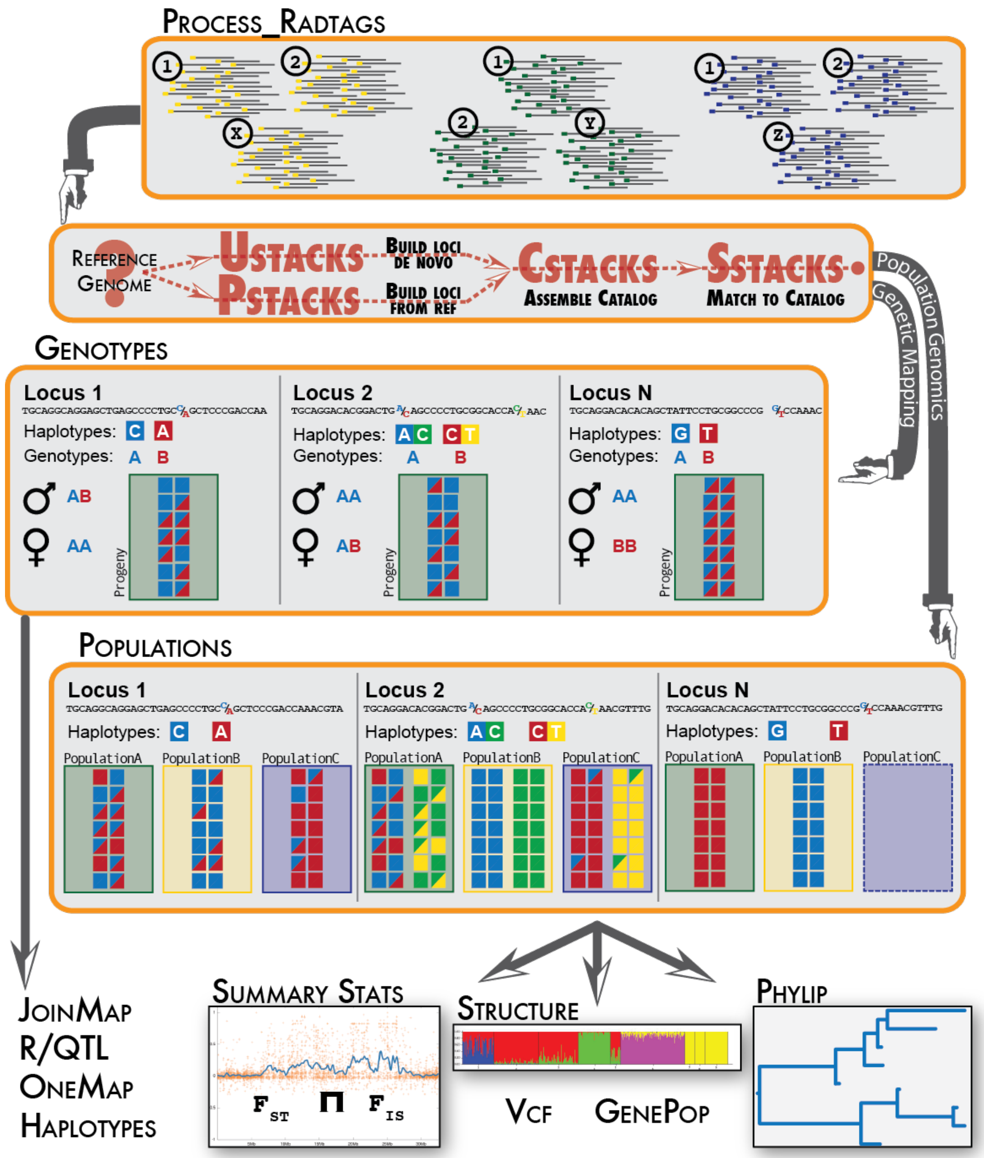
Estimating genetic diversity and population information from short read (ddRAD-seq) type data

03 – denovo_map

Chris Barratt (sDiv / Evolution and Adaptation)

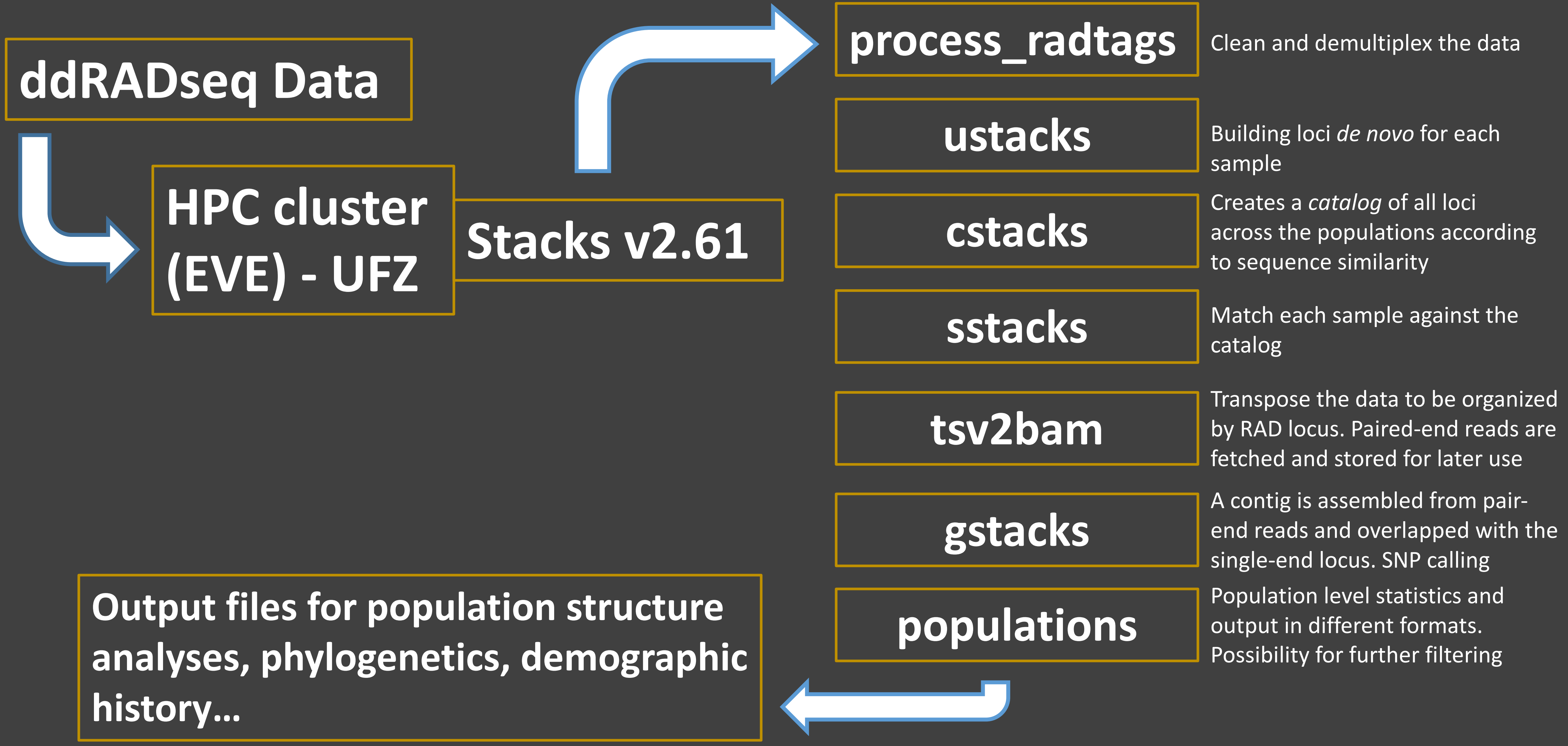
Laura Mendez (Evolution and Adaptation)

Assistant: Dimas Calderon (Evolution and Adaptation)



Rochette, Rivera-Colón & Catchen, 2019 - Mol. Ecology

ddRAD sequencing



ddRADseq Data

HPC cluster (EVE) - UFZ

Stacks v2.61

process_radtags Clean and demultiplex the data

ustacks Building loci *de novo* for each sample

cstacks Creates a *catalog* of all loci across the populations according to sequence similarity

sstacks Match each sample against the catalog

tsv2bam Transpose the data to be organized by RAD locus. Paired-end reads are fetched and stored for later use

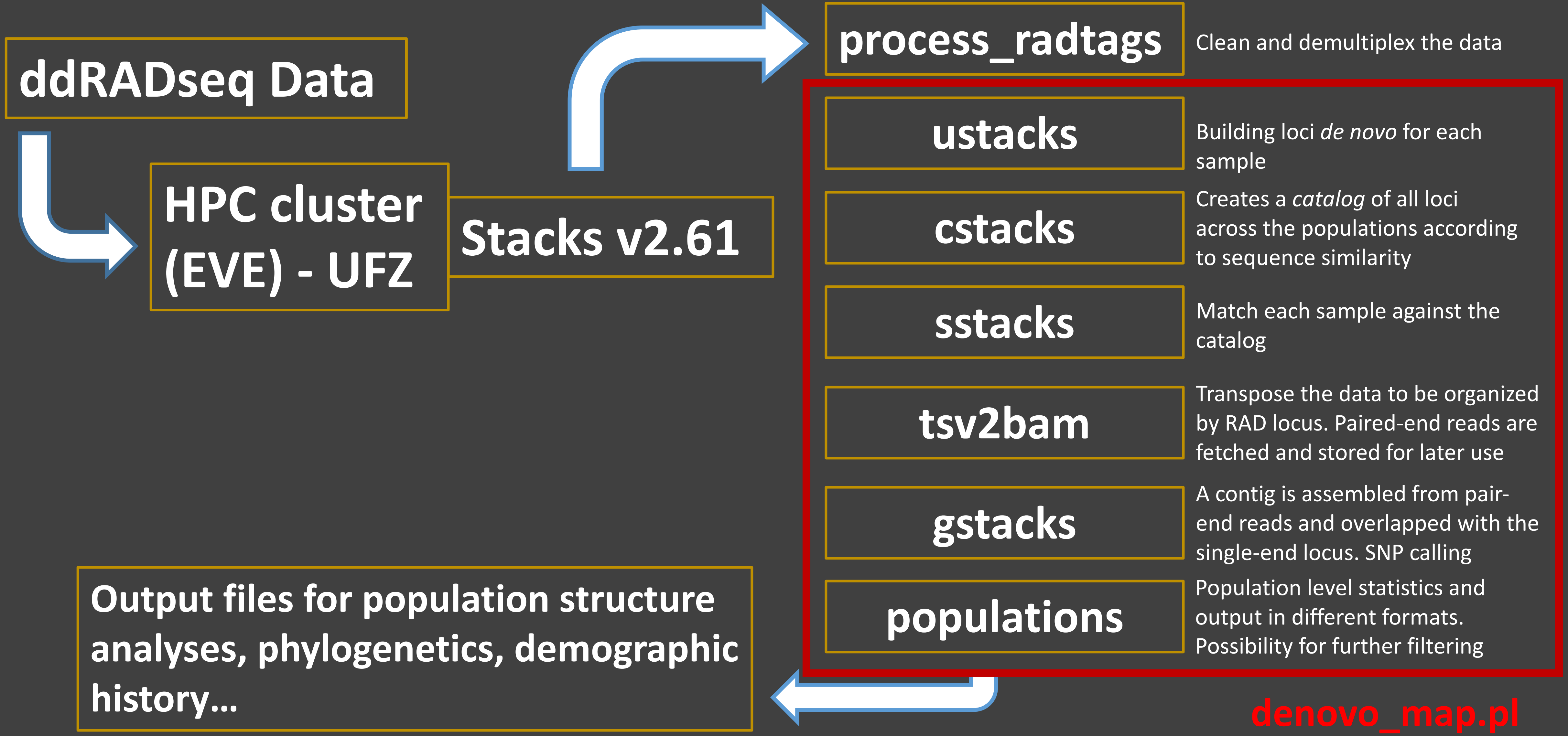
gstacks A contig is assembled from pair-end reads and overlapped with the single-end locus. SNP calling

populations Population level statistics and output in different formats. Possibility for further filtering

Output files for population structure analyses, phylogenetics, demographic history…

Rochette, Rivera-Colón & Catchen, 2019 - Mol. Ecology

ddRAD sequencing



ddRADseq Data

HPC cluster (EVE) - UFZ

Stacks v2.61

process_radtags Clean and demultiplex the data

ustacks Building loci *de novo* for each sample

cstacks Creates a *catalog* of all loci across the populations according to sequence similarity

sstacks Match each sample against the catalog

tsv2bam Transpose the data to be organized by RAD locus. Paired-end reads are fetched and stored for later use

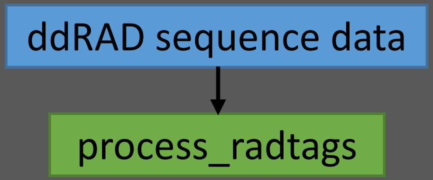
gstacks A contig is assembled from pair-end reads and overlapped with the single-end locus. SNP calling

populations Population level statistics and output in different formats. Possibility for further filtering

denovo_map.pl

Output files for population structure analyses, phylogenetics, demographic history…

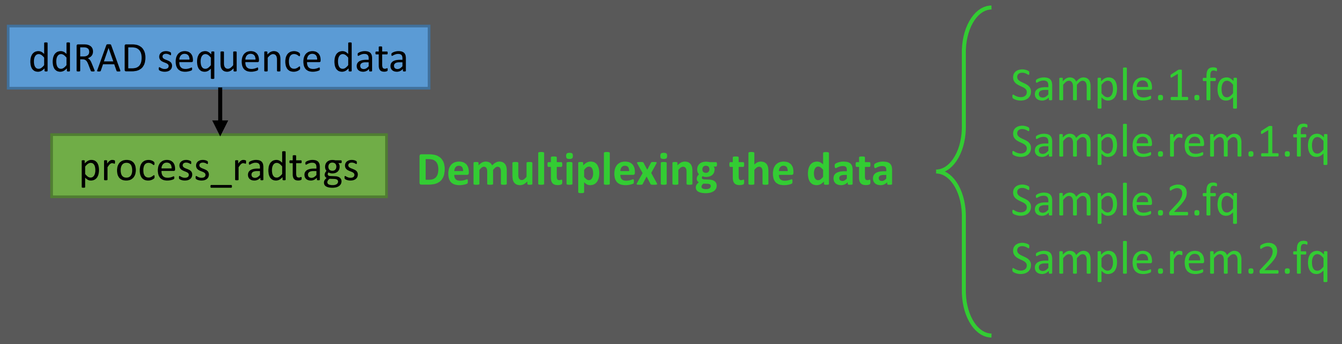
Bioinformatics Workflow



ddRAD sequence data

process_radtags

Bioinformatics Workflow



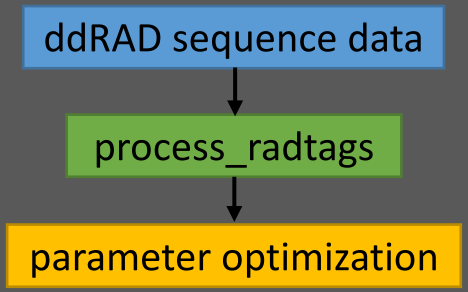
ddRAD sequence data

process_radtags

Demultiplexing the data

Sample.1.fq
Sample.rem.1.fq
Sample.2.fq
Sample.rem.2.fq

Bioinformatics Workflow

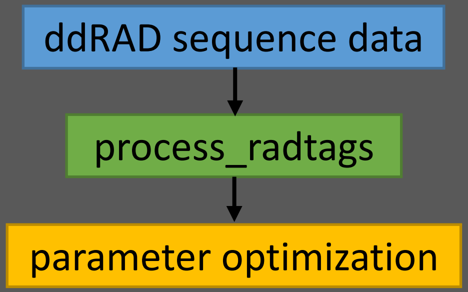


ddRAD sequence data

process_radtags

parameter optimization

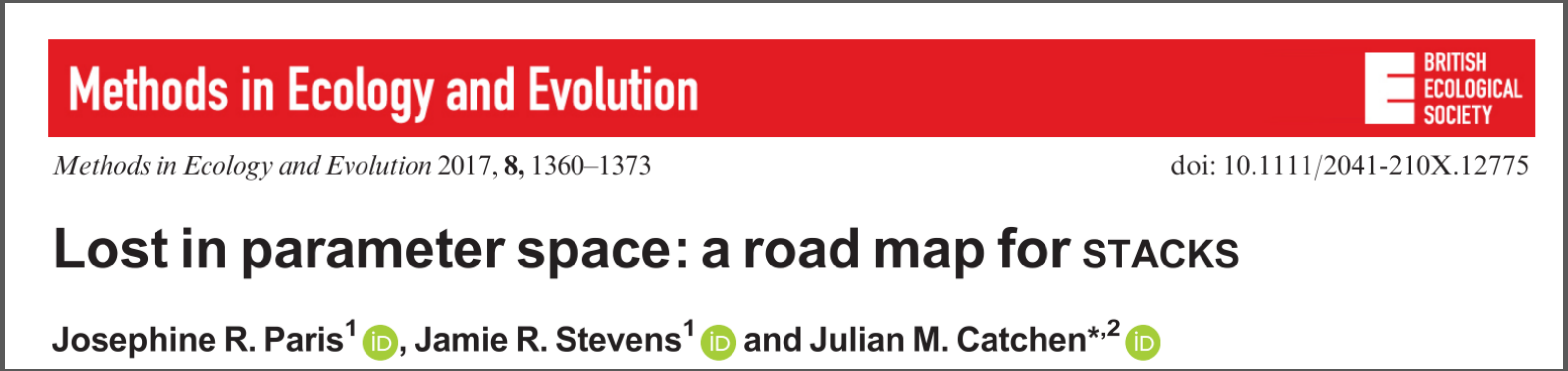
Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization



Methods in Ecology and Evolution

BRITISH ECOLOGICAL SOCIETY

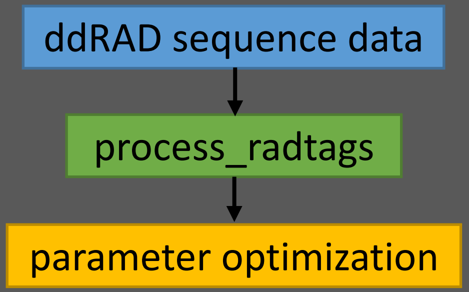
Methods in Ecology and Evolution 2017, **8,** 1360–1373

doi: 10.1111/2041-210X.12775

Lost in parameter space: a road map for STACKS

Josephine R. Paris[1], Jamie R. Stevens[1] and Julian M. Catchen*,[2]

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

m - depth coverage
M - mismatches allowed within indv. (ustacks)
n - mismatches allowed between indv. (cstacks)

“Find a balance between obtaining true polymorphism and introducing sequencing error”

m - Minimum depth coverage (minimum stack depth parameter) ustacks

- Controls the number of raw reads required to form an initial stack.
- If set to a value of 3 then three or more identical reads must be found to consider those reads a stack. If a stack is formed with only two reads, then those reads are set aside and a stack is not constructed.
- If this parameter is set too low, then reads with convergent sequencing errors are likely to be erroneously labeled as stacks.
- If this parameter too high, then true alleles will not be recorded and will drop out of the analysis.

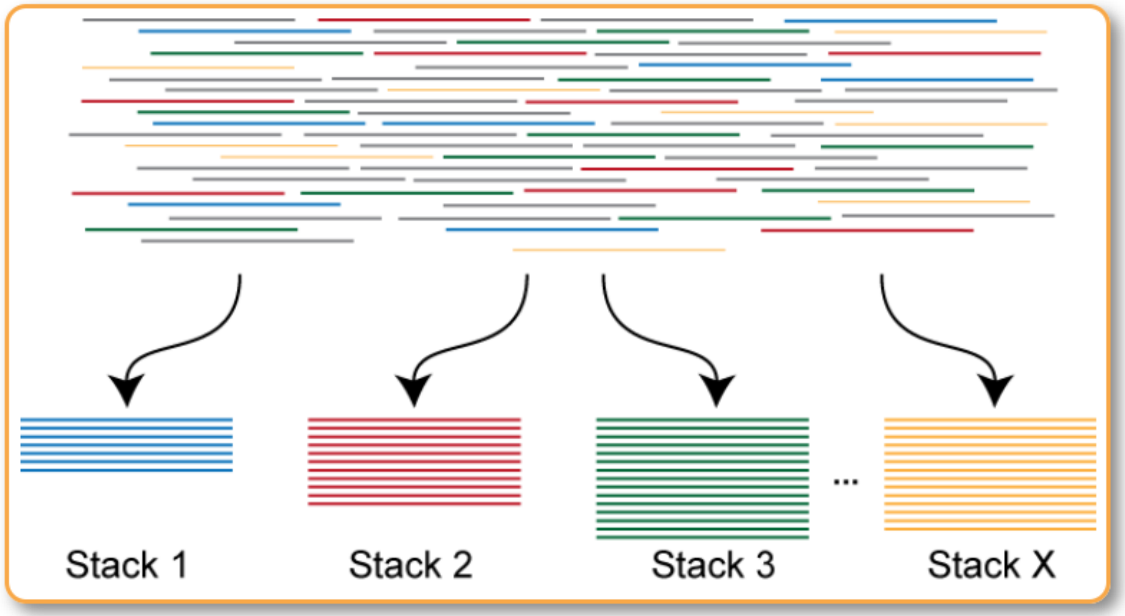


Figure 1. The initial stage of the **ustacks** *de novo* assembly algorithm forms exactly matching stacks from raw short-reads.

M - Mismatches allowed within indv. (Distance Allowed Between Stacks) ustacks

- The distance allowed between stacks parameter represents the number of nucleotides that may be different between two stacks in order to merge them. These nucleotide differences may be due to polymorphisms present between two alleles, or they may be due to sequencing error.
- If you set this parameter too low, then some loci will fail to be reconstructed. This means the SNPs contained in that locus will not be identified and this locus will appear as two loci to the remainder of the pipeline.
- Setting this parameter too high will allow repetitive sequence to chain together in to large, nonsensical loci.

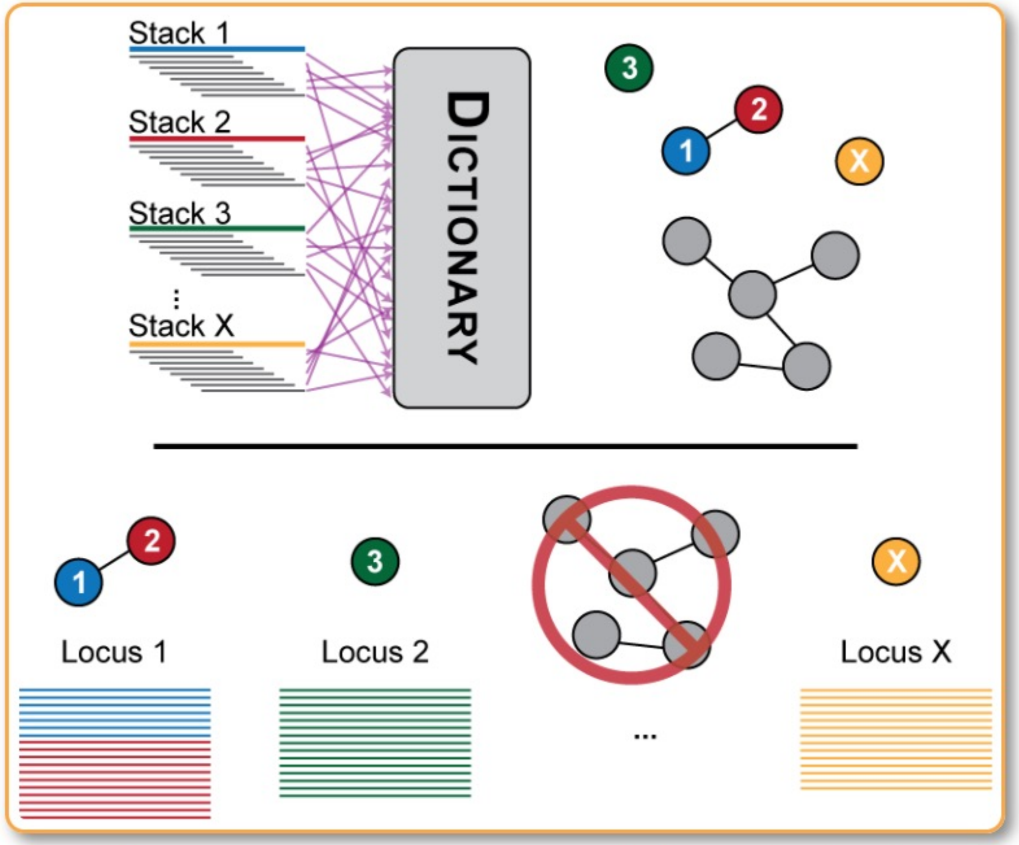


Figure 2. The consensus sequence from each stack is broken down into kmers in the second stage of the `ustacks` algorithm and stored in a dictionary. Two stacks that have a certain number of k-mers in common are considered as potentially matching and are aligned together. If the number of nucleotide mismatches are less than the distance allwoed between stacks the stacks are merged into a locus.

n - mismatches allowed between indv. (Distance Between Catalog Loci) cstacks

- Once loci are built per sample, then, the data from each individual will be merged into a **catalog** (by the **cstacks** program), which is meant to contain all the loci and alleles in the population.
- If the **distance between catalog loci** parameter is greater than 0, then **cstacks** will use the consensus sequence from each locus to attempt to merge loci together across samples.
- if you set this parameter too low there will be loci across individuals that are represented independently in the catalog that are truly the same locus.
- If you set this parameter too high, you will again allow loci close together in sequence space to chain together and create big, erroneous loci in the catalog.

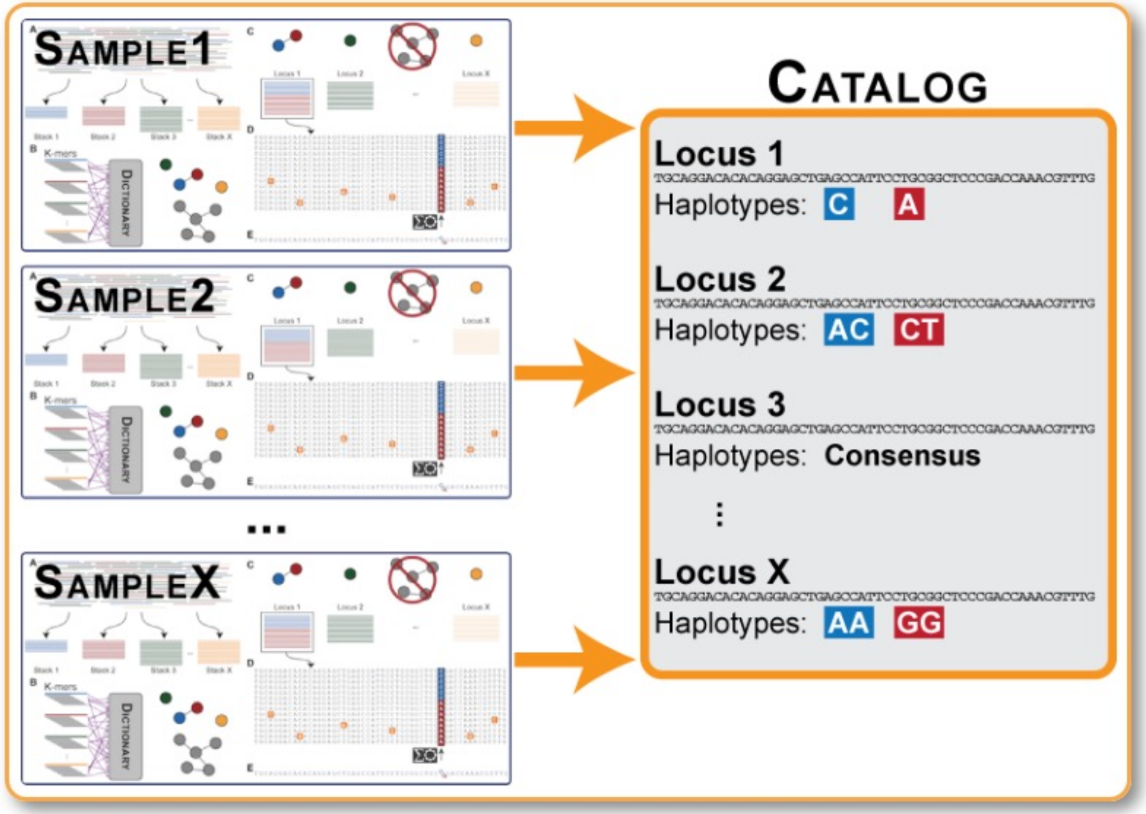
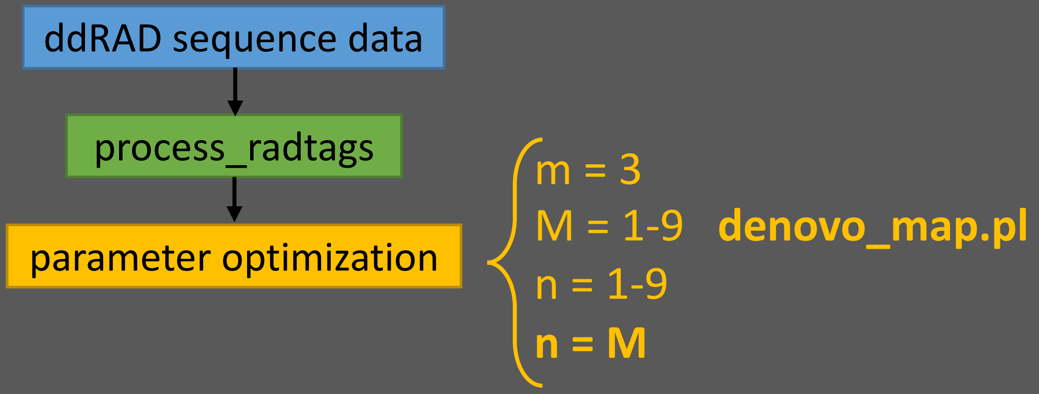

SAMPLE1
SAMPLE2
...
SAMPLEX
DICTIONARY
K-mers
CATALOG
Locus 1
TGCAGGACACACAGGAGCTGAGCCATTCCTGCGGCTCCCGACCAAACGTTTG
Haplotypes: C A
Locus 2
TGCAGGACACACAGGAGCTGAGCCATTCCTGCGGCTCCCGACCAAACGTTTG
Haplotypes: AC CT
Locus 3
TGCAGGACACACAGGAGCTGAGCCATTCCTGCGGCTCCCGACCAAACGTTTG
Haplotypes: Consensus
⋮
Locus X
TGCAGGACACACAGGAGCTGAGCCATTCCTGCGGCTCCCGACCAAACGTTTG
Haplotypes: AA GG

Figure 4. Catalog construction.

