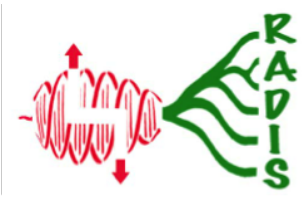

Analysis of RAD-seq data for InterSpecific phylogeny

***Astrid Cruaud*[1£], *Mathieu Gautier*[12£], *Jean-Pierre Rossi*[1], *Jean-Yves Rasplus*[1], *Jérôme Gouzy*[34]**

[1]*INRA, UMR1062 CBGP, F-34988 Montferrier-sur-Lez, France*
[2]*IBC, F-34095 Montpellier, France*
[3]*INRA, UMR441 LIPM, F-31326 Castanet Tolosan, France*
[4]*CNRS, UMR2594 LIPM, F-31326 Castanet Tolosan, France*
[£] *equal contributors*

version 1.0.0, 28 january 2016

Contents

1 Quickstart **2**

1.1 Install . . . 2
1.2 Requirements . . . 2
1.3 Quick overview of the workflow . . . 3
1.4 Disclaimer . . . 5
1.5 Citation . . . 5
1.6 Funding . . . 5

2 Step by Step tutorial **5**

2.1 Example data sets . . . 5
2.1.1 With barcodes added to reads 1 only . . . 6
2.1.2 With barcodes added to reads 1 and reads 2 . . . 6
2.2 Run RADIS . . . 6
2.2.1 Suggested directory structure . . . 6
2.2.2 Before running the analyses . . . 6
2.2.2.1 The `datadir` directory. . . . 6
2.2.2.2 The configuration file. . . . 7
2.2.2.3 The `outdir` directory. . . . 7
2.2.3 Running the analyses . . . 8

2.2.4 Follow the progress of the analysis and detect execution errors . . . 8
2.3 Dig into the results (Step 1, data cleaning) . . . 9
2.3.1 Step 0.0 Checking parameters (RADIS) . . . 10
2.3.2 Step 0.0 Dumping pairs of files (RADIS) . . . 10
2.3.3 Step 1.0 Read filtering (RADIS) . . . 10
2.3.4 Step 1.1 Demultiplexing of data (Stacks / process_radtags) . . . 10
2.3.5 Step 1.2 Trimming of reads (RADIS) . . . 11
2.3.6 Step 1.3 Removing PCR duplicates (Stacks / clone_filter) . . . 11
2.3.7 Step 1.4 Rename files (RADIS) . . . 12
2.4 Dig into the results (Step 2, data analysis) . . . 12
2.4.1 Step 2.1 Building individual loci (Stacks / ustacks) . . . 13
2.4.2 Step 2.2 Building catalog loci (Stacks / cstacks) . . . 13
2.4.3 Step 2.3 Selection of loci and samples and Step 2.4 Building phylip files (RADIS) . . . 15
2.4.4 Step 2.5 Phylogenetic analysis (RAxML) . . . 16
2.5 How can I mix data from many RAD libraries? . . . 16

3 Cited references **17**

1 Quickstart

1.1 Install

`RADIS` is available from : http://www1.montpellier.inra.fr/CBGP/software/RADIS/

To install `RADIS`, download and unzip the tar.gz file :

```
tar -xvzf RADIS-1.0.0.tar.gz
```

A directory named `RADIS-1.0.0` is created, which contains five subdirectories and a quickstart file.

```
RADIS-1.0.0
  - bin # contains perl scripts
  - cfg # contains configuration files
  - datadir_example_double_barcode # contains example data (barcodes on reads 1 on 2)
  - datadir_example_single_barcode # contains example data (barcodes on reads 1 only)
  - lib # contains libraries required to run RADIS
  - quickstart # 4-step instructions to run RADIS
```

1.2 Requirements

`RADIS` has been tested on Linux platforms, including large computer clusters and multicore machines and should work on any standard UNIX-like environment. Depending on the level of depth and detail you want to explore in your analyses, data processing may require lot of your machine, both in terms of memory and

processor speed. More specifically, the process of evaluating multiple combinations of parameters can become time consuming.

RADIS relies on Stacks (Catchen *et al.*, 2011, 2013, http://catchenlab.life.illinois.edu/stacks/) for demultiplexing of data, removing PCR duplicates and building individual and catalog loci and RAxML (Statmatakis, 2006 a, b, http://www.exelixis-lab.org) for phylogenetic inferences, though other softwares may be used to analyse output data files. RADIS requires that R is installed.

- The last version of Stacks tested was version 1.32. This version can be downloaded from http://catchenlab.life.illinois.edu/stacks/source/stacks-1.32.tar.gz. The Stacks manual is available from http://creskolab.uoregon.edu/stacks/manual/index.php#files and a very useful tutorial can be found here http://creskolab.uoregon.edu/stacks/param_tut.php
- The last version of RAxML tested was version 8.2.4. RAxML can be downloaded from GitHub (https://github.com/stamatak/standard-RAxML)
- The last version of R tested was version 3.2.2 (2015-08-14) – "Fire Safety".

