Basic Local Alignment Search Tool

Martin Tötsches

14.06.2010

Table of contents

History of BLAST

- the program was designed by *Stephen W. Altschul*, *Warren Gish*, *Webb Miller*, *Eugene W. Myers* and *David J. Lipman*
- it was published in 1990 in the Journal of Molecular Biology *(J. Mol. Biol.)*
- it is one of the most common and best programs to compare new sequences *(amino acid or DNA)* with existing sequences in a database

global Alignments

- global Alignments completely align two given sequences
- a general technique to determine this type of Alignment is the *Needleman-Wunsch algorithm*
- an example with *GACTA* and *ACT*:

G	A	C	T	A
-	A	C	T	-

local Alignments

- a local Alignment is the best possible Alignment of two *substrings* of two given Sequences
- a general technique to determine this type of Alignment is the *Smith-Waterman algorithm*
- an example with *PQRAXABCSTVQ* and *XYAXBACSLL*:

A	X	A	B	-	C	S
A	X	-	B	A	C	S

- **BLAST** uses local Alignments

Score functions

- scoring functions usually asign numerical values to specific operations which are executed to align the given sequences
- in our case we have to asign values to *match*, *mismatch* and *Indel(Insertion or Deletion)*
- asigned scores (example): *match* = *+1*, *mismatch* = *-1* and *Indel* = *-2*

Score functions

G A C T A
\- A C T -

- the score for the above example would be:
  (-2) + 1 + 1 + 1 + (-2) = -1

important

BLAST does not allow Indels!
For nucleotide sequences **BLAST** uses the following scores
match = +5 mismatch = -4 Indel = NOT ALLOWED

Abstract

- the scoring according to amino acid sequences is more complicated than the scoring for nucleotide sequences
- $\longrightarrow$ development and usage of PAM-matrices
  (PAM = Point Accepted Mutation)
- PAM is a measure for the evolutionary distance between two amino acid sequences
- the developed matrices reflect the physiochemical properties of specific groups of amino acids

A matrix for DNA-sequences

	A	T	C	G
A	5	-4	-4	-4
T	-4	5	-4	-4
C	-4	-4	5	-4
G	-4	-4	-4	5

Remark

BLAST uses this score matrix for nucleotide sequences.

The PAM 120 matrix

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

Development of the PAM matrix

1. sequences who are nearly 85% identical are aligned
2. reconstruction of phylogenetic trees

Development of the PAM matrix

1. sequences who are nearly 85% identical are aligned
2. reconstruction of phylogenetic trees
3. count the substitutions of one amino acid by each other amino acid $\longrightarrow$ substitution matrix

Development of the PAM matrix

1. sequences who are nearly 85% identical are aligned
2. reconstruction of phylogenetic trees
3. count the substitutions of one amino acid by each other amino acid $\longrightarrow$ substitution matrix
4. calculate the mutability for each amino acid

Development of the PAM matrix

1. sequences who are nearly 85% identical are aligned
2. reconstruction of phylogenetic trees
3. count the substitutions of one amino acid by each other amino acid $\longrightarrow$ substitution matrix
4. calculate the mutability for each amino acid
5. generate a Mutation Data Matrix with the two formulas:

Development of the PAM matrix

1. sequences who are nearly 85% identical are aligned
2. reconstruction of phylogenetic trees
3. count the substitutions of one amino acid by each other amino acid $\longrightarrow$ substitution matrix
4. calculate the mutability for each amino acid
5. generate a Mutation Data Matrix with the two formulas:

$$M_{ij} = \begin{cases} \frac{m_j * A_{ij}}{\sum_{(i)} A_{ij}}, & \text{if } i \neq j \\ 1 - m_j, & \text{otherwise} \end{cases}$$

Development of the PAM matrix

1. sequences who are nearly 85% identical are aligned
2. reconstruction of phylogenetic trees
3. count the substitutions of one amino acid by each other amino acid $\longrightarrow$ substitution matrix
4. calculate the mutability for each amino acid
5. generate a Mutation Data Matrix with the two formulas:

$$M_{ij} = \begin{cases} \frac{m_j * A_{ij}}{\sum_{(i)} A_{ij}}, & \text{if } i \neq j \\ 1 - m_j, & \text{otherwise} \end{cases}$$

Methods

- maximal segment pair *(MSP)*
- local maximal segment pair *(LMSP)*
- approximation of MSP scores
- implementation

MSP

Definition

A MSP is the highest scoring pair of identical length segments chosen from two sequences.