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

m = 3

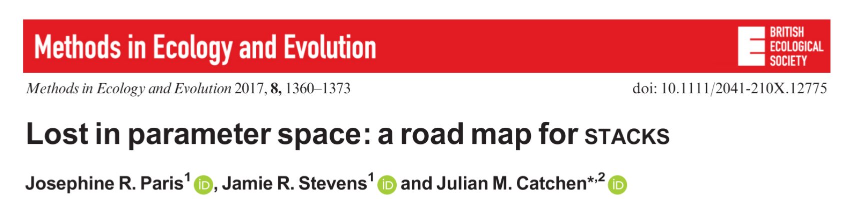
M = 1-9

n = 1-9

n = M

denovo_map.pl

“Find a balance between obtaining true polymorphism and introducing sequencing error”



Methods in Ecology and Evolution

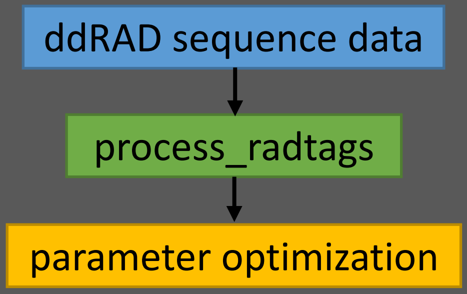
BRITISH ECOLOGICAL SOCIETY

Methods in Ecology and Evolution 2017, **8,** 1360–1373 doi: 10.1111/2041-210X.12775

Lost in parameter space: a road map for STACKS

Josephine R. Paris[1], Jamie R. Stevens[1] and Julian M. Catchen*[,2]

Bioinformatics Workflow



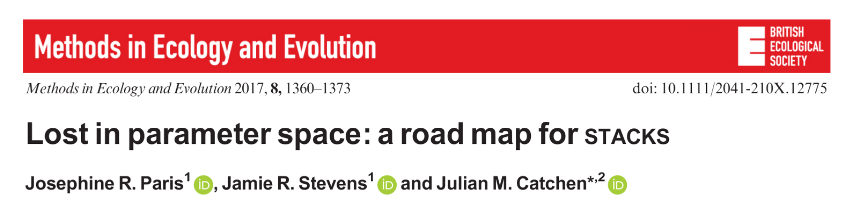
ddRAD sequence data

process_radtags

parameter optimization

m = 3
M = 1-9 **denovo_map.pl -> populations -r 0.80**
n = 1-9
n = M

“a locus must be found in 80% of individuals of a single population to be processed”



Methods in Ecology and Evolution

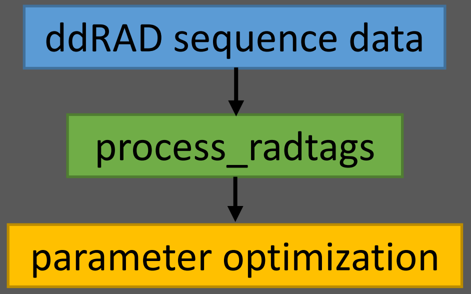
BRITISH ECOLOGICAL SOCIETY

Methods in Ecology and Evolution 2017, **8,** 1360–1373 doi: 10.1111/2041-210X.12775

Lost in parameter space: a road map for STACKS

Josephine R. Paris[1], Jamie R. Stevens[1] and Julian M. Catchen*,[2]

Bioinformatics Workflow


ddRAD sequence data

process_radtags

parameter optimization

- Number of r80 polymorphic loci
- Number of new polymorphic found for each iteration of M

“Find a balance between obtaining true polymorphism and introducing sequencing error”

Methods in Ecology and Evolution

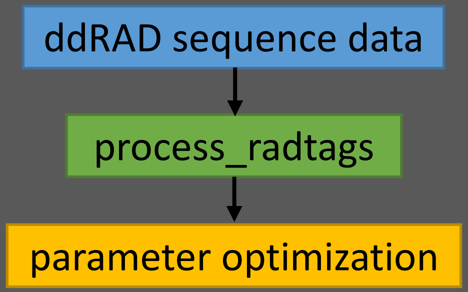
BRITISH ECOLOGICAL SOCIETY

Methods in Ecology and Evolution 2017, **8,** 1360–1373 doi: 10.1111/2041-210X.12775

Lost in parameter space: a road map for STACKS

Josephine R. Paris[1], Jamie R. Stevens[1] and Julian M. Catchen[*,2]

Bioinformatics Workflow



ddRAD sequence data

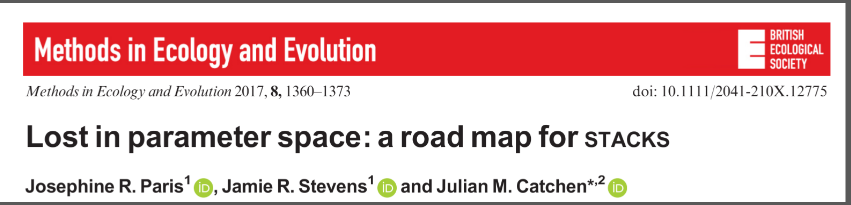
process_radtags

parameter optimization

- Number of r80 polymorphic loci
- Number of new polymorphic found for each iteration of M

“Find a balance between obtaining true polymorphism and introducing sequencing error”

~ 14 hours with a population map including 14 individuals



Methods in Ecology and Evolution

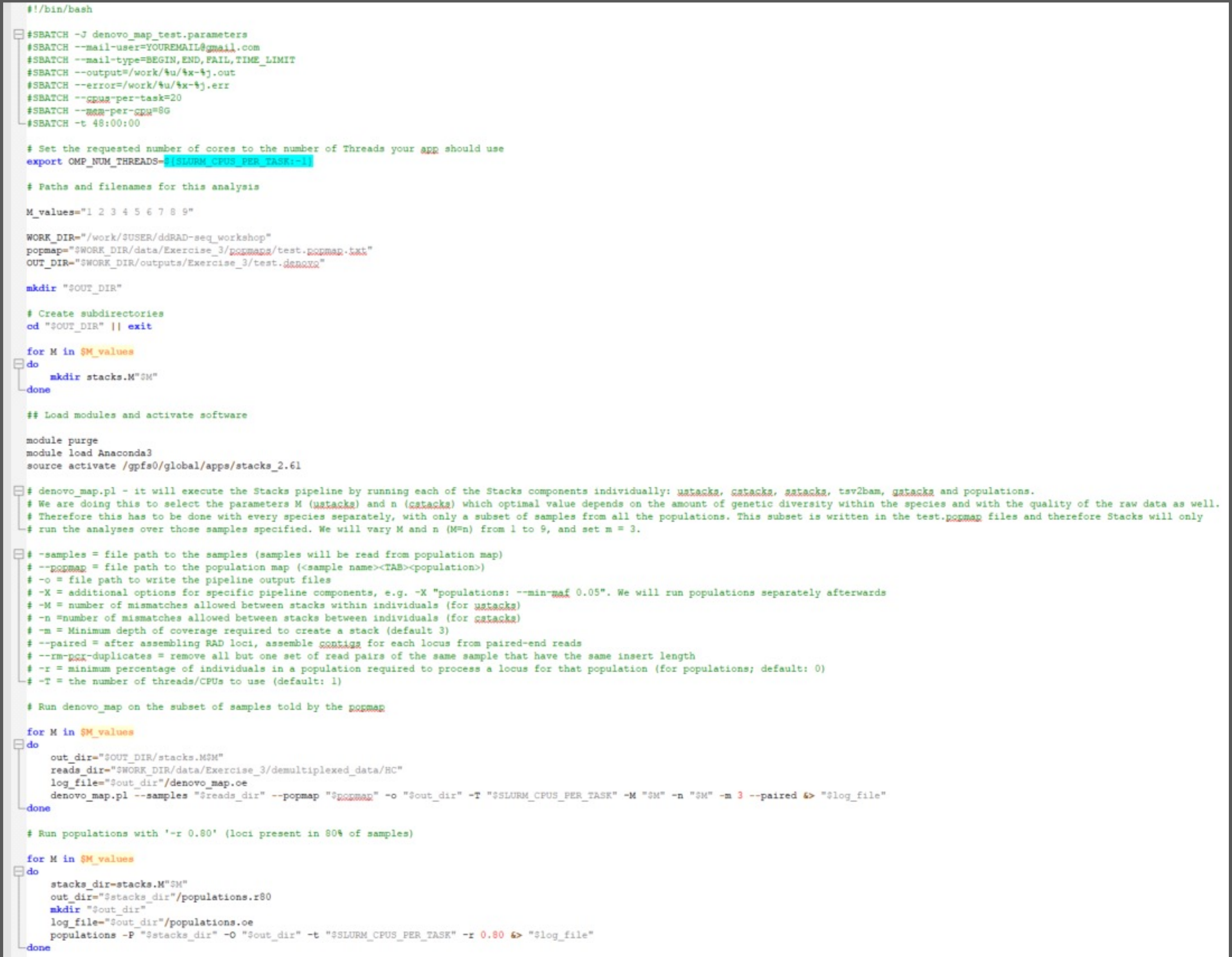
BRITISH ECOLOGICAL SOCIETY

Methods in Ecology and Evolution 2017, **8,** 1360–1373 doi: 10.1111/2041-210X.12775

Lost in parameter space: a road map for STACKS

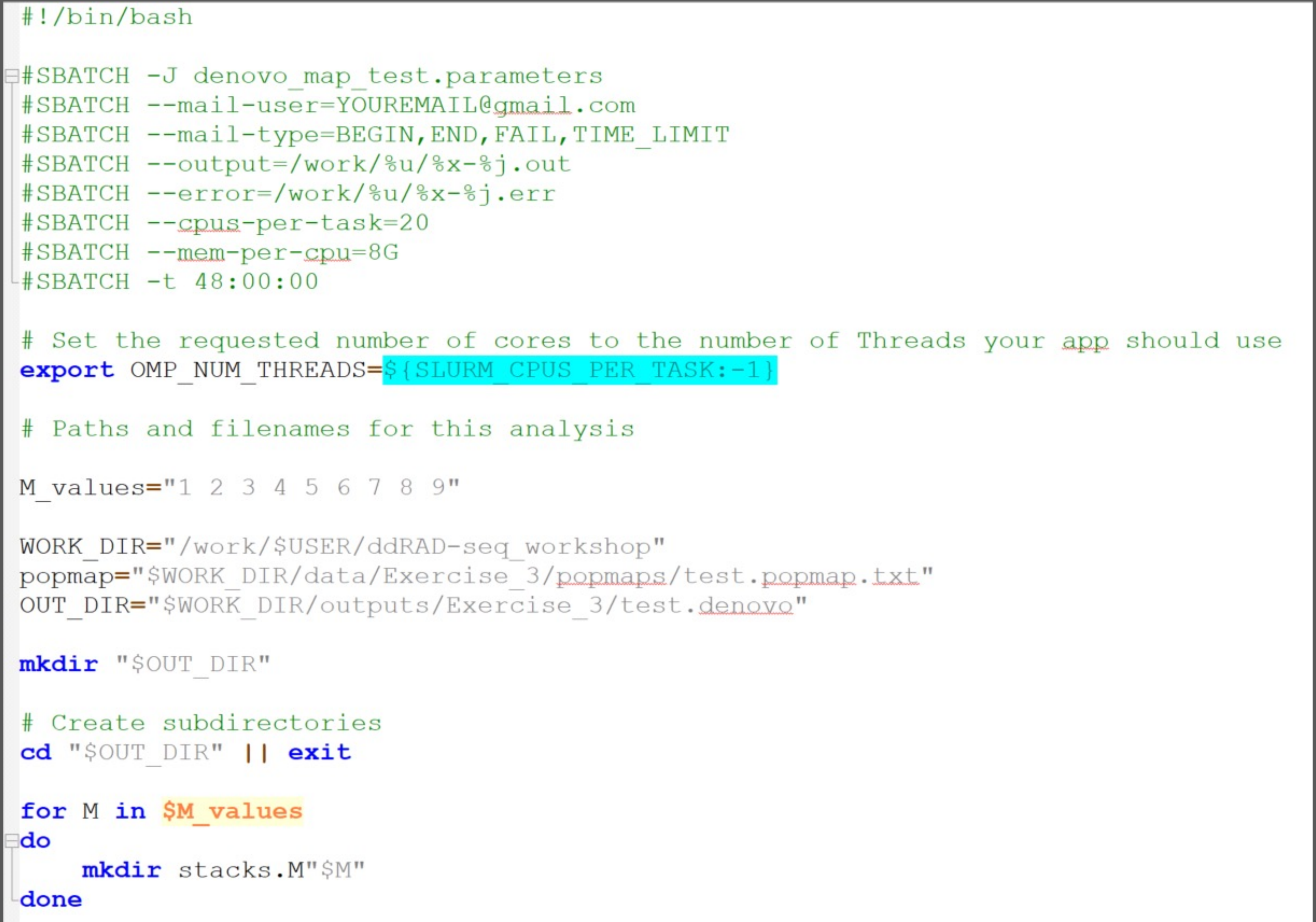
Josephine R. Paris[1], Jamie R. Stevens[1] and Julian M. Catchen*[,2]

