

# Heuristic Alignment and Searching

Mark Voorhies

3/30/2011

# Types of alignments

**Global Alignment** Each letter of each sequence is aligned to a letter or a gap (*e.g.*, Needleman-Wunsch).

**Local Alignment** An optimal pair of subsequences is taken from the two sequences and globally aligned (*e.g.*, Smith-Waterman).

# Types of alignments

**Global Alignment** Each letter of each sequence is aligned to a letter or a gap (*e.g.*, Needleman-Wunsch).

**Local Alignment** An optimal pair of subsequences is taken from the two sequences and globally aligned (*e.g.*, Smith-Waterman). *This tends to be more biologically relevant.*

The implementation of local alignment is the same as for global alignment, with a few changes to the rules:

The implementation of local alignment is the same as for global alignment, with a few changes to the rules:

- Initialize edges to 0 (*no penalty for starting in the middle of a sequence*)

The implementation of local alignment is the same as for global alignment, with a few changes to the rules:

- Initialize edges to 0 (*no penalty for starting in the middle of a sequence*)
- The maximum score is never less than 0, and no pointer is recorded unless the score is greater than 0 (*note that this implies negative scores for gaps and bad matches*)

The implementation of local alignment is the same as for global alignment, with a few changes to the rules:

- Initialize edges to 0 (*no penalty for starting in the middle of a sequence*)
- The maximum score is never less than 0, and no pointer is recorded unless the score is greater than 0 (*note that this implies negative scores for gaps and bad matches*)
- The trace-back starts from the highest score in the matrix and ends at a score of 0 (*local, rather than global, alignment*)





# Timing CLUSTALW

Timing CLUSTALW from the command line:

```
for i in 50 100 150 200 250 300 350 400 450; do
  head -n $i -q G217B_iron.fasta Pb01_iron.fasta > temp.fasta;
  time clustalw -infile=temp.fasta -type=DNA -align;
done
```

# Timing CLUSTALW

Timing CLUSTALW from the command line:

```
for i in 50 100 150 200 250 300 350 400 450; do
  head -n $i -q G217B_iron.fasta Pb01_iron.fasta > temp.fasta;
  time clustalw -infile=temp.fasta -type=DNA -align;
done
```

The output looks like this:

```
Sequences (1:2) Aligned. Score: 0
Guide tree file created: [temp.dnd]
```

```
There are 1 groups
Start of Multiple Alignment
```

```
Aligning...
Group 1:                               Delayed
Alignment Score 7238
```

```
CLUSTAL-Alignment file created [temp.aln]
```

```
real 0m3.400s
user 0m3.388s
sys 0m0.012s
```

# Timing CLUSTALW

You can copy the timing results into Excel.



# Timing CLUSTALW

You can fit the timing results to a curve in Excel.

$$y = Ax^B \quad (1)$$

$$\log y = \log Ax^B \quad (2)$$

$$= \log A + B \log x \quad (3)$$

$$= A' + B \log x \quad (4)$$

You can fit the timing results to a curve in Excel.

