DIAMOND user manual

Benjamin Buchfink
buchfink@gmail.com

October 2014

1 Introduction

diamond is a local aligner for mapping DNA query sequences against a protein reference database (blastx alignment mode). Query sequences are accepted in FASTA format and translated in all reading frames with low complexity segments being masked using the BLAST SEG algorithm. Significant matches with a bit score above 50 are written to disk in the BLAST tabular output format.

diamond is primarily designed to run on high memory server machines, but is able to run on computers with as little as 4 GB of memory. Currently Unix-based operating systems and Intel/AMD-compatible hardware architectures are supported.

2 Installation

diamond is deployed as C++ source code. Compiling diamond from source is currently only supported on Unix-based operating systems.

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 use the binary created in the source directory or pass `--prefix=DIR` to the `configure` script to choose a different installation directory.

3 Basic command line use

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 <input file> --db <database file> --threads <CPU threads>
```

This will create a binary `diamond` database file with the specified name and the appended extension `.dmnd`. The input file has to contain protein sequences in FASTA format.

An alignment task may then be initiated using the `blastx` command like this:

```
diamond blastx --db <database file> --query <query file> --out <output file>
--threads <CPU threads> --tmpdir <temporary directory>
```

The query file must contain DNA sequences in FASTA format. 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 substantially increase the program's memory usage.

4 Memory usage

The program has three parameters that control its memory usage, which should be adjusted to the target machine for optimal performance.

`--tmpdir` *<directory>* This directory will be used for temporary storage. It 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.

`--block-size, -b` **This is an option to the makedb command.** It sets the 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.

`--index-chunks, -c` The number of chunks for processing the seed index. Higher numbers will reduce memory usage but also performance. The default value is 4. It is not recommended to increase this value further, but instead use the `--block-size` option if memory usage is too high. This value may be set to 1 for maximum performance.

5 Advanced options

`--max-target-seqs, -k` The maximum number of target sequences per query to report alignments for. Default is 25.

`--min-score` Minimum bit score to report a match. The default is 50.

`--sensitive` Trigger the sensitive alignment mode with a 16x9 seed shape configuration.

`--band` Dynamic programming band for seed extension. This is automatically set to 15% of the respective query sequence length, but may be increased for more accurate alignments of longer sequences.

6 Output

Output files are written in the default BLAST tabular format (option `-outfmt 6` of BLAST).

Alignments are scored using the BLOSUM62 matrix with a gap existence penalty of 11 and gap extension penalty of 1.

Diamond uses the classical formula for the expected value given in Karlin & Altschul, 1990. More recent BLAST versions use a more elaborate computation, so the reported e-values will not be identical to BLAST.

BLAST also uses composition based statistics for its scores by default, which is an advanced computation that slightly modifies alignment raw scores and bit scores. Bit scores reported by `diamond` are identical with BLAST scores when composition based statistics are disabled (option `-comp_based_stats 0` of BLAST).