01_denovo_map_test.parameters



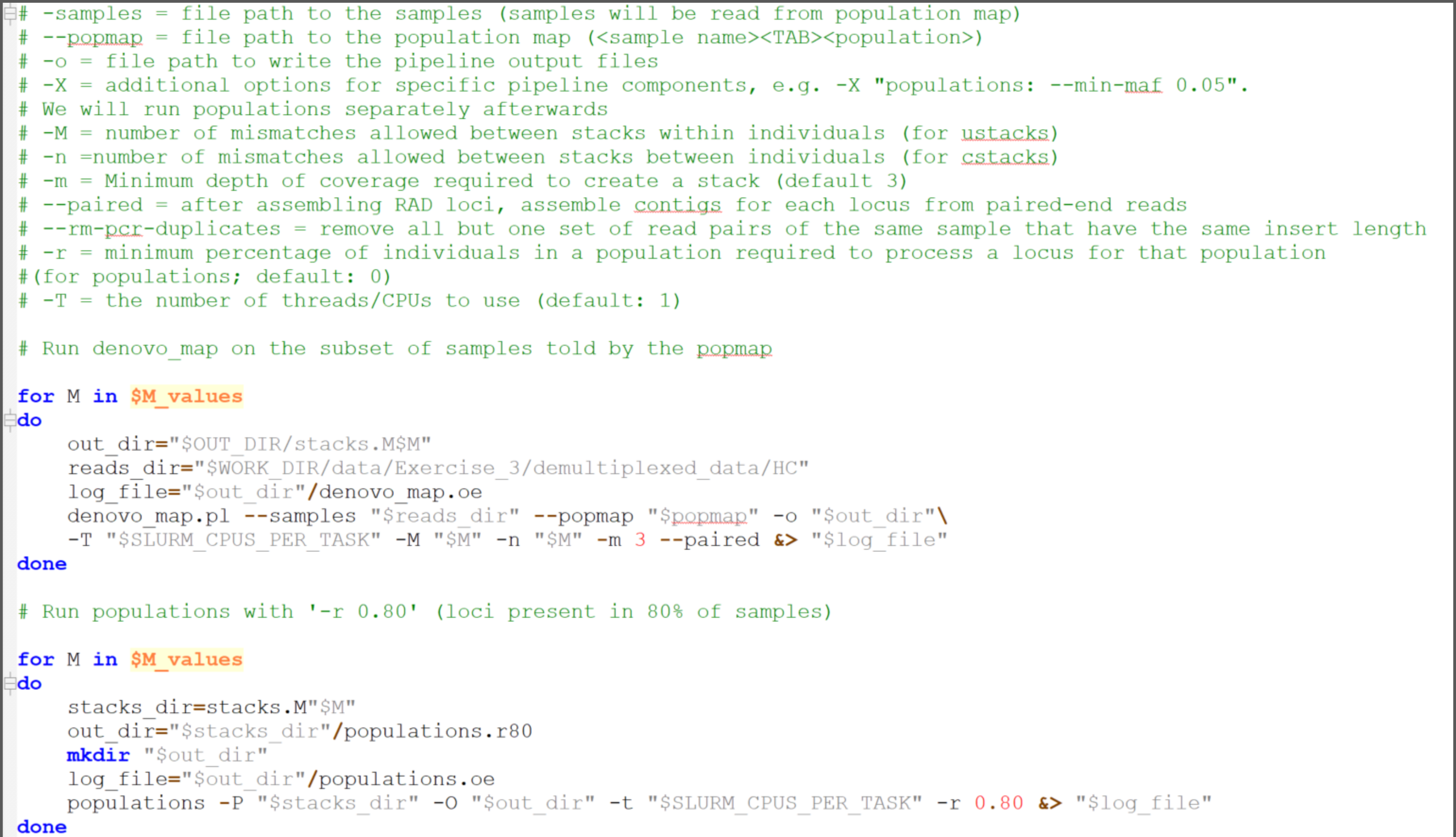
```
#!/bin/bash

#SBATCH -J denovo_map_test.parameters
#SBATCH --mail-user=YOUREMAIL@gmail.com
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT
#SBATCH --output=/work/%u/%x-%j.out
#SBATCH --error=/work/%u/%x-%j.err
#SBATCH --cpus-per-task=20
#SBATCH --mem-per-cpu=8G
#SBATCH -t 48:00:00

# Set the requested number of cores to the number of Threads your app should use
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK:-1}

# Paths and filenames for this analysis

M_values="1 2 3 4 5 6 7 8 9"

WORK_DIR="/work/$USER/ddRAD-seq_workshop"
popmap="$WORK_DIR/data/Exercise_3/popmaps/test.popmap.txt"
OUT_DIR="$WORK_DIR/outputs/Exercise_3/test.denovo"

mkdir "$OUT_DIR"

# Create subdirectories
cd "$OUT_DIR" || exit

for M in $M_values
do
    mkdir stacks.M"$M"
done

## Load modules and activate software

module purge
module load Anaconda3
source activate /gpfs0/global/apps/stacks_2.61

# denovo_map.pl - it will execute the Stacks pipeline by running each of the Stacks components individually: ustacks, cstacks, sstacks, tsv2bam, gstacks and populations.
# We are doing this to select the parameters M (ustacks) and n (cstacks) which optimal value depends on the amount of genetic diversity within the species and with the quality of the raw data as well.
# Therefore this has to be done with every species separately, with only a subset of samples from all the populations. This subset is written in the test.popmap files and therefore Stacks will only
# run the analyses over those samples specified. We will vary M and n (M=n) from 1 to 9, and set m = 3.

# -samples = file path to the samples (samples will be read from population map)
# --popmap = file path to the population map (<sample name><TAB><population>)
# -o = file path to write the pipeline output files
# -X = additional options for specific pipeline components, e.g. -X "populations: --min-maf 0.05". We will run populations separately afterwards
# -M = number of mismatches allowed between stacks within individuals (for ustacks)
# -n =number of mismatches allowed between stacks between individuals (for cstacks)
# -m = Minimum depth of coverage required to create a stack (default 3)
# --paired = after assembling RAD loci, assemble contigs for each locus from paired-end reads
# --rm-pcr-duplicates = remove all but one set of read pairs of the same sample that have the same insert length
# -r = minimum percentage of individuals in a population required to process a locus for that population (for populations; default: 0)
# -T = the number of threads/CPUs to use (default: 1)

# Run denovo_map on the subset of samples told by the popmap

for M in $M_values
do
    out_dir="$OUT_DIR/stacks.M$M"
    reads_dir="$WORK_DIR/data/Exercise_3/demultiplexed_data/HC"
    log_file="$out_dir"/denovo_map.oe
    denovo_map.pl --samples "$reads_dir" --popmap "$popmap" -o "$out_dir" -T "$SLURM_CPUS_PER_TASK" -M "$M" -n "$M" -m 3 --paired &> "$log_file"
done

# Run populations with '-r 0.80' (loci present in 80% of samples)

for M in $M_values
do
    stacks_dir=stacks.M"$M"
    out_dir="$stacks_dir"/populations.r80
    mkdir "$out_dir"
    log_file="$out_dir"/populations.oe
    populations -P "$stacks_dir" -O "$out_dir" -t "$SLURM_CPUS_PER_TASK" -r 0.80 &> "$log_file"
done
```

01_denovo_map_test.parameters

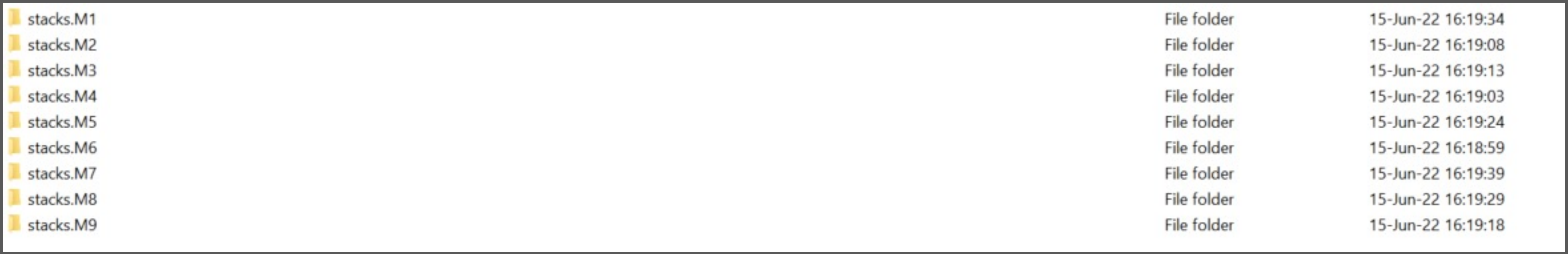


```bash
#!/bin/bash

#SBATCH -J denovo_map_test.parameters
#SBATCH --mail-user=YOUREMAIL@gmail.com
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT
#SBATCH --output=/work/%u/%x-%j.out
#SBATCH --error=/work/%u/%x-%j.err
#SBATCH --cpus-per-task=20
#SBATCH --mem-per-cpu=8G
#SBATCH -t 48:00:00

# Set the requested number of cores to the number of Threads your app should use
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK:-1}

# Paths and filenames for this analysis

M_values="1 2 3 4 5 6 7 8 9"

WORK_DIR="/work/$USER/ddRAD-seq_workshop"
popmap="$WORK_DIR/data/Exercise_3/popmaps/test.popmap.txt"
OUT_DIR="$WORK_DIR/outputs/Exercise_3/test.denovo"

mkdir "$OUT_DIR"

# Create subdirectories
cd "$OUT_DIR" || exit

for M in $M_values
do
    mkdir stacks.M"$M"
done
```

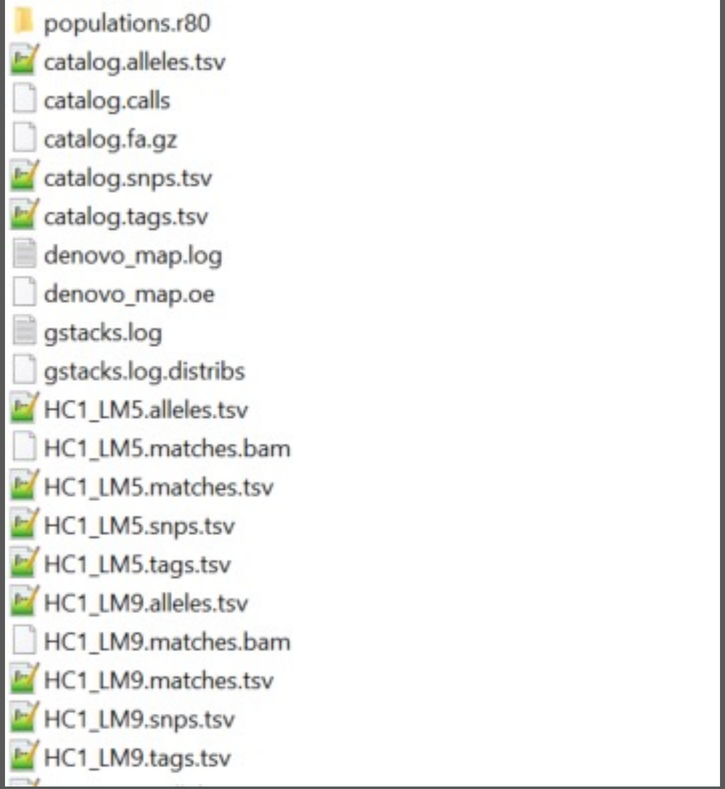
01_denovo_map_test.parameters



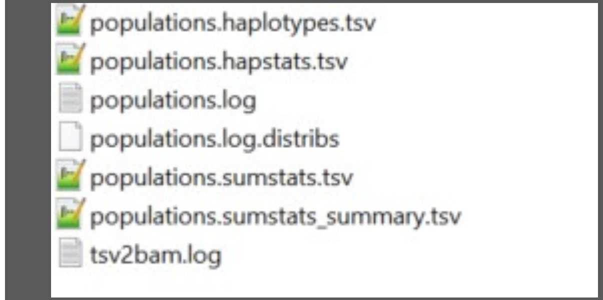
```
# -samples = file path to the samples (samples will be read from population map)
# --popmap = file path to the population map (<sample name><TAB><population>)
# -o = file path to write the pipeline output files
# -X = additional options for specific pipeline components, e.g. -X "populations: --min-maf 0.05".
# We will run populations separately afterwards
# -M = number of mismatches allowed between stacks within individuals (for ustacks)
# -n =number of mismatches allowed between stacks between individuals (for cstacks)
# -m = Minimum depth of coverage required to create a stack (default 3)
# --paired = after assembling RAD loci, assemble contigs for each locus from paired-end reads
# --rm-pcr-duplicates = remove all but one set of read pairs of the same sample that have the same insert length
# -r = minimum percentage of individuals in a population required to process a locus for that population
#(for populations; default: 0)
# -T = the number of threads/CPUs to use (default: 1)

# Run denovo_map on the subset of samples told by the popmap

for M in $M_values
do
    out_dir="$OUT_DIR/stacks.M$M"
    reads_dir="$WORK_DIR/data/Exercise_3/demultiplexed_data/HC"
    log_file="$out_dir"/denovo_map.oe
    denovo_map.pl --samples "$reads_dir" --popmap "$popmap" -o "$out_dir"\
    -T "$SLURM_CPUS_PER_TASK" -M "$M" -n "$M" -m 3 --paired &> "$log_file"
done

# Run populations with '-r 0.80' (loci present in 80% of samples)

for M in $M_values
do
    stacks_dir=stacks.M"$M"
    out_dir="$stacks_dir"/populations.r80
    mkdir "$out_dir"
    log_file="$out_dir"/populations.oe
    populations -P "$stacks_dir" -O "$out_dir" -t "$SLURM_CPUS_PER_TASK" -r 0.80 &> "$log_file"
done
```



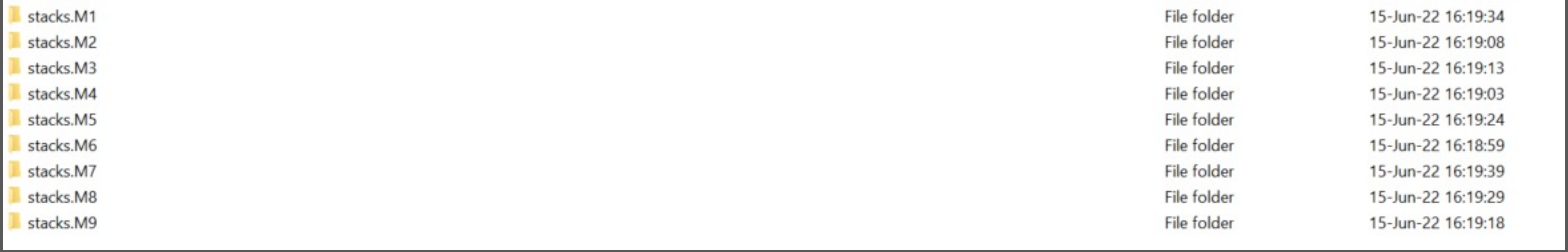
stacks.M1	File folder	15-Jun-22 16:19:34
stacks.M2	File folder	15-Jun-22 16:19:08
stacks.M3	File folder	15-Jun-22 16:19:13
stacks.M4	File folder	15-Jun-22 16:19:03
stacks.M5	File folder	15-Jun-22 16:19:24
stacks.M6	File folder	15-Jun-22 16:18:59
stacks.M7	File folder	15-Jun-22 16:19:39
stacks.M8	File folder	15-Jun-22 16:19:29
stacks.M9	File folder	15-Jun-22 16:19:18



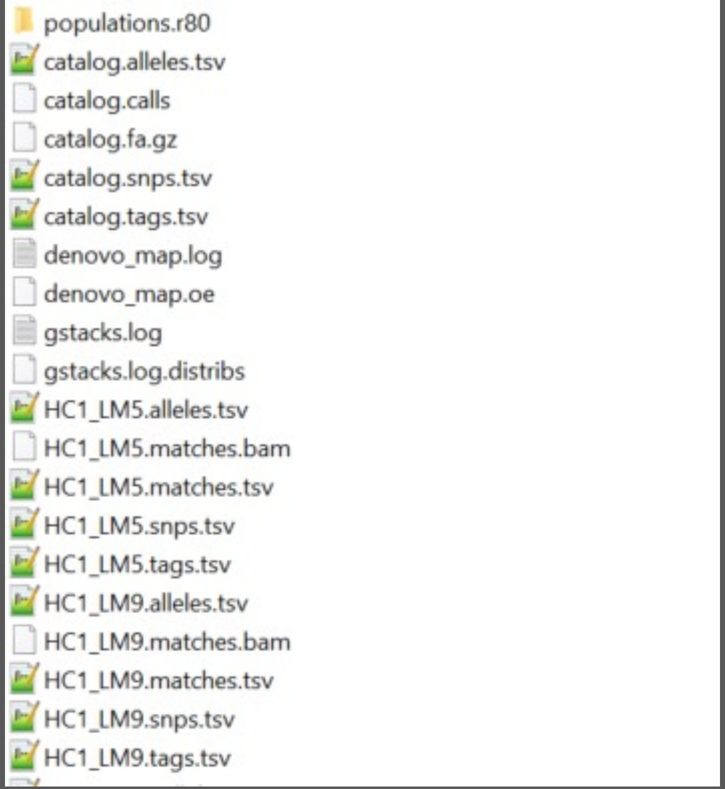
populations.r80
catalog.alleles.tsv
catalog.calls
catalog.fa.gz
catalog.snps.tsv
catalog.tags.tsv
denovo_map.log
denovo_map.oe
gstacks.log
gstacks.log.distribs
HC1_LM5.alleles.tsv
HC1_LM5.matches.bam
HC1_LM5.matches.tsv
HC1_LM5.snps.tsv
HC1_LM5.tags.tsv
HC1_LM9.alleles.tsv
HC1_LM9.matches.bam
HC1_LM9.matches.tsv
HC1_LM9.snps.tsv
HC1_LM9.tags.tsv



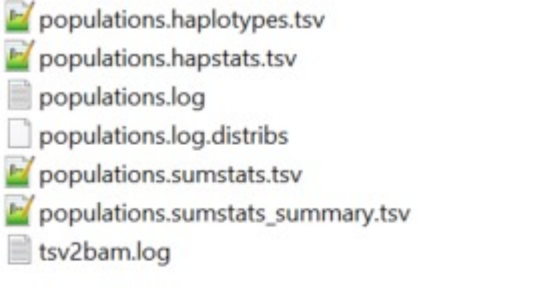
populations.haplotypes.tsv
populations.hapstats.tsv
populations.log
populations.log.distribs
populations.sumstats.tsv
populations.sumstats_summary.tsv
tsv2bam.log



stacks.M1	File folder	15-Jun-22 16:19:34
stacks.M2	File folder	15-Jun-22 16:19:08
stacks.M3	File folder	15-Jun-22 16:19:13
stacks.M4	File folder	15-Jun-22 16:19:03
stacks.M5	File folder	15-Jun-22 16:19:24
stacks.M6	File folder	15-Jun-22 16:18:59
stacks.M7	File folder	15-Jun-22 16:19:39
stacks.M8	File folder	15-Jun-22 16:19:29
stacks.M9	File folder	15-Jun-22 16:19:18



