DIAMOND vs BLAST
Sequence Alignment Software

Sheila Podell

November 9, 2015

Why use Diamond?

- Sequence comparisons on really large data sets
  - Nucleotide or protein query versus protein database
- Advantages
  - MUCH faster than BLASTX or BLASTP
  - Sensitivity similar to BLAST
- Disadvantages
  - Slower than BLAST on small data sets
  - Lower sensitivity on distant matches (e-value >1e-5)
  - Can’t be used for BLASTN (nucleotide versus nucleotide)

Why is DIAMOND faster?

- Double Index Alignment
  - Seed list and locations calculated for both query and reference sequences (BLAST indexes reference only)
- Spaced kmer (word) seeds
  - Longer total length, subset of discontinuous positions
- Simplified kmer amino acid alphabet (11 vs 20)
- Smaller binary database reference file (1/2 size)
- Optimized RAM usage avoids VM delays
- Seeds selected by simple exact match for final extension using Smith-Waterman alignment

How to get it

- Algorithm description
  - Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Methods. 2015 Jan;12(1): 59-60
- Documentation and downloading
  - http://ab.inf.uni-tuebingen.de/software/diamond/
- Installation
  - Linux 64-bit binary available
  - OSX compile from source (wasn't difficult)
  - Clear, detailed instructions on website

How to use it

- Format database
  - took ~30 mins for Genbank nr (54 million sequences)

```
diamond makedb --in nr.faa -d nr --threads 16
```

- Run program (separate commands for output)

```
diamond blastp -d nr.dmnd -q query.faa -a matches.daa –t tmpdir
diamond view -a matches.daa -o matches.tab -f tab
diamond view -a matches.daa -o matches.sam -f sam --compress 1
```

- Very clear instructions, options similar to BLAST
  - blastx/p, number of matches reported, gap penalties, e-value or bitscore cutoffs, low complexity masking, threads
  - additional options: sensitivity profile, memory usage

DIAMOND versus BLAST comparisons

- Published claims (BLASTX, 100bp Illumina reads)
  - 20,000 times faster than BLASTX in “default” mode
    - Found 80-90% BLASTX matches at threshold e-value < 1e-5
  - 2,500 times faster than BLASTX in “sensitive” mode
    - Found 94% BLASTX matches at threshold e-value < 1e-5
- My own experience (BLASTP, full length proteins)
  - 50X faster in default mode, 15X faster in sensitive mode
  - Differences between numbers of matches are smaller than statistical “noise” from uneven database representation at e-value < 1e-5

