length	99
Number of sequences in database	17878

The Karlin-Altschul Equation



A minor constant

Scaling factor

Expected number of alignments

$$E = kmne^{-\lambda S}$$

Normalized score

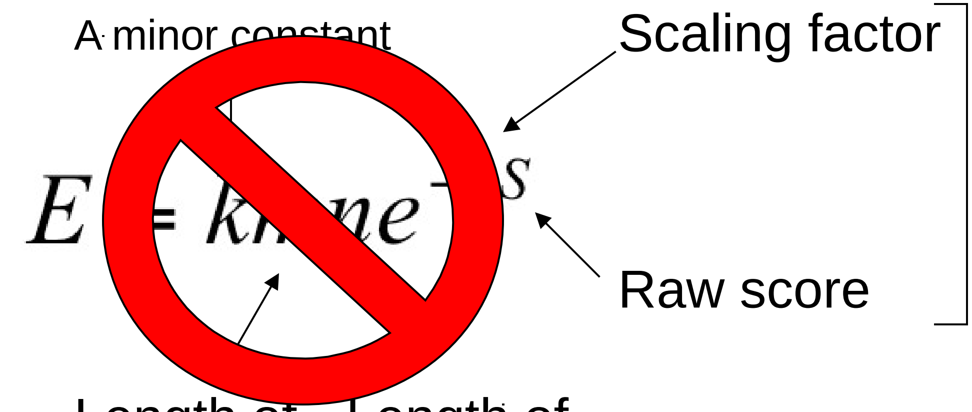
Raw score

Length of query

Length of database

Search space

The Karlin-Altschul Equation



A minor constant

Scaling factor

Normalized score

Expected number of alignments

Raw score

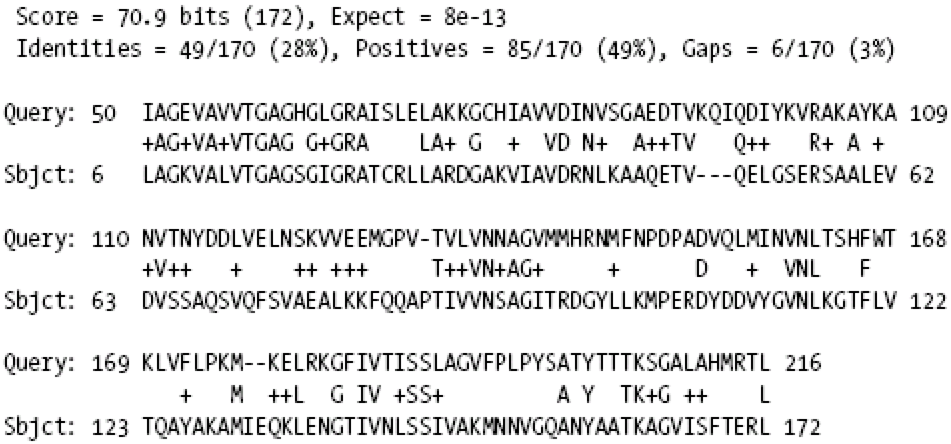
Length of query

Length of database

Search space

ACTUAL vs. EFFECTIVE LENGTHS

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



```
 Score = 70.9 bits (172), Expect = 8e-13
 Identities = 49/170 (28%), Positives = 85/170 (49%), Gaps = 6/170 (3%)

Query: 50  IAGEVAVVTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQIQDIYKVRAKAYKA 109
            +AG+VA+VTGAG G+GRA    LA+ G  +  VD N+  A++TV   Q++   R+ A +
Sbjct: 6   LAGKVALVTGAGSGIGRATCRLLARDGAKVIAVDRNLKAAQETV---QELGSERSAALEV 62

Query: 110 NVTNYDDLVELNSKVVEEMGPV-TVLVNNAGVMMHRNMFNPDPADVQLMINVNLTSHFWT 168
            +V++   +    ++ +++     T++VN+AG+     +      D   +  VNL   F
Sbjct: 63  DVSSAQSVQFSVAEALKKFQQAPTIVVNSAGITRDGYLLKMPERDYDDVYGVNLKGTFLV 122

Query: 169 KLVFLPKM--KELRKGFIVTISSLAGVFPLPYSATYTTTKSGALAHMRTL 216
               +   M  ++L  G IV +SS+         A Y  TK+G ++    L
Sbjct: 123 TQAYAKAMIEQKLENGTIVNLSSIVAKMNNVGQANYAATKAGVISFTERL 172
```

$$E = Km'n'e^{-\lambda S}$$

The ‘expected HSP length’

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

$$1 = Kmne^{-\lambda S}$$

$$\ln(1/Kmn) = \ln(e^{-\lambda S})$$

$$-\ln(Kmn) = -\lambda S$$

$$\ln(Kmn) = \lambda S$$

$$l = \ln(Kmn)/H$$

Dependent on search space

Recall that ***H*** is nats/aligned residue, thus $\lambda S = Hl$

$$\ln(Kmn)/H = l$$

$$l = \ln(Kmn) / H$$

ACGTGTGCGCAGTGTCGCGTGTGCACACTATAGCC

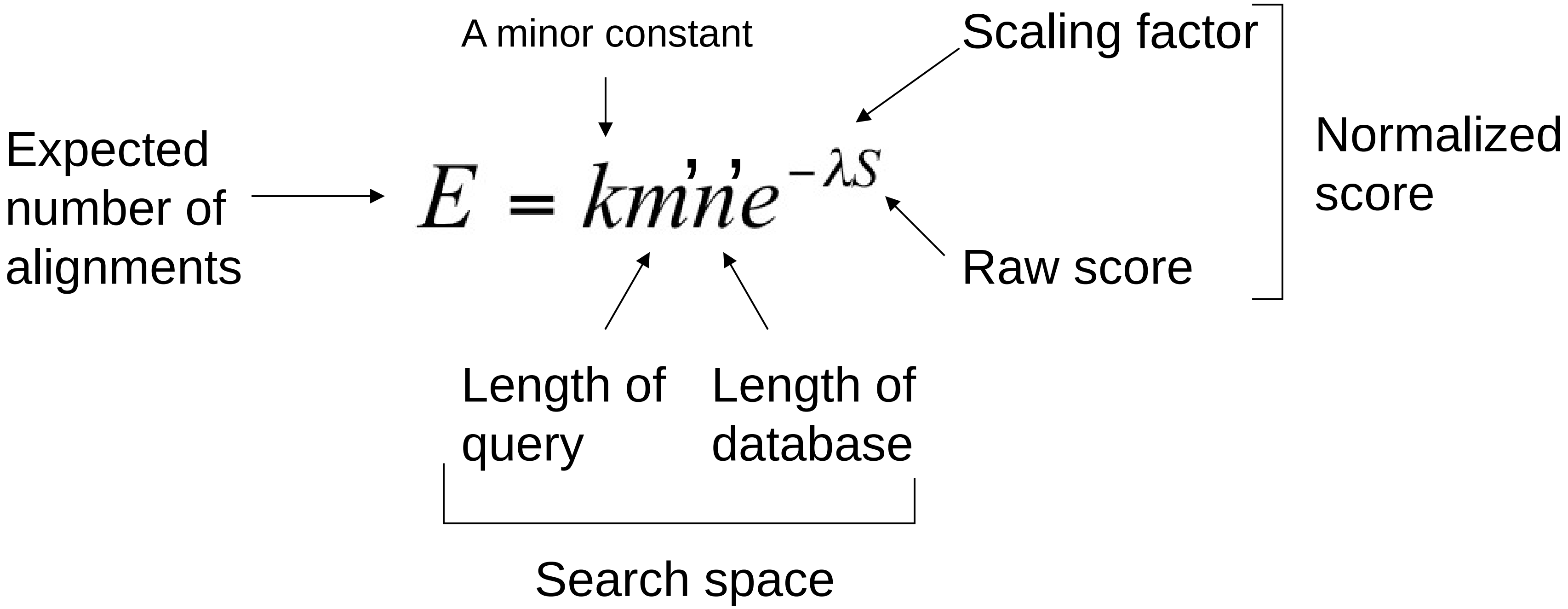
Actual length (m)

effective length(*m’*) = *m –l*

effectve length (*n’*) = total length db – num_seqs*l

What happens if m’ < 0 ?