1.3 Quick overview of the workflow

The package contains three Perl scripts (stored in RADIS-1.0.0/bin) :

- RADIS_step1_data_cleaning.pl, which allows filtering, demultiplexing and trimming of data. PCR duplicates are then removed and files are renamed according to user specification.
- RADIS_step2_data_analysis.pl, which allows building individual loci, building catalog loci, selecting loci and samples and building phylip files for phylogenetic analysis. Phylip files are then analysed using RAxML. Note that the workflow can be stopped before RAxML analyses using the flag --without-raxml.
- RADIS.pl, which allows running the full pipeline. Note that the workflow can be stopped before RAxML analyses using the flag --without-raxml.

Before running any analysis you must:

- Provide a correspondence between barcodes (or combination of barcodes) used to tag your samples and sample codes (see examples in RADIS-1.0.0/datadir_example_single_barcode/barcodes_lib_names.txt or RADIS-1.0.0/datadir_example_double_barcode/barcodes_lib_names.txt)

---

barcodes_lib_names.txt file specification :

barcode used to tag reads 1	barcode used to tag reads 2	sample code
AAACAA	ACCTA	JRAS02472_0195
ATGAAG	CGTAC	TRIC00013_0199

Column separator = tab

– Remove the second column if no barcode were used for reads 2.

– You can mix barcodes of any length.

– Note that **sample codes MUST be different from each other** to avoid troubles during renaming.

– **REMOVE empty lines** (pay attention to remove blank lines at the end of files)

– Avoid pipe as well as single and double quotes in sample codes

– Open file in terminal before running `RADIS` to make sure file format is correct :

`wc -l barcodes_lib_names.txt` # will count samples, check if result corresponds to your expectations

`cat -A barcodes_lib_names.txt` # (for Linux) file, including field separators, is printed on screen you should see ^I (tab) between columns and a $ at the end of each line and nothing else, if not see below

`cat -te barcodes_lib_names.txt` # (for Mac) file, including field separators, is printed on screen you should see ^I (tab) between columns and a $ at the end of each line and nothing else, if not see below

In case results of the wc and cat commands are not correct it may be because Windows or Mac produce different "end of line" (EOL) characters as compared to Linux. Here are some tricks to avoid headaches trying to understand what is wrong:

`mac2unix barcodes_lib_names.txt` # will change format from Mac native to full UNIX

`dos2unix barcodes_lib_names.txt` # will change format from Windows to UNIX

`sed -i '/^$/d' barcodes_lib_names.txt` # will remove extra spaces introduced by Mac

N.B. you may need to install the program `dos2unix` on your computer

---

- Fill up a configuration file with parameters values and absolute paths to external softwares (i.e. `Stacks` and `RAxML`). A template is provided in `RADIS-1.0.0/cfg/RADIS.cfg`. As much as possible, comments (starting with a hash #) are added to explain the purpose of the command (Nota: everything after the # is ignored by the shell, **DO NOT remove the # from the .cfg file**).

Once the `barcodes_lib_names.txt` and `.cfg` files are ready, scripts are run as follows (optional parameters between brackets) :

```
($RADIS/bin/RADIS_step1_data_cleaning.pl --datadir rooted_directory_name --outdir
rooted_directory_name [--cfg filename] [--ncpus integer] [--interleave_list_of_files]
> stdoutfilename) 2> stderrfilename


($RADIS/bin/RADIS_step2_data_analyses.pl --listoffastqfiles 'quoted string' --outdir
rooted_directory_name [--cfg filename] [--ncpus integer] [--without-raxml]
> stdoutfilename) 2> stderrfilename


($RADIS/bin/RADIS.pl --datadir rooted_directory_name --outdir rooted_directory_name
[--cfg filename] [--ncpus integer] [--interleave_list_of_files] [--without-raxml]
> stdoutfilename) 2> stderrfilename
```

with:

```
--cfg # full path to the configuration file

--datadir # full path to the input directory which should contain sequence files
#(.fq, .fq.gz, fastq, fastq.gz) and a barcode/sample code correspondence file

--interleave_list_of_files # use this flag when the ls command on the datadir directory
```

```
#groups all read1 files and then all read2 files (expected output from ls is
#file1_read1, file1_read2, file2_read1, file2_read2, etc)

--listoffastqfiles 'quoted string' # list of purified reads obtained after data cleaning.
#A text file listing absolute path to chosen fq_1 files can be provided (.fof).

--ncpus # integer, number of cpus

--outdir # full path to the output directory.

--without-raxml # use this flag to stop the pipeline before RAxML analyses
#(after .phy files are created)

stdoutfilename # will contain a report on the execution of all tasks

stderrfilename # will contain warning messages and execution errors
```

1.4 Disclaimer

RADIS is delivered as it is (and without any warranty). If you find any bug, or have some suggestions to improve the pipeline and better fit your needs, feel free to do it and, please, keep us posted!

1.5 Citation

If you use RADIS please cite also all external programs used by the pipeline. Many thanks!

Suggested citation : "Analyses were performed using the Perl pipeline RADIS (Cruaud *et al.*, submitted) that relies on Stacks (Catchen *et al.*, 2011, 2013) for demultiplexing of data, removing PCR duplicates and building individual and catalog loci and RAxML (Statmatakis, 2006 a, b) for phylogenetic inferences".

