STACKS_WORKFLOW

An integrated workflow to streamline STACKS analyses on RAD/GBS data

About STACKS

The STACKS analysis pipeline (http://creskolab.uoregon.edu/stacks/) is the de facto tool for SNP discovery in Genotyping By Sequencing (GBS) and Restriction-site Associated DNA sequencing (RAD) studies when no reference genome is available. Upon starting to use STACKS, it is highly suggested to read the two official STACKS papers. These articles are listed at the bottom of the official page and contain the keyword *Stacks* in their title.

About the STACKS Workflow

This STACKS Workflow aims at making the use of the STACKS pipeline easier and more structured so that people tasked with analysing GBS or RAD data and possessing limited UNIX/Linux experience can jump on the analysis wagon faster. It was developped with the needs of our research group in mind with an emphasis on non-model species studies. We make no claim about its usefulness to other groups or in other contexts, but we still believe it may be of use to some.

The workflow has been tested with version 1.10 and earlier versions of STACKS under Linux (Ubuntu 12.04 to 13.10) and MacOSX.

Licence

Overview of the steps

Step 0 - Install and prepare the workflow
Step 1 - Download raw datafiles (Illumina lanes)
Step 2 - Extract individual data with process_radtags
Step 3 - Rename samples
Step 4 - STACKS pipeline (ustack/pstacks, cstack, sstack, populations/genotypes)
Step 5 - Filters

The workflow

Step 0 - Install and prepare the workflow

a) Download and install the most recent version of this workflow

- From the terminal, run:

```
cd ~/Desktop
wget https://github.com/enormandeau/stacks\_workflow/archive/master.zip
unzip master.zip
```

If you have `git` installed, you can do the same faster with:

```
git clone https://github.com/enormandeau/stacks_workflow
```

Use the extracted or cloned folder as your working directory for the rest of the project. All the commands in this manual are launched from that directory.

b) Download and install STACKS

- http://creskolab.uoregon.edu/stacks/
- Unzip the archive
- From within the STACKS folder, run:

```
./configure
make
sudo make install
```

Step 1 - Download raw datafiles (Illumina lanes)

a) Put them in the **02-raw** folder of the stacks_workflow folder

- NOTE: All file names MUST end with **.fastq.gz**

b) Prepare the **lane_info.txt** file automatically

- From the stacks_workflow folder, run:

```
./00-scripts/01_prepare_lane_info.sh
```

Step 2 - Extract individual data with process_radtags

a) Prepare a file named **sample_information.csv** using the same format found in the **example_sample_information.csv** file in the **01-info_files** folder. Also save this file in the **01-info_files** folder. This file will be used to extract the samples and rename the sample files automatically. The first column contains the EXACT name of the data file for the lane of each sample. The second column contains the barcode sequence of each sample. The third column contains the population name of each sample. The fourth column contains the name of the sample (do not include the population name or abbreviation in the sample name). The fifth column contains a number identifying the populations. Columns three and four are treated as text, so they can contain either text or numbers. Other columns can be present after the fifth one and will be ignored. However, it is crucial that the five first columns respect the format in the example file exactly. Be especially careful not to include errors in this file, for example mixing lower and capital letters in population or sample names (eg: Pop01 and pop01), since these will be treated as two different populations.

b) Launch process_radtags with:

```
./00-scripts/02_process_radtags.sh <trimLength> <enzyme>
```

Where: - trimLength = length to trim all the sequences
- enzyme = name of enzyme (run **process_radtags**, without options, for a list of the supported enzymes)

Step 3a - Rename samples

a) To rename and copy the samples, run:

```
./00-scripts/03_rename_samples.sh
```

b) Join samples that should go together

- Go to 04-all_samples and join the .fq files that should go together with the `cat` command
- Remove partial .fq files that have been joined
- Remove individuals with too few sequences (optional)

Step 3b - (Optional) Align reads to a reference genome

a) Install bwa

b) Download reference genome to the 01-info_files

c) Index reference genome, run:

```
bwa index -p genome -a bwtsw ./01-info_files/<genome reference>
```

d) copy files:

```
cp genome.* 01-info_files
```

e) Align samples:

```
for i in $(ls -1 04-all_samples/*.fq)
do
    name=$(basename $i)
    bwa aln -n 5 -k 3 -t 2 ./01-info_files/genome $i | \
    bwa samse -r "@RG\tID:'$name'\tSM:'$name'\tPL:Illumina" \
        ./01-info_files/genome - $i ./04ln-all_samples/$name.sam; \
done
```

Step 4 - STACKS (ustack/pstacks, cstack, sstack, populations/genotypes)

a) Prepare population info file

- To prepare a template of that file, run:

```
./00-scripts/04_prepare_population_map_template.sh
```

b) Open the stacks script in the 00-scripts folder and edit the options

c) Run the STACKS programs, in order:

- ustacks (or pstacks for reference assisted):

```
./00-scripts/stacks_1a_ustacks.sh
```

or (if you are using a reference genome):

```
./00-scripts/stacks_1b_pstacks.sh
```

- cstacks:

```
./00-scripts/stacks_2_cstacks.sh
```

- sstacks:

```
./00-scripts/stacks_3_sstacks.sh
```

- populations or genotypes:

```
./00-scripts/stacks_4_populations.sh
```

Step 5 - Filters

- Use ./00-scripts/05_filterStacksSNPs.py to filter your SNPs. To print the documentation, type:

```
./00-scripts/05_filterStacksSNPs.py
```

- Launch the script, example:

```
./00-scripts/05_new_filterStacksSNPs.py \
    -i 05-stacks/batch_1.sumstats.tsv \
    -o filtered.tsv \
    -P 01-info_files/population_map.txt \
    -p 2 -x 1 -H 0.7 -a 0.05 -A 0 -f -0.3 -F 0.8 -s 10
```