The Karlin-Altschul Equation



$$E = km'n'e^{-\lambda S}$$

Expected number of alignments

A minor constant

Scaling factor

Raw score

Normalized score

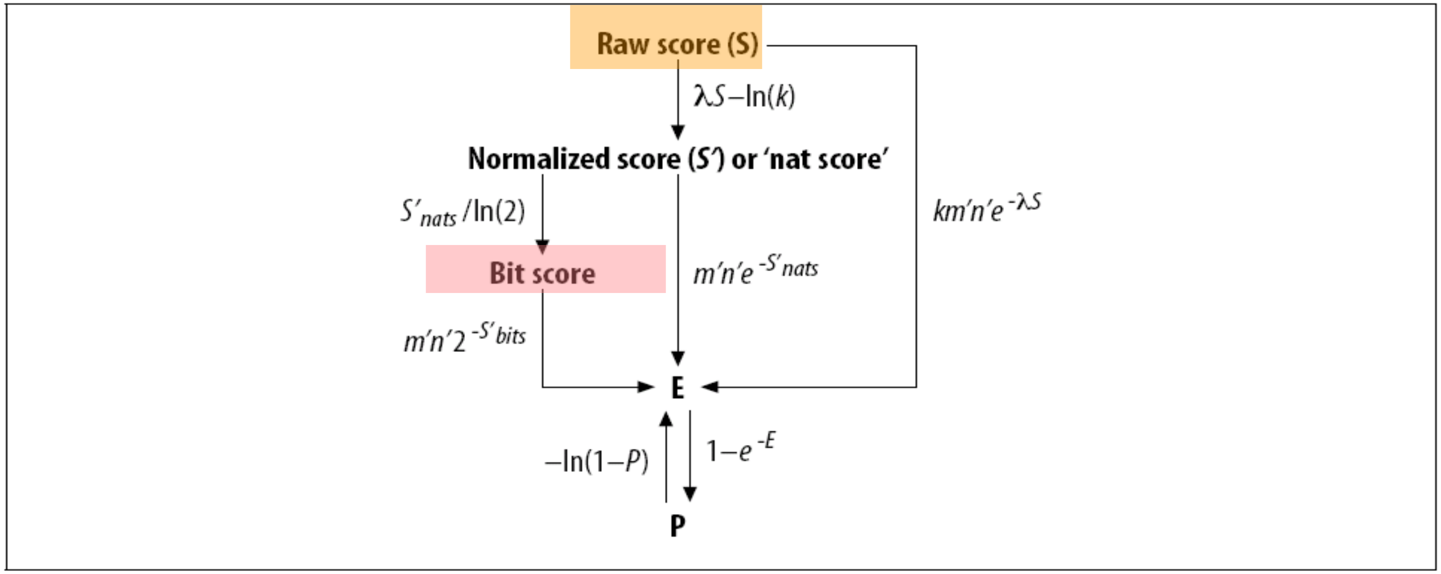
Length of query

Length of database

Search space

Converting a raw score to a bit score

```
>gi|23098447|ref|NP_691913.1| (NC_004193) Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146
VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
          VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10
VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```



Converting a raw score to a bit score

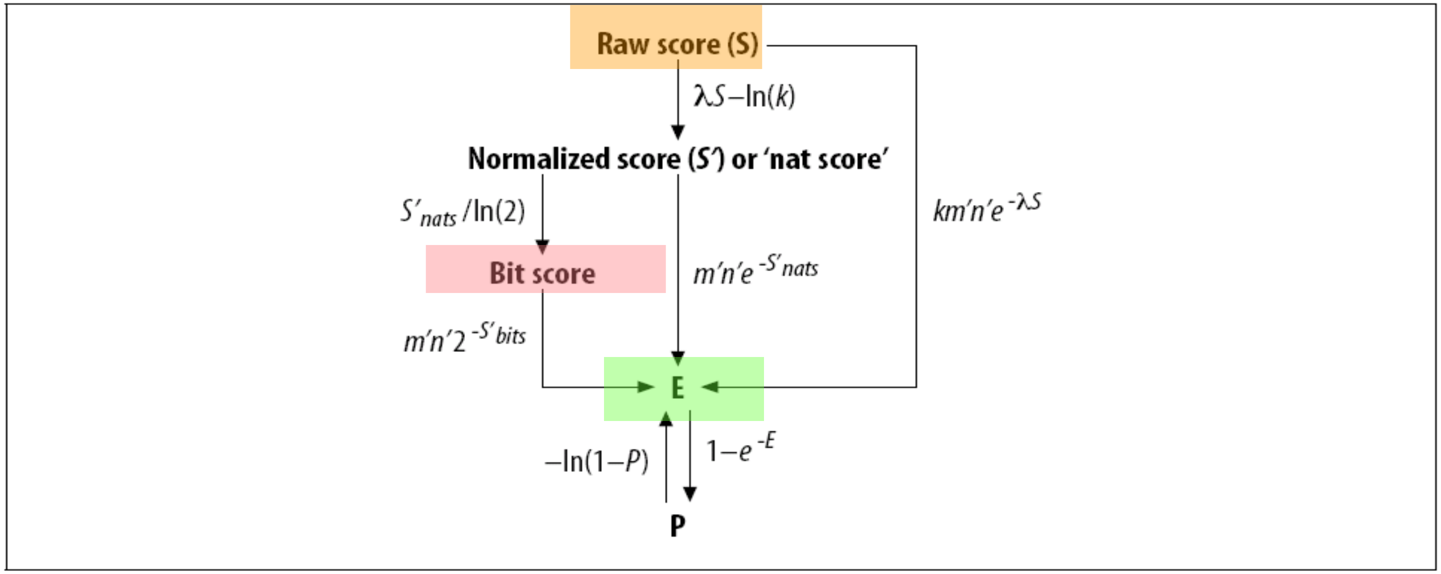
$$S'_{nats} = \lambda_{nats} S_{raw} - \ln K$$

$$S'_{bits} = \lambda_{bits} S_{raw} - \ln K$$

$$S'_{bits} = S'_{nats} / \ln(2)$$

Converting a raw score or a bit score to an Expect

```
>gi|23098447|ref|NP_691913.1| (NC_004193) Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146
VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
           VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10
VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```



Converting a raw score or a bit score to an Expect

$$E = Km'n'e^{-\lambda S}$$

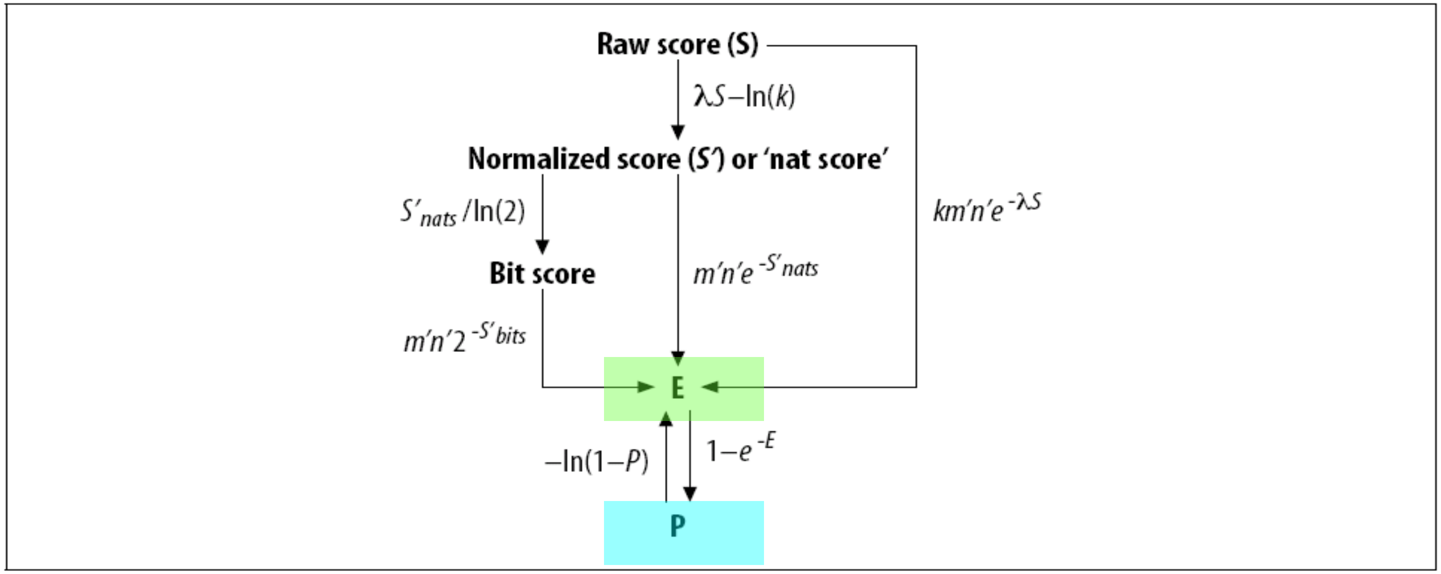
$$E = m'n'e^{-S_{nats}^{'}}$$

$$E = m'n'2^{-S_{bits}^{'}}$$

$$S_{nats}' = \lambda_{nats} S_{raw} - \ln K$$

Converting an Expect to a WU-BLAST P value

```
>gi|23098447|ref|NP_691913.1| (NC_004193) Length = 253

 Score = 38.9 bits (89), Expect = 3e-05
 Identities = 17/40 (42%), Positives = 26/40 (64%)
 Frame = -1

Query: 4146
VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
           VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10
VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```



Converting an Expect to a WU-BLAST P value

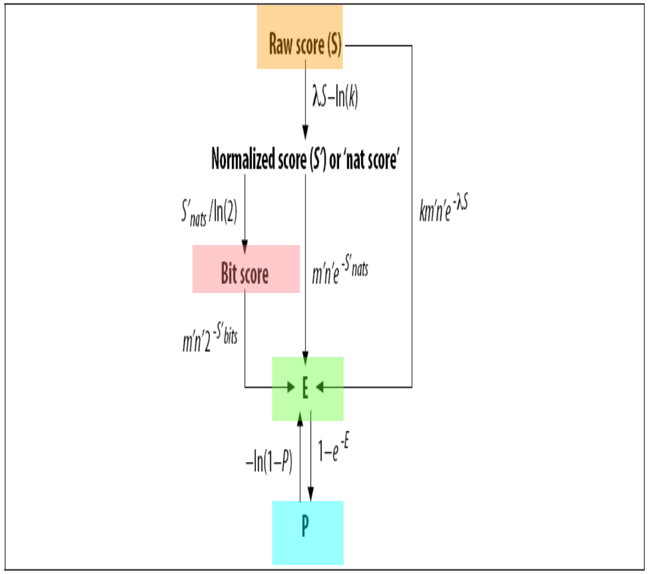
$$P = 1 - e^{-E}$$

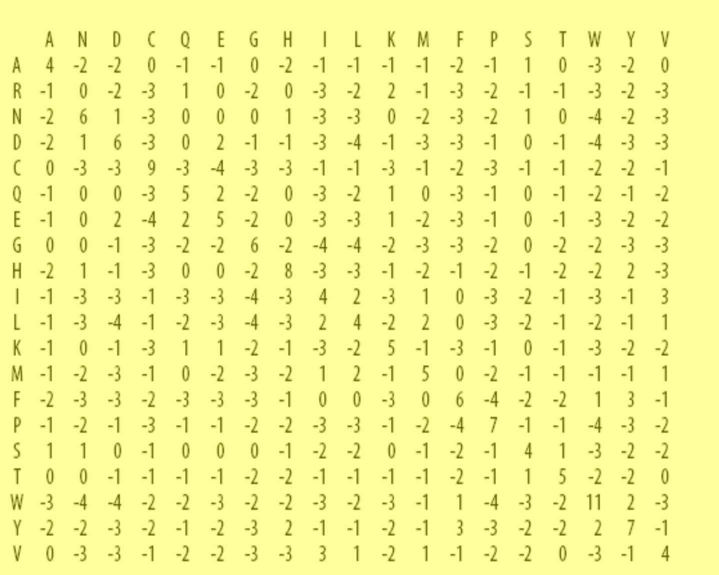
$$E = -\ln(1 - P)$$

Note that E ~= P if either value < 1e^{-5}

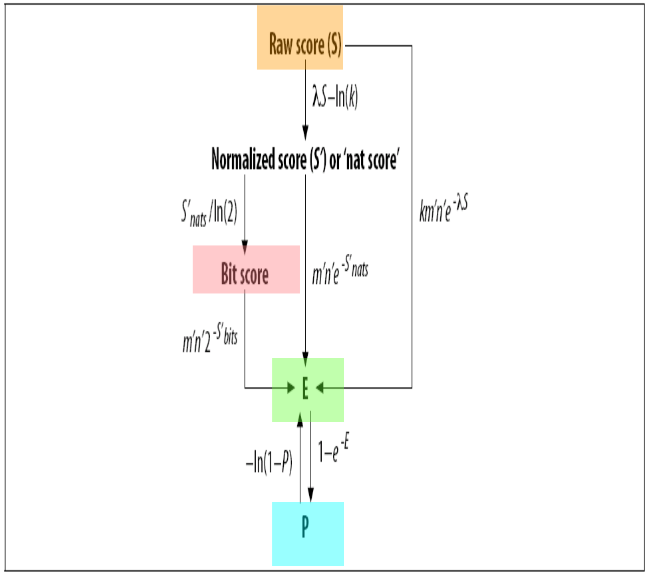
Review: where the parts of an HSP come from, and what they mean

