

# Serial BLAST Searching

Ian Korf, The Wellcome Trust Sanger Institute

## Abstract

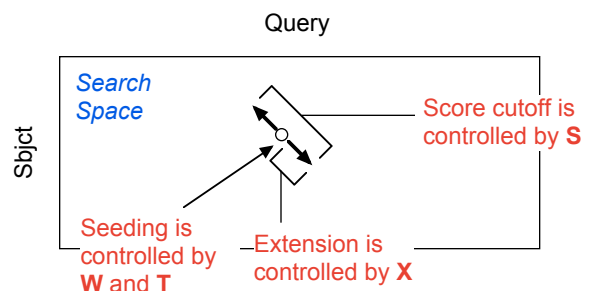
The translating BLAST algorithms are powerful tools for finding protein-coding genes because they identify amino acid similarities in nucleotide sequences. Unfortunately, these kinds of searches are computationally intensive and often represent bottlenecks in sequence analysis pipelines. Tuning parameters for speed can make the searches much faster, but one risks losing low-scoring alignments. However, high scoring alignments are relatively resistant to such changes in parameters, and this fact makes it possible to use a serial strategy where a fast, insensitive search is used to pre-screen a database for similar sequences, and a slow, sensitive search is used to produce the sequence alignments. The experiments presented here demonstrate that serial searching improves speed at almost no cost to sensitivity and can improve both speed and sensitivity.

## BLAST Background

BLAST is fast because it does not explore the entire search space between two sequences. The overall procedure simplified is:

1. Seeding
2. Extension
3. Alignment score cutoff

The speed and sensitivity of BLAST are controlled by parameters that operate at each of these stages.



## Seeding and Speeding

The most important parameters for increasing the speed of BLAST searches are wordsize ( $W$ ), word score threshold ( $T$ ), and 2-hit initiation ( $HITDIST$ ). Simply put, reducing the number of potential alignments makes BLAST run faster.

Increasing  $T$  is the simplest way to reduce the number of seeds. The following word pairs have a score of 12 in BLOSUM62 and would not be considered if  $T$  was raised to 13.

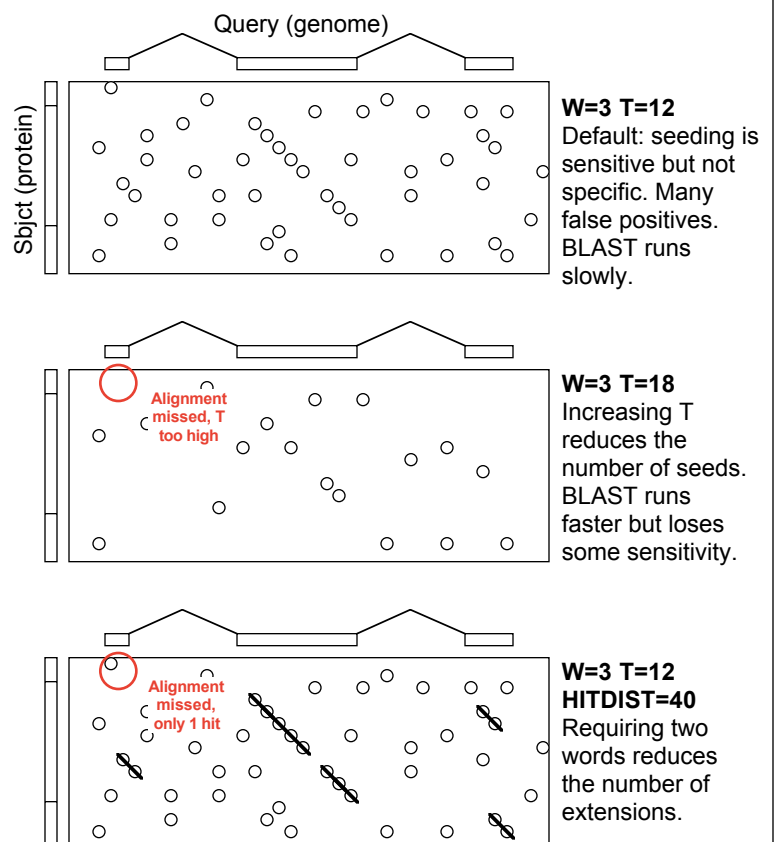
Query word:	AIL	QEK	LWP
Database word:	AIL	QER	MWD

Setting  $T$  very high, such as  $T=999$  limits seeds to matching words (like BLASTN searches).

Scaling up  $T$  and  $W$ , for example from  $W=3$   $T=12$  to  $W=4$   $T=16$ , reduces seeds because amino acid pairs have an average negative score.

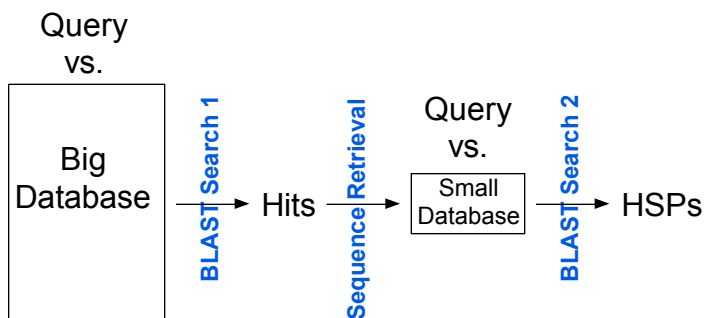
Using  $HITDIST$  reduces the number of potential alignments by requiring 2 word hits within an ungapped window.

