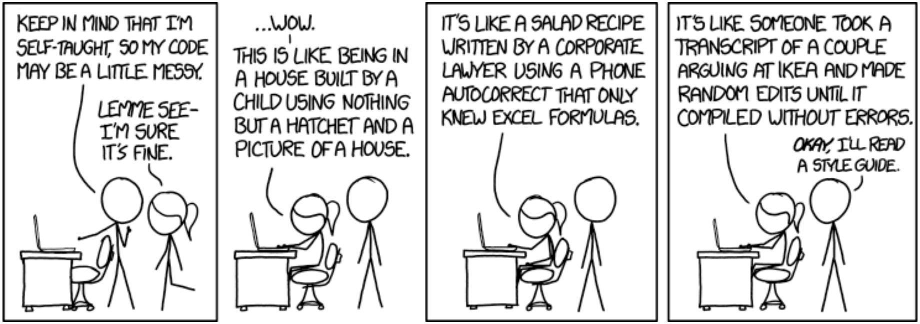

KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE– I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

https://xkcd.com/1513/

DIAMOND: Fast protein alignment

Don L. Armstrong

Institute for Genomic Biology, Computing Genomes for Reproductive Health, University of Illinois, Urbana-Champaign

October 18, 2015

Code and slides are here:

http://dla2.us/p/dmnd2015

The Problem

- mRNA sequences
- No clear reference genome/database
- How to annotate them?

The Problem

- mRNA sequ
- No clear ref
- How to ann

No Reference Genome

- Blind mole rat (*Spalax*)
- No reference genome
- Close to mouse/rat, but not close enough to use mouse/rat directly
- Placenta sequences; how to annotate them?

The Problem

- mRNA sequ
- No clear ref
- How to ann

Environmental Samples

- Permafrost
- Many un-cultured, un-sequenced bacteria
- How to annotate what genes they are expressing?

Basic Solution

- Annotate mRNA against proteins in the nr or in a suitable reference proteome
- Six possible translations of nucleotide into amino acid
- Take all possible sub-sequences
- Hash into reference, extend match
- Pick best match

Previous Contenders

- Blastx
- Rapsearch
- mBlast
- many, many, more

Diamond Methodology Advances

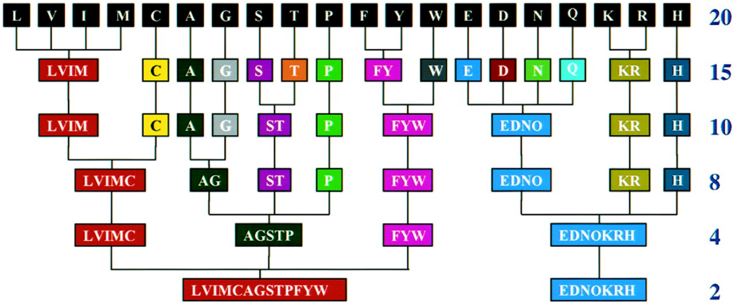
- Seed and Extend
- Reduced Alphabet
- Spaced Seeds with Specific Seed Shape
- Double Indexing

Seed and Extend

- Calculate an index
- Look up matching indices in the database
- Local string alignment using Smith-Waterman
- Looks like blast, right?

Reduced Alphabet

- LVI M C G STA P F Y W KREDNQ
- Smaller index sizes — less memory usage
- Greater sensitivity — seed more likely to match
- More likelihood of useless extensions — only the seed matched


L V I M C A G S T P F Y W E D N Q K R H
20
LVIM C A G S T P FY W E D N Q KR H
15
LVIM C A G ST P FYW EDNQ KR H
10
LVIMC AG ST P FYW EDNQ KR H
8
LVIMC AGSTP FYW EDNQKRH
4
LVIMCAGSTPFYW EDNQKRH
2

[1]

Spaced Seeds with Specific Seed Shape

- Spaced seeds are longer seeds in which only a subset of the positions are used
- For example, if
    - the sequence was ABCDEFGHI
    - the seed shape was 11100010
    - then you would query into the index with ABCG
- Originally presented in PatternHunter[2]
- Why is this better than consecutive seeds?

Consecutive Seeds vs Spaced Seeds

- Target Sequence: ABCDEFGHIJK
- Sequenced Sequence: ABCZEFYHIXK
- Seed Shape: 11100010 (4) and Consecutive: 1111 (4)

Pathological example

Shift	Spaced	Consecutive
0	ABCF=ABCF	ABCD≠ABCZ
1	BCDG≠BCZY	BCDE≠BCZE
2	CDEH≠CZEH	CDEF≠CZEF
3	DEFI≠ZEFI	DEFG≠ZEFY
4	EFGJ≠EFYW	EFGH≠EFYH
5	FGHK≠FYHK	FGHI≠FYHI
6		GHIJ≠YHIW
7		HIJK≠HIWK

- Spaced seed matches once
- Consecutive seed never matches
- Consecutive seed does more comparisons and may match repeatedly

Optimal Spaced Seed

- Fewest overlaps with shifted seed
- Longer seeds are better
- Equivalent weight
- Use dynamic programming to calculate optimal seed for given length

DIAMOND Seeds (Fast)