```
>gi|23098447|ref|NP_691913.1| (NC_004193) Length = 253

Score = 38.9 bits (89), Expect = 3e-05
Identities = 17/40 (42%), Positives = 26/40 (64%)
Frame = -1

Query: 4146
VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
          VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10
VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```



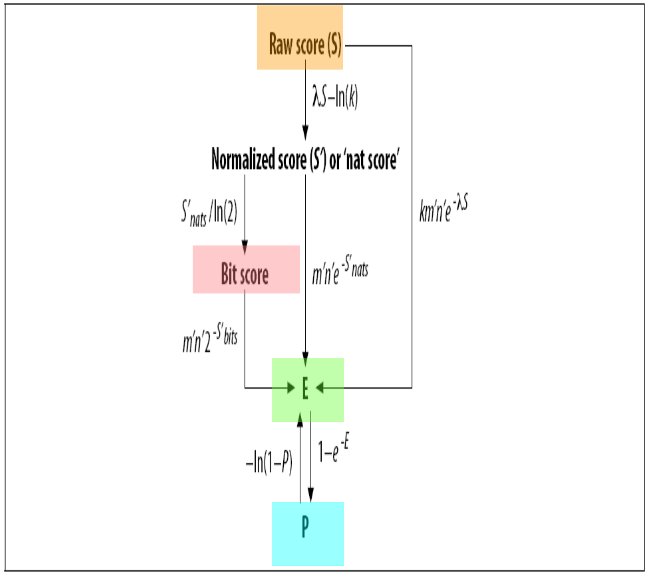


	A	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2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



Why use Karlin-Altschul statistics?
Why not just stop with the raw score?

Why use Karlin-Altschul statistics?
Why not just stop with the raw score?



Scores is fine, if you are only interested
In the top score… when to stop?

How to compare scores produced using two different
scoring schemes?
Bit score provide a common currency for scores,
i.e. 52 bits is 52 bits is 52 bits.

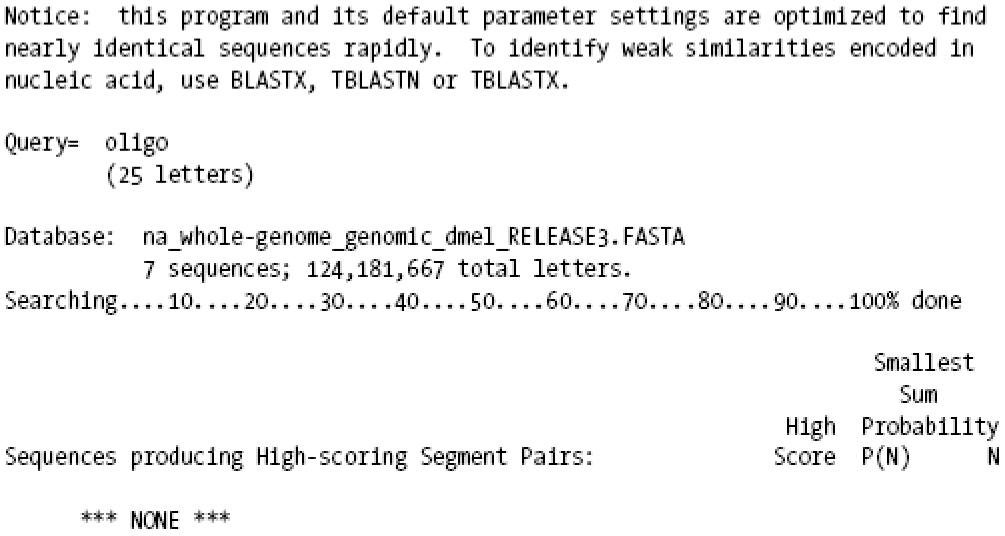
Scores don’t reflect database size; Expects do.

K-A stats is a bit like stoichiometry: Score ~ weight
λ ~ Avogadro's’ number
E ~ mass

Where Did My Oligo Go?

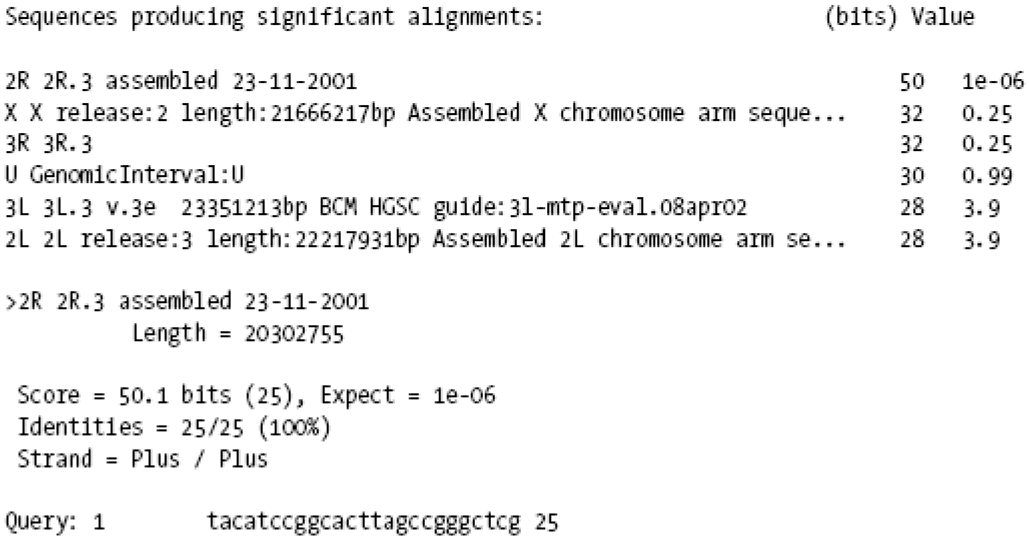
TACATCCGGCACTTAGCCGGGCTCG

WU-BLASTN



```
Notice:  this program and its default parameter settings are optimized to find
nearly identical sequences rapidly.  To identify weak similarities encoded in
nucleic acid, use BLASTX, TBLASTN or TBLASTX.

Query=  oligo
        (25 letters)

Database:  na_whole-genome_genomic_dmel_RELEASE3.FASTA
           7 sequences; 124,181,667 total letters.
Searching....10....20....30....40....50....60....70....80....90....100% done

                                                                     Smallest
                                                                       Sum
                                                              High  Probability
Sequences producing High-scoring Segment Pairs:              Score  P(N)      N

      *** NONE ***
```

NCBI-BLASTN



```
Sequences producing significant alignments:                      (bits) Value

2R 2R.3 assembled 23-11-2001                                          50   1e-06
X X release:2 length:21666217bp Assembled X chromosome arm seque...    32   0.25
3R 3R.3                                                               32   0.25
U GenomicInterval:U                                                   30   0.99
3L 3L.3 v.3e  23351213bp BCM HGSC guide:3l-mtp-eval.08apr02           28   3.9
2L 2L release:3 length:22217931bp Assembled 2L chromosome arm se...    28   3.9

>2R 2R.3 assembled 23-11-2001
          Length = 20302755

 Score = 50.1 bits (25), Expect = 1e-06
 Identities = 25/25 (100%)
 Strand = Plus / Plus

Query: 1        tacatccggcacttagccgggctcg 25
```

Table 7-3. Selected WU-BLASTN parameters and values from the search shown in Example 7-5

Parameter	Value
λ	0.104 nats (gapped)
k	0.0151 nats (gapped)
H	0.0600 nats/aligned residue
m	25 (length of the query sequence)
n	124,181,667(number of letters in the database)
Number of sequences in database	7

Table 7-4. Selected NCBI-BLASTN parameters and values from the search shown in Example 7-5

Parameter	Value
λ	1.37 nats (gapped)
k	0.711 nats (gapped)
H	1.31 nats/aligned residue
m	25 (length of the query sequence)
n	124,181,667 (number of letters in the database searched)
Number of sequences in database	7

Table 4-1. Nucleotide scoring schemes

Match	Mismatch	Expected score	λ (bits)	H (bits)	% ID
+10	-10	-5	0.158	0.793	75
+1	-1	-0.5	1.58	0.791	75
+1	-2	-1.25	1.92	1.62	95
+1	-3	-2	1.98	1.89	99
+5	-4	-1.75	0.277	0.519	65

$$S_{E=1} = \ln(Kmn) / \lambda$$

NCBI ~ 15
WU-BLAST ~170

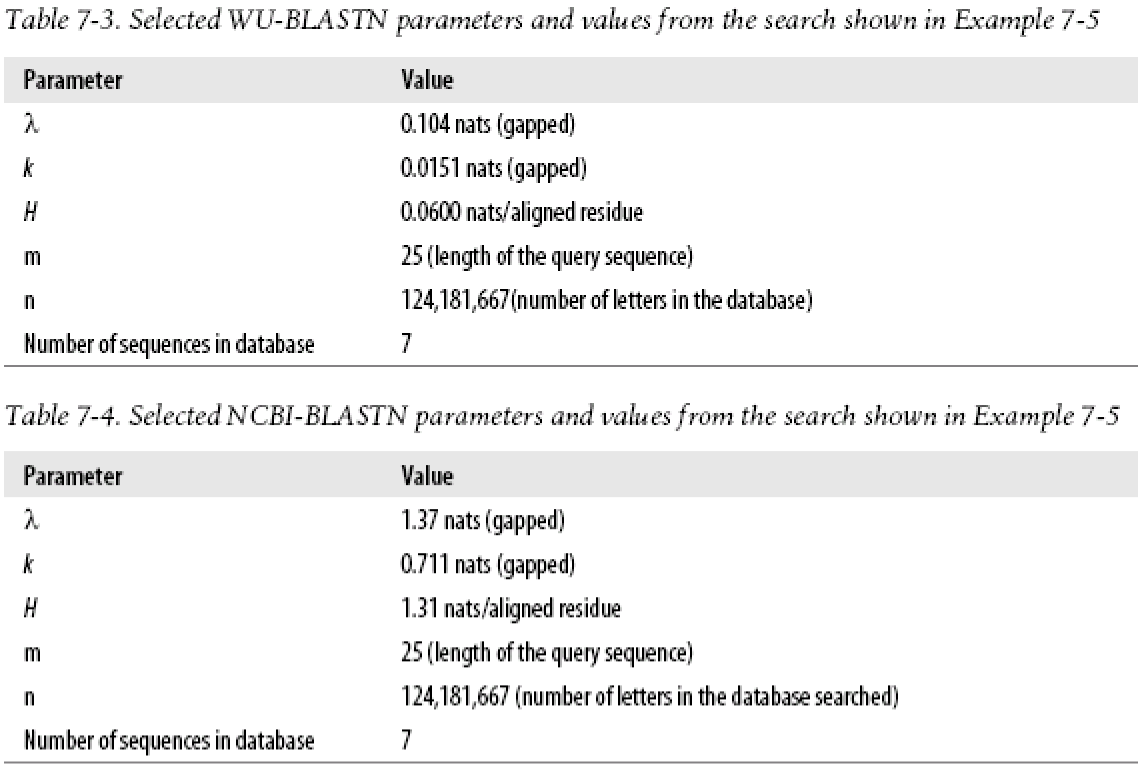


Table 7-3. Selected WU-BLASTN parameters and values from the search shown in Example 7-5