$$y = Ax^B \quad (1)$$

$$\log y = \log Ax^B \quad (2)$$

$$= \log A + B \log x \quad (3)$$

$$= A^0 + B \log x \quad (4)$$

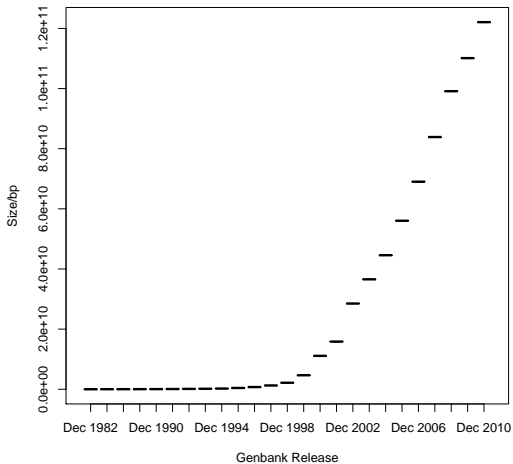
Here is an R script that does the same thing:

```
data <- read.csv("timings.csv", header = FALSE, col.names = c("t", "n"))
x <- log(data$n*80)
y <- log(data$t/60)
f <- lm(y ~ x)
x0 <- 0:40000
a <- exp(f$coeff[1])
b <- f$coeff[2]
pdf("ClustalwTimings.pdf")
plot(data$n*80, data$t/60, xlab = "length/bp", ylab = "time/minutes",
      main = "CLUSTALW_timings_on_Intel_Core2_T7300@2.00GHz,_32bit")
points(x0, a*x0^b, col = "blue", type = "l")
legend("topleft", c("y=-(1.8e-9)x^(2.08)"), col = "blue", lty = 1)
dev.off()
```



# $O(MN)$ time is too slow!

source: <ftp://ftp.ncbi.nih.gov/genbank/gbrel.txt>





## Why BLAST?

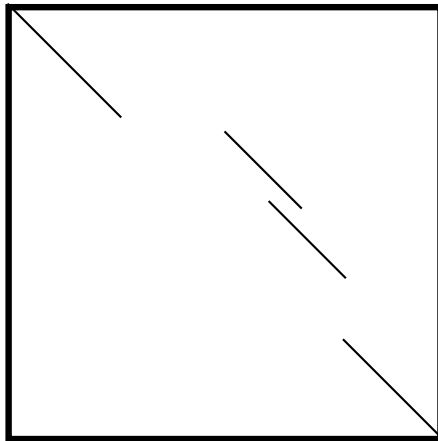
- Fast, heuristic approximation to a full Smith-Waterman local alignment
- Developed with a statistical framework to calculate expected number of false positive hits.
- Heuristics biased towards "biologically relevant" hits.

# Seeding searches

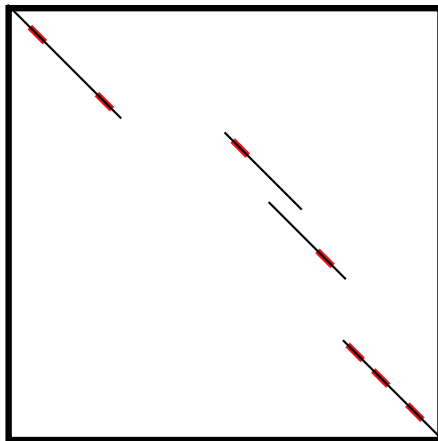
Most of the magic in a sequence-search tool lives in its indexing scheme

Program	Purpose	Indexing
BLAST	Database searching	Target indexing, 3aa or 11nt words
BLAT	mRNA mapping	Query indexing
BOWTIE	RnaSeq	Specialized index for low quality, short
e-PCR	Simulated PCR	Annealing-oriented index