1.6 Funding

This work was based upon financial support received from the division "Plant Health and the Environment" of the French National Institute for Agricultural Research (INRA) to AC (Tricho-NG project) and by the French National Research Agency (ANR) project "TriPTIC" to AC and JYR.

2 Step by Step tutorial

This tutorial is meant to show to both beginners and most experienced users how to use RADIS.

2.1 Example data sets

Two example data sets are provided with the package. Please note that these data sets have been extracted from much larger data sets to speed up processing ($<$ 1 min on 8-cores of a 16-cores Linux, 2.9GHz, 64GB RAM computer). Obtained results are thus meaningless.

2.1.1 With barcodes added to reads 1 only

Datadir = `RADIS-1.0.0/datadir_example_single_barcode` [fq.gz files + barcodes_lib_names.txt file]
Configuration file = `RADIS-1.0.0/cfg/RADIS_example_single_barcode.cfg`

This data set is a subset of the Cruaud *et al.* (2014) data set on *Carabus* beetles for which the experimental design was : digestion with the *PstI* enzyme and ligation of P1 adaptors containing 5 or 6- bp barcodes. Paired-end sequencing (2*100 nt) on a single lane of a HiSeq 2000 flowcell. The `.fq.gz` files provided contain some of the reads obtained for four of the 31 samples sequenced. The `barcodes_lib_names.txt` file gives the correspondence between the barcodes used and the sample codes.

2.1.2 With barcodes added to reads 1 and reads 2

Datadir = `RADIS-1.0.0/datadir_example_double_barcode` [fq.gz files + barcodes_lib_names.txt file]
Configuration file = `RADIS-1.0.0/cfg/RADIS_example_double_barcode.cfg`

This data set is a subset of an unpublished data set on *Trichogramma* wasps for which the experimental design was : digestion with the *PstI* enzyme and ligation of P1 and P2 adaptors containing 5 or 6 bp- barcodes. Paired-end sequencing (2*125 nt) on a single lane of a HiSeq 2000 flowcell. The `.fq.gz` files contain some of the reads obtained for four of the 40 samples sequenced. The `barcodes_lib_names.txt` file gives the correspondence between the combination of barcodes used and the sample codes.

2.2 Run RADIS

2.2.1 Suggested directory structure

You can of course adopt whatever structure you like before starting your analyses. This one will fit with commands provided in the tutorial to run `RADIS` on the example data.

```
home
  - user
     - RADIS-1.0.0
           - bin
           - cfg
           - datadir_example_double_barcode
           - datadir_example_single_barcode
           - lib
           - quickstart
```

2.2.2 Before running the analyses

Please note that `RADIS` will not provide any quality control checks on your raw sequence data. If you want to check whether your data has any problems of which you should be aware before doing the analyses, we suggest the use of the software `FastQC` developed by the Bioinformatics Group of the Babraham Institute (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/).

2.2.2.1 The `datadir` directory. Before starting the analysis, you must create a datadir directory in which you will store:

- the raw Illumina data
- the correspondence list between barcodes (or combination of barcodes) and sample codes (`barcodes_lib_names.txt` file).

You need to list the content of the `datadir` directory before running the analysis to check whether the flag `--interleave_list_of_files` must be added to the command.

As an example, you can list the content of one of the example datadir directories by typing the following command.

```
ls /home/user/RADIS-1.0.0/datadir_example_single_barcode
```

Screenshot of the results you should get:

```
barcodes_lib_names.txt  example_single_barcode_R1_001.fq.gz
example_single_barcode_R1_002.fq.gz  example_single_barcode_R2_001.fq.gz
example_single_barcode_R2_002.fq.gz
```

Here, R1 fastq files are listed first, then come R2 fastq files. Therefore, to analyse this example data set, you need to add the flag `--interleave_list_of_files` to the command.

When you list files in your own directory, you may get something like this instead:

```
barcodes_lib_names.txt  example_single_barcode_R1_001.fq.gz
example_single_barcode_R2_001.fq.gz  example_single_barcode_R1_002.fq.gz
example_single_barcode_R2_002.fq.gz
```

In that case (fastq files with paired-end reads 1 and 2 follow each other in the list), the flag `--interleave_list_of_files` is not required.

To provide a correspondence between barcodes and sample codes you can use one of the example `barcodes_lib_names.txt` files as template (`RADIS-1.0.0/datadir_example_single_barcode/barcodes_lib_names.txt` or `RADIS-1.0.0/datadir_example_double_barcode/barcodes_lib_names.txt`). Note that sample codes **MUST** be different from each other to avoid problems when files are renamed. You can mix barcodes of any length.

2.2.2.2 The configuration file. Before starting the analysis, you must also fill up a configuration file to set parameters and path to external softwares. You can use the `RADIS-1.0.0/cfg/RADIS.cfg` file as template to set up the parameters for your analysis. As much as possible, comments (starting with a hash #) are added to explain the purpose of the command (Nota: everything after the # is ignored by the shell, **DO NOT** remove the # from the .cfg file).

2.2.2.3 The `outdir` directory. Lastly, you need to create an `outdir` directory in which results will be automatically stored. You may for example create as many configuration files as needed and store the results in different outdir directories. As an example, you can create an outdir directory for the analysis of the "single barcode data set" in the `/home/user/RADIS-1.0.0/` directory by typing the following command:

```
mkdir /home/user/RADIS-1.0.0/outdir_example_single_barcode_fullpipeline
```

2.2.3 Running the analyses

For now, you can either execute:

- the `RADIS.pl` script, to run the full pipeline.

