Pamir Manual

Pinar Kavak

November 10, 2016

Contents

1 Getting Started

1.1 Installation

Pamir can be obtained from `https://bitbucket.org/compbio/pamir`

1.1.1 Prerequisite

Pamir relies on specific version of the following tools:

- g++ 4.8.2 or newer version
  - To use `Set Cover` strategy to resolve multi-mappings, boost library and g++ >= 4.9.0 are required for for lambda expressions and other features from C++14.
  - For Set Cover you also need to type
    export BOOST_INCLUDE= the/BOOST/version/include/ (directory of BOOST in your machine).
- Python 2.7 or newer version (for the package `argparse`)
- mrsFAST version 3.3.11 or newer.

1.1.2 Details and Troubleshooting

In most of cases, type

```
git clone --recursive git@bitbucket.org:compbio/pamir.git
cd pamir && make
```

In case you only want minimal functions of Pamir without installing newer version of g++ and boost library, just type

```
make -Bj basic
```

1.2 Running Pamir

Pamir (Insertion Discovery tool for Whole Genome Sequencing Data) detects novel sequence insertions based on one-end anchors (OEA) and orphans from paired-end Whole Genome Sequencing (WGS) reads.

Note that reference genome is required for running Pamir in addition to sequencing or mapping data.

1.2.1 Project Name

To run Pamir you have to specify a project name such that Pamir will create a folder to store the results and intermediate files. You need to specify project name by `-p`.

1.2.2 Data Preparation

Required Information. Two information are required for running Pamir :

1. Reference Genome: You need to provide the reference genome in single fasta file by specifying the parameters `-r` or `--reference`.
2. Masking File: You can provide a file for masking reference genome. For example, you can ask Pamir to ignore events in repeat regions by giving `-m repeat.mask` . When you only want to consider events in genic regions, use `-m genic.region --invert-masker` and Pamir will mask those regions not in the given file.

Read Length. Now Pamir only accepts WGS datasets in which two mates of all reads are of equal length.

1.2.3 Sequencing Data

Pamir can take either FASTQ and SAM files as its input. It has three different options to accept inputs:

- **SAM/by mrsFAST-best-search**: A paired-end mapping result of your WGS data which satisfies the following conditions:
  - Two mates from a read are grouped together.
  - All mates are of equal length.

  For example, a *best-mapping* SAM be a valid input file for Pamir . You can specify by `--files mrsfast-best-search=wgs.sam`. You can give multiple best-mapping files too, by comma separated or just the folder directory that includes the inputs. You can specify by
  `--files mrsfast-best-search=sample1.sam,sample2.sam,sample3.sam` or
  `--files mrsfast-best-search=directory/to/sample_best_mapping_sam_files/`
- **FASTQ**: Pamir also accepts FASTQ format as the input data once it is a single gzipped file such that two (equal-length) mates of a read locate consecutively. You can specify by giving `--files fastq=wgs.fastq.gz`.
- **Alignment file SAM/BAM**: Pamir also accepts any other alignment output sorted by readname. Alignment output can be in SAM or BAM format. You can specify by `--files alignment=wgs.sam or --files alignment=wgs.bam`.

1.2.4 MrsFAST Parameters

Pamir uses mrsFAST for multi-mapping the orphan and OEA reads obtained from the best-mapping output. You can give your own mrsFAST parameters or Pamir will use the default values. Some of the parameters you may want to update are :

- –**mrsfast-n**: Maximum number of mapping loci of anchor of an OEA. Anchor with higher mapping location will be ignored. 0 for considering all mapping locations. (Default = 50)
- –**mrsfast-threads**: Number of the threads used by mrsFAST-Ultra for mapping. (Default = 1)
- –**mrsfast-errors**: Number of the errors allowed by mrsFAST-Ultra for mapping. In default mode Pamir does not give any error number to mrsFAST-Ultra, in which case it calculates the error value as 0.06 x readlength. (Default = -1)
- –**mrsfast-index-ws**: Window size used by mrsFAST-Ultra for indexing the reference genome. (Default = 12)

1.2.5 Other Parameters User can Define

- –**num-worker**: Number of independent prediction jobs to be created. You can define this parameter according to your core number. (Default = 1)
- –**resume**: Restart pipeline of an existing project from the stage that has not been completed yet.
- –**assembler**: The assembler to be used in orphan assembly stage (Options: velvet, minia, sga. Default = velvet).

1.3 Results

Pamir generates a VCF file for detected novel sequence insertions. You can run genotyping for each sample after obtaining the VCF file by:

python **genotyping.py** projectFolder/aftersetcover_PASS.sorted reference.fa.masked sample1_FASTQ_1.fq sample1_FASTQ_2.fq readlength mrsfast-min mrsfast-max projectFolderDirectory

1.4 Example Commands

- To start a new analysis from a mrsfast-best mapping result SAM file:

  $ ./pamir.py -p my_project -r ref.fa --files mrsfast-best-search=sample.sam

- To make a pooled-run with multiple samples separated by comma:

  $ ./pamir.py -p my_project -r ref.fa --files mrsfast-best-search=sample.sam,sample2.sam,sample3.sam

- To make a pooled-run with multiple samples which are in a folder called SAMPLEFOLDER:

  $ ./pamir.py -p my_project -r ref.fa --files mrsfast-best-search=SAMPLEFOLDER

- To start from another mapping tool's alignment result SAM/BAM file:

  $ ./pamir.py -p my_project -r ref.fa --files alignment=sample.bam

- To start from a gzipped fastq file,

  $ ./pamir.py -p my_project -r ref.fa --files fastq=sample.fastq.gz

- To ignore regions in a mask file (e.g., repeat regions),

  $ ./pamir.py -p my_project -r ref.fa -m repeat.txt --files mrsfast-best-search=sample.sam

- To analyze events only in some regions of the reference genome (e.g., genic regions),

  $ ./pamir.py -p my_project -r ref.fa -m genic.region --invert-mask --files mrsfast-best-search=sample.sam

- To make sure that mrsFAST will not report the mapping locations of an OEA read more after the 30th location:

  $ ./mistrvar.py -p my_project -r ref.fa --mrsfast-n 30 --files mrsfast-best-search=sample.sam

- To specify the core number for mrsFAST during multi-mapping of OEAs:

  $ ./pamir.py -p my_project -r ref.fa --mrsfast-threads 8 --files mrsfast-best-search=sample.sam

- To speed up the prediction process by defining the independent prediction jobs according to available core numbers:

  $ ./pamir.py -p my_project -r ref.fa --num-worker 20 --files mrsfast-best-search=sample.sam

- To specify the assembler as sga for orphan assembly and also the number of prediction jobs will be 20:

  $ ./pamir.py -p my_project -r ref.fa --num-worker 20 --assembler sga --files mrsfast-best-search=sample.sam

- To resume from the previously finished stage:

  $ ./pamir.py -p my_project --resume

1.5 Example Invalid Commands

The following commands do not satisfy requirements of Pamir and will fail pamir.py:

- Project name is missing:

  $./pamir.py -r ref.fa --files alignment=sample.sam

- Reference genome file is missing:

  $./pamir.py -p my_project --files alignment=sample.sam

- Incorrect path of the mask file:

  $./pamir.py -p my_project -m non-exist-mask-file --files alignment=sample.sam

- No input sequencing files:

  $./pamir.py -p my_project -r ref.fa

- Multiple sequencing sources:

  $./pamir.py -p my_project -r ref.fa --files mrsfast-best-search=sample.sam fastq=sample2.fastq.gz