Parameter	Value
λ	0.104 nats (gapped)
k	0.0151 nats (gapped)
H	0.0600 nats/aligned residue
m	25 (length of the query sequence)
n	124,181,667(number of letters in the database)
Number of sequences in database	7

Table 7-4. Selected NCBI-BLASTN parameters and values from the search shown in Example 7-5

Parameter	Value
λ	1.37 nats (gapped)
k	0.711 nats (gapped)
H	1.31 nats/aligned residue
m	25 (length of the query sequence)
n	124,181,667 (number of letters in the database searched)
Number of sequences in database	7

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

$$1 = Kmne^{-\lambda S}$$

$$\ln(Kmn) = \lambda S$$

$$\ln(Kmn) / \lambda = S_{raw}$$

So how long would an oligo have to be to generate a score of 15 or 170?

$$l = \ln(Kmn) / H$$

$l_{ncbi}=16$

$l_{\text{wu-BLAST}=294}$

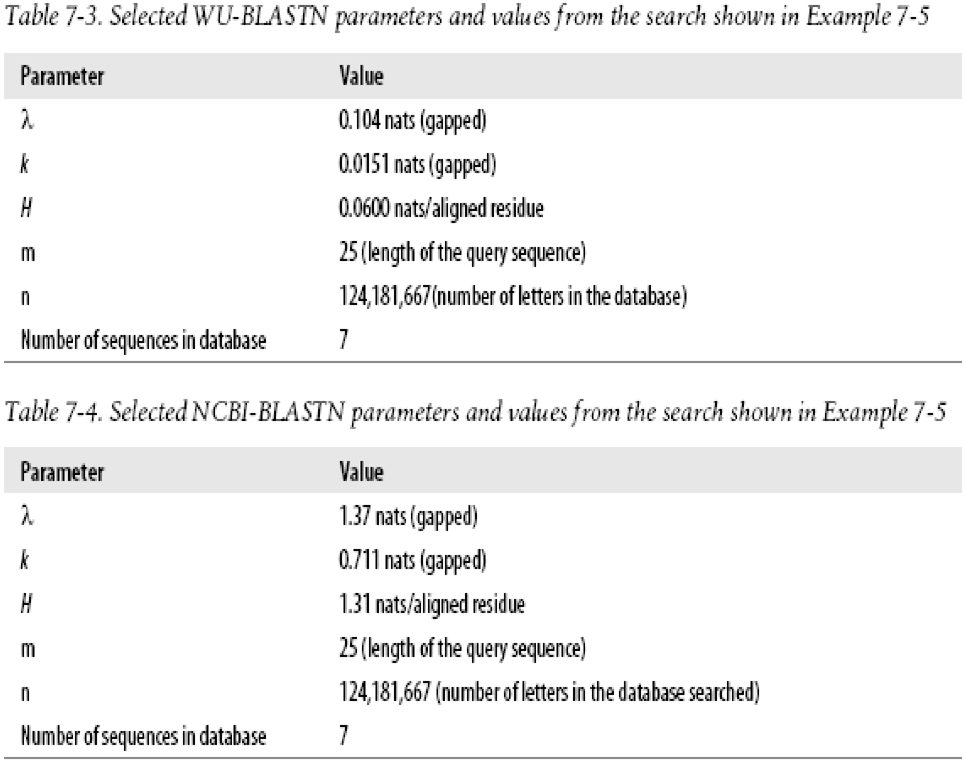
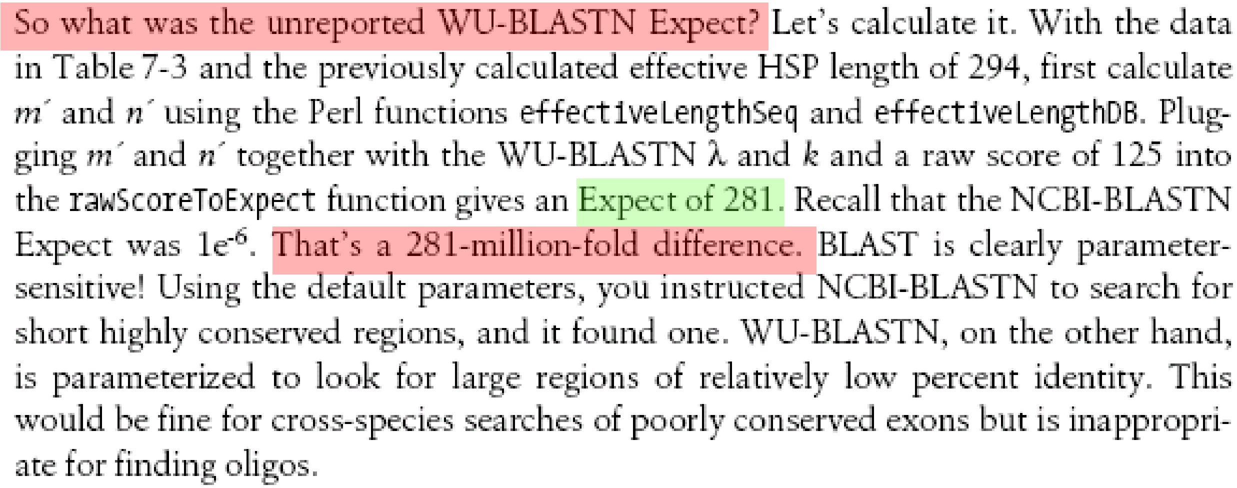


Table 7-3. Selected WU-BLASTN parameters and values from the search shown in Example 7-5

Parameter	Value
λ	0.104 nats (gapped)
k	0.0151 nats (gapped)
H	0.0600 nats/aligned residue
m	25 (length of the query sequence)
n	124,181,667(number of letters in the database)
Number of sequences in database	7

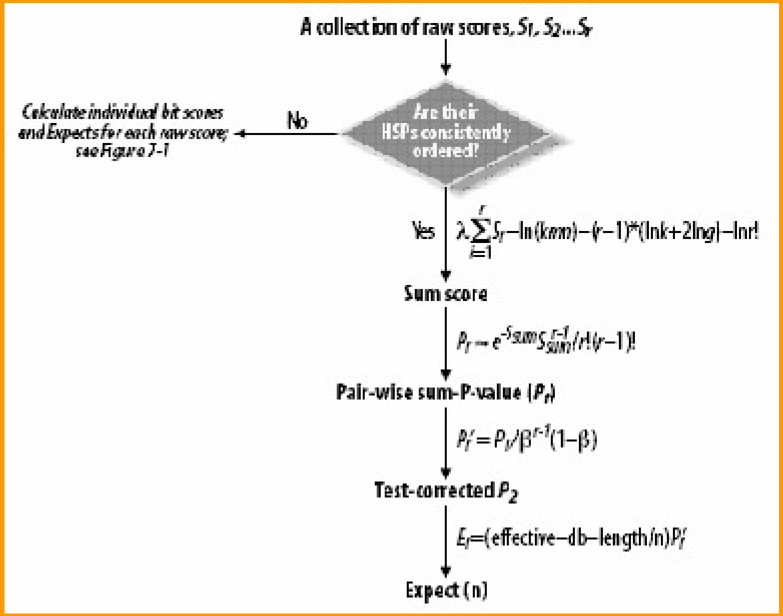
Table 7-4. Selected NCBI-BLASTN parameters and values from the search shown in Example 7-5

Parameter	Value
λ	1.37 nats (gapped)
k	0.711 nats (gapped)
H	1.31 nats/aligned residue
m	25 (length of the query sequence)
n	124,181,667 (number of letters in the database searched)
Number of sequences in database	7



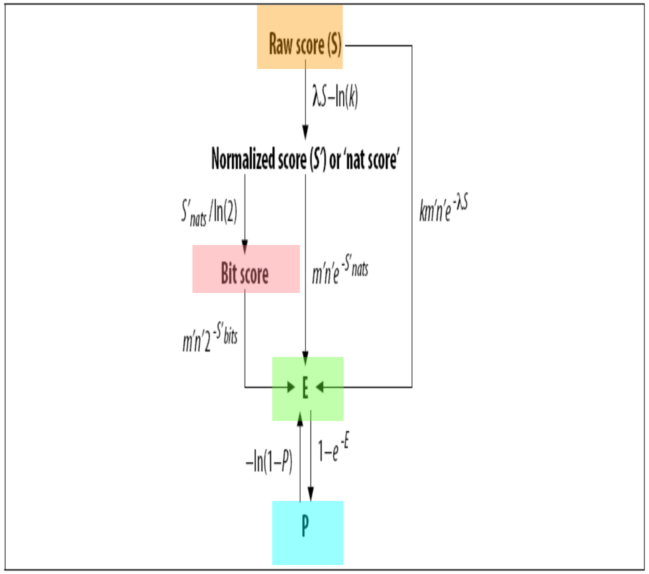
So what was the unreported WU-BLASTN Expect? Let's calculate it. With the data in Table 7-3 and the previously calculated effective HSP length of 294, first calculate m' and n' using the Perl functions `effectiveLengthSeq` and `effectiveLengthDB`. Plugging m' and n' together with the WU-BLASTN λ and k and a raw score of 125 into the `rawScoreToExpect` function gives an Expect of 281. Recall that the NCBI-BLASTN Expect was 1e^{-6}. That's a 281-million-fold difference. BLAST is clearly parameter-sensitive! Using the default parameters, you instructed NCBI-BLASTN to search for short highly conserved regions, and it found one. WU-BLASTN, on the other hand, is parameterized to look for large regions of relatively low percent identity. This would be fine for cross-species searches of poorly conserved exons but is inappropriate for finding oligos.