As an example, to run this script on the "single barcode data set" from `/home/user/RADIS-1.0.0/`, type the following command:

```
(./bin/RADIS.pl --datadir /home/user/RADIS-1.0.0/datadir_example_single_barcode --outdir
 /home/user/RADIS-1.0.0/outdir_example_single_barcode_fullpipeline --cfg
 /home/user/RADIS-1.0.0/cfg/RADIS_example_single_barcode.cfg --interleave_list_of_files
 > stdout_fullpipeline) 2> stderr_fullpipeline
```

Note: the flags `--ncpus` (number of cpus) and `--without-raxml` (to stop the pipeline before RAxML analyses) may be added.

- the `RADIS_step1_data_cleaning.pl` script for filtering, demultiplexing and trimming of data, removal of PCR duplicates and renaming of files according to sample codes.

As an example, to run this script on the "single barcode data set" from `/home/user/RADIS-1.0.0/`, type the following command:

```
(./bin/RADIS_step1_data_cleaning.pl --datadir
/home/user/RADIS-1.0.0/datadir_example_single_barcode --outdir
/home/user/RADIS-1.0.0/outdir_example_single_barcode_datacleaning --cfg
/home/user/RADIS-1.0.0/cfg/RADIS_example_single_barcode.cfg --interleave_list_of_files
> stdout_datacleaning) 2> stderr_datacleaning
```

Note: the flag `--ncpus` (number of cpus) may be added.

To execute the `RADIS_step2_data_analysis.pl` script, you need to wait for the output data of the data-cleaning step.

2.2.4 Follow the progress of the analysis and detect execution errors

You can view the current status and progress of your analysis, in the `stdout` file (which is created in the current directory, i.e. `/home/user/RADIS-1.0.0/` for the example we are using).

Screenshot of the first lines of the `stdout_fullpipeline` file :

```
./bin/RADIS.pl --datadir /home/cruaud/RADIS-1.0.0/datadir_example_single_barcode
--outdir /home/cruaud/RADIS-1.0.0/outdir_example_single_barcode_fullpipeline
--cfg /home/cruaud/RADIS-1.0.0/cfg/RADIS_example_single_barcode.cfg
--interleave_list_of_files
================================================================================
```

```
      RADIS step1: data cleaning    Fri Jan 29 18:13:18 CET 2016
================================================================================
Step 0.0: Checking parameters.........................done
Step 0.0: Dumping pairs of files.....................started
Step 0.0: Dumping pairs of files......................done
Step 1.0: Read filtering (4 files)...................started
```

The `stdout` file will always start with a reminder of the command you used.

The `stderr` file will list warning message and execution error

Screenshot of the first lines of the `stderr_fullpipeline` file

```
Read1                                                                    Read2
example_single_barcode_R1_001.fq.gz          example_single_barcode_R2_001.fq.gz
example_single_barcode_R1_002.fq.gz          example_single_barcode_R2_002.fq.gz
Warning: consider adding the --interleave_list_of_files flag in case of incorrect pairing
```

In the `stderr` file you can see whether reads1 and 2 are correctly paired (and if the flag `--interleave_list_of_files` was required or not)

2.3 Dig into the results (Step 1, data cleaning)

Output files and subdirectories are created in the `outdir` directory :

The directory structure you should get in the `outdir` directory once the data-cleaning step of the pipeline is complete ("single barcode" data set) is detailed below:

```
- 1.0.statsgoodreads.tsv      # statistics on read filtering
- 1.1.statsprocesstags.tsv    # statistics for the process_radtags step
- 1.3.countclonefilter.tsv    # statistics on the number of PCR duplicates
- 1.4.statspurereads.tsv      # number of reads per sample once the data-cleaning step
                                #is complete
- 5                      # contains data for samples tagged with 5bp-barcodes
    - 01_DMP                 # contains fq files for DeMultiPlexed R1 and R2
    - 01_REM                 # contains fq files REMoved from the analysis
    - 02_TRIM                # contains fq files for TRIMmed R1 and R2
    - 03_FIL                 # contains files with the number of FILtered PCR duplicates
    - 04_PUR                 # contains PURified R1 and R2 (ready for data-analysis)
    - process_radtags.log    # output from the process_radtags programm
- 6                      # contains data for samples tagged with 6bp-barcodes
    - 01_DMP
    - 01_REM
    - 02_TRIM
    - 03_FIL
    - 04_PUR
```

```
    - process_radtags.log
- barcodes5bp.pools           # list of 5-bp barcodes
- barcodes6bp.pools           # list of 6-bp barcodes
- Good_R1.fq                  # R1 that pass the filtering step
- Good_R2.fq                  # R2 that pass the filtering step
```

