The `Geno-PanPipe` **Software Package**

Daniel Ortiz Martínez

Mathematics and Computer Science Department

University of Barcelona

Table of Contents

Introduction

- Execution of `bam` file pipelines entails many difficulties:
    - Downloading of very large files
    - Combination of tools with different input requirements
    - Existence of dependencies between tools
    - Tools may need to be added or removed
    - Each tool has specific computational requirements
    - Pipeline may need to be executed for hundreds of files
    - Parallelism should be exploited when possible
    - ...

- `Geno-PanPipe` provides a `PanPipe` module as well as a set of utilities to tackle these problems

Package Overview

Package Dependencies

- Shell Bash

- Python

- Conda

- PanPipe (https://daormar.github.io/panpipe/)

- Database download clients

- Slurm Workload Manager (optional)

Package Installation

- Obtain the package using git:

```
git clone https://daormar.github.io/geno-panpipe/
```

- Change to the directory with the package's source code and type:

```
./reconf
./configure --with-panpipe=<DIR>
make
make install
```

NOTE 1: argument of `--with-panpipe` option is used to indicate the directory where PanPipe was built

NOTE 2: use `--prefix` option of `configure` to install the package in a custom directory

- `--with-panpipe=<DIR>`: sets location `PanPipe` of package

- `--with-icgcscor=<DIR>`: sets location of ICGC's score client

- `--with-aspera=<DIR>`: enables Aspera Connect download client

- `--with-egadecrypt=<DIR>`: location of EGA decryptor tool

Functionality

- Execution of pipelines processing normal-tumor bam files

- Automate processing of all of the samples of a dataset

- Handle file downloading as part of pipeline execution

- Databases
    - EGA
    - ICGC

- Download clients
    - `aspc`
    - `score-client`
        - Amazon cloud
        - Collaboratory cloud
    - `pyega3`

Implemented Analysis Steps

- `bam` **file downloading**:
    - `download_ega_{norm|tum}_bam`
    - `download_ega_asp_{norm|tum}_bam`
    - `download_aws_{norm|tum}_bam`
    - `download_collab_{norm|tum}_bam`
- `bam` **file manipulation**:
    - `sort_{norm|tum}_bam`
    - `index_{norm|tum}_bam`
    - `delete_bam_files`

- **Small Indels and Single Nucleotide Variant Callers**:
  - `manta_germline`
  - `manta_somatic`
  - `platypus_germline`
  - `strelka_germline`
  - `strelka_somatic`

- **Copy Number Variant Callers**:
    - cnvkit
    - facets
    - sequenza
    - wisecondorx

Implemented Analysis Steps

- **Structural Variant Callers**:
  - `delly, parallel_delly`
  - `lumpy, parallel_lumpy`
  - `parallel_svtyper`

- **MSI Analyzers**:
  - `msisensor`

Main Tools and File Formats

- `query_ega_metadata`

- `query_icgc_metadata`

- `analyze_dataset`

- Extracts information from EGA metadata

- Main input parameters:
  - `-s <string>`: file with sample information
  - `-a <string>`: file with analysis information
  - `-t <string>`: file with study information
  - `-p <string>`: file listing Aspera box content
  - `-f <int>`: output format

- Extracts information from ICGC metadata

- Main input parameters:
  - -d <string>: file with donor information
  - -a <string>: file with aws manifest
  - -t <string>: table file in json format
  - -f <int>: output format:

- Uses metadata information to automate analysis of a whole dataset

- Main input parameters:
    - -pfile <string>: file with pipeline steps to be performed
    - -r <string>: file with reference genome
    - -m <string>: file with metadata, one entry per line

- Implements a `PanPipe` module for analyzing `bam` files

- Functions can be classified in 3 groups:
    - Download of `bam` files
    - Manipulation of `bam` files
    - Bioinformatics analysis (SNV, CNV and SV callers, MSI analyzers)

- Reference genome operations:
  - `filter_contig_from_genref`
  - `gen_bed_for_genome`

- Data preparation for analysis steps:
  - `convert_snppos_to_snpgcc`
  - `create_snv_pos_ascat`
  - `gen_wisecondorx_ref`

- **EGA/ICGC metadata**: information regarding a whole dataset that is typically spread out in a set of files

- **Analysis metadata**: file providing all the information of a given dataset that is relevant to automate its analysis

- **Analysis automation script**: file with a sequence of commands automating the analysis of a dataset

- Sample information (`Sample_File.map`)
  - contains file name info

- Analysis information (`Analisys_Sample_meta_info.map`)
  - contains donor and phenotype information

- Study information (`Study_analysis_sample.map`)
  - contains EGA id information

- Aspera box content (`dbox_content`)

- Donor information (`donor.<study_name>.tsv`)
  - contains gender information

- AWS manifest (`manifest.aws-virginia.<code>.tsv`)
  - contains object id, file name and donor id information