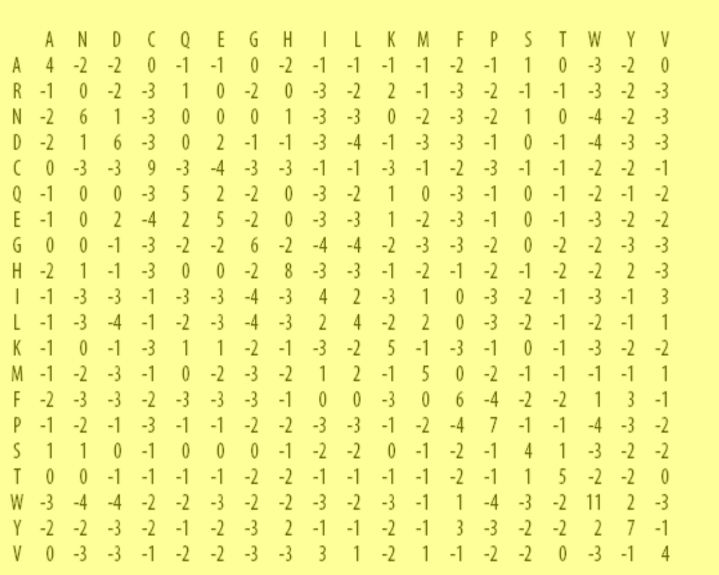
Sum Statistics


A collection of raw scores, $S_1, S_2 \ldots S_r$
Are their HSPs consistently ordered?
No
Calculate individual bit scores and Expects for each raw score; see Figure 7-1
Yes
$\lambda \sum_{i=1}^{r} S_r - \ln(kmn) - (r-1)*(\ln k + 2\ln g) - \ln r!$
Sum score
$P_r \sim e^{-S_{sum}} S_{sum}^{r-1} / r!(r-1)!$
Pair-wise sum-P-value (P_r)
$P_r' = P_r / \beta^{r-1}(1-\beta)$
Test-corrected P_2
$E_r = (\text{effective}-\text{db}-\text{length}/n) P_r'$
Expect (n)

Review: where the parts of an HSP come from, and what they mean

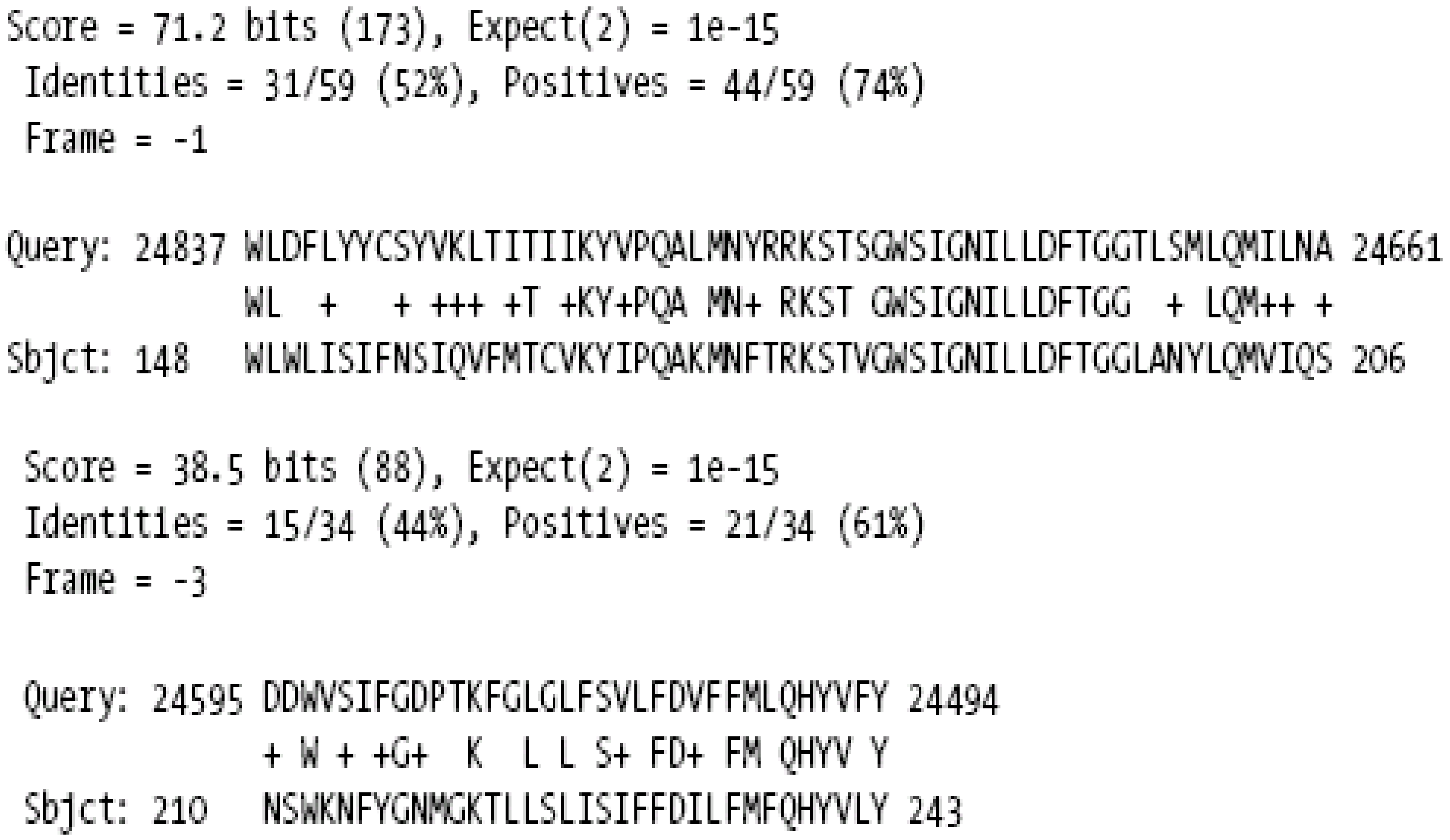
```
>gi|23098447|ref|NP_691913.1| (NC_004193) Length = 253

Score = 38.9 bits (89), Expect = 3e-05
Identities = 17/40 (42%), Positives = 26/40 (64%)
Frame = -1

Query: 4146 VTGAGHGLGRAISLELAKKGCHIAVVDINVSGAEDTVKQI 4027
            VTGA  G+G+AI+   A +G  + V D+N  GA+  V++I
Sbjct: 10 VTGAASGMGKAIATLYASEGAKVIVADLNEEGAQSVVEEI 49
```





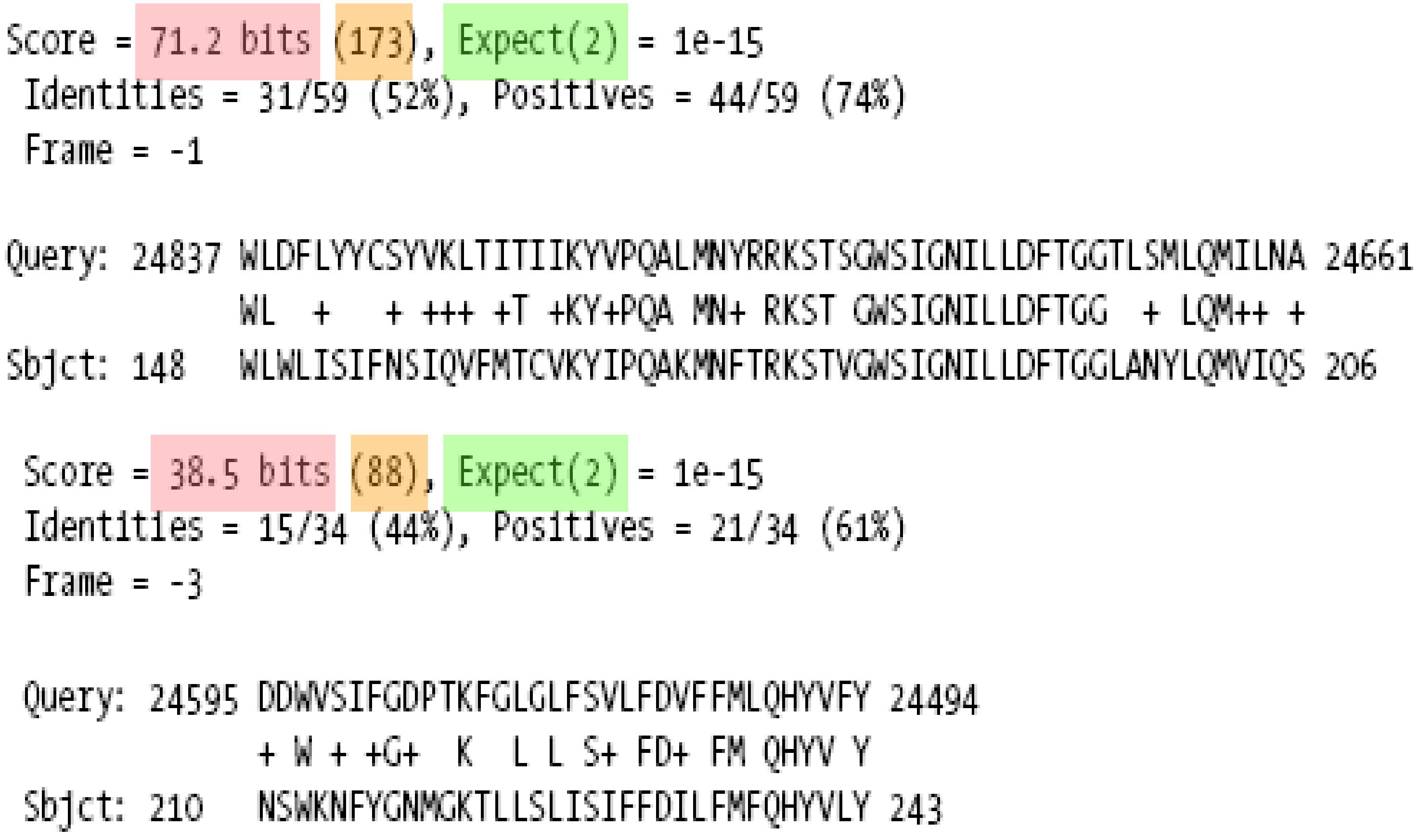
	A	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2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

What’s different about this BLAST Hit ?