If barcodes are used to tag reads 1 and reads 2 `RADIS` creates a subdirectory named `PBC` (for "Paired BarCode") instead of a directory for each barcode length (see below, the directory structure you should get for the analysis of a "double barcode" data set) :

```
- 1.0.statsgoodreads.tsv
- 1.1.statsprocesstags.tsv
- 1.3.countclonefilter.tsv
- 1.4.statspurereads.tsv
- barcodes.pairs # barcode pairs used to tag R1 and R2 (data_cleaning step)
- Good_R1.fq
- Good_R2.fq
- PBC       # contains data for samples tagged with pairs of barcodes (data_cleaning step)
    - 01_DMP
    - 01_REM
    - 02_TRIM
    - 03_FIL
    - 04_PUR
    - process_radtags.log
```

Hereafter we detail each step of the process following the content of the `stdout` file

2.3.1 Step 0.0 Checking parameters (RADIS)

The configuration file is checked by `RADIS` for coherence and integrity

2.3.2 Step 0.0 Dumping pairs of files (RADIS)

The output of the ls command on the datadir directory is printed in the `stderr` file. You can thus check if the flag `--interleave_list_of_files` is required or not

2.3.3 Step 1.0 Read filtering (RADIS)

Reads that failed to pass the filtering step of the Illumina's CASAVA pipeline (filtering=Y) are discarded. The number of processed `fq.gz` files is indicated between parentheses in the `stdout` file. A file named `1.0.statsgoodreads.tsv` that includes statistics on the number of filtered reads is created in the `outdir` directory.

2.3.4 Step 1.1 Demultiplexing of data (Stacks / process_radtags)

The program `process_radtags` from the software `Stacks` is used to demultiplex data and remove bad quality reads that escaped filtering.

A file named `1.1.statsprocesstags.tsv`, that includes the number of reads 1 and 2 retained per barcode/sample is created in the `outdir` directory.

If barcodes are used to tag reads 1 only, as many directories as barcode length classes are created in the `outdir` directory, each of them being labelled with the barcode length (*e.g.* two directories named 5 and 6 respectively are created in `/home/user/RADIS-1.0.0/outdir_example_single_barcode_datacleaning` for the example "single barcode" data set).

Within these directories, two subdirectories and one file are created:

- `01_REM` that contains REMoved reads (i.e bad quality reads, reads with unrecognized barcodes or restriction sites etc.)
- `01_DMP` that contains DeMultiPlexed "good" reads.
- The `process_radtags.log` file which is the log file created by `process_radtags`

If barcodes are used to tag reads 1 and reads 2 `RADIS` creates a subdirectory named `PBC` instead of a directory for each barcode length.

2.3.5 Step 1.2 Trimming of reads (RADIS)

`RADIS` used the values of `radis_nttrim_read1_5p`, `radis_nttrim_read1_3p`, `radis_nttrim_read2_5p` and `radis_nttrim_read2_3p` provided in the configuration file to remove nucleotids from the 5' and 3' ends of the reads 1 and 2.

Output files are stored in the `02_TRIM` (for TRIMming) directory.

Note that even if you use barcodes of different length, reads are automatically cut to the same length (shortest)

2.3.6 Step 1.3 Removing PCR duplicates (Stacks / clone_filter)

The program `clone_filter` from the software `Stacks` is used on the trimmed reads 1 and 2 to remove PCR duplicates (*i.e.* Read1 / Read2 pairs that match perfectly).

A directory named `03_FIL` (for PCR clone FILtering) is created, in which files generated by `clone_filter` (`*clone_filter.log`), provide statistics on the number of PCR clones for each barcode/sample.

By the way, `RADIS` creates a file named `1.3.countclonefilter.tsv` in the `outdir` directory that summarised the distribution of PCR duplicates for the whole data set and allows calculation of the percentage of reads lost because of PCR duplicates.

`1.3.countclonefilter.tsv` file interpretation :

NumClones	Count	Intrepretation
1	101773	101773 reads were represented only once in the data set
2	1296	1296 clones where found in which reads were represented twice
3	413	413 clones where found in which reads were represented three times

Nota - sum of the second column will give you the total number of reads retained after PCR duplicates removal (should be equal to the total number of purified reads as reported in the `1.4.statspurereads.tsv` file, see below) - to obtain the total number of reads in the whole data set before clone filtering you can multiply the two first columns and add up the results (`1*101773 + 2*1296 + 3*413`) (this should be equal to the total number of reads obtained after `process_radtags`) - here, the percentage of reads lost because of

the presence of PCR duplicates is `[1-(101773 + 1296 + 413)] / (1*101773 + 2*1296 + 3*413) *100`. To have a best idea of the number of reads lost you may also compared the results of `process_radtags` with the number of reads reported in the `1.4.statspurereads.tsv` file (see below).

2.3.7 Step 1.4 Rename files (RADIS)

Output sequence files renamed after the sample codes are stored in the `04_PUR` directory (for PURified reads). A file named `1.4.statspurereads.tsv` that reports on the number of pure reads per sample (*i.e* reads remaining after demultiplexing, cleaning and PCR clone removal) is created in the `outdir` directory.

The data-cleaning step is now complete. The "pure" reads 1 outputs will be the inputs for the data-analysis step. Note that assembly of reads 2 is not yet implemented in `RADIS`.

2.4 Dig into the results (Step 2, data analysis)

If you run the whole pipeline, results of the data-analysis step are stored in the `ASSEMBLY_STACKS` subdirectory of the `outdir` directory. If you want to run the data-analysis step on the "pure" reads generated by the data-cleaning step, you have to execute the `RADIS_step2_data_analysis.pl` script.

