1 Introduction

DIAMOND is a BLAST-compatible local aligner for mapping translated DNA query sequences against a protein reference database (BLASTX alignment mode). The speedup over BLAST is up to 20,000 on short reads at a typical sensitivity of 90-99% relative to BLAST depending on the data and settings.

DIAMOND is actively supported and developed. Support inquiries, feedback and feature requests are welcome.

Contact: Benjamin Buchfink (developer)
Publication: Buchfink, B., Xie, C. & Huson, D.H. *Nature Methods* doi:10.1038/nmeth.3176 (2014).

Please cite the publication when using the program, both standalone and as part of a larger pipeline.

2 Installation

DIAMOND runs on the Intel 64/AMD64 hardware architecture and Unix-based operating systems. It is primarily designed for high memory server machines, but is able to run on a standard desktop computer with 4 GB of memory.

The program may be downloaded in Linux 64 bit binary format for immediate use. Alternatively, the program can be compiled from source (see 10).

3 Basic command line use

We assume to have a protein database file in FASTA format named `nr.faa` and a file of DNA reads that we want to align named `reads.fna`.

In order to set up a reference database for DIAMOND, the `makedb` command needs to be executed with the following command line:

```
$ diamond makedb --in nr.faa -d nr
```

This will create a binary DIAMOND database file with the specified name (`nr.dmnd`). The alignment task may then be initiated using the `blastx` command like this:

```
$ diamond blastx -d nr -q reads.fna -a reads -t <temporary directory>
```

The temporary directory should point to a fast local disk with a lot of free space. It is possible to omit this option, this will however increase the program's memory usage substantially.

The output file here is specified with the `-a` option and named `reads.daa`. It is generated in DAA (DIAMOND alignment archive) format. Other formats can be generated using the `view` command. For instance, the following command will generate BLAST tabular format from the DAA file and save it to disk:

```
$ diamond view -a reads.daa -o reads.m8
```

4 Commands

Commands are issued as the first parameter on the command line and set the task to be run by the program.

Command	Description
makedb	Create DIAMOND formatted reference database from a FASTA input file.
blastp	Align protein query sequences against a protein reference database.
blastx	Align translated DNA query sequences against a protein reference database.
view	Generate formatted output from DAA files.

5 General options

Option	Short	Default	Description
--db	-d		Path to DIAMOND database file (not including the file extension `.dmnd`).
--threads	-p	max	Number of CPU threads.

6 `makedb` options

Option	Short	Default	Description
--in			Path to protein reference database file in FASTA format (may be gzip compressed).
--block-size	-b	2	Block size in billions of sequence letters to be processed at a time. The default value of 2 is chosen for the program to run on a machine with 32

			GB of memory. When using a high-memory server, it is recommended to increase this number for better performance. A value of 0.4 will allow the program to run on a machine with 4 GB of memory.

7 Alignment options

Option	Short	Default	Description
--query	-q		Path to query input file in FASTA or FASTQ format (may be gzip compressed).
--daa	-a		Path to output file in DAA format (extension `.daa` will be appended).
--tmpdir	-t	/dev/shm	Directory to be used for temporary storage. This should point to a fast local disk with a lot of free space. Omitting this option will keep temporary information in memory and increase the program's memory usage.
--max-target-seqs	-k	25	The maximum number of target sequences per query to keep alignments for.
--top			Keep alignments within the given percentage range of the top alignment score for a query (overrides –max-target-seqs option).
--evalue	-e	0.001	Maximum expected value to keep an alignment.
--min-score		-	Minimum bit score to keep an alignment. Setting this option will override the --evalue parameter.
--sensitive			Trigger the sensitive alignment mode with a 16x9 seed shape configuration.
--band		auto	Dynamic programming band for seed extension. This corresponds to the maximum length of gaps that can be found in alignments.
--gapopen		11	Gap open penalty (see 11.1).
--gapextend		1	Gap extension penalty (see 11.1).
--matrix		BLOSUM62	Scoring matrix (see 11.1).
--seg		yes	Enable SEG masking of low complexity segments in the query (yes/no).
--index-chunks	-c	4	The number of chunks for processing the seed index. Bigger numbers will reduce memory usage but also performance. This value may be set to 1 for maximum performance, but should not be set higher than the default.

8 View options

Option	Short	Default	Description
--daa	-a		Path to input file in DAA format.
--out	-o	stdout	Path to output file.
--outfmt	-f	tab	Format of output file. tab = BLAST tabular format sam = SAM format
--compress		0	Compression for output file (0=none, 1=gzip).

9 FAQ

DIAMOND is slower than claimed in the paper, even slower than BLAST.
The DIAMOND algorithm is designed for the alignment of large datasets. The algorithm is not efficient for a small number of query sequences or only a single one of them, and speed will be low. BLAST is recommend for small datasets.

Can several copies of DIAMOND be run in parallel?
It is possible, but not recommended. The algorithm is more efficient if you allocate more memory to a single task. If you need to process several files, performance will be better if you run DIAMOND on them sequentially.

Reads imported into MEGAN lack taxonomic or functional assignment.
MEGAN requires mapping files which need to be downloaded separately at the MEGAN website and configured to be used.

10 Compiling from source

The Boost libraries (version 1.53.0 or higher) are required for compilation. It is recommended to have Boost installed by your system administrator prior to installing DIAMOND. Alternatively, the package includes a script called install-boost which will download and install a local copy of Boost for the user.

To compile DIAMOND from source, invoke the following commands on the shell:

```
$ tar xzf diamond.tar.gz
$ cd diamond
$ ./configure
$ make
$ make install
```

Alternatively, for having a local copy of Boost installed as well:

```
$ tar xzf diamond.tar.gz
$ cd diamond
$ ./install-boost
$ ./configure --with-boost=boost
$ make
$ make install
```

This will install the DIAMOND binary to /usr/local/bin and requires write permission to that directory. You may also pass --prefix=DIR to the configure script to choose a different installation directory.

11 Appendix

11.1 Scoring matrices

Matrix	Supported values for (gap open)/(gap extend)
BLOSUM45	(10-13)/3; (12-16)/2; (16-19)/1
BLOSUM50	(9-13)/3; (12-16)/2; (15-19)/1
BLOSUM62	(6-11)/2; (9-13)/1
BLOSUM80	(6-9)/2; 13/2; 25/2; (9-11)/1
BLOSUM90	(6-9)/2; (9-11)/1
PAM250	(11-15)/3; (13-17)/2; (17-21)/1
PAM70	(6-8)/2; (9-11)/1
PAM30	(5-7)/2; (8-10)/1