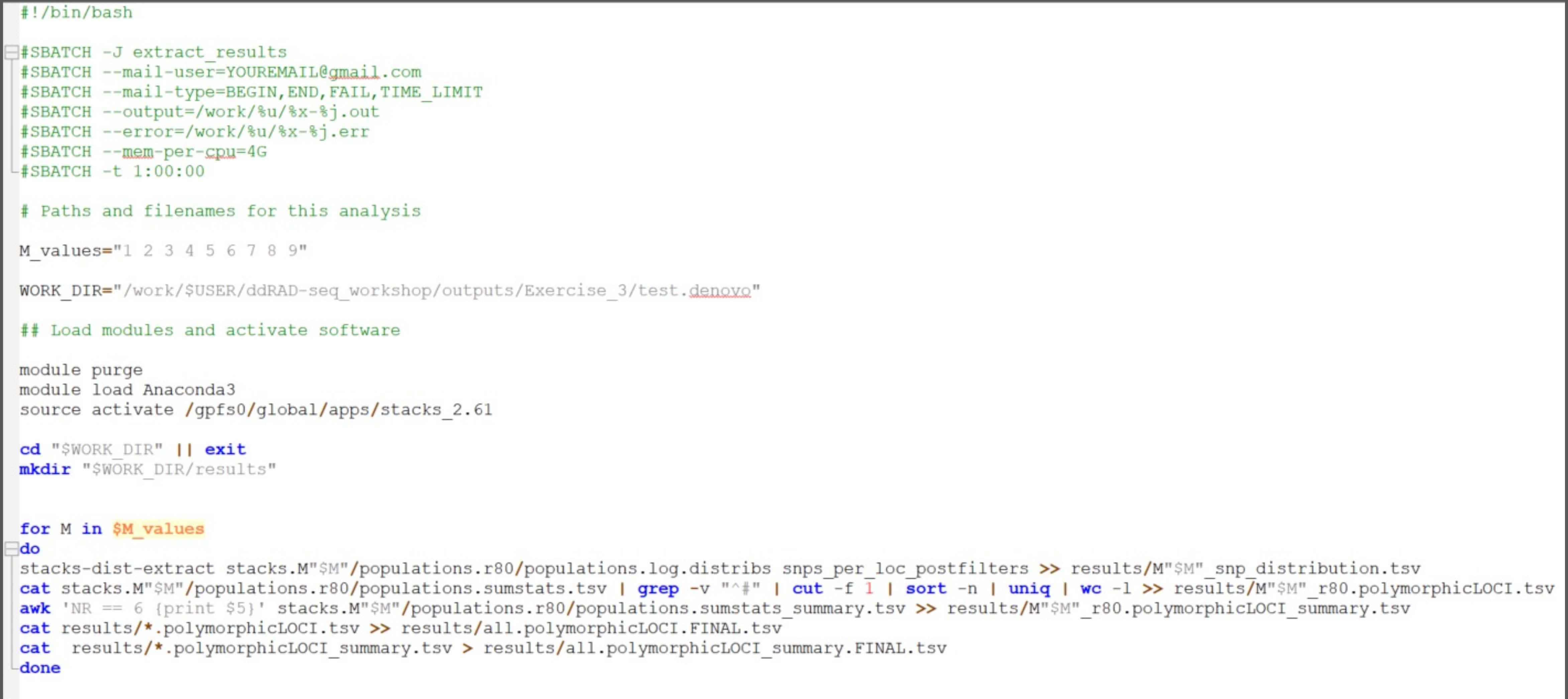
```
populations.r80
catalog.alleles.tsv
catalog.calls
catalog.fa.gz
catalog.snps.tsv
catalog.tags.tsv
denovo_map.log
denovo_map.oe
gstacks.log
gstacks.log.distribs
HC1_LM5.alleles.tsv
HC1_LM5.matches.bam
HC1_LM5.matches.tsv
HC1_LM5.snps.tsv
HC1_LM5.tags.tsv
HC1_LM9.alleles.tsv
HC1_LM9.matches.bam
HC1_LM9.matches.tsv
HC1_LM9.snps.tsv
HC1_LM9.tags.tsv
```



```
populations.haplotypes.tsv
populations.hapstats.tsv
populations.log
populations.log.distribs
populations.sumstats.tsv
populations.sumstats_summary.tsv
tsv2bam.log
```

Extract this information:
Number of r80 polymorphic loci
- Number of new polymorphic found for each iteration of M

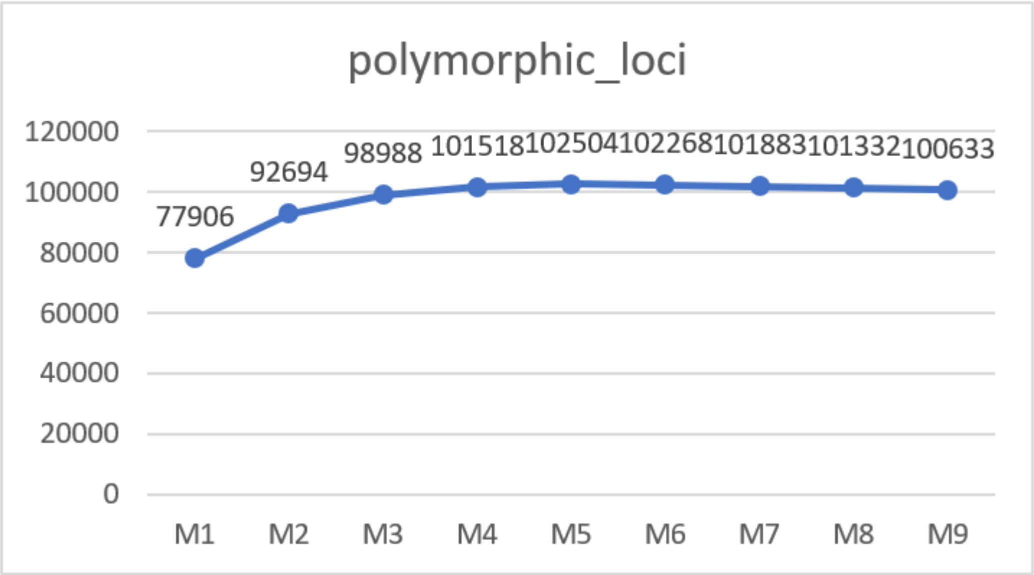
02_extract_results

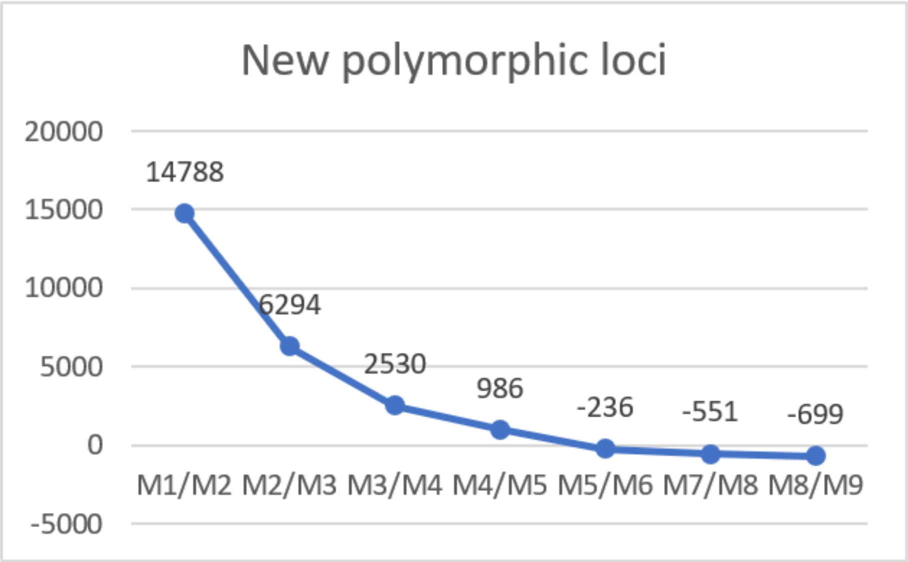


```
#!/bin/bash

#SBATCH -J extract_results
#SBATCH --mail-user=YOUREMAIL@gmail.com
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT
#SBATCH --output=/work/%u/%x-%j.out
#SBATCH --error=/work/%u/%x-%j.err
#SBATCH --mem-per-cpu=4G
#SBATCH -t 1:00:00

# Paths and filenames for this analysis

M_values="1 2 3 4 5 6 7 8 9"

WORK_DIR="/work/$USER/ddRAD-seq_workshop/outputs/Exercise_3/test.denovo"

## Load modules and activate software

module purge
module load Anaconda3
source activate /gpfs0/global/apps/stacks_2.61

cd "$WORK_DIR" || exit
mkdir "$WORK_DIR/results"


for M in $M_values
do
stacks-dist-extract stacks.M"$M"/populations.r80/populations.log.distribs snps_per_loc_postfilters >> results/M"$M"_snp_distribution.tsv
cat stacks.M"$M"/populations.r80/populations.sumstats.tsv | grep -v "^#" | cut -f 1 | sort -n | uniq | wc -l >> results/M"$M"_r80.polymorphicLOCI.tsv
awk 'NR == 6 {print $5}' stacks.M"$M"/populations.r80/populations.sumstats_summary.tsv >> results/M"$M"_r80.polymorphicLOCI_summary.tsv
cat results/*.polymorphicLOCI.tsv >> results/all.polymorphicLOCI.FINAL.tsv
cat  results/*.polymorphicLOCI_summary.tsv > results/all.polymorphicLOCI_summary.FINAL.tsv
done
```

Bioinformatics

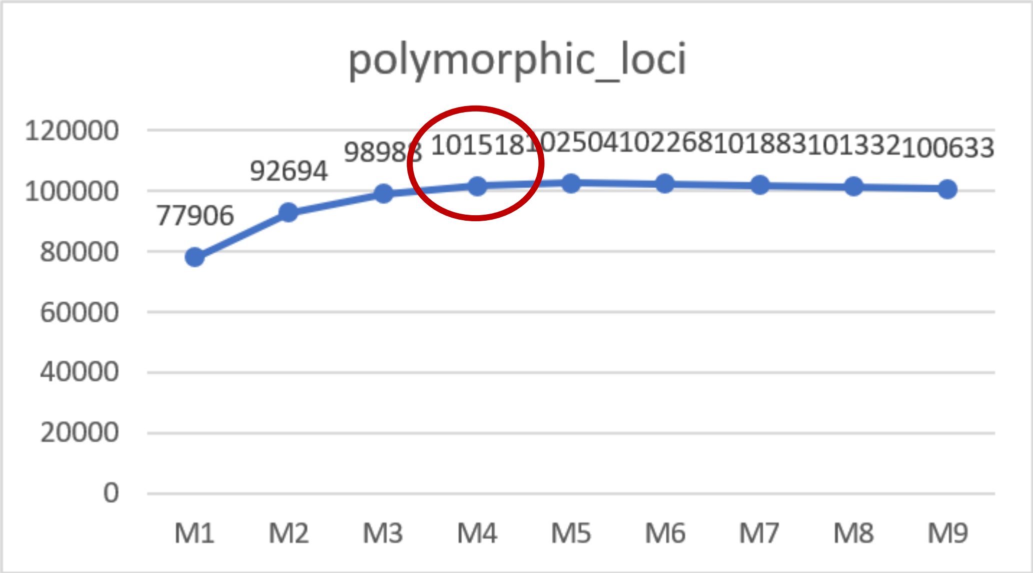
Bismarckia nobilis

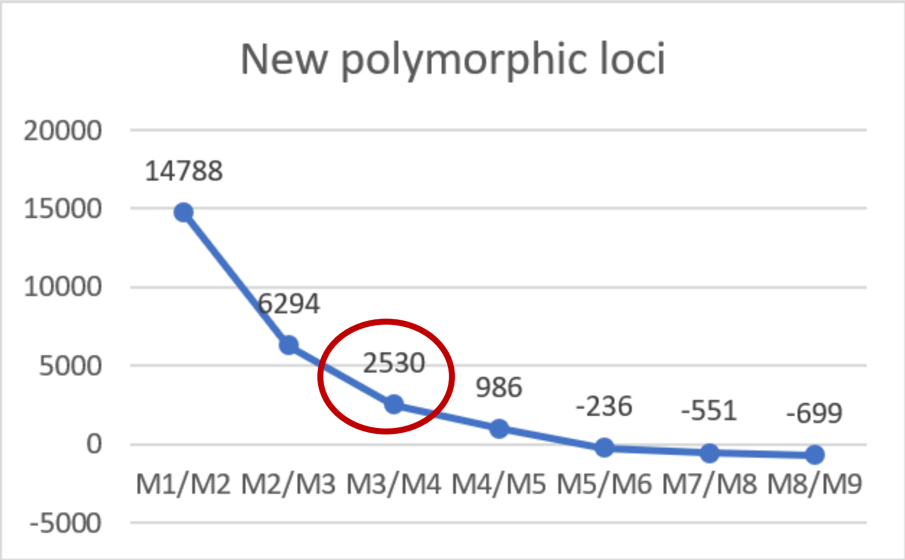




Bioinformatics

Bismarckia nobilis

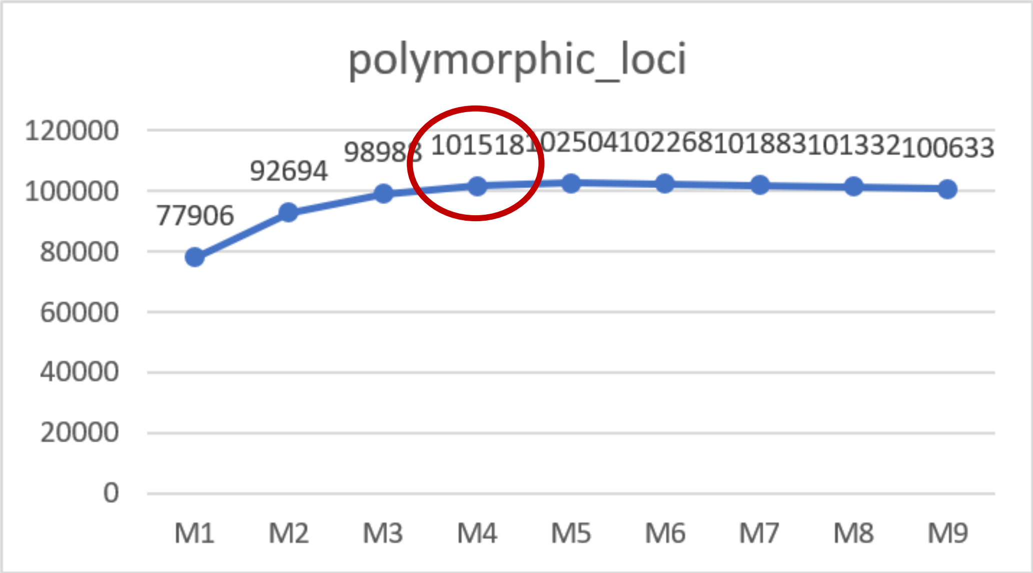


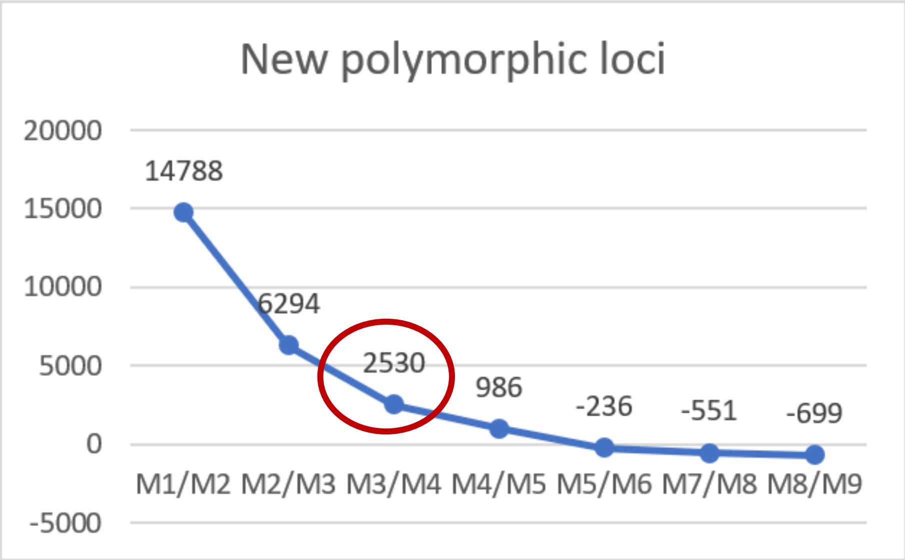


"Find a balance between obtaining true polymorphism and introducing sequencing error"

