denovo_map.pl

The **denovo_map.pl** program will execute the Stacks pipeline by running each of the Stacks components individually. It is the simplest way to run Stacks and it handles many of the details, such as sample numbering and loading data to the MySQL database, if desired. The program performs several stages, including:

1. Running **ustacks** on each of the samples specified, building loci and calling SNPs *de novo* in each sample.

2. Executing **cstacks** to create a *catalog* of all loci across the population (or from just the parents if processing a genetic map). Loci are matched up across samples according to sequence similarity.

3. Next, **sstacks** will be executed to match each sample against the catalog. In the case of a genetic map, the parents and progeny are matched against the catalog.

4. In the case of a population analysis, the **populations** program will be run to generate population-level summary statistics. If you specified a population map (`-O` option) it will be supplied to **populations**. If you are analyzing a genetic map, the **genotypes** program will be executed to generate a set of markers and a set of initial genotypes for export to a linkage mapping program.

5. Computation is now complete. If database interaction is enabled, **denovo_map.pl** will upload the results of each stage of the analysis: individual loci, the catalog, matches against the catalog, and genotypes or sumamry statistics into the database.

6. Lastly, if database interaction is enabled, **index_radtags.pl** will be run to build a database index to speed up access to the database and enable web-based filtering.

Specifying Samples

The raw data for each sample in the analysis has to be specified to Stacks. **All of your samples have to be sepcified together for a single run of the pipeline.** There are two ways to do this in **denovo_map.pl**. First, each sample can be specified separately on the command line. If processing a genetic map, each parent and/or progeny can be specified using `-p` and `-r`, respectively. If analyzing a population, each sample can be specified using `-s` (see examples below).

The second way to specify samples is to use a population map. In this case, you specify the path to the directory containing all samples using the `--samples` option, and then **denovo_map.pl** will read the contents of the population map file (specified with the `-O` option) and search for each specified sample in the `--samples` directory.

Using a population map

A population map is an optional file that contains assignments of each of your samples to a particular population. See the manual for more information on how they work. The **denovo_map.pl** program will not directly use the file (beyond potentially reading the sample names out of it), however it will upload the population mappings to the database for use in the web interface, if enabled. It is the **populations** program that acutally uses the population map for statistical calculations, and **denovo_map.pl** will provide the map to the **populations** program. You can run

the **populations** program by hand, specifying other population maps as you like, after the pipeline completes its first execution.

Using the Database (and Web Interface)

If you have the database and web interface installed (MySQL and the Apache Webserver) then **denovo_map.pl** can upload the output from the pipeline to the database for viewing in a web browser. To do so:

1. Pick a *batch ID*, a number to represent the batch of data processed by Stacks. Specify it to the pipeline with the `-b` option.

2. Choose a database name and specify it to **denovo_map.pl** using the `-B` option. For the database to be displayed by the web interface it needs to have a "*_radtags*" suffix, e.g. `fishstudy_radtags` (this allows the web interface to ignore other databases that may be present in your server unrelated to Stacks).

3. You can specify a description of this batch of data by specifying it with the `-D` option. The description will be displayed in the web interface, it can help you differentiate between different runs of the pipeline, with for example, different parameter choices.

4. If the database already exists, you are ready to start **denovo_map.pl**. If the database does not yet exist, you can specify the `--create_db` option to create it. If it already exists, but you want to delete and recreate it, specify the `--overw_db` option.

Different batches can be loaded into the database by running **denovo_map.pl** repeatedly with different batch IDs and specifying the same database. This is convenient to collect a set of pipeline runs, with different parameter settings, in one place for comparison.

> If you do not wish to use the database, specify the `-S` option to **denovo_map.pl** and all database interactions will be disabled. If you want to load data into the database after a **denovo_map.pl** execution, you can do so using the **load_radtags.pl** program.

Passing additional arguments to Stacks component programs

The Stacks component programs contain a lot of possible options that can be invoked. It would be impractical to expose them all througth the **denovo_map.pl** wrapper program. Instead, you can pass additional options to internal programs that **denovo_map.pl** will execute using the `-X`. To use this option, you specify (in quotes) the program the option goes to, followed by a colon (":"), followed by the option itself. For example, `-X "populations:--fstats"` will pass the `--fstats` option to the **populations** program. Another example, `-X "populations:-r 0.8"` will pass the `-r` option, with the argument `0.8`, to the **populations** program. **Each option should be specified separately with `-x`**. See below for examples.

When not to use **denovo_map.pl**

There are a few reasons to run the pipeline manually instead of using the **denovo_map.pl** wrapper.

1. If you have a very large number of samples, you may not want to put them all in the catalog. In a population analysis, all of the samples specified to **denovo_map.pl** will be loaded into the catalog. In a *de novo* analysis, each sample added to the catalog will also add a small fraction of error to the catalog. With a very large number of samples, the error can overwhelm true loci in the population. In this case you may only want to load a subset of each population in your analysis.

2. Again, if you have a lot of samples, you may want to speed your analysis by splitting up your samples and running them on a number of nodes in a cluster. In this case, you would have to queue up **ustacks** to run on different nodes with different samples. This can't be done using **denovo_map.pl**.

Program Options

```
denovo_map.pl -p path -r path [-s path] -o path [-t] [-m min_cov] [-M mismatches] [-n mismatches] [-T num_t
```

[-O popmap] [-B db -b batch_id -D "desc"] [-S -i num] [-e path] [-d] [-h]

b — batch ID representing this dataset (an integer, e.g. 1, 2, 3).

o — path to write pipeline output files.

O — if analyzing one or more populations, specify a pOpulation map.

A — if processing a genetic map, specify the cross type, 'CP', 'F2', 'BC1', 'DH', or 'GEN'.

