"I like being surrounded by good ideas. Every single time you walk past something you like, you get a ***blast*** of happy chemicals to the brain, and I like that."
-- Douglas Coupland

Are there other sequences like this one?

1) Huge public databases - GenBank, Swissprot, etc.

2) Sequence comparison is the most powerful and reliable method to determine evolutionary relationships between genes

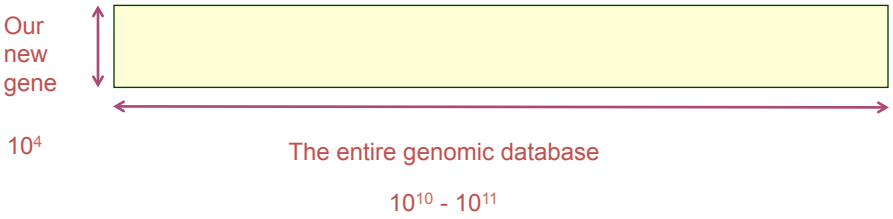
3) Similarity searching is based on alignment

4) So why not just do local alignment against the whole database???

Given a newly discovered gene,

- Does it occur in other species?
- How fast does it evolve?

Assume we try Smith-Waterman:

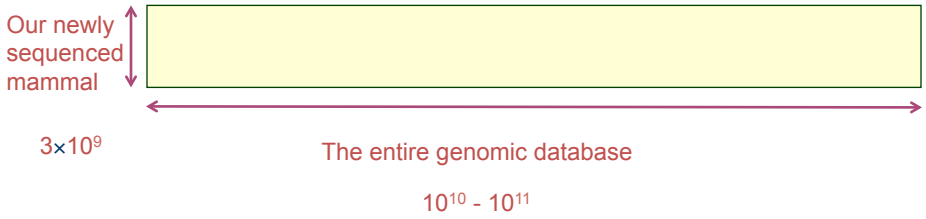


- Consider task of searching SWISS-PROT:
  - Say query sequence is 362 amino acids long
  - SWISS-PROT (v38) contains 29,085,255 amino acids
- Local alignment via Smith-Waterman:
  - $O(10^{10})$ matrix operations!
  - If each operation is 1/1000 of a second, this search takes 115.7 ***days!***
  - With modern processor and an optimized implementation, this search may take only ~10 minutes.
- What if need to do 1000 searches a day?!?
  - That requires a search to be < 1.4 minutes apiece.

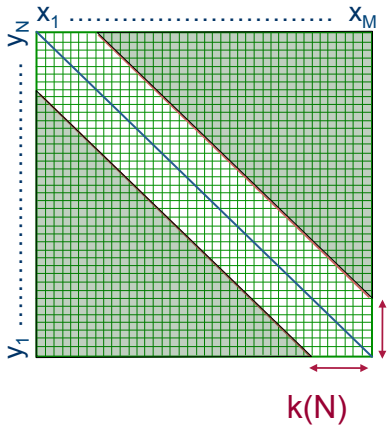
Given a newly sequenced organism,

- Which subregions align with other organisms?
  - Potential genes
  - Other biological characteristics

Assume we try Smith-Waterman:



Bounded Dynamic Programming



Ways of reducing alignment algorithm demands:

- Can reduce space at cost of time, but this trades MORE compute for less memory.
- Methods for bounding the DP to reduce time AND space, but at the cost of perhaps *missing* the best alignment.

Heuristic algorithms for local search

BLAST and **FASTA*** provide rapid similarity searching

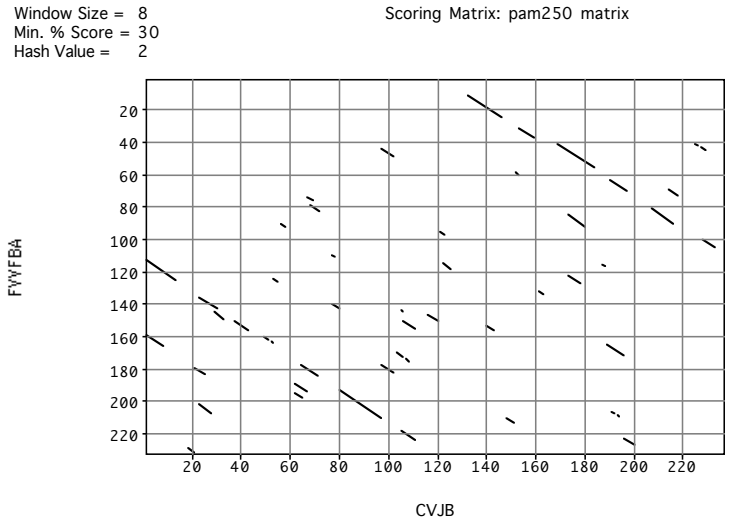
rapid = approximate (heuristic)

