ParDRe

Reference Manual

Authors:
Jorge González-Domínguez
Bertil Schmidt

Institution:
Parallel and Distributed Architectures Group
Institute of Computer Science
Johannes Gutenberg University Mainz, Germany

Date:
June 8, 2017

Contents

1 Introduction **2**

2 Installation **3**

3 Execution **4**

Introduction

ParDRe is a parallel tool to remove duplicate reads. Duplicate reads can be seen as identical or nearly identical sequences with some mismatches. This tool will let the users to avoid the analysis of not necessary reads, reducing the time of subsequent procedures with the dataset (e.g., assemblies, mappings, etc.).

The tool is implemented with MPI in order to exploit the parallel capabilities of multicore clusters. It is faster than multithreaded counterparts (end of 2015) for the same number of cores and, thanks to the message-passing technology, it can be executed on clusters.

The *ParDRe* tool is developed by the Parallel and Distributed Architectures Group at the Johannes Gutenberg University in Mainz. The tool is distributed as free software and publicly available under the GPLv3 license at:

`http://sourceforge.net/projects/pardre`

The corresponding license file is shipped with the software but can also be accessed via:

`http://www.gnu.org/licenses/gpl-3.0.en.html`

If you want to reuse the code, please ensure compliance to the aforementioned license and a proper attribution/citation of the original work/authors.

Installation

To complete the installation of *ParDRe* follow these steps:

1. Untar the archive and move into the *ParDRe* directory.
2. Update the file *Makefile* of the root directory in order to indicate the correct path and libraries for the MPI compiler installed in your system.
3. Type `make` to build *ParDRe*.

Execution

ParDRe can be executed with any MPI running command (e.g., `mpirun`, `mpiexec`). The only compulsory argument is the input dataset (or two datasets in case of paired-end execution). It works with `fasta` and `fastq` formats. The possible arguments for the program are:

- *-i*. Compulsory. String with the path to the sequence file in FASTA/FASTQ format.
- *-p*. Compulsory in paired-end scenarios. String with the path to the second sequence file in FASTA/FASTQ format for paired-end scenarios.
- *-z*. Optional. To specify whether the input and output are compressed with .gz extension. If the value is higher than 0, it indicates the compression level. Its by default value is 0.
- *-o*. Optional. String with the path to the output sequence file in FASTA/FASTQ format. Its by-default value is the same as the input file followed by `NonDup`.
- *-r*. Optional. String with the path to the second output sequence file in FASTA/FASTQ format for paired-end scenarios. Its by-default value is the same as the input file followed by `NonDup`.
- *-m*. Optional. Integer with the number of allowed mismatches to identify two reads as equivalent. Its by-default value is 0.
- *-l*. Optional. Integer with the prefix length for computation. *ParDRe* organizes the reads in clusters according to their first *-l* characters. Then, it compares the reads of the same cluster. Higher prefix-length usually leads to shorter computation but can miss some duplicates. The loss of accuracy is observed when removing near-duplicated reads. Reads in the same cluster have exactly the same prefix, i.e., mismatches are not allowed in the prefix. The longer the prefix, more near-duplicated can be missed. Lets use as example an scenario where we try to compress near-duplicated reads with up to one mismatch. If we use a prefix of length 20 to compare two reads that only have one mismatch in position 16, *ParDRe* stores them in different clusters, they are never compared, and both reads will be in the output. Otherwise, with a prefix of length 15 *ParDRe* compares them and, as only one base is different, discards one of them. Its by-default value is 20.
- *-c*. Optional. Integer with the number of bases to compare for each sequence (starting from the beginning), default is equal to the sequence length (all bases are compared). It must be equal or higher to the prefix length.
- *-d*. Optional. ParDRe also supports the option of only removing optical duplicates. In this case this parameter is an integer that indicates the Euclidean distance used to identify which duplicates are optical. If this parameter is not specified, ParDRe will remove all duplicates or near-duplicates, either optical or PCR.

- *-b*. Optional. The computation is divided by blocks. Integer with the number of reads that are read in each block. Its by-default value is 50000.
- *-t*. Optional. Integer with the number of threads per MPI process. The best configuration process/threads depends on the characteristics of the input datasets and the machine. In general, it is not advisable to use many threads per process. Its by-default value is 1.

For instance, the following command removes the duplicates of identical reads (no mismatches) of a single-end dataset on 8 cores using one MPI process per core and prefix length of 15.

```
mpirun -n 8 ./ParDRe -i dataset1.fastq -l 15
```

The following command shows a similar example but for paired-end computing, allowing two mismatches and using two threads per process with prefix length of 20:

```
mpirun -n 4 ./ParDRe -i dataset1.fastq -p dataset2.fastq -m 2 -t 2
```