About the STACKS Workflow

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. This STACKS Workflow aims at making the use of the STACKS pipeline easier and more structured so that people with GBS or RAD projects and limited UNIX/Linux experience can jump on the analysis wagon faster. It is being developped with the needs of our research group in mind and we make no claim about its use to other groups or in other contexts.

The workflow has been tested with version 1.04 of STACKS.

Licence

Overview of the steps

Step 0 - Install and prepare the workflow
Step 1 - Download raw datafiles (Illumina lanes)
Step 2 - Extract individual information with process_radtags
Step 3 - Rename samples and make links
Step 4 - STACKS (ustack/pstacks, cstack, sstack, populations/genotypes)
Step 5 - Filters
Step 6 - Format for population genetics

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
```

Use the extracted folder (`stacks_workflow-master`) as your working directory for the rest of the project. If you just updated the workflow, please use the MANUAL.pdf file that comes with that new version.

b) Download and install STACKS
- http://creskolab.uoregon.edu/stacks/
- Unzip
- From within the STACKS folder, run:

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

Step 1 - Download raw datafiles (Illumina lanes)

a) Put them in the **raw** folder of the gbs_workflow
- NOTE: All file names MUST end with **.fastq.gz**

b) Prepare the **lane_info.txt** file automatically
- From the gbs_workflow folder, run:

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

c) Prepare the **sample_info.txt** file using the format of **example_sample_info.txt**. This file will be used to extract the samples and rename the sample files in an intelligible manner. 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. 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 (Pop01 and pop01), since these will be treated as two different populations.

Step 2 - Extract individual information with process_radtags

a) Prepare a sample information file **sample_info.txt** and put it in the **01-info_files folder**. This file should contain one line by sample and two columns. The first column contains the exact name of the lane in which the individual is found (see the **lane_info.txt** file in the **01-info_files folder** for an example of the format) and the sequence of the tag for that individual in the second column.

b) Launch process_radtags with:

TODO use discarted reads at each step rather than treating the whole file each time

```
./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 list)

Step 3a - Rename samples and make links

a) To rename and copy the samples, run:

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

b) Join samples that should go together

TODO Implement neat way of doing this
- Go to 04-all_samples and join the .fq files that should go together
- Remove partial .fq files that have been joined
- Remove individuals with too few sequences (optional)

Step 3b - 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
```

d) Aligned samples, run

```
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) Rename the template file to **population_map.txt** and remove **.fq** extensions in columns 1
c) Open the stacks script in the 00-scripts folder and edit the options
d) 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_filterStacksSNPs.py ./05-stacks/batch_1.sumstats.tsv 2 1
0.6 0.05 -0.3 0.3 8
```

Step 6 - Format for population genetics

... in development ...