## Seeding Examples



## The Serial Strategy

Serial searching adds an additional layer of thresholding to the general BLAST strategy but operates at the *sequence level* rather than the *alignment level*. The serial strategy has 3 parts: search 1, sequence retrieval, search 2.



Sequence retrieval is trivial because recent versions of WU-BLAST include *xdget* for retrieving sequences from BLAST databases.

Not shown here, but segmenting large sequences improves speed and sensitivity of serial searches.

## Searches and Serial Searches

**Table 1: The effect of seeding parameters on BLASTX searches.** Search of *Saccharomyces cerevisiae* chromosome I against all GenBank amino acid sequences. **WU-BLAST** and **NCBI-BLAST** defaults are colored as is a **recommended** parameter set.

		Search 1			Search 2		
W	T	Word Hits	Speed	Alignment Sensitivity	Sequence Sensitivity	40% Aln. Sensitivity	40% Seq. Sensitivity
<b>3</b>	<b>12</b>	<b>1</b>	<b>1.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
3	15	1	2.7	72.4	98.3	83.5	99.8
3	18	1	3.8	62.8	97.6	74.8	99.8
3	999	1	3.9	61.1	97.2	71.9	99.8
4	16	1	3.4	75.1	98.6	88.1	99.9
4	20	1	13.7	38.0	90.4	50.9	99.5
4	999	1	27.9	23.8	84.2	30.4	98.8
5	20	1	8.7	53.4	96.7	70.3	99.9
5	25	1	42.0	20.5	80.2	27.5	99.0
5	999	1	124.3	10.5	64.8	13.1	96.6
3	12	2	3.9	56.7	97.6	68.7	99.6
<b>3</b>	<b>13</b>	<b>2</b>	<b>6.1</b>	<b>41.5</b>	<b>95.6</b>	<b>52.2</b>	<b>99.6</b>
3	15	2	12.9	24.1	90.5	29.2	99.3
3	18	2	22.0	18.7	87.2	20.6	98.8
3	999	2	29.6	17.6	86.0	18.8	98.5
<b>4</b>	<b>16</b>	<b>2</b>	<b>19.4</b>	<b>25.5</b>	<b>91.9</b>	<b>31.7</b>	<b>99.2</b>
4	20	2	129.6	8.1	64.8	9.1	95.9
4	999	2	266.3	5.1	47.3	7.4	93.6
5	20	2	51.6	13.7	82.0	13.0	98.7
5	25	2	337.7	4.0	38.8	6.7	90.9
5	999	2	620.6	2.4	23.0	5.9	84.6

## Genefinding with TBLASTX

As more and more genomes are sequenced, TBLASTX will become an increasingly powerful tool for finding genes, especially those missed by traditional gene finders. What is the best way to run TBLASTX between genomes?

**Table 2: The effect of seeding parameters on TBLASTX searches.** Search of 500 *Caenorhabditis elegans* genes vs. *Caenorhabditis briggsae* draft genome. Sensitivity and specificity calculated at the nucleotide level with respect to the *C. elegans* annotations.

		Word Hits	Search 1		Search 2		Total Speed
W	T		SN	SP	SN	SP	
3	12	1	<b>95.0</b>	47.9	95.5	46.4	1.0
3	14	2	94.5	49.7	<b>95.4</b>	46.9	<b>2.3</b>
4	16	2	91.8	63.0	94.9	48.5	11.1
5	25	1	90.6	66.9	94.5	50.0	23.6
4	20	2	88.1	81.1	94.1	50.8	38.7
5	999	1	88.7	75.2	<b>94.2</b>	51.5	<b>60.8</b>

Improved sensitivity and speed

Huge speed increase with little loss in sensitivity

## Conclusions

The serial strategy greatly improves the speed of BLASTX searches. If one is only interested in alignments  $\geq 40\%$  identity (which makes sense given the default BLOSUM62 scoring matrix), even very fast parameters miss almost nothing. As demonstrated by the TBLASTX experiments, serial searching can improve both speed and sensitivity.

### Adaptations for NCBI-BLAST

- NCBI-BLAST supports word sizes of 2 or 3, so you're limited to changing *T* (-f option).
- Sequences are retrieved from BLAST databases with the *fastacmd* program but the *-I* flag specifying a gi-list may be a better solution since you don't need to create a second stage database.
- You can increase sensitivity in the second stage by turning off 2-hit initiation with -A 0.

### Nuts and Bolts

All BLAST searches employed WU-BLAST version 15-May-2002 and included the following parameters: *B=10000000 V=10000000 hspmax=0 filter=seg+xnu*. In those searches employing 2-hit initiation, *HITDIST=30* was used. TBLASTX searches included: *nogaps altscore="" any -999" altscore="" any \* -999"* to ensure there were no stop codons. The *C. briggsae* genome was segmented into 20 Kb segments to reduce second stage search space. All experiments were conducted on a Sun V880 with 4 x 750 MHz cpus and 8 GB RAM. A more detailed manuscript has been submitted to Bioinformatics (preprint available soon at <http://dna.cs.wustl.edu>).