As an example, to run this script on the "pure" reads generated by the data-cleaning step on the "single barcode data set" type the following commands in `/home/user/RADIS-1.0.0/`:

```
mkdir /home/user/RADIS-1.0.0/outdir_example_single_barcode_dataanalysis
#create a new outdir directory

(./bin/RADIS_step2_data_analyses.pl --listoffastqfiles
'/home/user/RADIS-1.0.0/outdir_example_single_barcode_datacleaning/*/04_PUR/*fq_1'
--outdir /home/user/RADIS-1.0.0/outdir_example_single_barcode_dataanalysis --cfg
/home/user/RADIS-1.0.0/cfg/RADIS_example_single_barcode.cfg
> stdout_dataanalysis) 2> stderr_dataanalysis
```

Note: the flag `--ncpus` (number of cpus) and `--without-raxml` (to stop the pipeline before `RAxML` analyses) may be added. After the `--listoffastqfiles` flag you must provide path to "pure" reads 1 files you want to include in the analysis (when running the whole pipeline, `RADIS.pl` takes all files contained in the `04_PUR` subdirectory). In the command above, all "pure" reads 1 generated by the data-cleaning analysis of the "single barcode" data set will be processed. If you want to restrict the analysis to some samples, you have to find a way to list path to all samples you want to keep. The easiest way is probably to create a "file of filenames (`.fof` extension)"

As an example, to analyse only three samples from the "single barcode data set" you may type the following commands in `/home/user/RADIS-1.0.0/`:

```
ls outdir_example_single_barcode_datacleaning/5/04_PUR/JRAS039*.fq_1 > mylist.fof
```

-Path to the first selected samples are added to the file `mylist.fof` which is created in the current directory (`/home/user/RADIS-1.0.0/`)

```
ls outdir_example_single_barcode_datacleaning/6/04_PUR/JRAS03787.fq_1 >> mylist.fof
```

-Path to the other selected sample is added to the file `mylist.fof` which is created in the current directory (`/home/user/RADIS-1.0.0/`)

```
more mylist.fof # print on screen the content of the mylist.fof file
#you should have the following output
outdir_example_single_barcode_datacleaning/5/04_PUR/JRAS03952.fq_1
outdir_example_single_barcode_datacleaning/5/04_PUR/JRAS03954.fq_1
outdir_example_single_barcode_datacleaning/6/04_PUR/JRAS03787.fq_1
```

to run the data-analysis step on these samples :

```
(./bin/RADIS_step2_data_analyses.pl --listoffastqfiles 'mylist.fof' --outdir
/home/user/RADIS-1.0.0/outdir_example_single_barcode_dataanalysis --cfg
/home/user/RADIS-1.0.0/cfg/RADIS_example_single_barcode.cfg
> stdout_dataanalysis) 2> stderr_dataanalysis
```

-Note: this analysis is meaningless as i) parameters in the `RADIS_example_single_barcode.cfg` are not adapted anymore (`radis_nsample_min`=2,4). ii) `RAxML` will not build up any tree with three taxa only... The idea was only to show how to create a `.fof` file.

Output files and subdirectories of the data-analysis step are created in the `outdir` directory

The directory structure you should get in the `outdir` directory once the data-analysis step is complete is detailed below:

```
ASSEMBLY_STACKS (if the full pipeline is executed) or outdir_dataanalysis
  - M2 # results for the first value of M (ustacks) tested (here, M=2)
    - 01_TAGS      # outputs of ustacks (TAGS files)
    - 02_CATALOG   # output of cstacks (CATALOG of loci) and some statistics
    - n4_03_SEL    #.phy files generated for the first value of n (cstacks) (here n=4)
                   #and all sample/loci SELection criteria + results of the RAxML analyses
    - n6_03_SEL    #.phy files generated for the second value of n (cstacks) (here n=6)
                   #and all sample/loci SELection criteria + results of the RAxML analyses
    - NO_PBLOCI-3 #.phy files , RAxML outputs for the analysis without ProBlematic LOCI
                    #(-3 : npbloci_cutoff=3)
  - M3 # contains the results for the second value of M (ustacks) tested (here, M=3)
    - 01_TAGS
    - 02_CATALOG
    - n4_03_SEL
    - n6_03_SEL
    - NO_PBLOCI-3
```

Here again, you can follow the progress of your analysis in the `stdout` file

2.4.1 Step 2.1 Building individual loci (Stacks / ustacks)

"Purified" reads 1 are processed individually by `ustacks` and a set of loci is produced for each sample. For each value of the parameter `stacks_ustacks_M` specified in the configuration file, a directory named `M$M` (`$M` = value of M) is created in the `outdir` directory. A directory named `01_TAGS` in which `ustacks` outputs (`*alleles.tsv`, `*tags.tsv`, `*snps.tsv`) are stored is created in each `M$M` directories.

