Tutorial 10:

Run Diamond Docker on the datasets for Kraken2

Background

What is DIAMOND?

- A sequence aligner for protein and translated DNA searches
- Designed for high performance analysis of big sequence data
- Has an alignment sensitivity that matches BLAST
- Features various sensitivity modes:
    - Default (fast)
    - Sensitive
    - Very-sensitive
    - Ultra-sensitive

How Does DIAMOND Work?

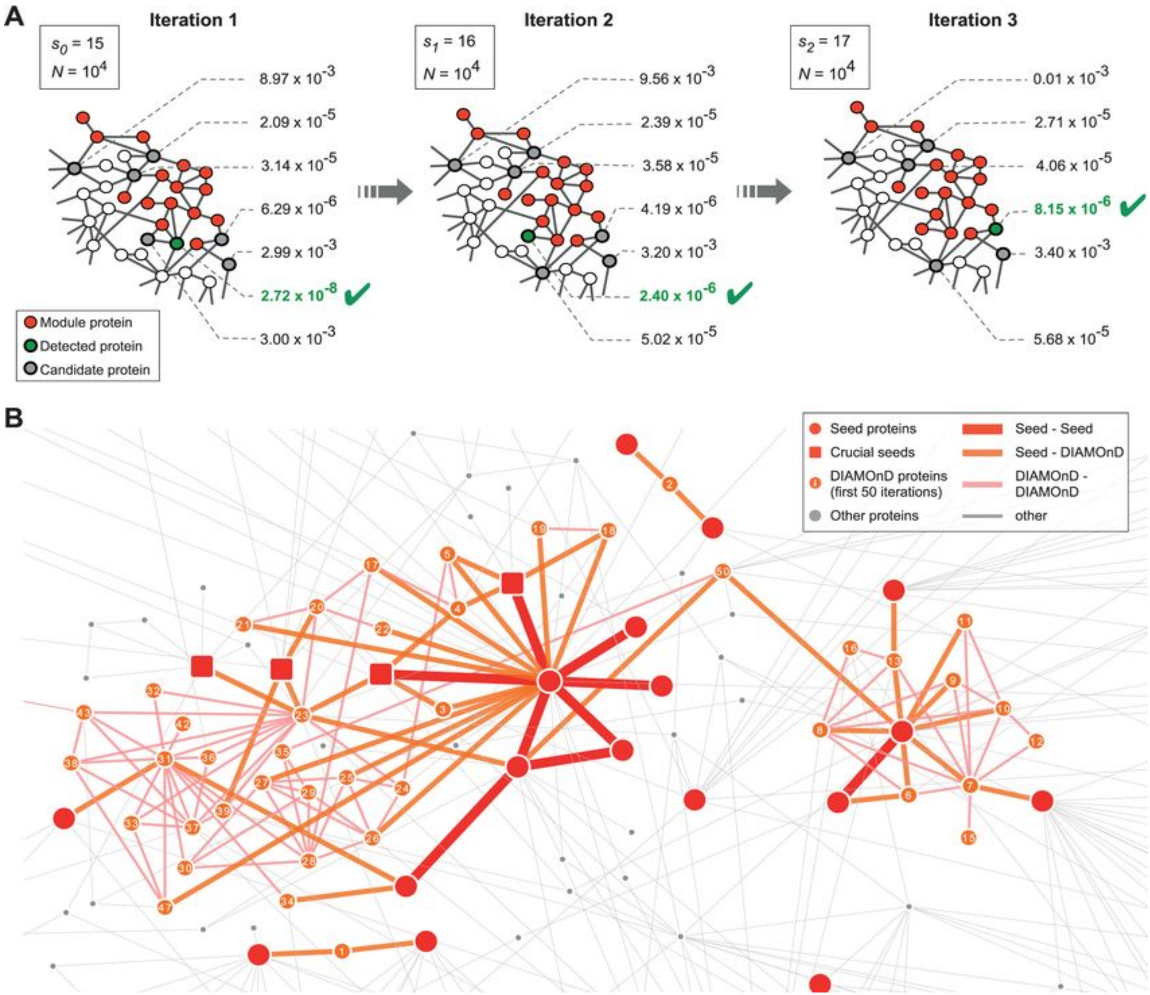
Double Indexing Approach:

- Database of seed-location pairs are built for query and reference sequences.
    - First index organizes and stores information about the sequences.
    - Second index is for querying sequences provided by the user.
- Matching seeds are paired through a hash join technique.