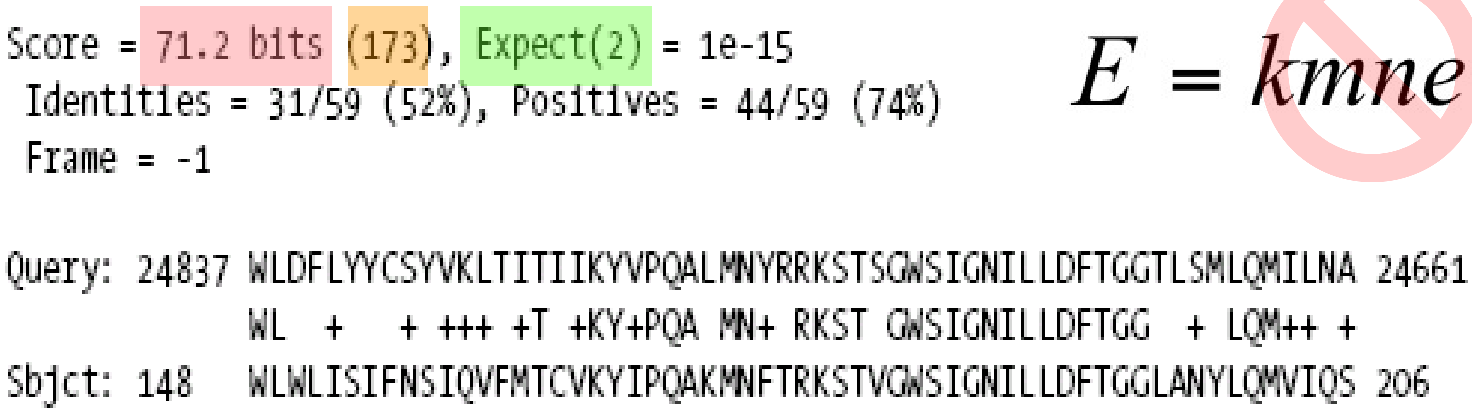
```
Score = 71.2 bits (173), Expect(2) = 1e-15
 Identities = 31/59 (52%), Positives = 44/59 (74%)
 Frame = -1

Query: 24837 WLDFLYYCSYVKLTITIIKYVPQALMNYRRKSTSGWSIGNILLDFTGGTLSMLQMILNA 24661
             WL  +   + +++ +T +KY+PQA MN+ RKST GWSIGNILLDFTGG  + LQM++ +
Sbjct: 148   WLWLISIFNSIQVFMTCVKYIPQAKMNFTRKSTVGWSIGNILLDFTGGLANYLQMVIQS 206

 Score = 38.5 bits (88), Expect(2) = 1e-15
 Identities = 15/34 (44%), Positives = 21/34 (61%)
 Frame = -3


 Query: 24595 DDWVSIFGDPTKFGLGLFSVLFDVFFMLQHYVFY 24494
              + W + +G+  K  L L S+ FD+ FM QHYV Y
 Sbjct: 210   NSWKNFYGNMGKTLLSLISIFFDILFMFQHYVLY 243
```

What’s different about this BLAST Hit ?



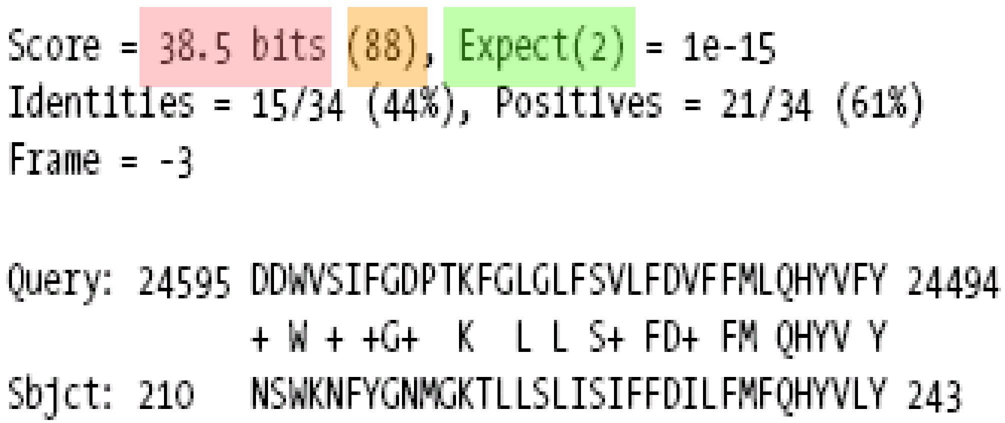
```
Score = 71.2 bits (173), Expect(2) = 1e-15
 Identities = 31/59 (52%), Positives = 44/59 (74%)
 Frame = -1

Query: 24837 WLDFLYYCSYVKLTITIIKYVPQALMNYRRKSTSGWSIGNILLDFTGGTLSMLQMILNA 24661
             WL  +   + +++ +T +KY+PQA MN+ RKST GWSIGNILLDFTGG  + LQM++ +
Sbjct: 148   WLWLISIFNSIQVFMTCVKYIPQAKMNFTRKSTVGWSIGNILLDFTGGLANYLQMVIQS 206

 Score = 38.5 bits (88), Expect(2) = 1e-15
 Identities = 15/34 (44%), Positives = 21/34 (61%)
 Frame = -3


 Query: 24595 DDWVSIFGDPTKFGLGLFSVLFDVFFMLQHYVFY 24494
              + W + +G+  K  L L S+ FD+ FM QHYV Y
 Sbjct: 210   NSWKNFYGNMGKTLLSLISIFFDILFMFQHYVLY 243
```

What’s different about this BLAST Hit ?



```
Score = 71.2 bits (173), Expect(2) = 1e-15
 Identities = 31/59 (52%), Positives = 44/59 (74%)
 Frame = -1

Query: 24837 WLDFLYYCSYVKLTITIIKYVPQALMNYRRKSTSGWSIGNILLDFTGGTLSMLQMILNA 24661
             WL  +   + +++ +T +KY+PQA MN+ RKST GWSIGNILLDFTGG  + LQM++ +
Sbjct: 148   WLWLISIFNSIQVFMTCVKYIPQAKMNFTRKSTVGWSIGNILLDFTGGLANYLQMVIQS 206
```

$$E = kmne^{-\lambda S}$$

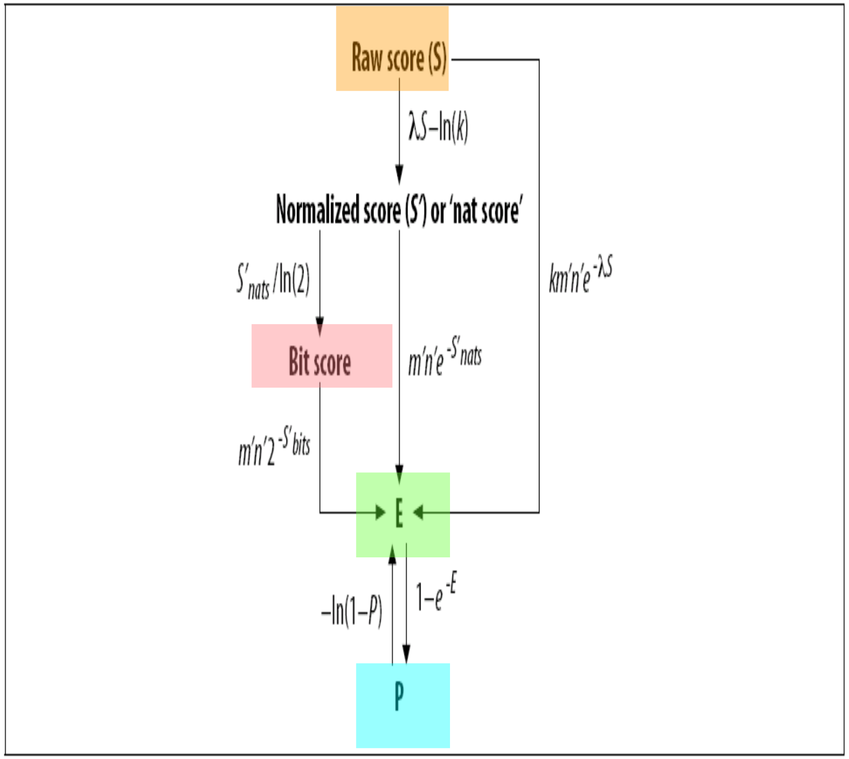


```
 Score = 38.5 bits (88), Expect(2) = 1e-15
 Identities = 15/34 (44%), Positives = 21/34 (61%)
 Frame = -3

 Query: 24595 DDWVSIFGDPTKFGLGLFSVLFDVFFMLQHYVFY 24494
              + W + +G+  K  L L S+ FD+ FM QHYV Y
 Sbjct: 210   NSWKNFYGNMGKTLLSLISIFFDILFMFQHYVLY 243
```

$$P_r = e^{-S_{sum}} S_{sum}^{\,r-1} / r!(r-1)!$$

Sum Statistics

BLAST uses two distinct methods to calculate an Expect



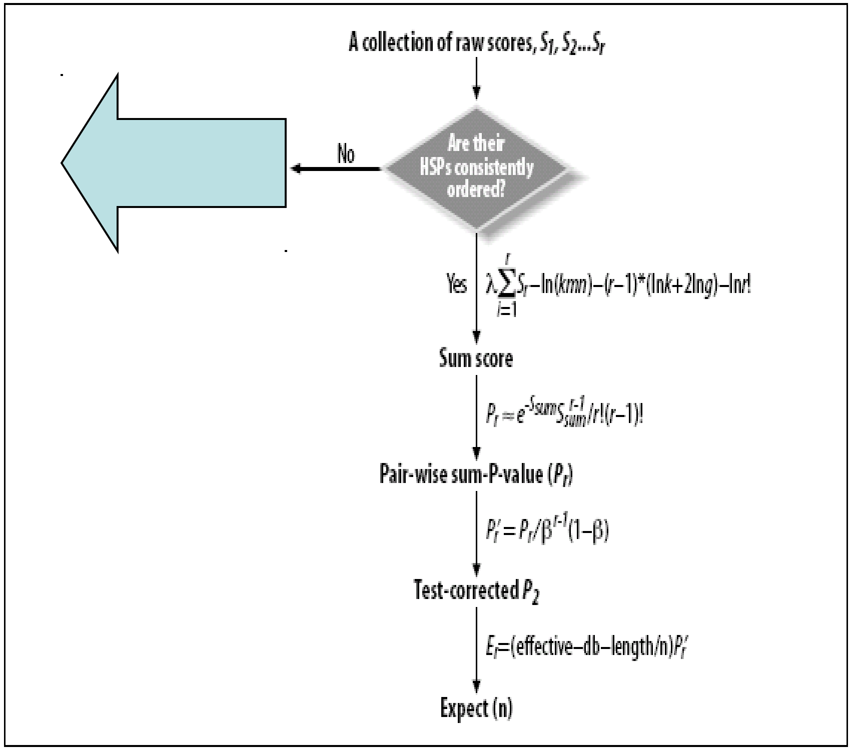
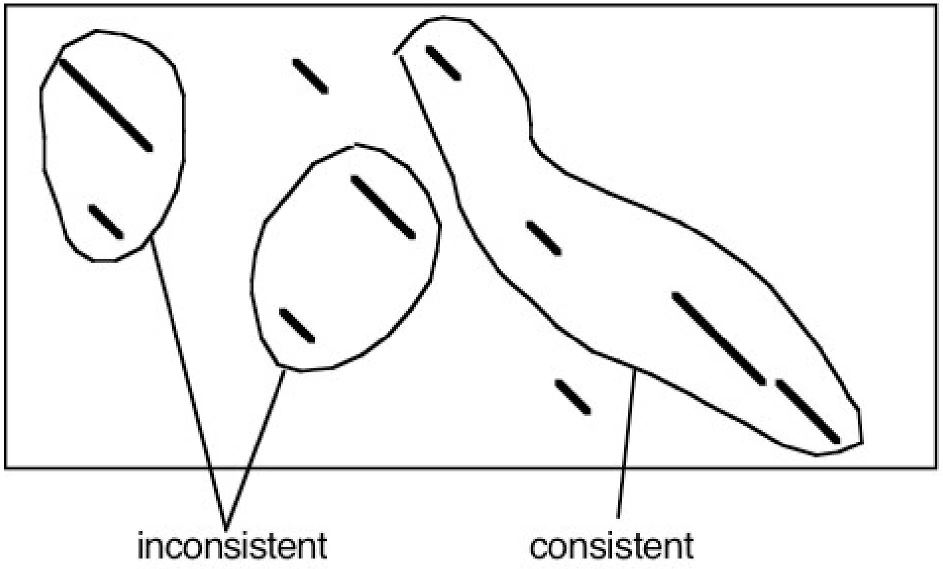

