PEANUT Documentation

Release 1.3.6

Johannes Koester

July 16, 2015

Contents

PEANUT is a read mapper for DNA or RNA sequence reads. Read mapping is the process of aligning biological DNA or RNA sequencing reads to a known reference genome.

By exploiting the massive parallelism of modern graphics processors and a novel index datastructure (the q-group index), PEANUT achieves supreme speed compared to current state of the art read mappers like BWA MEM, Bowtie2 and RazerS3 while maintaining their accuracy. PEANUT thereby allows to report both only the best hit or all hits of a read.

When using PEANUT, please cite our article:

> Köster J, Rahmann S (2014). Massively parallel read mapping on GPUs with the q-group index and PEANUT. PeerJ 2:e606 http://dx.doi.org/10.7717/peerj.606

CHAPTER 1

Requirements

- A POSIX compatible OS (e.g. Linux, MacOS X, FreeBSD)
- an NVIDIA GPU with up-to-date proprietary drivers and at least 1.5GB RAM (this may change in the future as the AMD drivers become more mature)
- Python >= 3.2
- Numpy >= 1.7
- Cython >= 0.19
- PyOpenCL >= 2013.1

CHAPTER 2

Installation

If not already present, you will need the proprietary NVIDIA drivers, Python3, setuptools, Cython, Numpy and PyOpenCL. On Ubuntu or Debian based systems, the NVIDIA drivers can be installed with:

```
$ sudo apt-get install nvidia-current
```

which requires admin rights. With admin rights, you should install setuptools, Cython and Numpy via:

```
$ sudo apt-get install python3-setuptools python3-numpy cython
```

Without admin rights, you can use a userspace Python 3 distribution like https://store.continuum.io/cshop/anaconda.

You can install PEANUT using the easy_install3 tool provided by setuptools. All missing dependencies will be installed automatically:

```
$ easy_install3 --user peanut
```

When installing manually from setup.py, just execute:

```
$ python3 setup.py install --user
```

To update PEANUT, issue:

```
$ easy_install3 -U peanut --user
```

CHAPTER 3

Usage

PEANUT will be available as a command line tool. To index a reference genome *genome.fasta*, issue the following:

```
$ peanut index genome.fasta genome.index.hdf5
```

To map paired end reads *reads.1.fastq* and *reads.2.fastq* onto the indexed reference, use the following invocation:

```
$ peanut map --threads 8 --insert-size 200 --insert-size-error 50 genome.index.hdf5 reads.1.fastq rea
```

Here, an insert size or 200 with a tolerance of 50 is expected. Defining the expected insert size is required for PEANUT to be able to detect properly paired reads. Setting the wrong insert size here can lead to reduced performance since PEANUT will try to rescue reads not properly paired by performing additional alignments. As can be seen, PEANUT outputs hits in the SAM format. Hence, output has to be piped into samtools to obtain a BAM file.

Per default, PEANUT reports the best and all equally good hits of a read. Alternatively, it can be configured to report a desired number of strata of equally good hits:

```
$ peanut map --strata <N> ...
```

Here, N is the number of desired strata, with N=all telling PEANUT to report all hits of a read down to a given error tolerance.

PEANUT buffers reads and hits in GPU and CPU memory. The default buffer settings of PEANUT are optimized for a GPU with at least 1.5 GB memory and a CPU with 16 GB memory (but 8 GB should do, too). You can lower both buffer sizes to adapt for weaker systems, e.g.:

```
$ peanut map --read-buffer 100000 --hits-buffer 500000 ...
```

This, however, can reduce performance since the amount of possible parallelism on the GPU is affected. For further help, invoke:

```
$ peanut --help
```

or visit http://peanut.readthedocs.org.

CHAPTER 4

FAQ

The following questions might be of general interest.

- Which resource requirements does PEANUT have?

  With default settings, it needs 16GB RAM for the CPU and a decent NVIDIA GPU with 1.5GB RAM.

- How many reads are mapped in one step?

  Per default, PEANUT maps one million reads per step. In case of paired end, half a million from each end are mapped. This influences the amount of memory used, and can be regulated as shown above.

- How do you decide which hit is the right one to report?

  The hit with the highest percent identity to the reference is reported as the best hit. With paired-end reads, percent identities of properly paired hits are summed for this decision. If no hit is properly paired, PEANUT tries to rescue the pair for the best hit by performing an alignment within the given expected insert size. When reporting more than one hit, they are sorted into equally scoring strata and the given number of strata is reported (see above).

- I would like to modify PEANUT. How should I start?

  You can download the source or checkout from Git (see above). Then modify anything you want, and issue the following, instructing Python to automatically rebuild and install everything that was changed:

  ```
  $ python3 setup.py install
  ```

CHAPTER 5

News

25 Apr 2015	Release 1.3.6 of PEANUT. Allow to choose device type from command line. This should enable PEANUT to run on other OpenCL devices. The nature of the algorithm suits best to GPUs, though.
9 Nov 2014	Release 1.3.5 of PEANUT. Added missing .pyx and .pxd files to source tarball distributed via Pypi.
5 Sep 2014	Release 1.3.2 of PEANUT. Fixed a bug causing an invalid OpenCL work group size with uneven number of reads (special thanks to Sean Li for reporting this).
18 Aug 2014	Release 1.3.1 of PEANUT. Added rescue mode for paired-end sequencing.
7 Jul 2014	Release 1.2 of PEANUT. Improved mapping quality estimate that reflects the original posterior probability like defined in the MAQ paper of Heng Li.
26 May 2014	Release 1.1 of PEANUT. Reduced memory usage (at most 1/2 if you are lucky).
16 May 2014	Release 1.0.3 of PEANUT. Changed the argument order for the map subcommand to agree with other mappers.
7 May 2014	Release 1.0.2 of PEANUT. More fixes for alignment selection in case of paired end reads. Fixed missing import and –query-buffer not being considered.
30 Apr 2014	Release 1.0.1 of PEANUT. Improved flag usage in SAM output. Rescaled mapping quality in accordance with the paper. Fixed rare cases of where the wrong mate alignment was chosen as the best alignment.

5.1 License

PEANUT is available under the `MIT license`.

5.2 Analysis Pipeline

The pipeline used for the analysis done in the paper can be obtained `here`.

5.3 Author

Johannes Köster

Genome Informatics, Institute of Human Genetics, Faculty of Medicine, University of Duisburg-Essen

johannes.koester@gmail.com

http://johanneskoester.bitbucket.org