Seed-and-Extend Algorithm:

- Identifies exact matches between sequences from the indexed database and the query index.
- Once seeds are identified, they are mapped to find longer, high-scoring alignments.

Seed-and-Extend Algorithm



Comparison

Comparison of DIAMOND and Kraken2

DIAMOND

- Uses a protein database and can query both DNA or amino acid sequences.
- Query algorithm: alignment using spaced seeds
- Taxonomic binning strategy: Lowest common ancestor

Kraken2

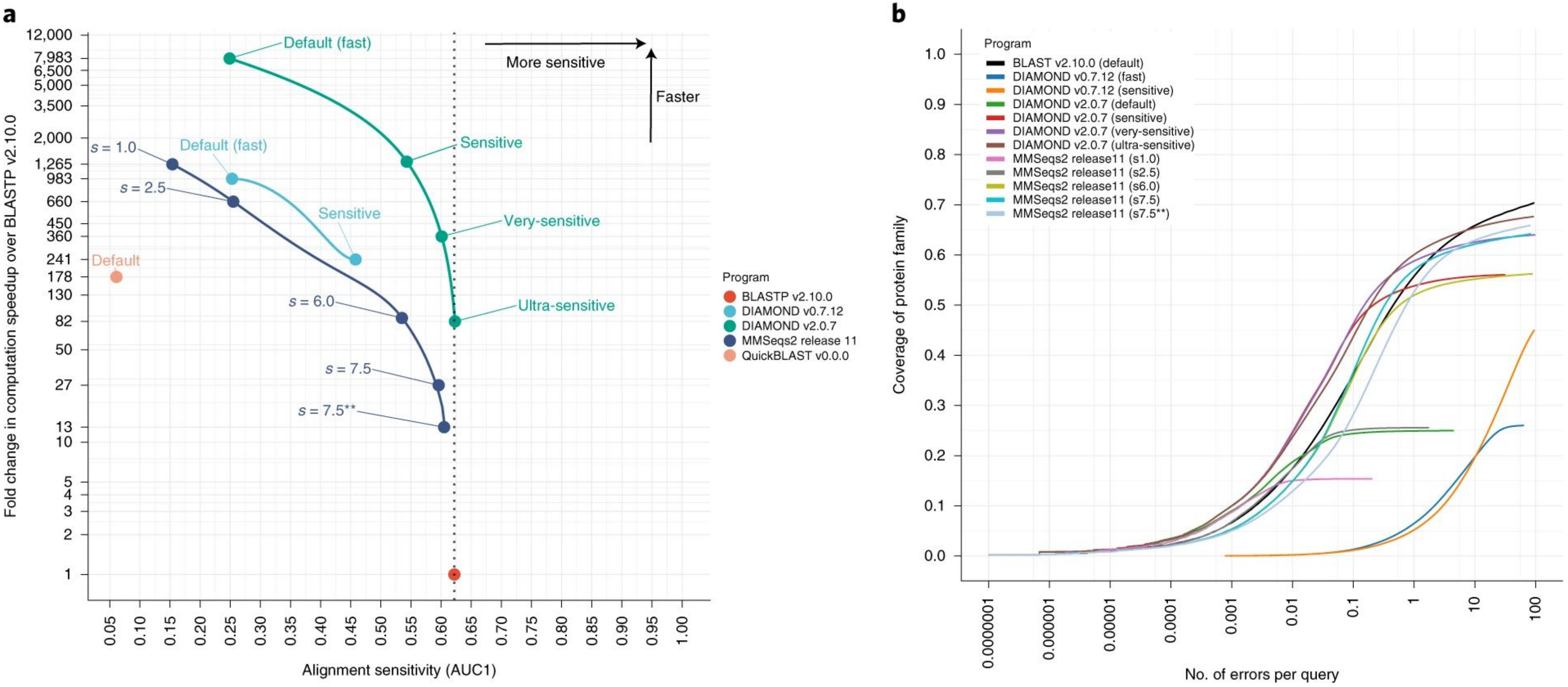
- Uses protein or DNA, whole genome or single locus databases
- Query algorithm: exact k-mer matching
- Taxonomic binning strategy utilizes the highest number of k-mer matches considering a root-to-leaf path

Comparison to Other Tools

- DIAMOND’s default mode is reported to be up to 20,000 times faster than BLASTX and reports about 80-90% of the matches that BLASTX would find.

- In sensitive mode, DIAMOND is about 2,500 times faster than BLASTX, finding more than 94% of all matches.

Comparison to Other Tools (Cont.)