Bioinformatics

Bismarckia nobilis

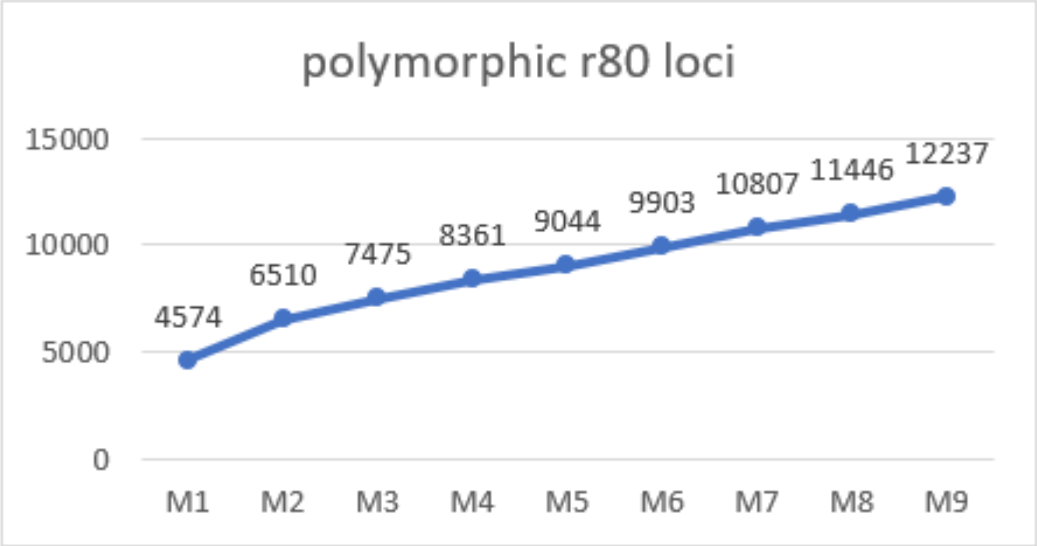




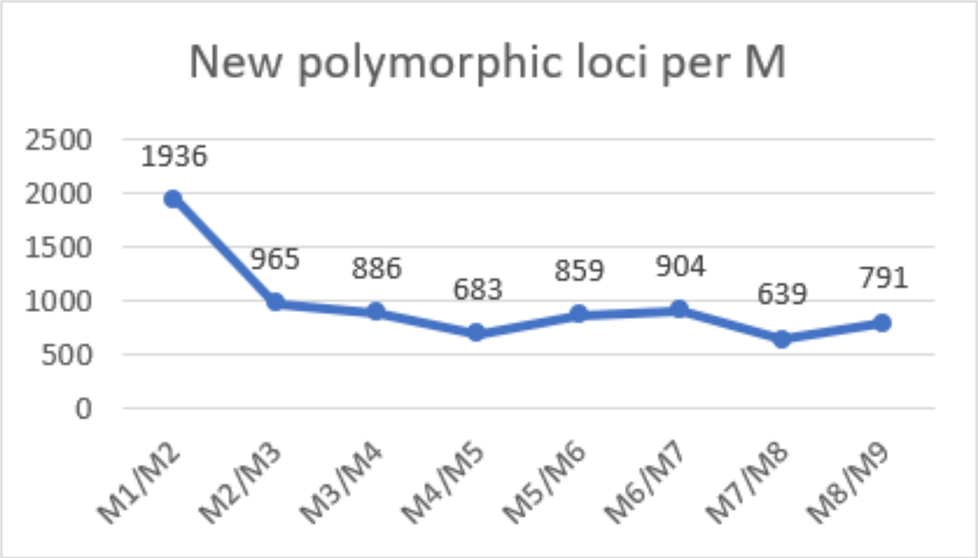
M = 4
M = n

Bioinformatics

Dypsis pinnatifrons



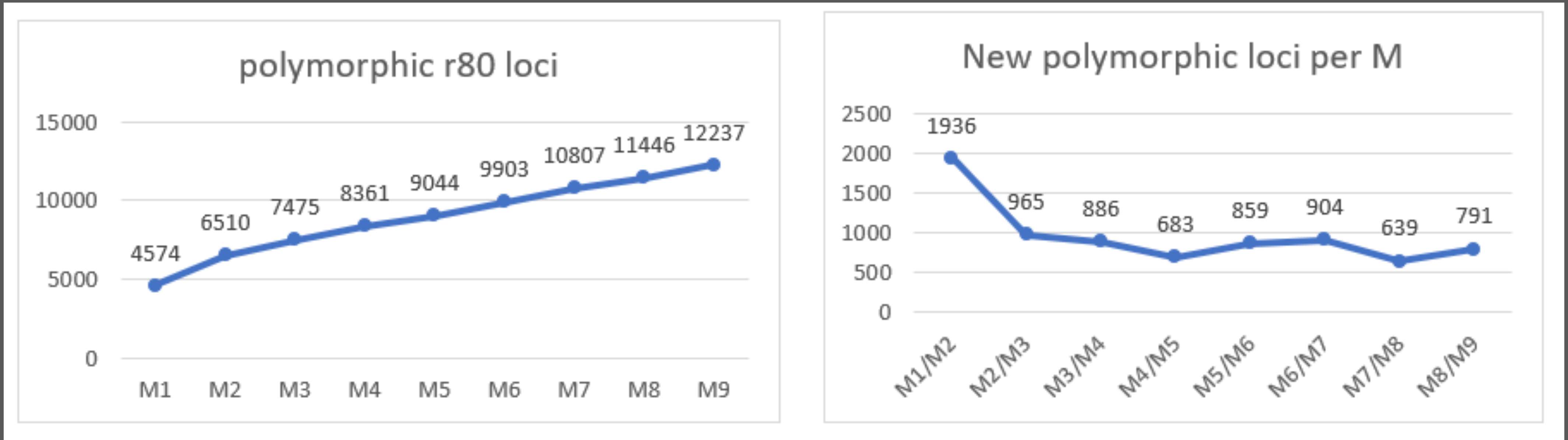
polymorphic r80 loci



New polymorphic loci per M

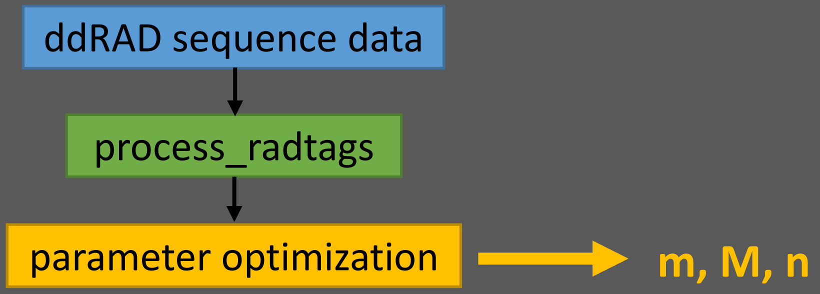
Bioinformatics

Dypsis pinnatifrons



M = 5
M = n

Bioinformatics Workflow



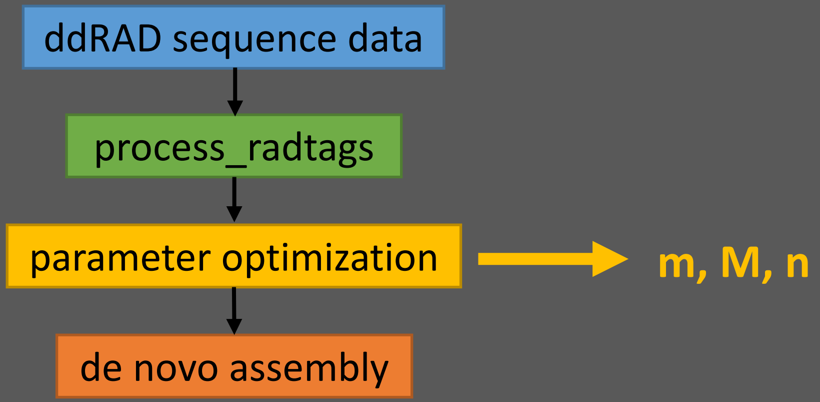
ddRAD sequence data

process_radtags

parameter optimization

m, M, n

Bioinformatics Workflow



ddRAD sequence data

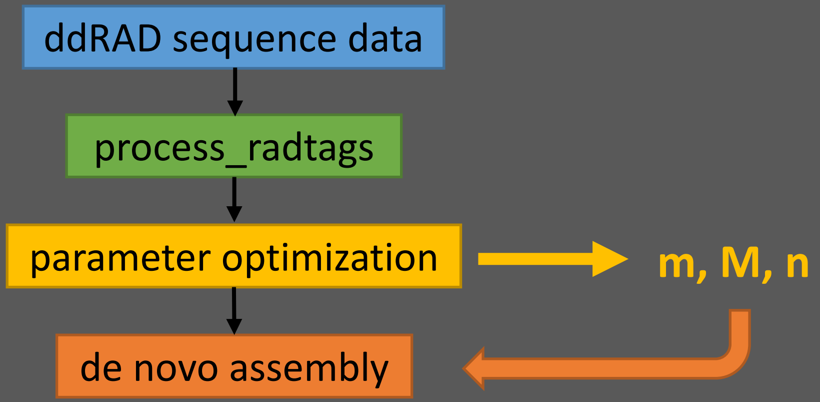
process_radtags

parameter optimization

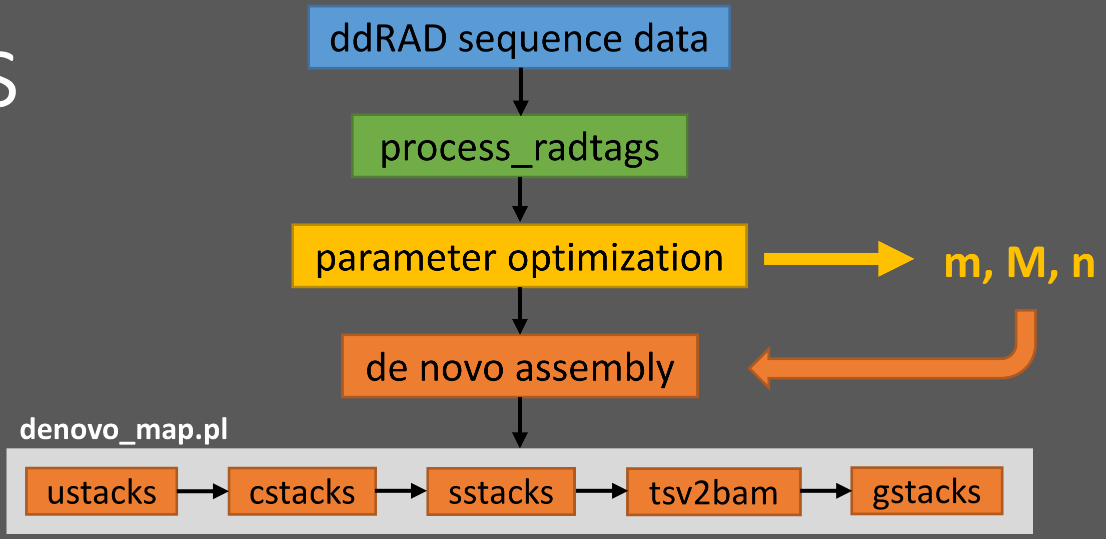
de novo assembly

m, M, n

Bioinformatics Workflow



Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

m, M, n

de novo assembly

denovo_map.pl

ustacks

cstacks

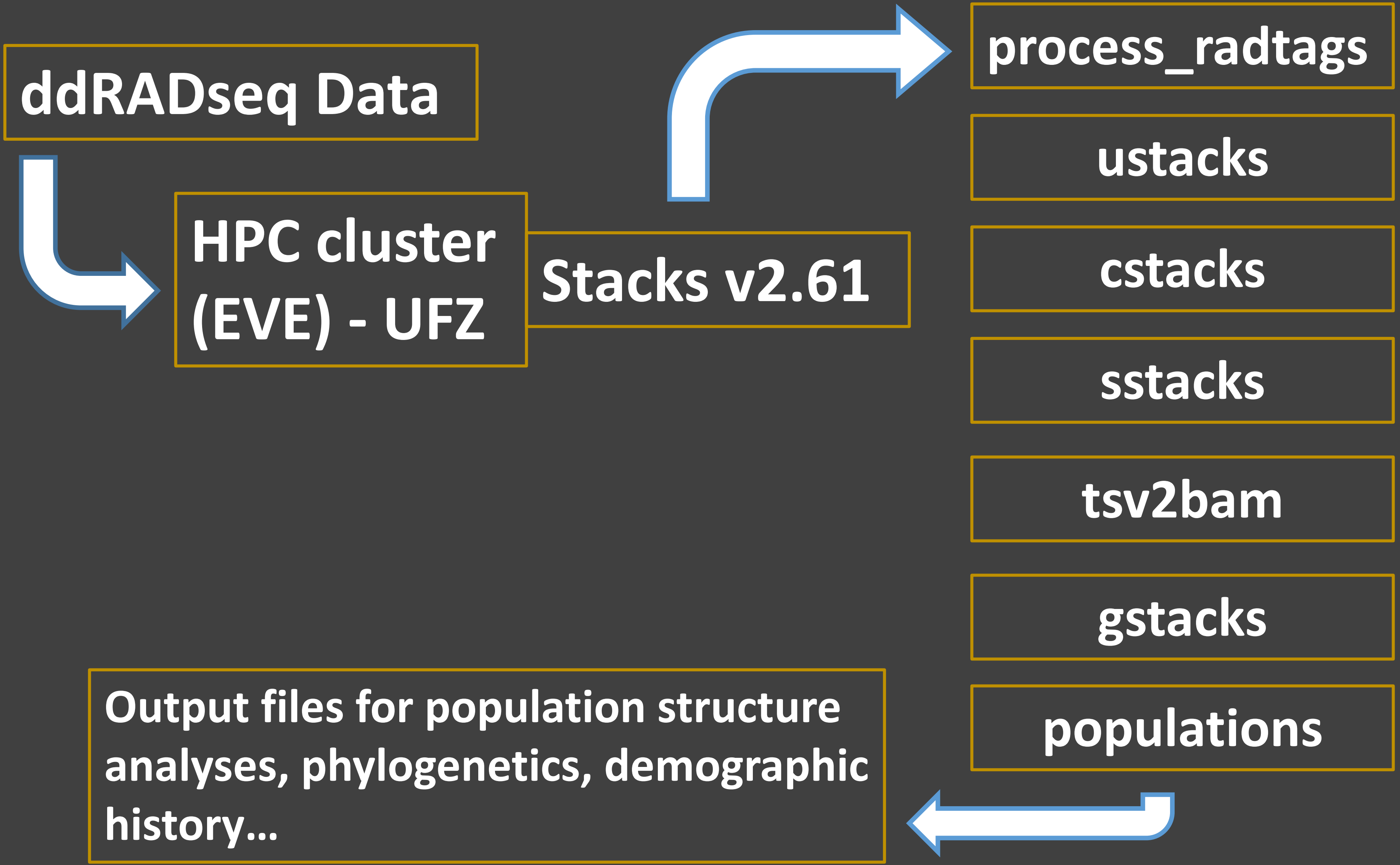
sstacks

tsv2bam

gstacks

Rochette, Rivera-Colón & Catchen, 2019 - Mol. Ecology

ddRAD sequencing



ddRADseq Data

HPC cluster (EVE) - UFZ

Stacks v2.61

process_radtags – Clean and demultiplex the data

ustacks – Building loci *de novo* for each sample

cstacks – Creates a *catalog* of all loci across the populations according to sequence similarity