Published BLASTX comparisons

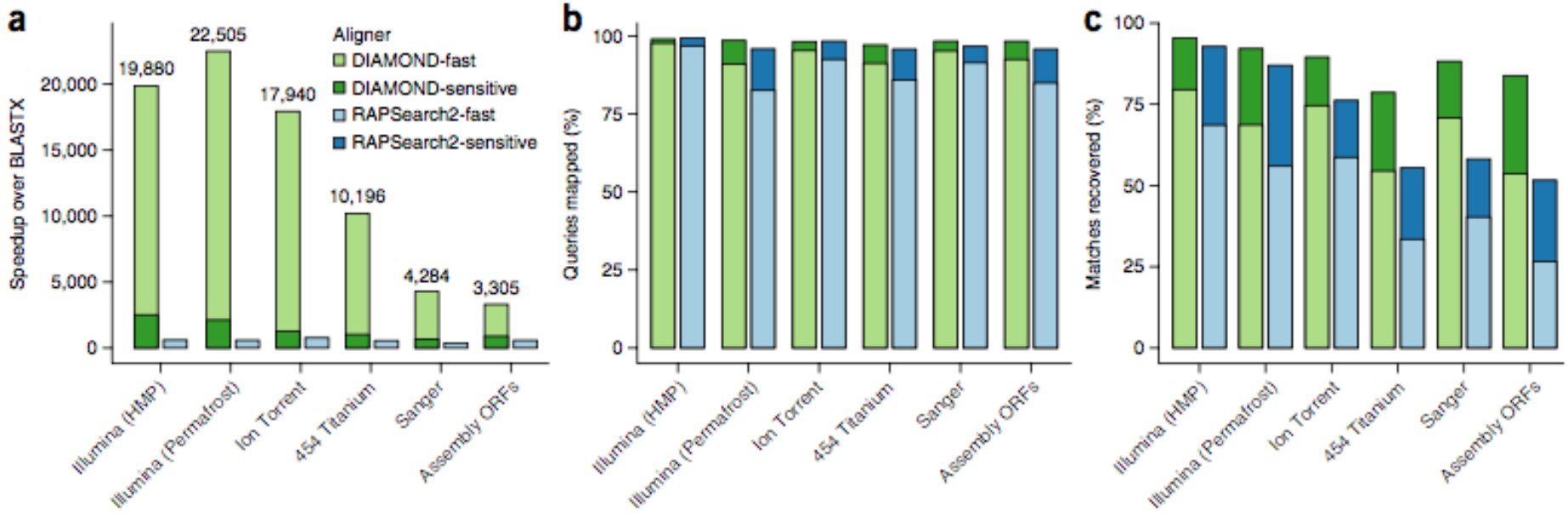


Figure 1 | Comparison of DIAMOND and RAPSearch2 against BLASTX for four sequencing technologies and for ORFs predicted from a bacterial assembly. (**a**) Fold speedup of each program over BLASTX. (**b**) Percentage (relative to BLASTX) of queries for which each program reports at least one alignment. (**c**) Percentage (relative to BLASTX) of matches recovered by each program. Only alignments with an expected value of ≤0.001 are considered. Programs were set to report alignments for up to 250 target sequences per read. Times are wall-clock times on a server using 48 cores and exclude one-time program startup overhead, which was <1 min for BLASTX and 5 min for DIAMOND-fast. HMP, Human Microbiome Project.

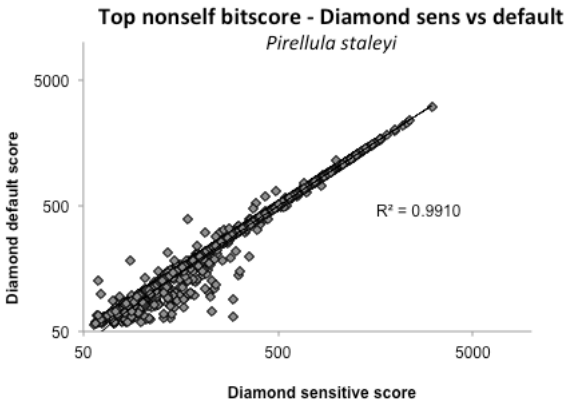
BLASTP versus DIAMOND
My own unpublished results

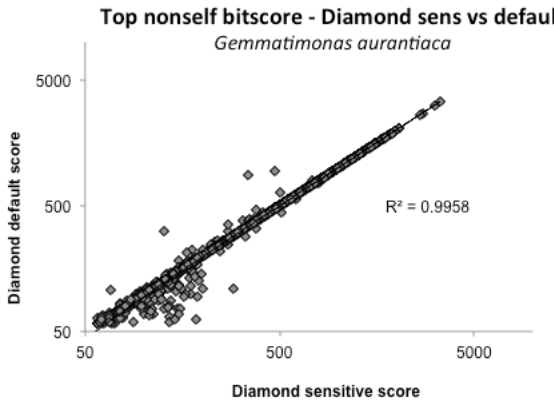
query	method	CPU time	wall time	speed vs ncbi	non-self matches eval 1e-5	missing non-self matches	pct missing matches	num LPI <0.6	pct LPI <0.6	pct GC
Pirellula staleyi	blastp	257:25:55	36:03:02	1	3979	738	15.6%	872	18.5%	58
query size = 4717	diamond_sens	15:05:08	1:57:30	18	4057	660	14.0%	965	20.5%	
	diamond	4:49:05	0:39:27	55	3853	864	18.3%	923	19.6%	
Gemmatimonas aurantiaca	blastp	204:53:56	29:52:20	1	3614	321	8.2%	983	25.0%	64
query size =3935	diamond_sens	15:45:23	2:06:27	14	3656	279	7.1%	1001	25.4%	
	diamond	4:55:35	0:39:49	45	3566	369	9.4%	968	24.6%	
Chloroflexus aurantiacus	blastp	210:03:14	34:03:05	1	3786	67	1.7%	198	5.1%	57
query size =3853	diamond_sens	15:33:13	2:01:00	17	3807	46	1.2%	180	4.7%	
	diamond	4:55:00	0:40:51	50	3788	65	1.7%	173	4.5%	
Bifidobacterium longum	diamond_sens	14:02:57	1:49:48	nd	2362	190	7.4%	70	2.7%	60
query size = 2552	diamond	4:33:32	0:37:43	nd	2345	207	8.1%	70	2.7%	
Fusobacterium nucleatum	diamond_sens	15:13:06	1:57:11	nd	1962	21	1.1%	50	2.5%	
query size = 1983	diamond	4:58:03	0:39:48	nd	1958	25	1.3%	49	2.5%	27
Buchnera aphidicola	diamond_sens	14:10:02	1:50:44	nd	561	3	0.1%	3	0.1%	
query size = 564	diamond	4:40:54	0:39:03	nd	560	4	0.2%	3	0.1%	26