Usage and Experiments

Usage

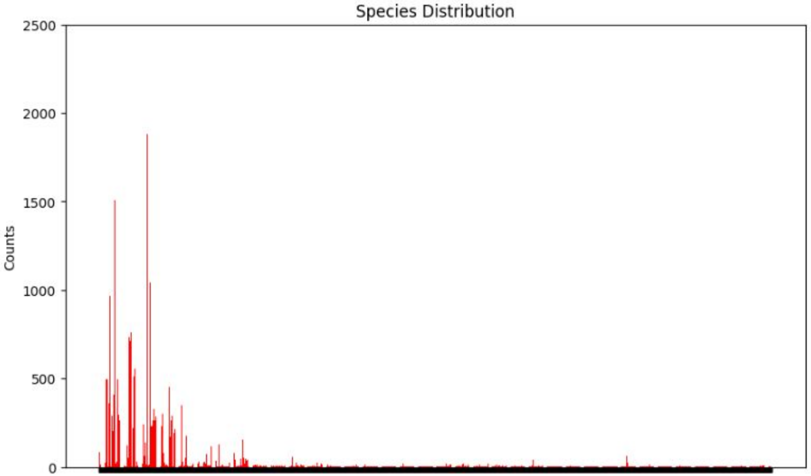
- singularity exec --bind /path/to/your/data:/data diamond_latest.sif /bin/bash
- diamond makedb --in /data/combined_uniprot_sprot_2_10.fasta.gz -d /data/sprotdb_2_10
- diamond blastp --ultra-sensitive -d sprotdb_2_10 -q /data/uniprot_sprot_part1.fasta.gz -o /data/output_file.m8 --threads 96

Dataset

- gene_refseq_uniprotkb_collab.gz
  - 176,513,537 accessions(ids)
  - no sequence
- uniprot_trembl.fasta.gz
  - 249,751,891 sequences
  - predicted data
- **uniprot_sprot.fasta.gz**
  - 570,830 sequences
  - curated data
- ETC
  - evol1.sorted.unmapped.R2.fastq.gz (17,692 sequences)
  - astral-scopedom-seqres-gd-all-2.08-2923-901-06.fa (1/11th, 27,776 sequences)

Dataset Analysis

- **uniprot_sprot.fasta.gz**
    - 570,830 sequences
    - curated data
    - 12,014 species
    - one sample species: 4,888
    - less than 10 samples species: 9,606



Run Diamond with sensitive settings

- default
- --sensitive
- --more-sensitive
- --ultra-sensitive

	time(s)	pairwise aligned(#)	Aligned queries(#)
default	12.555	4077	403
sensitive	65.239	5828	405
more	65.613	5828	405
ultra	64.898	5828	405

Query: evol1.sorted.unmapped.R2.fastq.gz
DB: uniprot_sprot

Run Diamond with sensitive settings

- default
- --sensitive
- --more-sensitive
- --ultra-sensitive

	time(s)	pairwise aligned(#)	Aligned queries(#)
default	4.226	5275	211
sensitive	22.264	5369	217
more	22.059	5369	217
ultra	69.202	5405	226

Query: 1/11th astral-scopedom-seqres-gd-all-2.08-2923-901-06.fa
DB: another 1/11th astral-scopedom-seqres-gd-all-2.08-2923-901-06.fa

Run Diamond with sensitive settings

- default
- --sensitive
- --more-sensitive
- --ultra-sensitive

	time(s)	pairwise aligned(#)	Aligned queries(#)
default	33.283	429380	30195
sensitive	152.994	625449	36799
more	189.506	638549	37119
ultra	907.452	676317	37963