A collection of raw scores, S1, S2...Sr
Are their HSPs consistently ordered?
No
Yes
λ∑_{i=1}^{r} Sr − ln(kmn) − (r−1)*(lnk+2lng) − lnr!
Sum score
Pr = e^-Ssum Ssum^(r-1)/r!(r−1)!
Pair-wise sum-P-value (Pr)
P′r = Pr/β^(r-1)(1−β)
Test-corrected P2
Er=(effective−db−length/n)P′r
Expect (n)

Figure 7-2. The essential calculations involved in deriving the aggregate Expect for a group of HSPs

Sum Statistics

Sum statistics increases the significance (decreases the E-value) for groups of consistent alignments.



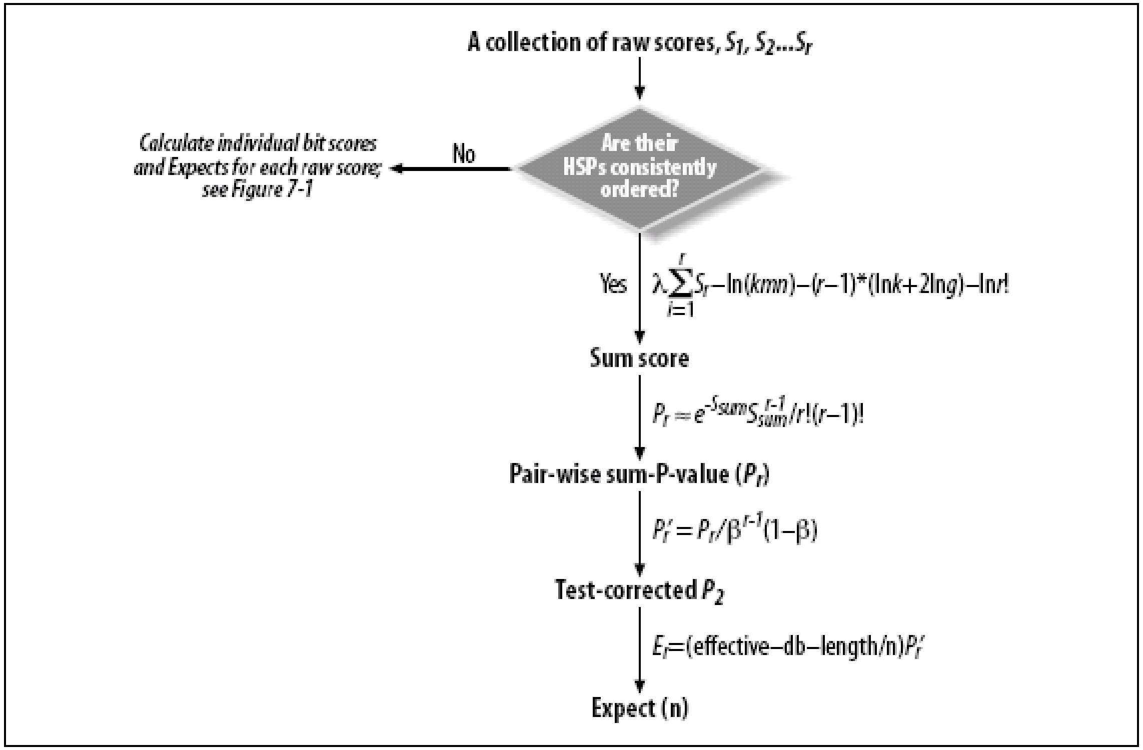

A collection of raw scores, $S_1, S_2 \ldots S_r$
Are their HSPs consistently ordered?
No
Calculate individual bit scores and Expects for each raw score; see Figure 7-1
Yes
$\lambda \sum_{i=1}^{r} S_r - \ln(kmn) - (r-1)*(\ln k + 2\ln g) - \ln r!$
Sum score
$P_r = e^{-S_{sum}} S_{sum}^{r-1} / r!(r-1)!$
Pair-wise sum-P-value (P_r)
$P_r' = P_r / \beta^{r-1}(1-\beta)$
Test-corrected P_2
E_r=(effective–db–length/n)P_r'
Expect (n)

Figure 7-2. The essential calculations involved in deriving the aggregate Expect for a group of HSPs

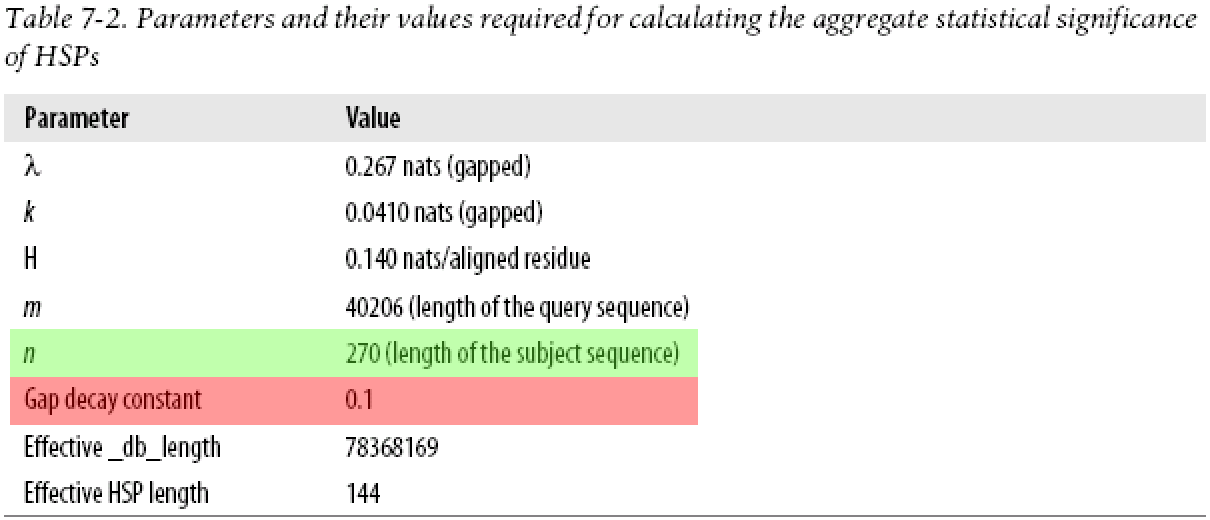


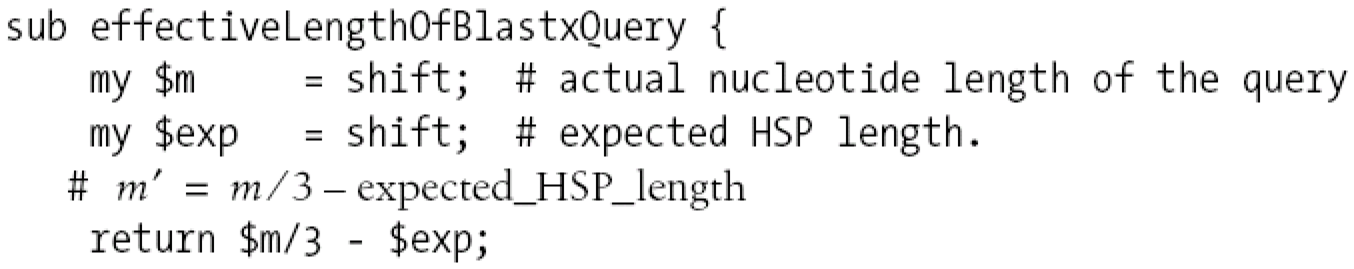
Table 7-2. Parameters and their values required for calculating the aggregate statistical significance of HSPs

Parameter	Value
λ	0.267 nats (gapped)
k	0.0410 nats (gapped)
H	0.140 nats/aligned residue
m	40206 (length of the query sequence)
n	270 (length of the subject sequence)
Gap decay constant	0.1
Effective _db_length	78368169
Effective HSP length	144

Sum Stats are ‘pair-wise’ in their focus

In other words, for the purposes of sum stat calculations
n = the length of the sbjct sequence; not the length on the db!