2.4.2 Step 2.2 Building catalog loci (Stacks / cstacks)

During this step, individual loci contained in `01_TAGS` are merged into a catalog of loci by the program `cstacks`. `cstacks` outputs for each value of the parameter `stacks_cstacks_n` specified in the configuration

file are stored in the directory `02_CATALOG` (`batch_$n.catalog.alleles.tsv`, `batch_$n.catalog.snps.tsv`, `batch_$n.catalog.tags.tsv`).

In addition, `RADIS` creates a binary (presence/absence) matrix for each catalog of loci (`batch_$n.catalog.matrix.tsv`), in which rows are sample codes and columns are loci ID in the catalog. For each locus, a given sample may be represented once (1), or more than once, or not at all (0). Statistics on each catalog of loci are stored in the files `batch_$n.catalog.matrix.tsv.summary`.

Screenshot of an example batch_$n.catalog.matrix.tsv.summary

```
Number of samples in catalog before sample selection= 4
Number of loci in catalog before loci selection= 2320

Summary statistics for the number of sample per locus

 Min.   :1.000
 1st Qu.:1.000
 Median :1.000
 Mean   :1.073
 3rd Qu.:1.000
 Max.   :4.000

Number of loci per sample
          counts
JRAS03787    486
JRAS03886    444
JRAS03952    730
JRAS03954    829

Summary statistics for the number of loci per sample

 Min.   :444.0
 1st Qu.:475.5
 Median :608.0
 Mean   :622.2
 3rd Qu.:754.8
 Max.   :829.0

Percentage of loci for which samples have more than one sequence
JRAS03787 JRAS03886 JRAS03952 JRAS03954
0.0000000 0.0000000 0.1724138 0.5603448

Percentage of loci for which samples have three or more  sequences
JRAS03787 JRAS03886 JRAS03952 JRAS03954
0.0000000 0.0000000 0.0000000 0.1724138

Number of loci in which at least 5% of the samples are represented= 2320
Number of loci in which at least 10% of the samples are represented= 2320
Number of loci in which at least 25% of the samples are represented= 2320
Number of loci in which at least 35% of the samples are represented= 134
Number of loci in which at least 50% of the samples are represented= 134
Number of loci in which at least 65% of the samples are represented= 31
```

```
Number of loci in which at least 75% of the samples are represented= 31
Number of loci in which at least 85% of the samples are represented= 4
Number of loci in which all samples are represented= 4
```

These statistics may help you to estimate the quality of your data, adjust the value of the `ustacks/cstacks` parameters and select loci and samples for further analyses

Note that three more files are generated at this step, `catstats.r` that contains the script used to obtain the summary statistics as well as `catstats.r.stdout` and `catstats.r.stderr` which report on its execution

2.4.3 Step 2.3 Selection of loci and samples and Step 2.4 Building phylip files (RADIS)

In this step you can perform loci and sample selection to build up data sets that fit your needs.

Indeed,

- Samples may be represented by very few sequences in the catalog of loci, because of low DNA quality or troubles during library preparation/sequencing.
- Samples may have no sequence for a given catalog locus. This can for example happen because of restriction site loss or if cstacks matching parameter is not permissive enough (n is set too low). In this latter case orthologous loci from multiple samples will not be merged together and you can have missing data for a collection of catalog loci (e.g. less than 50% of your samples may have sequences for locus ID 18).
- More than one locus in a single sample can match with a single catalog locus. This can for example happen when the two alleles of a single loci are merged but also because the cstacks matching parameter is too permissive (n is set too high). In this case, paralogs and non-homologous loci can be merged together. Note here that if samples have more than one sequence for a given loci (i.e. more than one ustacks loci ID is listed in the "Sequence ID" column of the `catalog.tags.tsv file`), only the ustacks consensus sequence of the first loci in the list is kept in the analysis).

For loci and samples selection you can play with the value of three parameters in the configuration file

- `radis_nloci_min` : samples must have sequence for at least `radis_nloci_min loci` to be kept in the analysis. Set to 1 if you want to keep all samples (even those with a large amount of missing data) (e.g. `radis_nloci_min = 10000` means that only samples with at least 10,000 loci will be kept in the analysis). Note that only one value can be provided for this parameter.
- `radis_nsample_min` : only locus for which at least `radis_nsample_min` samples have sequences will be kept in the analysis (*e.g.* `radis_nsample_min = 12` means that only loci for which at least 12 samples have a sequence will be kept in the analysis; note that samples having less than `radis_nsample_min` are ignored for loci selection). Note that a list of values can be provided for this parameter.
- `radis_npbloci_cutoff` = "potentially problematic" loci for which at least one sample has `radis_npbloci_cutoff` sequences or more will be removed from the analysis (*e.g.* `radis_npbloci_cutoff = 3` means that loci for which at least one sample has 3 or more sequences will be excluded from the analysis). Thus, this parameter allows users to remove loci in which paralogs and non-homologous sequences may have been merged. You can skip this part of the analysis by setting this parameter to 0. Note that only one value can be provided for this parameter.

For each value of the parameter `stacks_cstacks_n` specified in the configuration file, directories named `n$n_03_SEL` (for SELection) are created in which you will find output files labelled with the values of `stacks_ustacks_M`, `stacks_cstacks_n`, `radis_nsample_min` and `radis_nloci_min`. For example a filename with the label `M2n4S2L100` results from an analysis with `stacks_ustacks_M = 2`, `stacks_cstacks_n = 4`, `radis_nSample_min = 2` and `radis_nLoci_min = 100`.