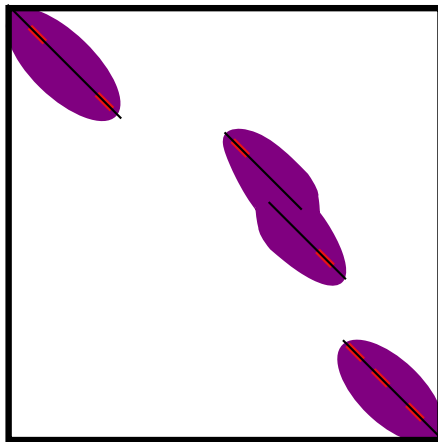
# BLAST: A quick overview



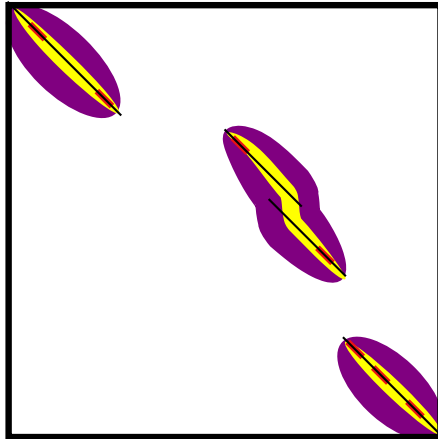
# BLAST: Seed from exact word hits



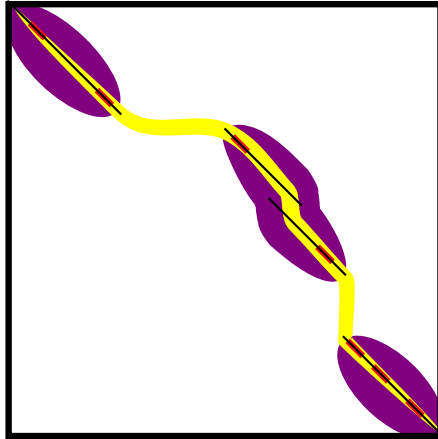
# BLAST: Myers and Miller local alignment around seed pairs



# BLAST: High Scoring Pairs (HSPs)



# Gapped BLAST: Merge neighboring HSPs



# How fast is BLAST?

The screenshot shows the NCBI BLAST web interface. The browser window title is "Nucleotide BLAST: Align two or more sequences using BLAST - Mozilla Firefox...". The page has a navigation bar with "blastn", "blastp", "blastx", "tblastn", and "tblastx" tabs. The main heading is "BLASTN programs: search nucleotide subjects using a nucleotide query. more...".

The interface is divided into sections for "Enter Query Sequence" and "Enter Subject Sequence".

**Enter Query Sequence:**

- Input field: "Enter accession number(s), gi(s), or FASTA sequence(s)"
- Buttons: "Clear", "Query subrange" (with "From" and "To" sub-inputs)
- Option: "Or, upload file" with a file path "/home/mvoorhie/Projects/Cc" and a "Browse..." button.
- Field: "Job Title" with a placeholder "Enter a descriptive title for your BLAST search".
- Checkbox: "Align two or more sequences" (checked).

**Enter Subject Sequence:**

- Input field: "Enter accession number, gi, or FASTA sequence"
- Buttons: "Clear", "Subject subrange" (with "From" and "To" sub-inputs)
- Option: "Or, upload file" with a file path "/home/mvoorhie/Projects/Cc" and a "Browse..." button.

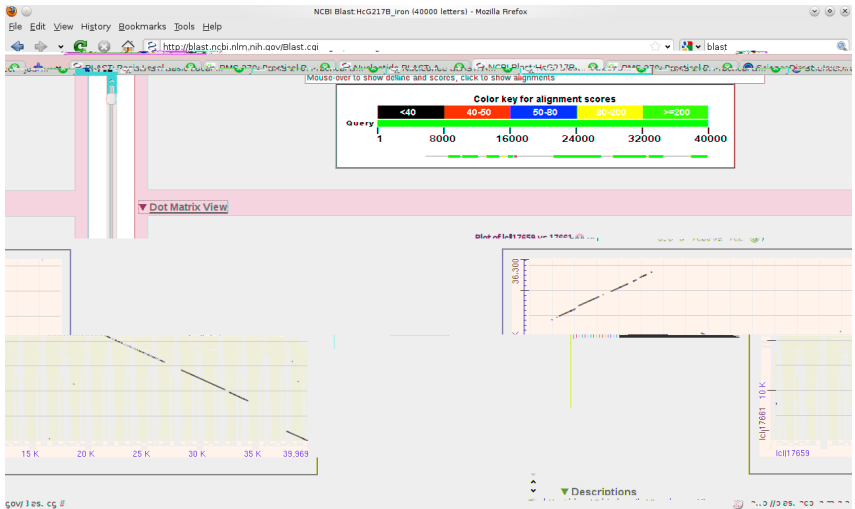
**Program Selection:**

- Section: "Optimize for"
- Radio buttons: "Highly similar sequences (megablast)" and "More dissimilar sequences (discontiguous megablast)".