- variable boundaries $\longrightarrow$ any length
- MSP provides a measure of local similarity
- allows the estimation of the statistical significance of the calculated scores under an appropriate *random sequence model*
  $\longrightarrow$ tractability to mathematical analysis

LMSP

Definition

A segment pair is locally maximal if its score cannot be improved either by extending or by shortening both segments.

Approximation of MSP scores

Definition

Let S be the threshold of the estimated MSP scores.

Definition

Let a word pair be a segment pair of fixed length w.

Definition

Let T be the threshold of the estimated word pair scores.

Approximation of MSP scores

- a scientist is only interested in those sequence entries with MSP scores over some cutoff score S
  1. sequences with high similarity
     $\longrightarrow$ biologically significant
  2. sequences with borderline scores
     $\longrightarrow$ helpful in distinguishing biologically interesting relationships
- **BLAST** searches the database for segments that contain a word pair with a score of at least T
  $\longrightarrow$ seeking for a word pair of fixed length w instead of searching the whole query minimizes time

Approximation of MSP scores

- a scientist is only interested in those sequence entries with MSP scores over some cutoff score S
  1. sequences with high similarity
     $\longrightarrow$ biologically significant
  2. sequences with borderline scores
     $\longrightarrow$ helpful in distinguishing biologically interesting relationships
- **BLAST** searches the database for segments that contain a word pair with a score of at least T
  $\longrightarrow$ seeking for a word pair of fixed length w instead of searching the whole query minimizes time
- any found hit is extended to determine if it is contained within a segment pair whose score is greater or equal to S

Approximation of MSP scores

- a scientist is only interested in those sequence entries with MSP scores over some cutoff score S
  1. sequences with high similarity
     $\longrightarrow$ biologically significant
  2. sequences with borderline scores
     $\longrightarrow$ helpful in distinguishing biologically interesting relationships
- **BLAST** searches the database for segments that contain a word pair with a score of at least T
  $\longrightarrow$ seeking for a word pair of fixed length w instead of searching the whole query minimizes time
- any found hit is extended to determine if it is contained within a segment pair whose score is greater or equal to S
- the lower the threshold T, the greater the chance that a segment pair with a score of at least S will contain a word pair with a score of at least T

Approximation of MSP scores

- a scientist is only interested in those sequence entries with MSP scores over some cutoff score S
  1. sequences with high similarity
     $\longrightarrow$ biologically significant
  2. sequences with borderline scores
     $\longrightarrow$ helpful in distinguishing biologically interesting relationships
- **BLAST** searches the database for segments that contain a word pair with a score of at least T
  $\longrightarrow$ seeking for a word pair of fixed length w instead of searching the whole query minimizes time
- any found hit is extended to determine if it is contained within a segment pair whose score is greater or equal to S
- the lower the threshold T, the greater the chance that a segment pair with a score of at least S will contain a word pair with a score of at least T

Implementation

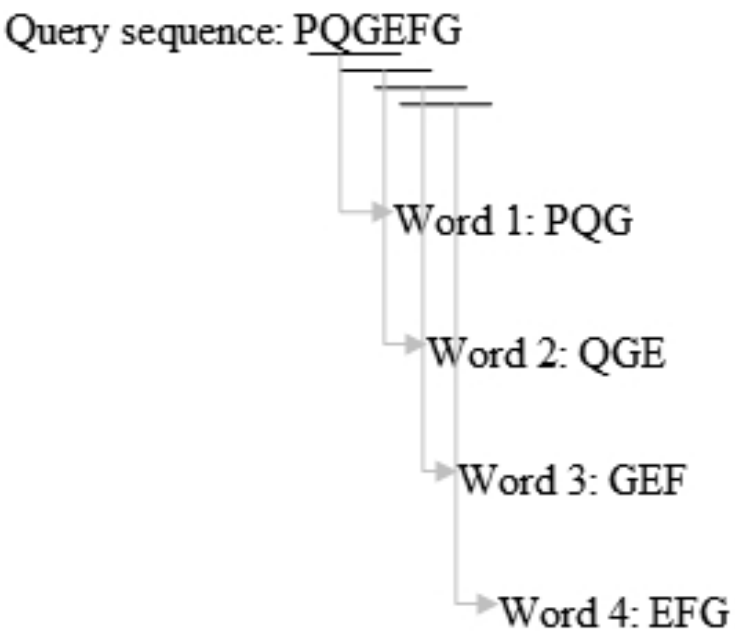
Algorithm

Three algorithmic steps have to be implemented:

1. compiling a list of high-scoring words from the query sequence
2. scanning the database for hits
3. extension of the hits

1. Creating the list of words

- example sequence: *PQGEFG*
- value for $w = 3$



1. Creating the list of words

- all the generated w-words from the query will be compared with all possible w-words and the individual scores will be calculated
- only those words with a score above or equal the threshold T are stored
- example: assume the first word of the previous slide $\longrightarrow$ PQG
- a comparison with PEG and PQA leeds to the scores 15 and 12
- if T is 13 PEG will be stored and PQA will be abandoned

1. Creating the list of words

Problem

DNA sequences are highly non-random, with locally biased base composition (e.g. A+T-rich regions), and repeated sequence elements and this has important consequences for the design of a DNA database search tool.
$\longrightarrow$ a database search will produce a copious output of matches with little interest

Solution: words generated by repetitive or unbalanced regions are removed from the query word list

2. Scanning the database

- the database is scanned for w-words that pair with w-words of the generated list of the query $\rightarrow$ BLAST-hits
- example for the query sequence: *PQGEFG*
- example for the database sequence: *PEGVVG*
- scanning process:

Query-sequence	P	Q	G	E	F	G
Database-sequence	**P**	**E**	**G**	V	V	G

2. Scanning the database

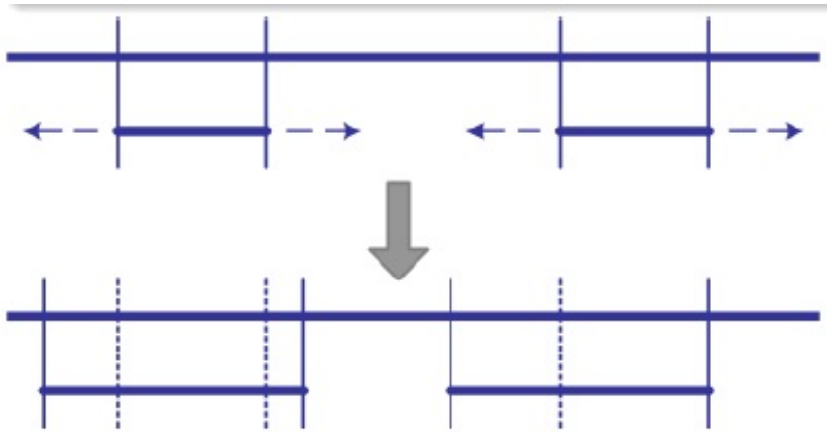
Problem

The scan of the database arouses a classic algorithmic problem, i.e. search a long sequence for all occurrences of certain short sequences.

There exist two main approaches for solving this problem:

1.
   - map each word to an integer between 1 an 20^w
   - a word can be uses as an index for an array
   - each database word leads directly to the corresponding hits
2.
   - usage of a deterministic finite automaton or finite state machine
     $\rightarrow$ utilization in BLAST because of efficiency

3. Extension of the found hits



- the hits from the database are extended in both directions
- the extensions terminate when a segment pair whose score falls a certain distance below the best score found for shorter extensions is found

Statistical significance

BLAST is able to return a value for the statistical significance of a found and extended hit in the database.