- JSON table file (`icgc_table.json`)
  - contains phenotype information

- Created with the query_ega_metadata tool

- Example entries:

```
EGAF00001664282 phenotype=Blood|Normal_blood gender=male ; EGAF00001664327 phenotype=Skin|
      Tumour_metastasis_to_local_lymph_node gender=male

EGAF00001670586 phenotype=Blood|Normal_blood gender=male ; EGAF00001664289 phenotype=Skin|
      Tumour_metastasis_to_local_lymph_node gender=male

EGAF00001664356 phenotype=Skin|Tumour_metastasis_to_distant_location gender=male ; EGAF00001670533
      phenotype=Blood|Normal_blood gender=male

EGAF00001661882 phenotype=Blood|Normal_blood gender=male ; EGAF00001661538 phenotype=Skin|
      Tumour_metastasis_to_local_lymph_node gender=male
...
```

Analysis Metadata (EGA Aspera)

- Created with the `query_ega_metadata` tool

- Example entries:

```
EGAD00001003388/PART_2/EGAZ00001300436_20170516_AWS_MELA_3c3ed66c-1505-4614-ac9d-575a6713b06a.bam.crypt
    phenotype=Blood|Normal_blood gender=male ; EGAD00001003388/PART_3/
    EGAZ00001300354_20170516_AWS_MELA_daf1ffd8-0a0f-4869-abc8-5be0b4fc1a21.bam.crypt phenotype=Skin|
    Tumour_metastasis_to_local_lymph_node gender=male

EGAD00001003388/PART_3/EGAZ00001303407_20170516_AWS_MELA_a197619e-f3e2-41f6-aef7-d1fadf3c1f5b.bam.crypt
    phenotype=Blood|Normal_blood gender=male ; EGAD00001003388/PART_2/
    EGAZ00001300389_20170516_AWS_MELA_3a9bf676-1a7b-4718-8396-fb36cc89b688.bam.crypt phenotype=Skin|
    Tumour_metastasis_to_local_lymph_node gender=male

EGAD00001003388/PART_3/EGAZ00001300416_20170516_AWS_MELA_f64eba46-d8a1-46f2-ba66-1b509e16c946.bam.crypt
    phenotype=Skin|Tumour_metastasis_to_distant_location gender=male ; EGAD00001003388/PART_3/
    EGAZ00001303394_20170516_AWS_MELA_7bb66858-7533-4f96-9cd4-41aae2fe18b2.bam.crypt phenotype=Blood|
    Normal_blood gender=male
...
```

- Created with the `query_icgc_metadata` tool

- Example entries:

```
34fa2369-424f-5886-9d23-6d19f8f15278 tumor female ; d759d07f-330c-5d0c-bd28-af72147dfb17 normal female

284f1424-d250-59cf-b105-da277b061e4a normal female ; e7e69d23-fb0d-5d3d-9027-ebf355053dbf tumor female

c42fffad-4ffd-59ba-93f1-2c573547369c normal female ; 3a33ef20-dfd0-50b0-afc2-38de9a5baa32 tumor female

37f076d6-fa64-5b5d-a0d0-b5cd7428d4a2 normal female ; 2c34270b-98d2-54b9-bdd3-068c6a9d858f tumor female
...
```

- Created with the `analyze_dataset` tool (`-p` option)

- At each entry (one per line), PanPipe's `pipe_exec` tool is used to analyze a normal-tumor bam file pair

- Entry example:

```
/home/dortiz/bio/software/bam-utils/bin/pipe_exec --pfile /home/dortiz/bio/software/bam-utils/share/bam
    -utils/examples/basic_test.ppl --outdir /mnt/raid/dortiz/bio/tasks/bam_analysis_testing_pipeline/
    d759d07f-330c-5d0c-bd28-af72147dfb17_34fa2369-424f-5886-9d23-6d19f8f15278 --sched SLURM -r /home/
    dortiz/bio/data/genome_references/refseq_hg19_filt.fa -extn d759d07f-330c-5d0c-bd28-af72147dfb17
    -extt 34fa2369-424f-5886-9d23-6d19f8f15278 -cr /home/dortiz/bio/data/genome_references/
    refseq_hg19_filt.fa.bed -egastr 50 -egacred /home/dortiz/bio/software/ega-download-client-python/
    dortiz_cred.json
```

Extending Package Functionality

- Focus on `bam_analysis.sh` module

- Two mechanisms:
  - Add new functions directly in `bam_analysis.sh`
  - Define a complementary module and import it in addition to `bam_analysis.sh`

Whole Pipeline Example

```
#import bam_analysis
#
download_ega_norm_bam cpus=1 mem=2048 time=10:00:00 stepdeps=none
download_ega_tum_bam cpus=1 mem=2048 time=10:00:00 stepdeps=none
index_norm_bam cpus=1 mem=1G time=4:00:00 stepdeps=afterok:download_ega_norm_bam
index_tum_bam cpus=1 mem=1G time=4:00:00 stepdeps=afterok:download_ega_tum_bam
manta_somatic cpus=8 mem=3G time=6:00:00 stepdeps=afterok:index_norm_bam,afterok:index_tum_bam
strelka_somatic cpus=8 mem=6G time=6:00:00 stepdeps=afterok:index_norm_bam,afterok:index_tum_bam,
        afterok:manta_somatic
msisensor cpus=8 mem=6G time=5:00:00 stepdeps=afterok:index_norm_bam,afterok:index_tum_bam
delete_bam_files cpus=1 mem=1G time=0:10:00 stepdeps=afterok:manta_somatic,afterok:strelka_somatic,
        afterok:msisensor
```