Tradeoff: sensitivity vs speed

sensitivity = # significant matches detected / # significant matches in the DB

*Note that FASTA here refers to a suite of alignment programs. It was the first of these approaches and established the now ubiquitously used FASTA file format! It is *unfortunate confusion* that they are named identically.

Recall our dot plots …



FASTA

1) Derived from logic of the dot plot
   - compute best diagonals from all frames of alignment
2) Word method looks for exact matches between words in query and test sequence
   - hash tables !!!
   - DNA words are usually 6 bases
   - protein words are 2 or 3 amino acids
   - only searches for diagonals in region of word matches = faster searching

Hash Tables: Basic Idea

- Use a key (arbitrary string or number) to index directly into an array – O(1) time to access records
  - A[“kreplach”] = “tasty stuffed dough”
  - Need a *hash function* to convert the key to an integer

	Key	Data
0	kim chi	spicy cabbage
1	kreplach	tasty stuffed dough
2	kiwi	Australian fruit

10

Optimal Hash Function

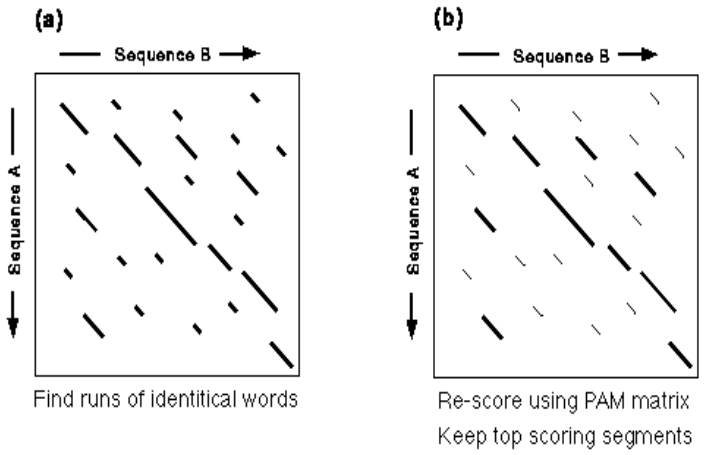
- The best hash function would distribute keys as evenly as possible in the hash table
- A collision occurs when two different keys hash to the same value. (Cannot store both data records in the same slot in array!)
- Collision resolution is necessary, but time consuming.

11

Hash Table tradeoffs

- Smaller keys = faster for small data, but more collisions for big data, which slows down look up.
- Larger keys = fewer collisions, but very large table (lots of memory) for big data.

FASTA Algorithm

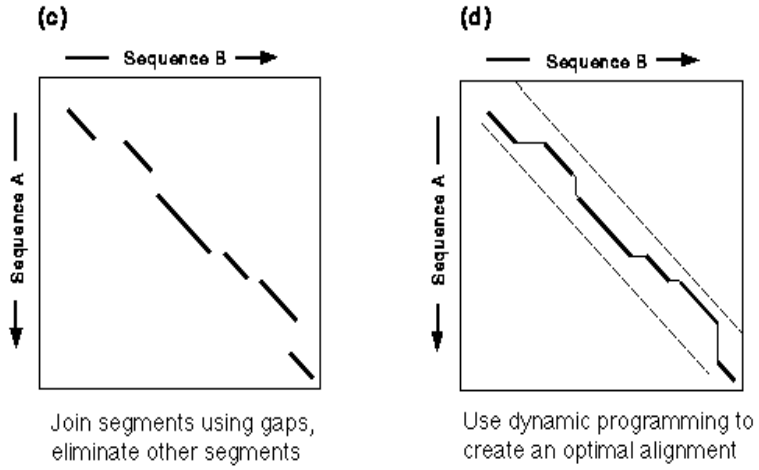


Makes Longest Diagonal

3) after all diagonals found, tries to join diagonals by adding gaps

4) computes alignments in regions of best diagonals

FASTA Alignments



BLAST Searches GenBank

[**BLAST**= **B**asic **L**ocal **A**lignment **S**earch **T**ool]