- when two sequences are compared the probability of finding a segment pair with a score greater or equal S is:

$$1 - e^{-(Kmne^{-\lambda S})}$$

- the probability of finding c or more distict segment pairs with a score greater or equal S is:

$$1 - e^{-(Kmne^{-\lambda S})} \sum_{i=0}^{c-1} \frac{y^i}{i!}$$

Complexity

Complexity

The expected-time computational complexity of the BLAST-algorithm is approximately:

$$\mathcal{O}(aW + bN + \frac{cNW}{20^w})$$

where:

W = the number of words generated

N = the number of residues in the database

a,b,c = constants

- W-term accounts for compiling the word list
- N-term covers the database scan
- NW-term is for extending the hits

Complexity

Complexity

The expected-time computational complexity of the BLAST-algorithm is approximately:

$$\mathcal{O}(aW + bN + \frac{cNW}{20^w})$$

where:
W = the number of words generated
N = the number of residues in the database
a,b,c = constants

- W-term accounts for compiling the word list
- N-term covers the database scan
- NW-term is for extending the hits

Performance of BLAST with "random sequences"

- the main question with regard to the performance of the algorithm is the selection of values for w, T and S
- therefore *Altschul et al.* generated one million pairs of "random protein sequences" of length 250 and subsequently searched the MSP for each using the PAM-120 matrix
- according to their investigations the most balancing values for w and T are:

	DNA-sequences	Protein-sequences
w	12	4
T	17	17

Performance of BLAST with "random sequences"

Results for the comparison of sequences with different w- and T-values

w	T	Implied % of MSPs missed by BLAST when S equals						
		45	50	55	60	65	70	75
3	14	20	16	12	10	8	6	5
4	16	18	14	11	8	6	5	4
	17	28	23	19	16	13	11	9
5	18	20	15	12	9	7	5	4

According to these results one may assume that the best value for T would be either 14 or 16.

Performance of BLAST with "random sequences"

- the number of words in the query-word list increases exponentially with decreasing T
- the values 14 or 16 for T lead to a crucial higher amount of words in the query-word list than a value of 17

Remark

If the amount of words in the query-word list increases it subsequently increases the needed CPU time exponentially.

$\rightarrow$ $T = 17$ balances the needed CPU time and the amount of probably missed MSPs

Performance of BLAST with homologous sequences

- comparison of proteins with other members of their respective superfamilies
  $\longrightarrow$ computing the true MSP scores
  $\longrightarrow$ computing the BLAST approximation with $w = 4$ and various settings of T
  $\Rightarrow$ comparison of the results with the ones from the previous random model
- two tests with BLAST:
  1. searching the globins with woolly monkey myoglobin
  2. comparing the mouse immunoglobin κ chain precursor V region with immunoglobin sequences

Performance of BLAST with homologous sequences

- comparison of proteins with other members of their respective superfamilies
  $\longrightarrow$ computing the true MSP scores
  $\longrightarrow$ computing the BLAST approximation with $w = 4$ and various settings of T
  $\Rightarrow$ comparison of the results with the ones from the previous random model
- two tests with BLAST:
  1. searching the globins with woolly monkey myoglobin
  2. comparing the mouse immunoglobin κ chain precursor V region with immunoglobin sequences
- both tests use $w = 4$ and $T = 17$

Performance of BLAST with homologous sequences

- comparison of proteins with other members of their respective superfamilies
  $\longrightarrow$ computing the true MSP scores
  $\longrightarrow$ computing the BLAST approximation with $w = 4$ and various settings of T
  $\Rightarrow$ comparison of the results with the ones from the previous random model
- two tests with BLAST:
  1. searching the globins with woolly monkey myoglobin
  2. comparing the mouse immunoglobin κ chain precursor V region with immunoglobin sequences
- both tests use $w = 4$ and $T = 17$

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs
   - in fact, it misses **43**

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs
   - in fact, it misses **43**

     $\longrightarrow$ the uniform pattern of conservation in the globins results in a relatively small number of high-scoring words between distantly related proteins

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs
   - in fact, it misses **43**
     $\longrightarrow$ the uniform pattern of conservation in the globins results in a relatively small number of high-scoring words between distantly related proteins
2. BLAST finds 33 subsequences containing MSPs with scores between 45 and 65

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs
   - in fact, it misses **43**
     $\longrightarrow$ the uniform pattern of conservation in the globins results in a relatively small number of high-scoring words between distantly related proteins
2. BLAST finds 33 subsequences containing MSPs with scores between 45 and 65
   - the random model suggests BLAST should miss about 8 MSPs

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs
   - in fact, it misses **43**
     $\longrightarrow$ the uniform pattern of conservation in the globins results in a relatively small number of high-scoring words between distantly related proteins
2. BLAST finds 33 subsequences containing MSPs with scores between 45 and 65
   - the random model suggests BLAST should miss about 8 MSPs
   - in fact, it misses only **2**