The bottom of the browser window shows a taskbar with a clock at 3:14 and a system tray with a volume icon.



# How fast is BLAST?



# How fast is BLAST?

```
time bl2seq -p blastn -i G217B_iron.fasta -j Pb01_iron.fasta -e 1e-6 > temp.blastn
```

```
real    0m0.342s  
user    0m0.080s  
sys     0m0.032s
```

# The basic flavors of BLAST

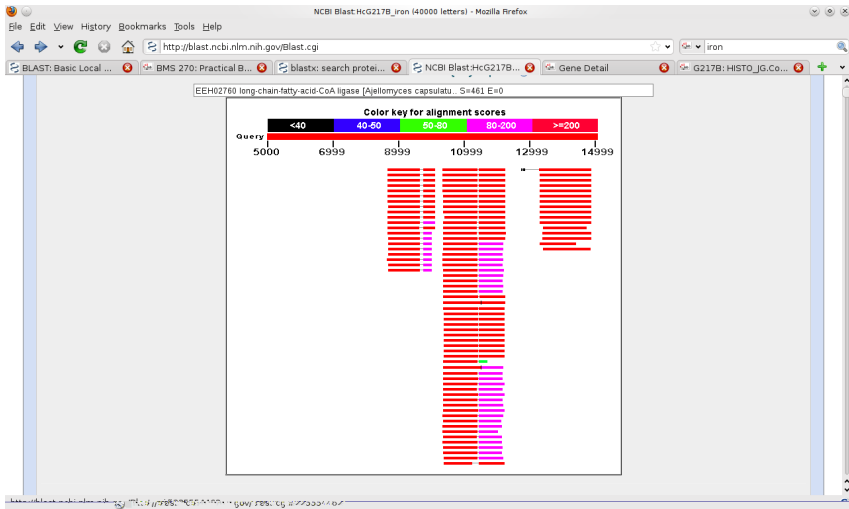
Target Query	Protein	DNA
Protein	BLASTP	TBLASTN
DNA	BLASTX	BLASTN TBLASTX

# BLASTX: Nucleotide query vs. Protein Database

The screenshot shows the NCBI BLASTX web interface. The browser title is "blastx: search protein databases using a translated nucleotide query - Mozilla Firefox". The address bar shows the URL: "http://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastx&BLAST\_PROGRAMS=blastx&PAGE\_TYPE=BlastSearch&SI...". The page content includes:

- Exclude** (Optional):  Models (XM/XP)  Uncultured/environmental sample sequences
- Entrez Query** (Optional): Enter an Entrez query to limit search
- BLAST** button
- Search database: **Non-redundant protein sequences (nr)** using **Blastx** (search protein databases using a translated nucleotide query)
- Show results in a new window
- Algorithm parameters**
  - General Parameters**
    - Max target sequences**: 100 (Select the maximum number of aligned sequences to display)
    - Expect threshold**: 10
    - Word size**: 3
    - Max matches in a query range**: 0
  - Scoring Parameters**
    - Matrix**: BLOSUM62
    - Gap Costs**: Existence: 11 Extension: 1
  - Filter**:  Low complexity regions
  - Mask**:  Mask for lookup table only
  - Mask lower case letters

# BLASTX: Nucleotide query vs. Protein Database



# Sometimes it's still worth running locally...

The screenshot shows a Mozilla Firefox browser window titled "NCBI Blast:HcG217B\_iron (40000 letters) - Mozilla Firefox". The address bar shows the URL "http://blast.ncbi.nlm.nih.gov/Blast.cgi". The page content includes the BLAST logo and navigation links. The main content area displays the search results for "HcG217B\_iron (40000 letters)".