- 111101011101111 (12)
- 111011001100101111 (12)
- 1111001001010001001111 (12)
- 111100101000010010010111 (12)

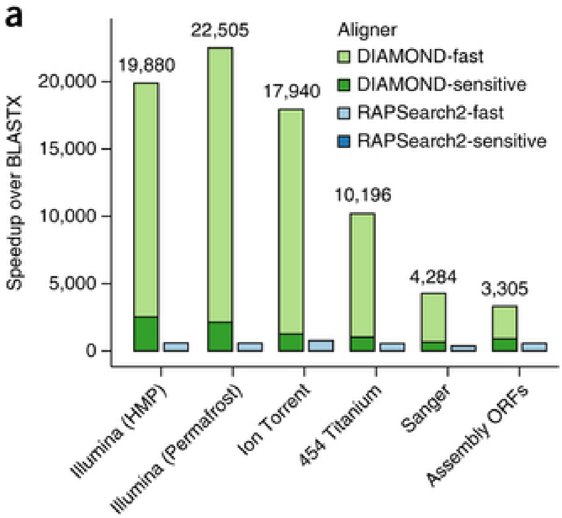
Double Indexing

- Blastx indexes the database
- Blastx runs the queries in input order
- DIAMOND indexes both the database and the queries
- DIAMOND runs queries in index order
- Why is this faster?

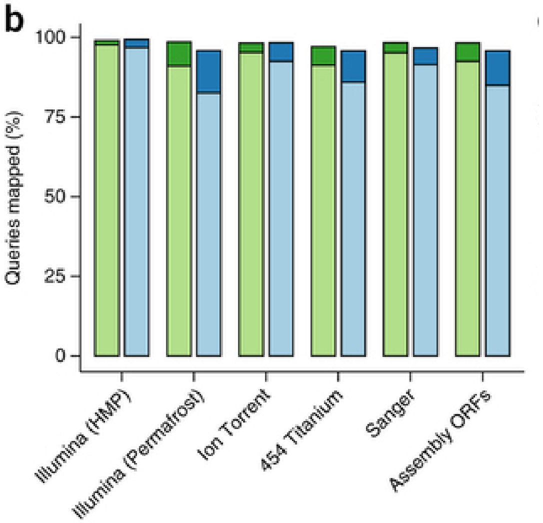
Double Indexing: Why it's faster

- Cache architecture
  - On CPU Cache – L1,L2
  - Shared CPU Cache L3
  - Much faster than main memory
- Each cache miss must hit main memory (must hit northbridge, which has significantly more latency than main cache, and takes hundreds of cycles)
- Dictionary Example: Is it faster to look up
  - "apple", "xylophone", "appliance", "xylem"
  - or "apple", "appliance", "xylem", "xylophone"?

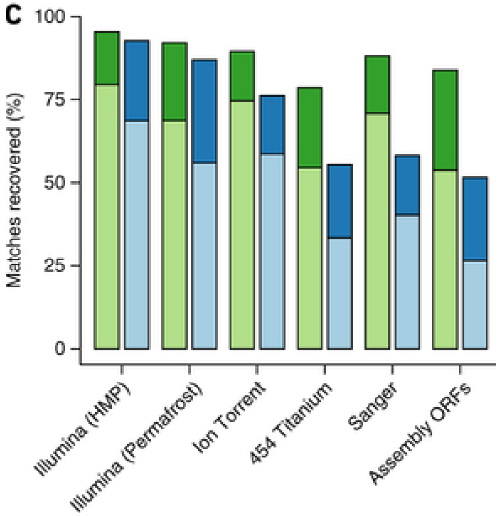
Speed of DIAMOND



Accuracy of DIAMOND: Any success



Accuracy of DIAMOND: Matches blastx



DIAMOND Usage

- Make the diamond database: `diamond makedb --in foo.fasta --db foo.dmnd;`
- Run the diamond query: `diamond blastx --db foo.diamond --threads 24 --query bar.fasta --daa bar_diamond.txt`

DIAMOND Output

- Standard BLASTx output
- Equivalent evalues and bit scores
- An example from *Spalax* (the top two proteins are isoforms):

query	match	% ident	length	# mm	gap	qst	qstp	sstart	sstop	evalue	score
c18_g1_i1	...065786	94.5	361	20	0	2	1084	992	1352	5.7e-203	704.9
c18_g1_i1	...081540	94.5	361	20	0	2	1084	940	1300	5.7e-203	704.9
c18_g1_i1	...142322	48.8	361	178	3	5	1078	944	1300	5.9e-99	359.4
c18_g1_i1	...039711	48.8	361	178	3	5	1078	936	1292	5.9e-99	359.4
c18_g1_i1	...141518	43.0	230	124	3	5	685	936	1161	1.7e-50	198.4

References

Murphy, L. R., Wallqvist, A. & Levy, R. M. Simplified amino acid alphabets for protein fold recognition and implications for folding. *Protein Eng. Des. Sel.* **13,** 149–152 (Mar. 2000).

Ma, B., Tromp, J. & Li, M. PatternHunter: faster and more sensitive homology search. *Bioinformatics* **18,** 440–445 (Mar. 2002).