Performance of BLAST with homologous sequences

1. BLAST finds 178 subsequences containing MSPs with scores between 50 and 80
   - the random model suggests BLAST should miss about 24 MSPs
   - in fact, it misses **43**
     $\longrightarrow$ the uniform pattern of conservation in the globins results in a relatively small number of high-scoring words between distantly related proteins
2. BLAST finds 33 subsequences containing MSPs with scores between 45 and 65
   - the random model suggests BLAST should miss about 8 MSPs
   - in fact, it misses only **2**

Performance of BLAST with homologous sequences

Performance on real data

In general, the distribution of mutations along sequences has been shown to be more clustered than predicted by a Poisson process, and thus the BLAST approximation should, on average, **perform better on real sequences than predicted by the random model**.

Performance of BLAST with two long DNA sequences

- comparison of a 73.360bp section of the human genome containing the β-like globin gene cluster with a corresponding 44.595bp section of the rabbit genome with $w = 12$
- the pair exhibits three main classes of locally similar regions

Performance of BLAST with two long DNA sequences

- comparison of a 73.360bp section of the human genome containing the β-like globin gene cluster with a corresponding 44.595bp section of the rabbit genome with $w = 12$
- the pair exhibits three main classes of locally similar regions
  1. genes
  2. long interspersed repeats
  3. certain anticipated weaker similarities

Performance of BLAST with two long DNA sequences

- comparison of a 73.360bp section of the human genome containing the β-like globin gene cluster with a corresponding 44.595bp section of the rabbit genome with $w = 12$
- the pair exhibits three main classes of locally similar regions
  1. genes
  2. long interspersed repeats
  3. certain anticipated weaker similarities

Performance of BLAST with two long DNA sequences

- BLAST finds 98 alignments scoring over 200 and 57 alignments scoring over 350
  $\longrightarrow$ 45 of 57 paired genes
  $\longrightarrow$ 12 of 57 include long interspersed repeat sequences
  $\longrightarrow$ remaining alignments appear because of intergene similarities
- with a change of the value of w to 8 an additional 32 alignments are found
  $\longrightarrow$ all of these fall into one of the three classes
  $\longrightarrow$ no essentially new information

Performance of BLAST with two long DNA sequences

- BLAST finds 98 alignments scoring over 200 and 57 alignments scoring over 350
  $\longrightarrow$ 45 of 57 paired genes
  $\longrightarrow$ 12 of 57 include long interspersed repeat sequences
  $\longrightarrow$ remaining alignments appear because of intergene similarities
- with a change of the value of w to 8 an additional 32 alignments are found
  $\longrightarrow$ all of these fall into one of the three classes
  $\longrightarrow$ no essentially new information

Comparison to FASTA

Comparing BLAST with parameters $w = 4$ and $T = 17$ to FASTP in its most sensitive mode ($ktup = 1$)

$\rightarrow$ BLAST is of comparable sensitivity

Comparison to FASTA

Comparing BLAST with parameters $w = 4$ and $T = 17$ to FASTP in its most sensitive mode ($ktup = 1$)

$\longrightarrow$ BLAST is of comparable sensitivity

$\longrightarrow$ BLAST generally yields fewer false-positives

Comparison to FASTA

Comparing BLAST with parameters $w = 4$ and $T = 17$ to FASTP in its most sensitive mode ($ktup = 1$)

$\rightarrow$ BLAST is of comparable sensitivity

$\rightarrow$ BLAST generally yields fewer false-positives

$\rightarrow$ BLAST is an order of magnitude faster

Comparison to FASTA

Comparing BLAST with parameters $w = 4$ and $T = 17$ to FASTP in its most sensitive mode ($ktup = 1$)

$\rightarrow$ BLAST is of comparable sensitivity

$\rightarrow$ BLAST generally yields fewer false-positives

$\rightarrow$ BLAST is an order of magnitude faster

References

1. Basic Local Alignment Search Tool
   Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers and David J. Lipman
   J. Mol. Biol. (1990) 215, 403-410
2. Sequence Analysis I+II Lecture notes
   Faculty of Technology, Bielefeld University
   Winter 2008/2009 and Summer 2009
3. BLAST-Handbook
   http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=handbook