**NCBI BLAST/blastx/Formatting Results - T73YCW0E01S**

[Edit and Resubmit](#) [Save Search Strategies](#) [Formatting options](#) [Download](#)

**An error has occurred on the server. Please, contact [blast-help@ncbi.nlm.nih.gov](mailto:blast-help@ncbi.nlm.nih.gov)**

**Informational Message: [blastxrv4.REAL]: Error: CPU usage limit was exceeded, resulting in SIGXCPU (24).**

**HcG217B\_iron (40000 letters)**

<b>Query ID</b>	lcl 2207	<b>Database Name</b>	nr
<b>Description</b>	HcG217B_iron	<b>Description</b>	All non-redundant GenBank CDS translations+PDB+S
<b>Molecule type</b>	nucleic acid		environmental samples from WGS projects
<b>Query Length</b>	40000	<b>Program</b>	BLASTX 2.2.25+ <a href="#">Citation</a>

**No significant similarity found. For reasons why, [click here](#)**

Other reports: [Search Summary](#)

**Sequence Viewer**

Done

$$E = kmne^{-S} \quad (5)$$

- $S$ : HSP score
- $E$ : Expected number of "random" hits in a database of this size scoring *at least*  $S$ .
- $m$ : Query length
- $n$ : Database size
- $k$ : Correction for similar, overlapping hits
- $\lambda$ : normalization factor for scoring matrix

$$E = kmne^{-S} \quad (5)$$

- $S$ : HSP score
- $E$ : Expected number of "random" hits in a database of this size scoring *at least*  $S$ .
- $m$ : Query length
- $n$ : Database size
- $k$ : Correction for similar, overlapping hits
- $\lambda$ : normalization factor for scoring matrix

A variant of this formula is used to generate sum probabilities for combined HSPs.



$$E = kmne^{-S} \quad (5)$$

- $S$ : HSP score
- $E$ : Expected number of "random" hits in a database of this size scoring *at least*  $S$ .
- $m$ : Query length
- $n$ : Database size
- $k$ : Correction for similar, overlapping hits
- $\lambda$ : normalization factor for scoring matrix

A variant of this formula is used to generate sum probabilities for combined HSPs.

$$p = 1 - e^{-E} \quad (6)$$

$$E = kmne^{-S} \quad (5)$$

- $S$ : HSP score
- $E$ : Expected number of "random" hits in a database of this size scoring *at least*  $S$ .
- $m$ : Query length
- $n$ : Database size
- $k$ : Correction for similar, overlapping hits
- $\lambda$ : normalization factor for scoring matrix

A variant of this formula is used to generate sum probabilities for combined HSPs.

$$p = 1 - e^{-E} \quad (6)$$

(If you care about the difference between  $E$  and  $p$ , you're already in trouble)

Important points:

- Extreme value distribution
- Assumption of infinite sequence length
- No rigorous framework for gap statistics (hmmer3 tries to fill this gap)

- BLAST is very fast, at the expense of not guaranteeing globally optimal results

# Summary

- BLAST is very fast, at the expense of not guaranteeing globally optimal results
- But the trade-offs that it makes are biased towards "biologically relevant" results

# Summary

- BLAST is very fast, at the expense of not guaranteeing globally optimal results
- But the trade-offs that it makes are biased towards "biologically relevant" results
- And it provides a statistical framework for evaluating its results.

# Summary

- BLAST is very fast, at the expense of not guaranteeing globally optimal results
- But the trade-offs that it makes are biased towards "biologically relevant" results
- And it provides a statistical framework for evaluating its results.
- We can, and should, treat our computer work as we would an experiment:
  - Document protocols and observations
  - Run positive and negative controls
  - Keep results organized and dated

# Homework

- Search your favorite proteins and collate interesting hits in one FASTA file per query { play with adding informative names and annotations (we will use these FASTA files tomorrow).
- Play with the BLAST book protocols (chapter 9) on the NCBI website
- Play with positive and negative controls (including permuted sequences)