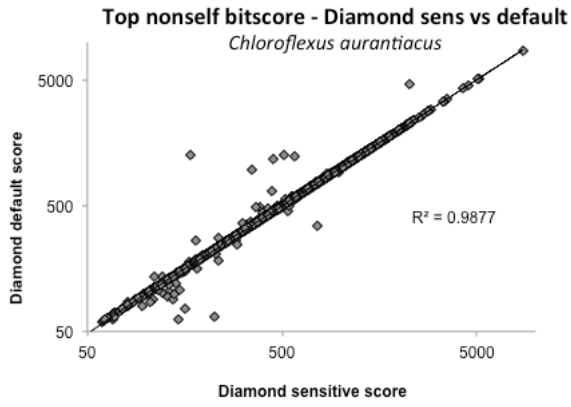
DarkHorse LPI scores < 0.6 identify phylogenetically unexpected matches, e.g. HGT

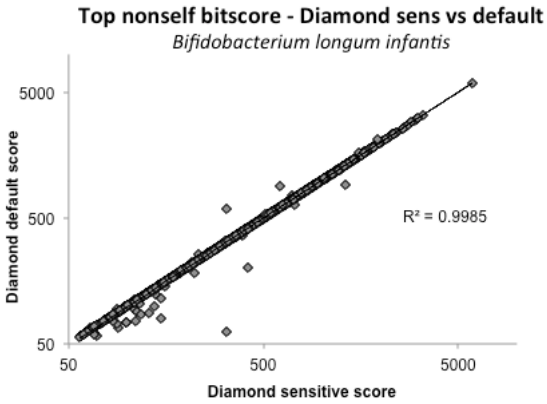
DIAMOND fast vs sensitive

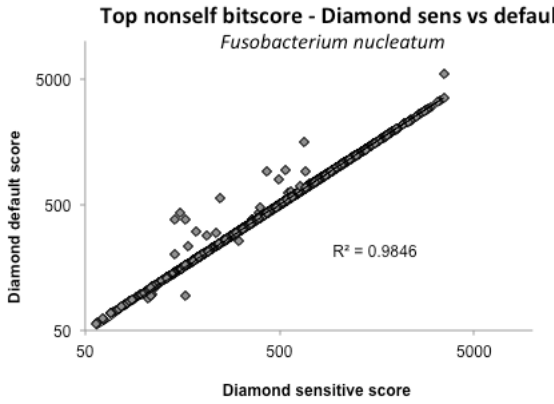
Which finds the "best" match?

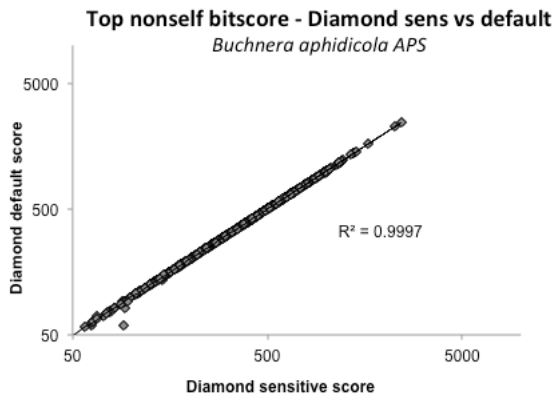












Conclusions

- DIAMOND really works.
- Easy to install and use.
- Even if you are using BLASTP instead of BLASTX, a 15- 50X speed up (versus 2,500-20,000X) can still be incredibly helpful on large data sets.
- If you do large blastx or blastp comparisons

Get it NOW!