The following files are created by `RADIS` and stored in the `n$n_03_SEL` directories

- `stacks_$M$n$nsample_min$nloci_min.sel` : which contains the list of the catalog ID of the loci that meet your selection criteria
- `stacks_$M$n$nsample_min$nloci_min.sel.phy` : Phylip formatted file with samples and loci that meet your selection criteria. Note that these files contain the full sequence of each locus, not only the SNP. Indeed, as underlined by Leaché et al., 2015, the analysis of the "full sequences rather than just SNPs is preferable from the perspectives of branch length and topological accuracy". In a pipeline, it seemed more relevant to keep the full sequence, but users may extract the SNP from the data sets using other programs if desired.
- `ana.$nsample_min.vs.$nloci_min.r` : that contain the script used for loci and sample selection and `ana.$nsample_min.vs.$nloci_min.r.stdout` and `ana.$nsample_min.vs.$nloci_min.r.stderr` that report on its execution.

Percentages of missing data for each phylip files are provided in the stdout file.

Results from the analysis without "potentially problematic" loci, are stored in a directory named `NO_PBLOCI-$radis_npbloci_cutoff`. Besides the `n$n_03_SEL` directories you will find three files `ana.npbloci-$radis_npbloci_cutoff.r` that contains the script used to remove "potentially problematic" loci and `ana.npbloci-$radis_npbloci_cutoff.r.stdout` and `ana.npbloci-$radis_npbloci_cutoff.r.stderr` which report on its execution.

2.4.4 Step 2.5 Phylogenetic analysis (RAxML)

If the flag `--without-raxml` was not included in the command, `RAxML` analyses are performed and classic output files from the program are stored in the `n$n_03_SEL` directories.

2.5 How can I mix data from many RAD libraries?

If different sets of barcodes have been used to tag samples, `.fastq.gz` files can be analysed at the same time. If the same set of barcodes was use to tag samples, you have to run the cleaning step on each raw data set first.

You can choose the samples you want to keep for the second step, or keep them all, up to you. Here again (as for the selection of specific samples after cleaning of raw data from a single library), the easiest way to provide a list of samples you want to include in the analysis is to create a "file of filenames (`.fof` extension)".

Let's imagine you want to merge some samples from the "single and double barcode" data sets, you should type the following commands (do not try to analyse the resulting data set, you will not get any homologous RAD tags...).

```
cd /home/user/RADIS-1.0.0/

ls outdir_example_single_barcode_datacleaning/5/04_PUR/JRAS039*.fq_1 > mylist.fof
# path to selected samples from the first library are added to the file "mylist.fof"
```

```
ls outdir_example_double_barcode_datacleaning/PBC/04_PUR/TRIC*.fq_1 >> mylist.fof
# path to selected samples from the second library are added to the file "mylist.fof"

more mylist.fof # print on screen the content of the mylist.fof file
#you should have the following output
#outdir_example_single_barcode_datacleaning/5/04_PUR/JRAS03952.fq_1
#outdir_example_single_barcode_datacleaning/5/04_PUR/JRAS03954.fq_1
#outdir_example_double_barcode_datacleaning/PBC/04_PUR/TRIC00013_0199.fq_1
#outdir_example_double_barcode_datacleaning/PBC/04_PUR/TRIC00027_1199.fq_1
#outdir_example_double_barcode_datacleaning/PBC/04_PUR/TRIC00027_3199.fq_1

mkdir outdir_combined  #outdir to store results of the data analysis part

(./bin/RADIS_step2_data_analyses.pl --listoffastqfiles 'mylist.fof' --outdir
/home/user/RADIS-1.0.0/outdir_combined --cfg /home/user/RADIS-1.0.0/cfg/RADIS_combined.cfg
> stdout_datacombinedanalysis) 2> stderr_datacombinedanalysis
```

3 Cited references

Catchen J., Amores A., Hohenlohe P., Cresko W., Postlethwait J. (2011). Stacks: building and genotyping loci de novo from short-read sequences. *G3: Genes, Genomes, Genetics*, 1 : 171-182.

Catchen J., Hohenlohe P.A., Bassham S., Amores A., Cresko W.A. (2013) Stacks: an analysis tool set for population genomics. *Molecular Ecology*, 22 : 3124-3140.

Cruaud A., Gautier M., Galan M., Foucaud J., Sauné L., Genson G., Dubois E., Nidelet S., Deuve T., Rasplus J.-Y. (2014). Empirical assessment of RAD sequencing for interspecific phylogeny. *Molecular Biology and Evolution* 31, 1272-1274.

Leaché A.D., Bandury B.L., Felsenstein J., Nieto-Montes de Oca A., Stamatakis A. (2015). Short tree, long tree, right tree, wrong tree: new acquisition bias corrections for inferring SNP phylogenies. *Systematic Biology.* 64, 1032-1047.

Stamatakis A. (2006 a) Phylogenetic models of rate heterogeneity: A High Performance Computing Perspective. *International Parallel and Distributed Processing Symposium (IPDPS 2006), Rhodes Island, Greece*, 8 pp.

Stamatakis A. (2006 b) RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22 : 2688-2690.