sstacks – Match each sample against the catalog

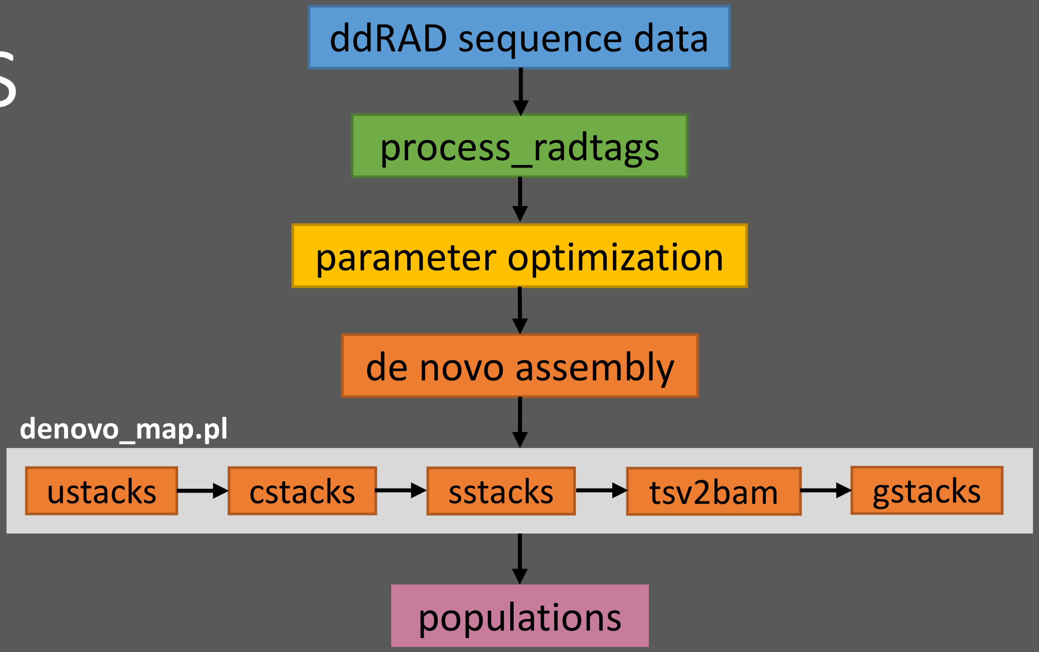
tsv2bam – Transpose the data to be organized by RAD locus. Paired-end reads are fetched and stored for later use

gstacks – A contig is assembled from pair-end reads and overlapped with the single-end locus. SNP calling

populations – Population level statistics and output in different formats. Possibility for further filtering

Output files for population structure analyses, phylogenetics, demographic history…

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

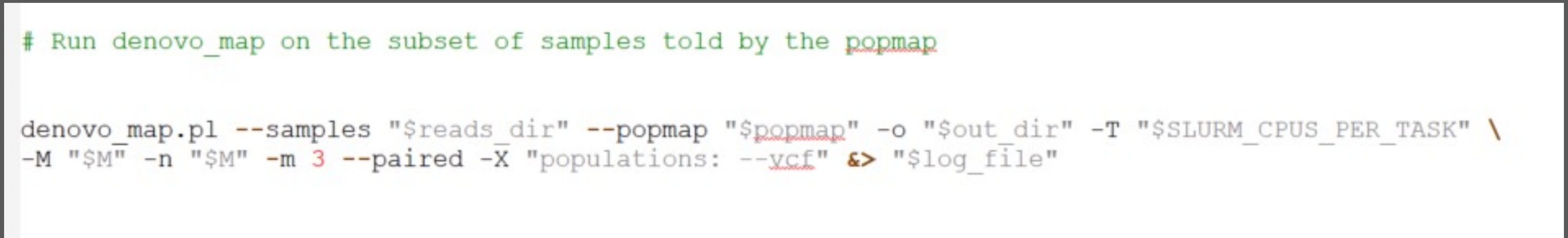
de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

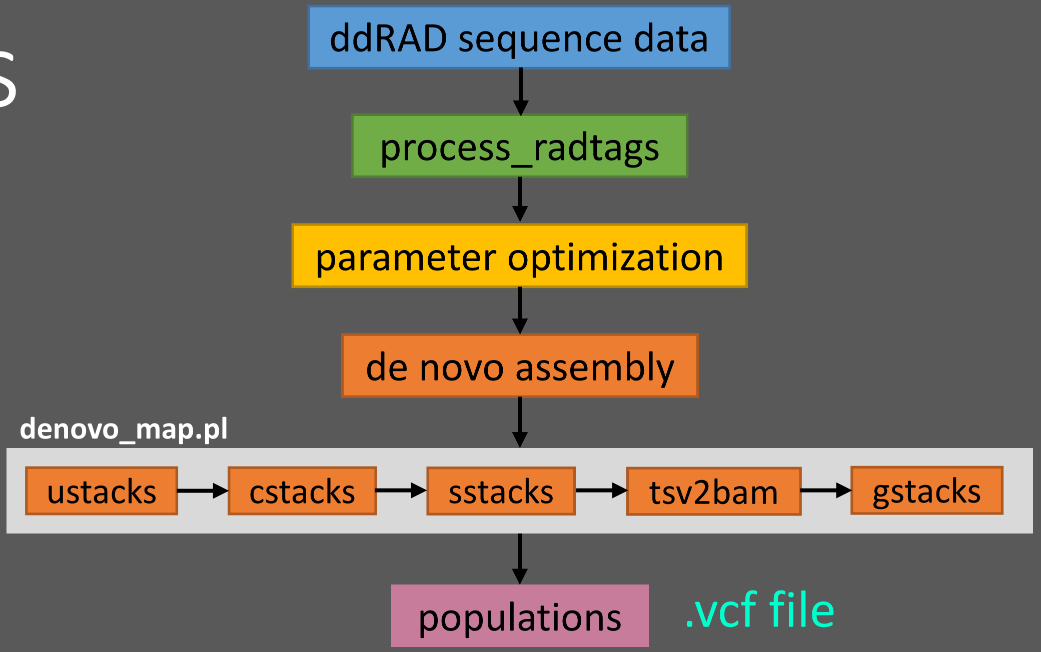
populations

03_denovo_map_full



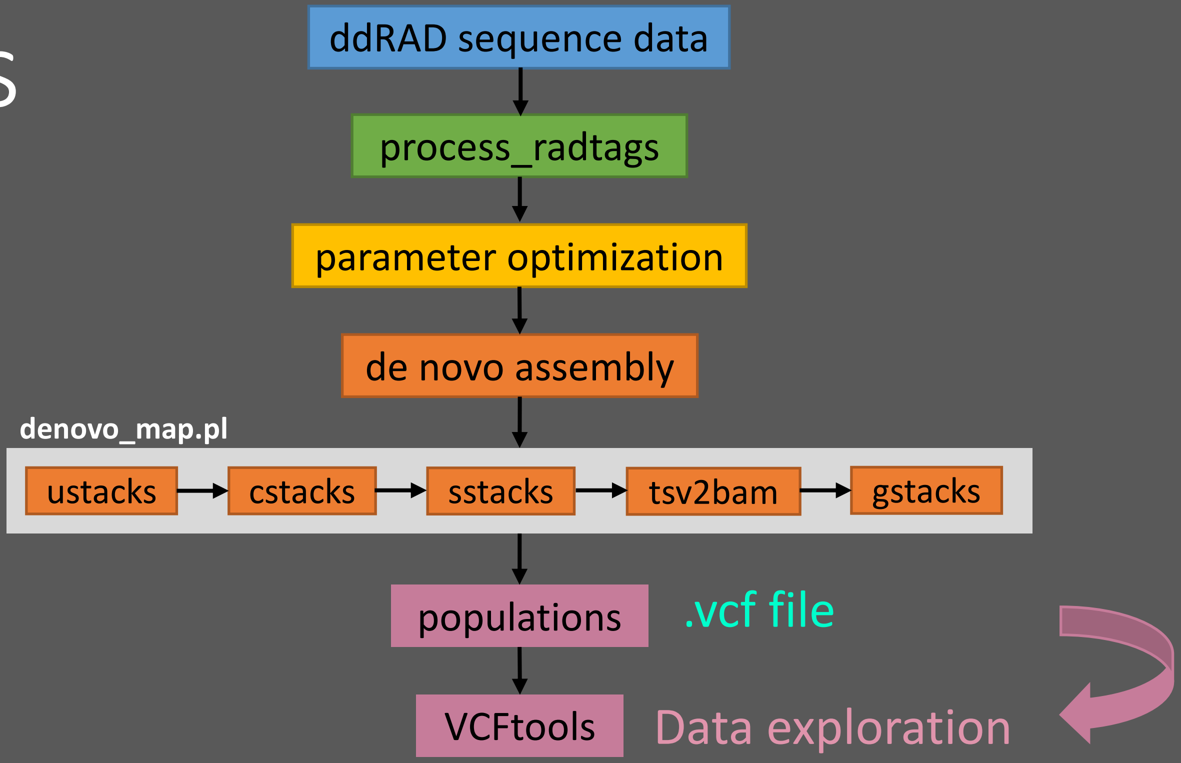
```
# Run denovo_map on the subset of samples told by the popmap

denovo_map.pl --samples "$reads_dir" --popmap "$popmap" -o "$out_dir" -T "$SLURM_CPUS_PER_TASK" \
-M "$M" -n "$M" -m 3 --paired -X "populations: --vcf" &> "$log_file"
```