T — specify the number of threads to execute.

e — executable path, location of pipeline programs.

d — perform a dry run. Do not actually execute any programs, just print what would be executed.

h — display this help message.

Specify each sample separately:

p — path to a FASTQ/FASTA file containing one set of parent sequences from a mapping cross.

r — path to a FASTQ/FASTA file containing one set of progeny sequences from a mapping cross.

s — path to a FASTQ/FASTA file containing an individual sample from a population.

Specify a path to samples and provide a population map:

--samples [path] — specify a path to the directory of samples (samples will be read from population map).

Stack assembly options:

m — specify a minimum number of identical, raw reads required to create a stack.

M — specify the number of mismatches allowed between loci when processing a single individual (default 2).

n — specify the number of mismatches allowed between loci when building the catalog (default 1).

--gapped — perform gapped assemblies in ustacks, cstacks, and sstacks (default: off).

Advanced (rarely used) options:

P — specify a minimum number of identical, raw reads required to create a stack in 'progeny' individuals.

N — specify the number of mismatches allowed when aligning secondary reads to primary stacks (default M+2).

t — remove, or break up, highly repetitive RAD-Tags in the ustacks program.

H — disable calling haplotypes from secondary reads.

Database options:

B — specify a database to load data into.

D — a description of this batch to be stored in the database.

S — disable recording SQL data in the database.

i — starting sample_id, this is determined automatically if database interaction is enabled.

--create_db — create the database specified by '-B' and populate the tables.

--overw_db — delete the database before creating a new copy of it (turns on --create_db).

SNP Model Options (these options are passed on to ustacks):

--bound_low [num] — lower bound for epsilon, the error rate, between 0 and 1.0 (default 0).

--bound_high [num] — upper bound for epsilon, the error rate, between 0 and 1.0 (default 1).

--alpha [num] — chi square significance level required to call a heterozygote or homozygote, either 0.1, 0.05 (default), 0.01, or 0.001.

Arbitrary command line options:

-X "program:option" — pass a command line option to one of the pipeline components, e.g.'-X "ustacks:--max_locus_stacks 4"'.

Example Usage

Genetic map

1. This example will run the pipeline for a genetic map, having specified two parents and three progeny. The pipeline will tell the internal programs to use 15 processors (`-T`) and the results will be uploaded to the `tut_radtags` database, after the database is created.

```
% denovo_map.pl -m 3 -M 3 -n 2 -T 15 -B tut_radtags -b 1 --create_db -A CP \
    -D "Genetic Map RAD-Tag Samples" \
    -o ./stacks \
    -p ./samples/f0_male.fq \
    -p ./samples/f0_female.fq \
    -r ./samples/progeny_01.fq \
    -r ./samples/progeny_02.fq \
    -r ./samples/progeny_03.fq
```

Population

1. This example will run the pipeline for a population analysis, having specified five samples to be analyzed. The pipeline will tell the internal programs to use 15 processors (`-T`) and the results will be uploaded to the `fishstudy_radtags` database after it is created.

```
% denovo_map.pl -m 3 -M 3 -n 2 -T 15 -B fishstudy_radtags -b 1 --create_db  \
    -D "Population RAD-Tag Samples, m:3;M:3;n:2" \
    -o ./stacks \
    -s ./samples/indv_01.fq \
    -s ./samples/indv_02.fq \
    -s ./samples/indv_03.fq \
    -s ./samples/indv_04.fq \
    -s ./samples/indv_05.fq
```

2. Now I can run the pipeline a second time, having incremented the batch ID, this time trying a different set of parameters. Once complete, both batch 1 and 2 will be avilable for viewing in the web interface.

```
% denovo_map.pl -m 5 -M 4 -n 2 -T 15 -B fishstudy_radtags -b 2  \
    -D "Population RAD-Tag Samples, m:5;M:4;n:2" \
    -o ./stacks \
    -s ./samples/indv_01.fq \
    -s ./samples/indv_02.fq \
    -s ./samples/indv_03.fq \
    -s ./samples/indv_04.fq \
    -s ./samples/indv_05.fq
```

3. In this example, I will supply a population map to **`denovo_map.pl`** containing the names of the samples I want to analyze, and instead of supplying the samples on the command line, I will just tell **`denovo_map.pl`** the directory the samples are located in. I will also use the `-x` option to tell **`populations`** to enable the calculation of F statistics, and `denovo_map.pl` will pick up the F statistics output and upload it to the database for viewing in the web interface.

```
% denovo_map.pl -m 3 -M 3 -n 2 -T 15 -B treestudy_radtags -b 1  \
    -D "Population RAD-Tag Samples, m:3;M:3;n:2" \
    -o ./stacks \
    -O ./treestudy_popmap \
    --samples ./samples \
    -X "populations:--fstats"
```

4. In this example, I will disable database interaction, but still supply a population map and tell **`populations`** to enable F statistics.

```
% denovo_map.pl -m 3 -M 3 -n 2 -b 1 -S -o ./stacks -O ./treestudy_popmap --samples ./samples -X "popu
```

The backslashes ("\") in the commands above cause the command to be continued onto the following line in the shell (it must be the last character on the line). This can be useful when typing a long command or when embedding a command in a shell script. If you are typing the command on a single line, do not include the backslashes.

Other Pipeline Programs

Raw Reads

- process_radtags
- process_shortreads
- clone_filter
- kmer_filter

Core

- ustacks
- pstacks
- cstacks
- sstacks
- genotypes
- populations
- rxstacks

Execution control

- denovo_map.pl
- ref_map.pl
- load_radtags.pl

Utilities

- index_radtags.pl
- export_sql.pl
- sort_read_pairs.pl
- exec_velvet.pl