(Altschul et. al. 1990)

The NCBI **BLAST** web server lets you compare your query sequence to various sections of GenBank:

- **nr** = non-redundant (main sections)
- **month** = new sequences from the past few weeks
- **ESTs**
- human, drosophila, yeast, or E.coli genomes
- proteins (by automatic translation)

- This is a VERY fast and powerful computer.

BLAST — Original Version

Dictionary (a Hash Table!):

All words of length k (~11)

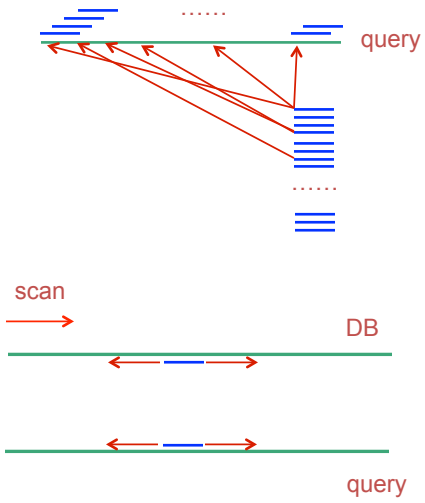
Alignment initiated between words of alignment score ≥ T (typically T = k)

Alignment:

Ungapped extensions until score below statistical threshold

Output:

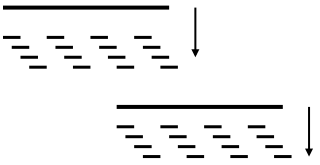
All local alignments with score > statistical threshold



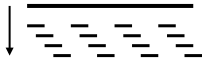
BLAST Word Matching

```
MEAAVKEEISVEDEAVDKNI
MEA
 EAA
  AAV
   AVK
    VKE
     KEE
      EEI
       EIS
        ISV
        ...
```

Break query into words:

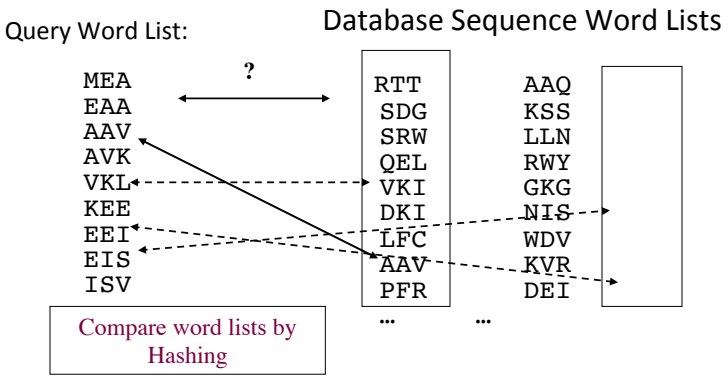


Break database sequences into words:

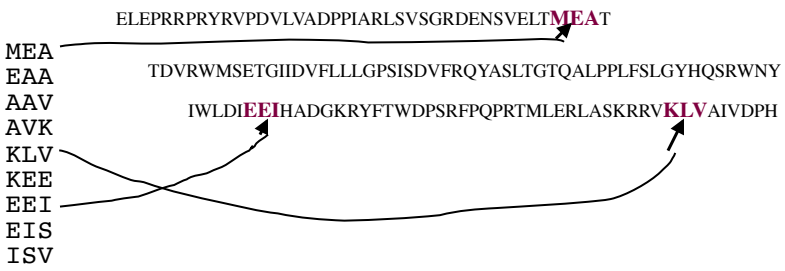


Calculate score of all possible variations for each word, keep all word variations scoring > T

Compare Word Lists

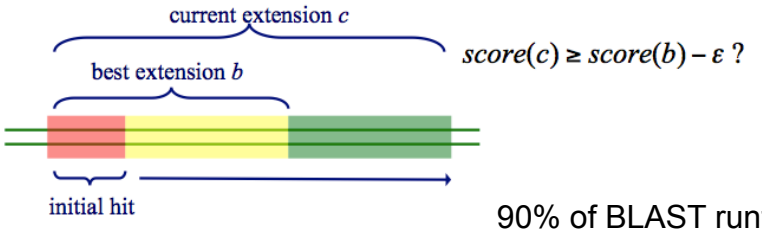


Find locations of matching words in database sequences



Extension

- Extend in both directions (without allowing gaps)
- Terminate extension in either direction when score falls below a certain distance of best score for shorter extension



90% of BLAST runtime is spent in this step!