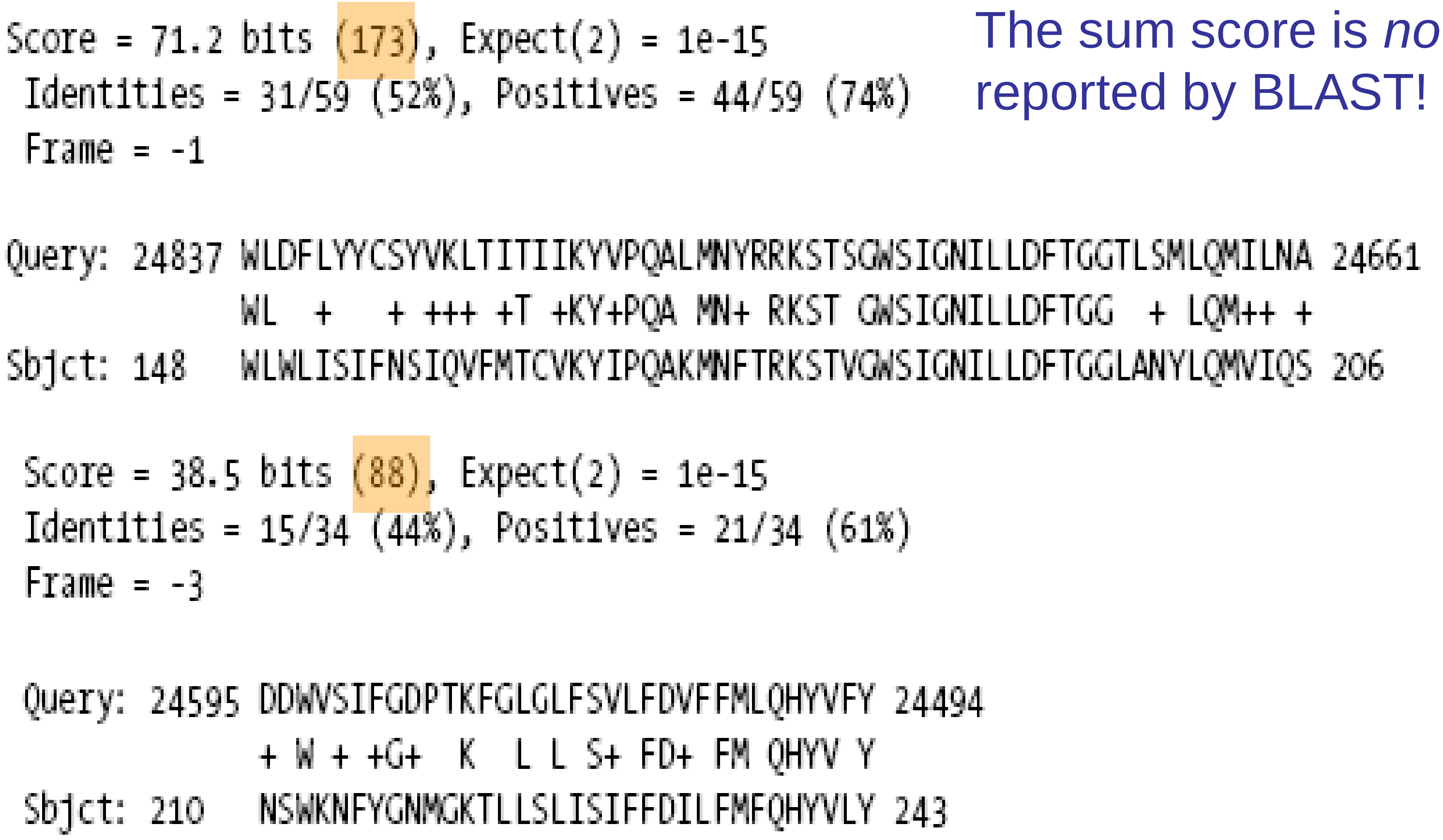
Actual Vs. effective lengths for BLASTX etc



```
sub effectiveLengthOfBlastxQuery {
    my $m      = shift;  # actual nucleotide length of the query
    my $exp    = shift;  # expected HSP length.
   #  m′  = m/3 – expected_HSP_length
    return $m/3 - $exp;
```

Sum Statistics are based on a ‘sum score’; rather than the raw score of the alignments

The sum score is *not* reported by BLAST!



```
Score = 71.2 bits (173), Expect(2) = 1e-15
 Identities = 31/59 (52%), Positives = 44/59 (74%)
 Frame = -1

Query: 24837 WLDFLYYCSYVKLTITIIKYVPQALMNYRRKSTSGWSIGNILLDFTGGTLSMLQMILNA 24661
             WL  +   + +++ +T +KY+PQA MN+ RKST GWSIGNILLDFTGG  + LQM++ +
Sbjct: 148   WLWLISIFNSIQVFMTCVKYIPQAKMNFTRKSTVGWSIGNILLDFTGGLANYLQMVIQS 206

 Score = 38.5 bits (88), Expect(2) = 1e-15
 Identities = 15/34 (44%), Positives = 21/34 (61%)
 Frame = -3

 Query: 24595 DDWVSIFGDPTKFGLGLFSVLFDVFFMLQHYVFY 24494
              + W + +G+  K  L L S+ FD+ FM QHYV Y
 Sbjct: 210   NSWKNFYGNMGKTLLSLISIFFDILFMFQHYVLY 243
```

Calculating a Sum score

$$S'_{nats} = \lambda S - \ln k$$

Equation 4-14.

$$S'_{sum} = \lambda \sum_{i=1}^{r} S_r$$

Equation 4-15.

$$S'_{sum} = \lambda \sum_{i=1}^{r} S_r - r \ln(kmn)$$

Equation 4-16.

$$S'_{sum} = \lambda \sum_{i=1}^{r} S_r - r \ln(kmn) + \ln(r!)$$

Equation 4-17.

$$S'_{sum} = \lambda \sum_{i=1}^{r} S_r - \ln(kmn) - (r-1) \cdot (\ln(k) + 2\ln(g)) - \log(r!)$$

Equation 4-18.

Converting a Sum score to an Expect(n)

$$P_r \approx e^{-S_{sum}} S_{sum}^{r-1} / r!(r-1)!$$

Equation 4-19.

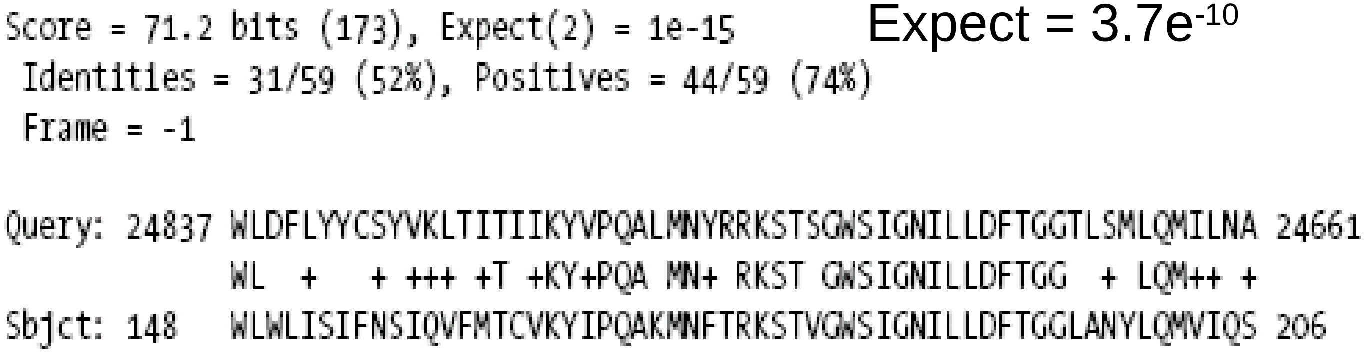
$$P'_r = P_r / \beta^{r-1}(1-\beta)$$

Equation 4-20.

$$\mathrm{Expect}(r) = (\mathrm{effective\_db\_length}/n) P'_r$$

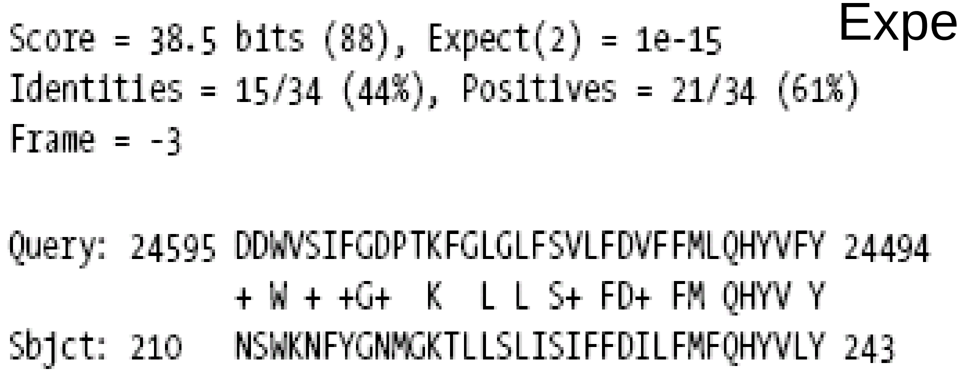
Equation 4-21.

Sum Statistics take home: buyer beware



```
Score = 71.2 bits (173), Expect(2) = 1e-15
 Identities = 31/59 (52%), Positives = 44/59 (74%)
 Frame = -1

Query: 24837 WLDFLYYCSYVKLTITIIKYVPQALMNYRRKSTSGWSIGNILLDFTGGTLSMLQMILNA 24661
             WL  +   + +++ +T +KY+PQA MN+ RKST GWSIGNILLDFTGG  + LQM++ +
Sbjct: 148   WLWLISIFNSIQVFMTCVKYIPQAKMNFTRKSTVGWSIGNILLDFTGGLANYLQMVIQS 206
```

Expect = 3.7e^{-10}



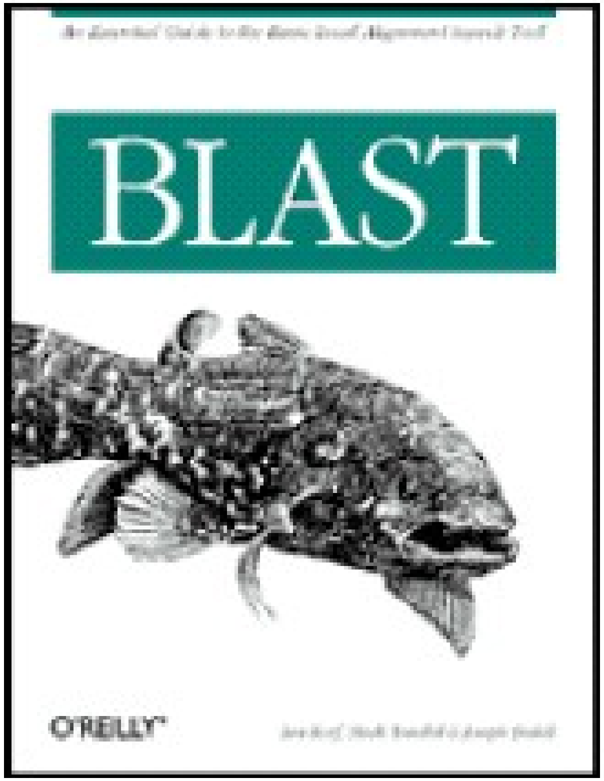
```
Score = 38.5 bits (88), Expect(2) = 1e-15
Identities = 15/34 (44%), Positives = 21/34 (61%)
Frame = -3

Query: 24595 DDWVSIFGDPTKFGLGLFSVLFDVFFMLQHYVFY 24494
             + W + +G+  K  L L S+ FD+ FM QHYV Y
Sbjct: 210   NSWKNFYGNMGKTLLSLISIFFDILFMFQHYVLY 243
```

Expect = 2.6e^{-8}

Best to calculate the ‘Expect(1)’ for each hit.

Which –hopefully– you now know how to do!

Enough BLAST for one day!


An Essential Guide to the Basic Local Alignment Search Tool
BLAST
O'REILLY
Ian Korf, Mark Yandell & Joseph Bedell