Query: 1/10th uniprot_sprot
DB: another 9/10th uniprot_sprot

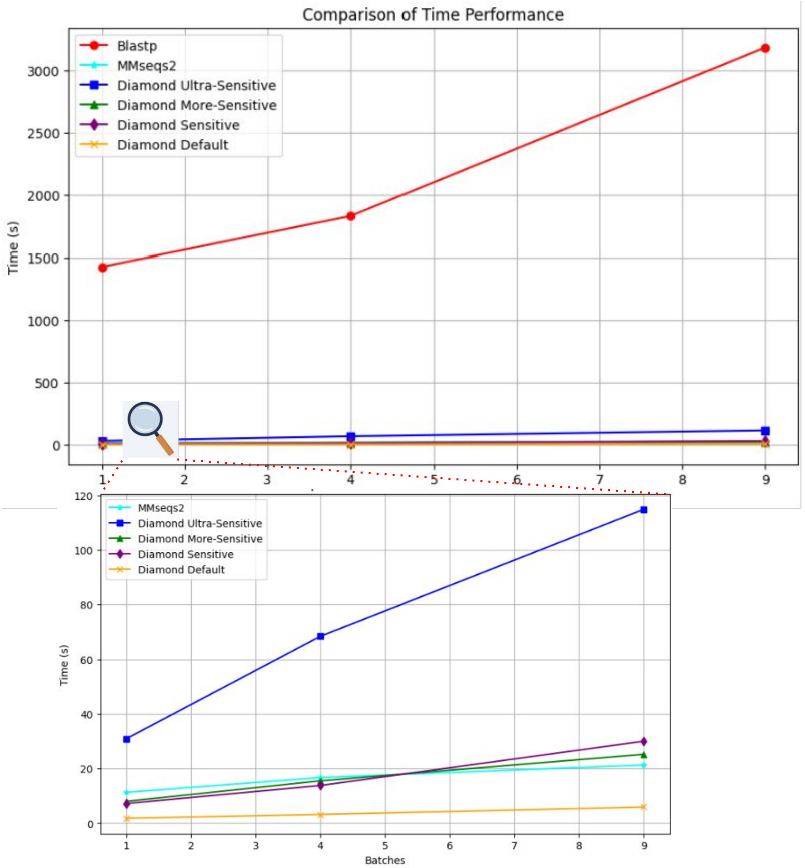
Comparison with Blast

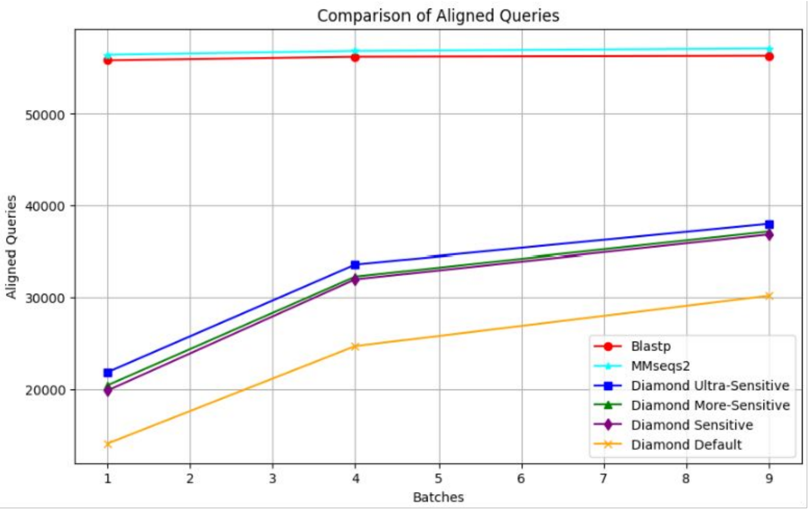
- Use 96 thread for blastp

	time(s)	pairwise aligned(#)	Aligned queries(#)
blastp	3181.1	9611230	56366
default	5.9	429380	30195
sensitive	30.0	625531	36799
more	25.2	638673	37123
ultra	114.9	676055	37962