BLAST — Original Version

Example:

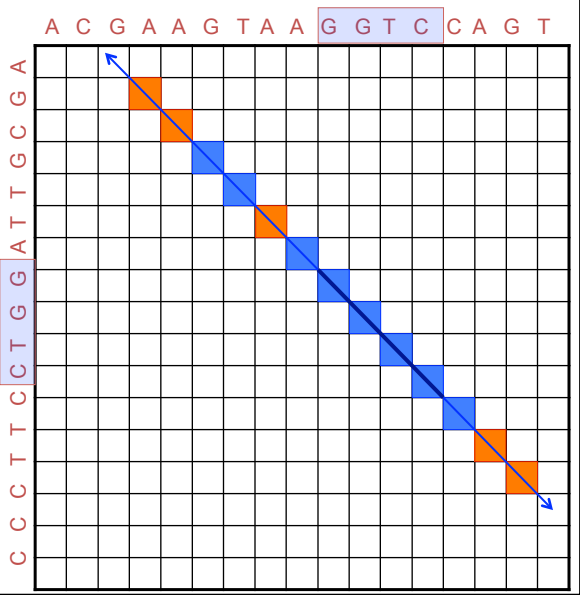
k = 4,
T = 4

The matching word GGTC initiates an alignment

Extension to the left and right with no gaps until alignment falls < 50%

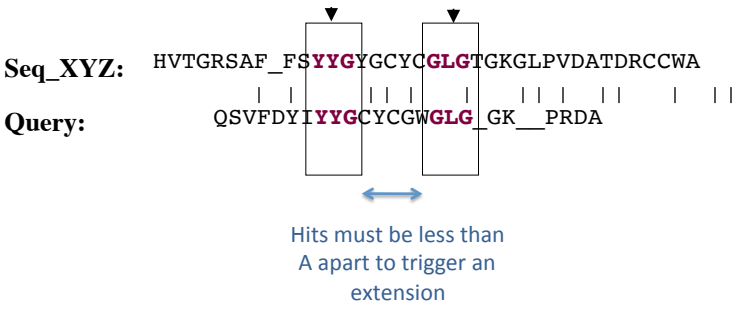
Output:

GTAAGGTCC
GTTAGGTCC



Gapped BLAST algorithm

- The NCBI's BLAST website now uses "gapped BLAST"
- This algorithm is more complex than the original BLAST
- It requires two word matches close to each other on a pair of sequences (i.e. with a gap) before it creates an alignment



•Use **two** word matches as anchors to build an alignment between the query and a database sequence.

•Then score the alignment.

HSPs are Aligned Regions

- The results of the word matching and attempts to extend the alignment are segments

  called HSPs (High-scoring Segment Pairs)

- **BLAST** often produces several short HSPs rather than a single aligned region

BLAST is Approximate

- BLAST makes similarity searches very quickly because it takes shortcuts.
  - looks for short, nearly identical "words"
- It also makes errors
  - misses some important similarities
  - makes many incorrect matches
    - easily fooled by repeats or skewed composition
    - Or evenly spaced mismatches!

Filters

Default filters remove low complexity from protein searches and known repeats (ie. *Alu*) from DNA searches.

Reduces how often gets fooled by low complexity regions, but at cost of making these regions essentially “unalignable”.

Word size

- Default word size
  - 11 bases for DNA
  - 3 for protein
- Short sequences will have few words
- Low quality sequence might have a sequencing error in every word
- Impacts sensitivity!
  - Shorter words => more sensitive => more words to expand => more runtime!

Flavors of Blast

Program	Query	Database
BLASTP	Protein	Protein
BLASTN	DNA	DNA
BLASTX	Translated DNA	Protein
TBLASTN	Protein	Translated DNA
TBLASTX	Translated DNA	Translated DNA