Query: 1/10th uniprot_sprot
DB: another 9/10th uniprot_sprot

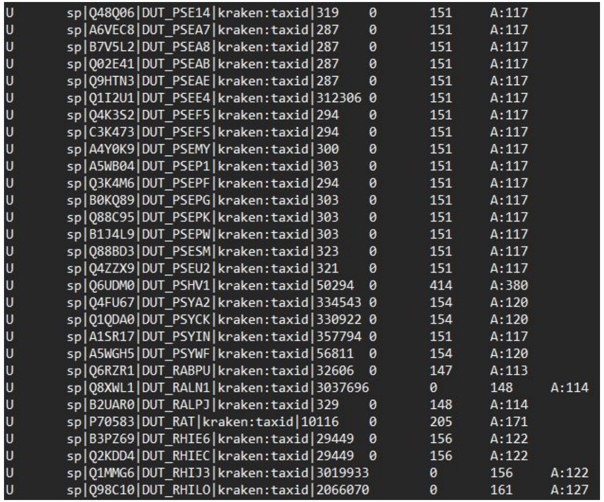
Incremental Comparison





Comparison with Kraken2

- Get texa for uniqrot_sprot file using ncbi api
    - Only able to get about 10%(57,083 seqs) of uniprot_sprot(about 8 hours to take)
- Build Kraken db with 9/10th of the file with texa
- Test 1/10th of the file(5,708 seqs) as a query
- Got unclassified result



```
U       sp|Q48Q06|DUT_PSE14|kraken:taxid|319    0       151     A:117
U       sp|A6VEC8|DUT_PSEA7|kraken:taxid|287    0       151     A:117
U       sp|B7V5L2|DUT_PSEA8|kraken:taxid|287    0       151     A:117
U       sp|Q02E41|DUT_PSEAB|kraken:taxid|287    0       151     A:117
U       sp|Q9HTN3|DUT_PSEAE|kraken:taxid|287    0       151     A:117
U       sp|Q1I2U1|DUT_PSEE4|kraken:taxid|312306 0       151     A:117
U       sp|Q4K3S2|DUT_PSEF5|kraken:taxid|294    0       151     A:117
U       sp|C3K473|DUT_PSEFS|kraken:taxid|294    0       151     A:117
U       sp|A4Y0K9|DUT_PSEMY|kraken:taxid|300    0       151     A:117
U       sp|A5WB04|DUT_PSEP1|kraken:taxid|303    0       151     A:117
U       sp|Q3K4M6|DUT_PSEPF|kraken:taxid|294    0       151     A:117
U       sp|B0KQ89|DUT_PSEPG|kraken:taxid|303    0       151     A:117
U       sp|Q88C95|DUT_PSEPK|kraken:taxid|303    0       151     A:117
U       sp|B1J4L9|DUT_PSEPW|kraken:taxid|303    0       151     A:117
U       sp|Q88BD3|DUT_PSESM|kraken:taxid|323    0       151     A:117
U       sp|Q4ZZX9|DUT_PSEU2|kraken:taxid|321    0       151     A:117
U       sp|Q6UDM0|DUT_PSHV1|kraken:taxid|50294  0       414     A:380
U       sp|Q4FU67|DUT_PSYA2|kraken:taxid|334543 0       154     A:120
U       sp|Q1QDA0|DUT_PSYCK|kraken:taxid|330922 0       154     A:120
U       sp|A1SR17|DUT_PSYIN|kraken:taxid|357794 0       151     A:117
U       sp|A5WGH5|DUT_PSYWF|kraken:taxid|56811  0       154     A:120
U       sp|Q6RZR1|DUT_RABPU|kraken:taxid|32606  0       147     A:113
U       sp|Q8XWL1|DUT_RALN1|kraken:taxid|3037696        0       148     A:114
U       sp|B2UAR0|DUT_RALPJ|kraken:taxid|329    0       148     A:114
U       sp|P70583|DUT_RAT|kraken:taxid|10116    0       205     A:171
U       sp|B3PZ69|DUT_RHIE6|kraken:taxid|29449  0       156     A:122
U       sp|Q2KDD4|DUT_RHIEC|kraken:taxid|29449  0       156     A:122
U       sp|Q1MMG6|DUT_RHIJ3|kraken:taxid|3019933        0       156     A:122
U       sp|Q98C10|DUT_RHILO|kraken:taxid|2066070        0       161     A:127
```

Resources

- About DIAMOND:

  Buchfink, B., Reuter, K., & Drost, H.-G. (2021, April 7). *Sensitive protein alignments at tree-of-life scale using Diamond*. Nature News. https://www.nature.com/articles/s41592-021-01101-x#Sec2

- DIAMOND Algorithm:

  Ghiassian, S. D., Menche, J., & Barabási, A.-L. (n.d.). *A disease module detection (diamond) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome*. PLOS Computational Biology. https://journals.plos.org/ploscompbiol/article?id=10.1371%2Fjournal.pcbi.1004120

- Comparison for DIAMOND and other tools:

  McIntyre, A.B.R., Ounit, R., Afshinnekoo, E. *et al.* Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biol* 18, 182 (2017). https://doi.org/10.1186/s13059-017-1299-7

Resources

- About source code and libraries:

  DerrickWood's Kraken 2 on GitHub: https://github.com/DerrickWood/kraken2

  Official NCBI BLAST+ Docker Image Documentation on GitHub: https://github.com/ncbi/blast_plus_docs

  soedinglab MMseqs2: Ultra-fast and sensitive sequence search and clustering suite on GitHub: https://github.com/soedinglab/MMseqs2

  DIAMOND: A sequence aligner for protein and translated DNA searches, designed for high performance analysis of big sequence data on GitHub by bbuchfink: https://github.com/bbuchfink/diamond