Bioinformatics Workflow



.vcf file

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

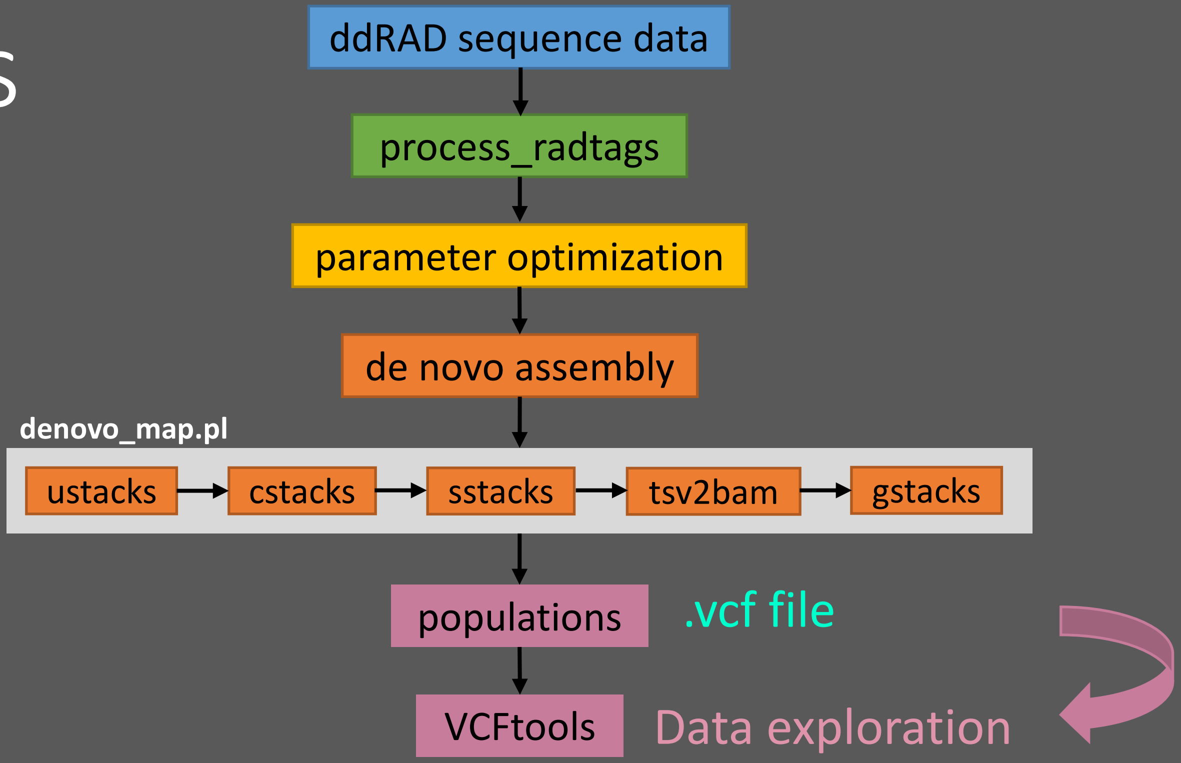
populations

.vcf file

VCFtools

Data exploration

Bioinformatics Workflow



--freq2 = allele frequency

--depth = mean depth per individual

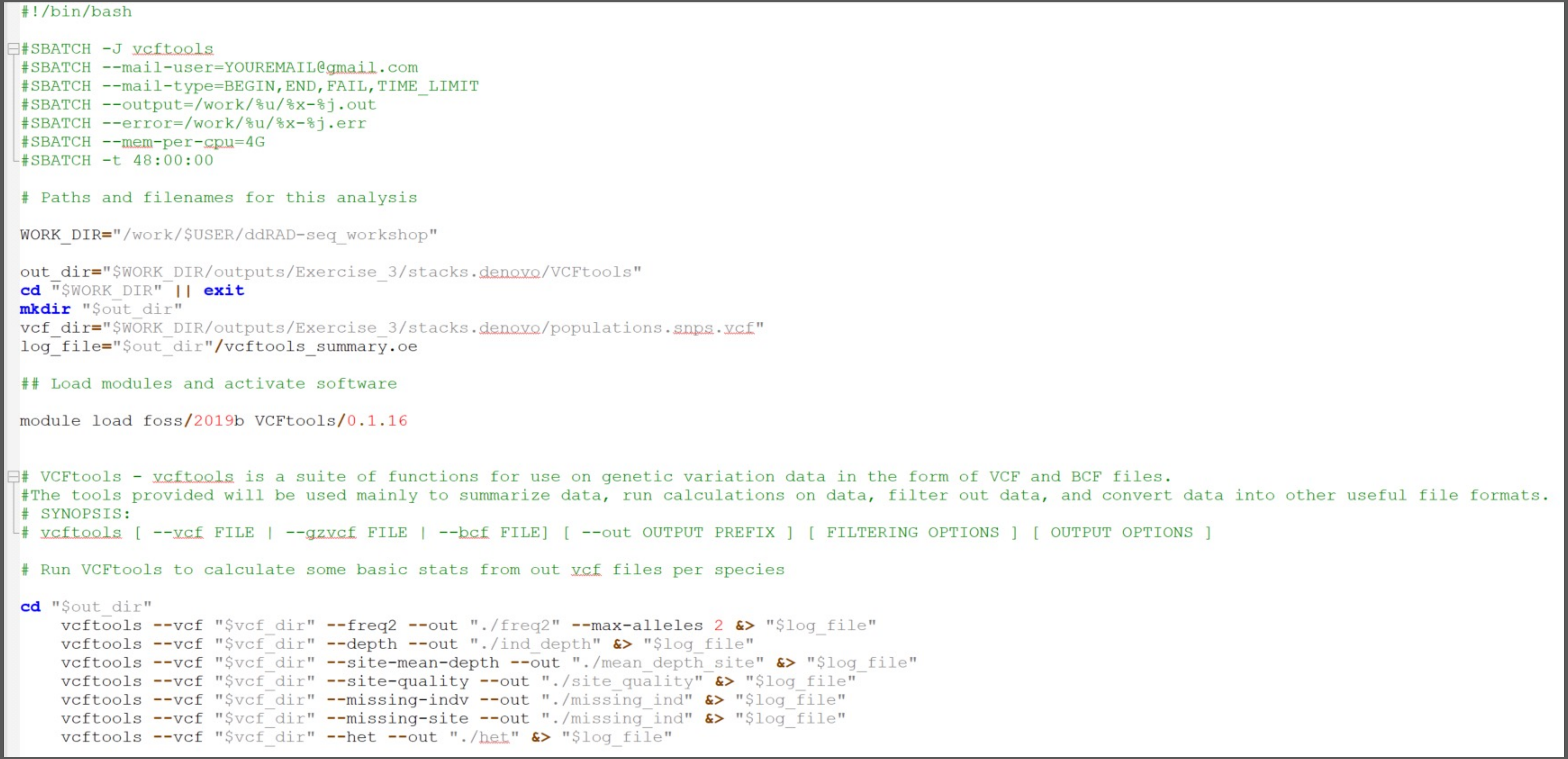
--site-mean-depth = mean depth per site

--missing-indv = proportion of missing data per individual

--missing-site = proportion of missing data per site

--het = heterozygosity and inbreeding coefficient per individual

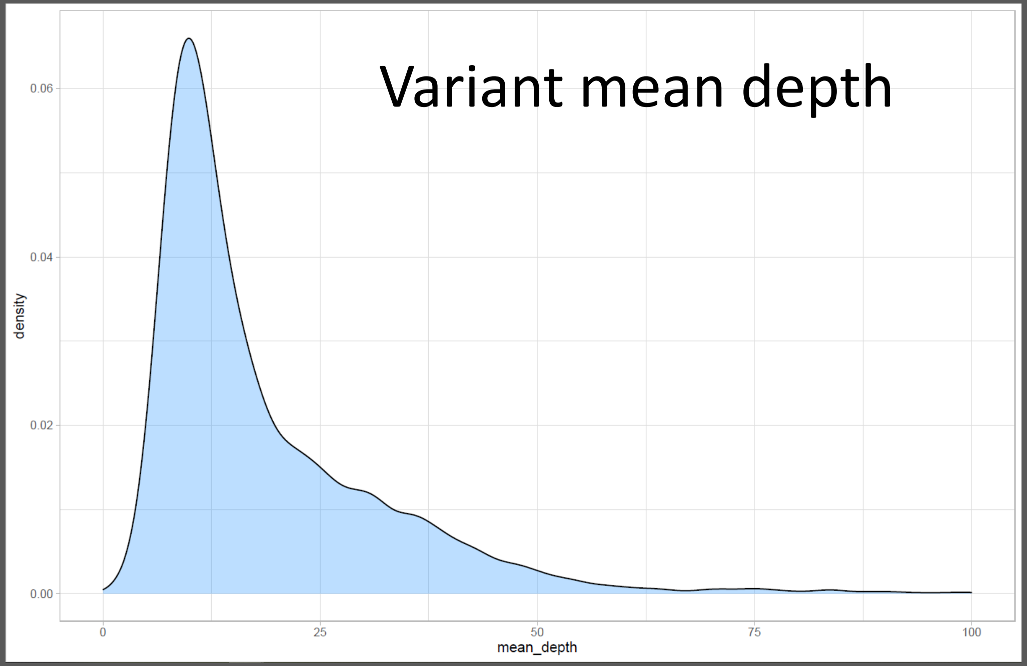
04_extract_results

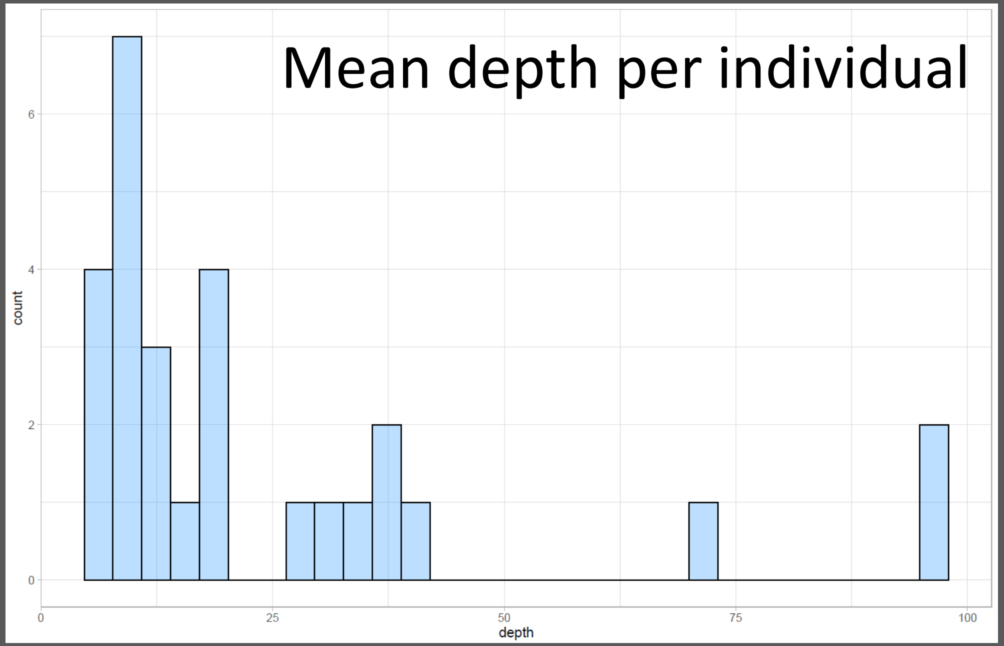


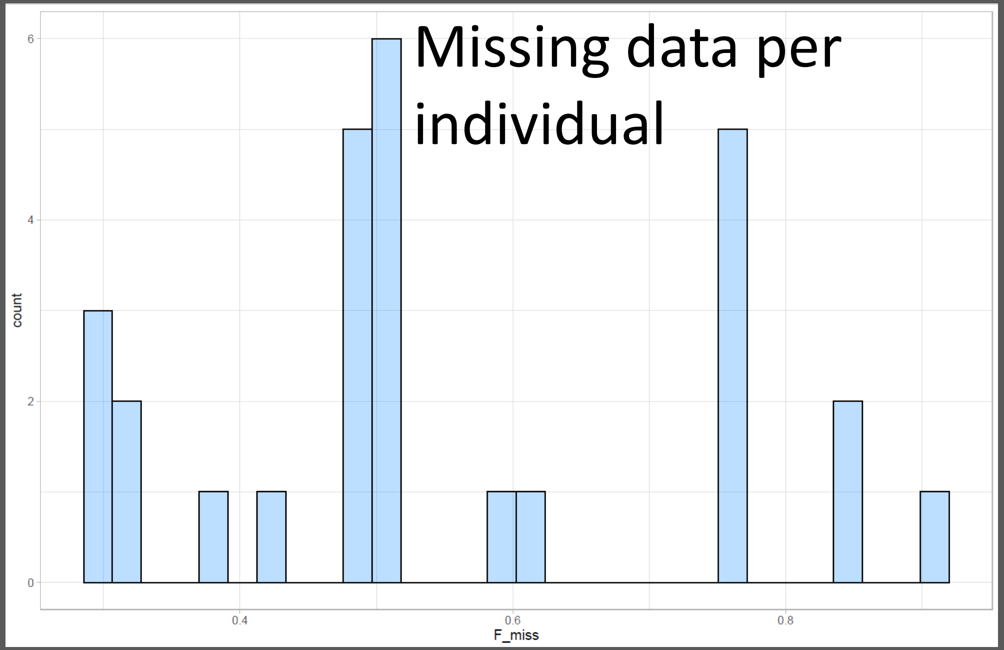
```
#!/bin/bash

#SBATCH -J vcftools
#SBATCH --mail-user=YOUREMAIL@gmail.com
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT
#SBATCH --output=/work/%u/%x-%j.out
#SBATCH --error=/work/%u/%x-%j.err
#SBATCH --mem-per-cpu=4G
#SBATCH -t 48:00:00

# Paths and filenames for this analysis

WORK_DIR="/work/$USER/ddRAD-seq_workshop"

out_dir="$WORK_DIR/outputs/Exercise_3/stacks.denovo/VCFtools"
cd "$WORK_DIR" || exit
mkdir "$out_dir"
vcf_dir="$WORK_DIR/outputs/Exercise_3/stacks.denovo/populations.snps.vcf"
log_file="$out_dir"/vcftools_summary.oe

## Load modules and activate software

module load foss/2019b VCFtools/0.1.16


# VCFtools - vcftools is a suite of functions for use on genetic variation data in the form of VCF and BCF files.
#The tools provided will be used mainly to summarize data, run calculations on data, filter out data, and convert data into other useful file formats.
# SYNOPSIS:
# vcftools [ --vcf FILE | --gzvcf FILE | --bcf FILE] [ --out OUTPUT PREFIX ] [ FILTERING OPTIONS ] [ OUTPUT OPTIONS ]

# Run VCFtools to calculate some basic stats from out vcf files per species

cd "$out_dir"
    vcftools --vcf "$vcf_dir" --freq2 --out "./freq2" --max-alleles 2 &> "$log_file"
    vcftools --vcf "$vcf_dir" --depth --out "./ind_depth" &> "$log_file"
    vcftools --vcf "$vcf_dir" --site-mean-depth --out "./mean_depth_site" &> "$log_file"
    vcftools --vcf "$vcf_dir" --site-quality --out "./site_quality" &> "$log_file"
    vcftools --vcf "$vcf_dir" --missing-indv --out "./missing_ind" &> "$log_file"
    vcftools --vcf "$vcf_dir" --missing-site --out "./missing_ind" &> "$log_file"
    vcftools --vcf "$vcf_dir" --het --out "./het" &> "$log_file"
```

Bioinformatics

Borassus madagascariensis
28 individuals



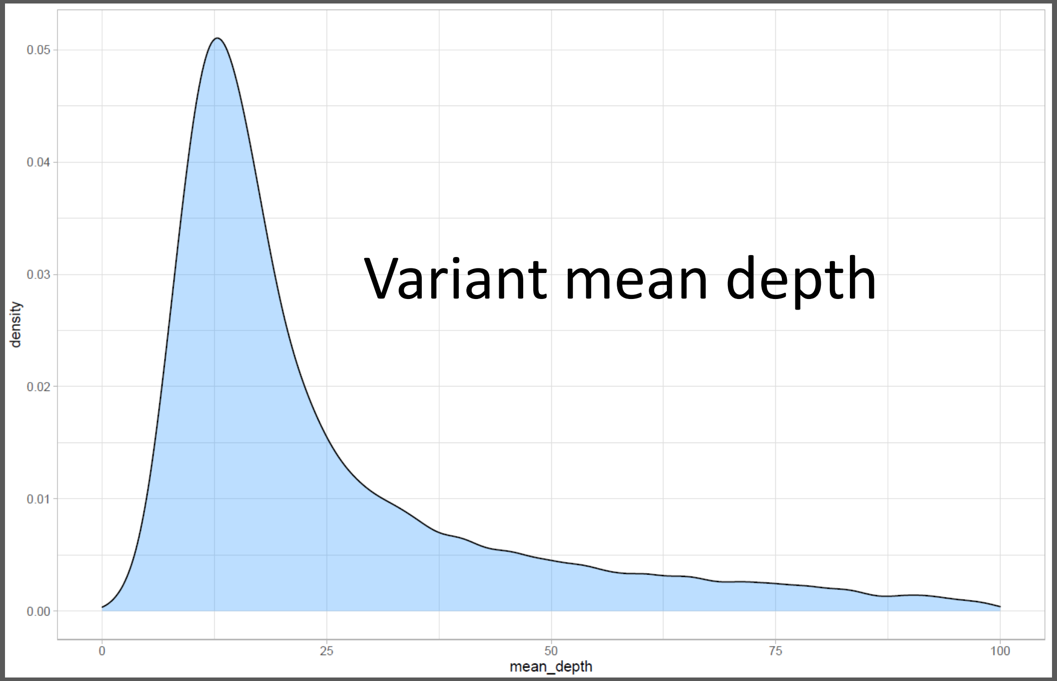


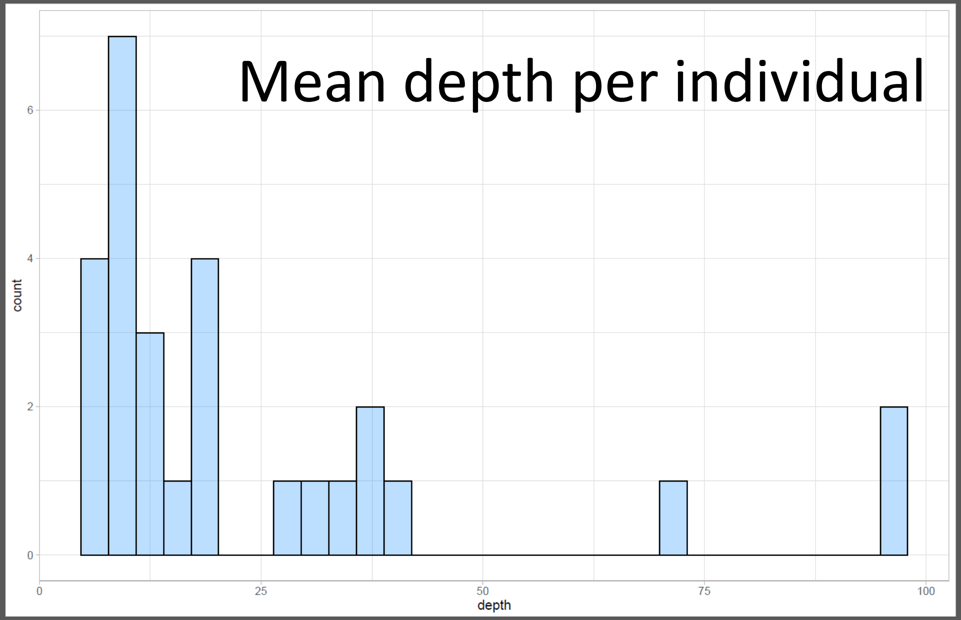


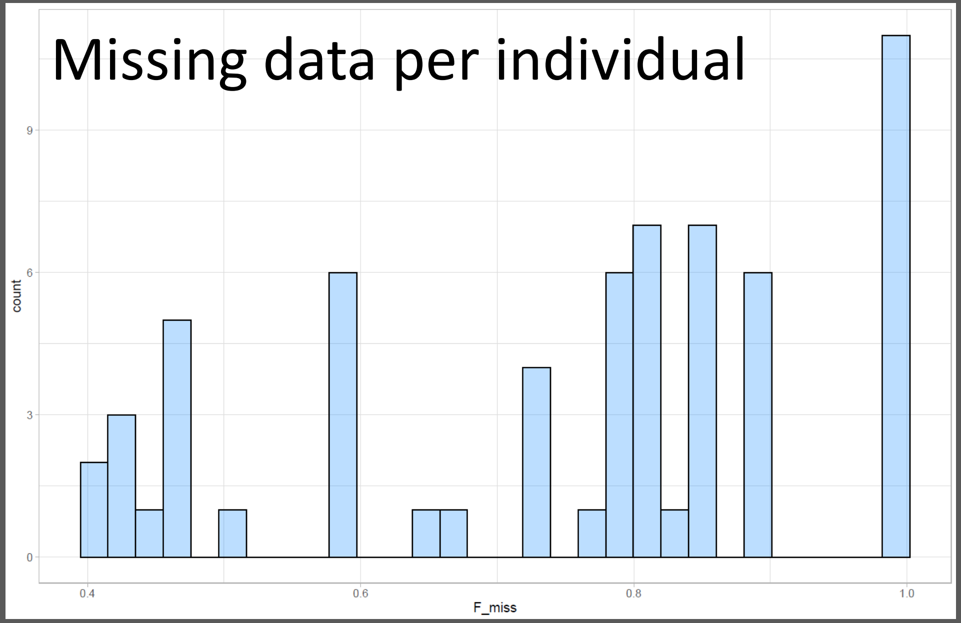
Bioinformatics

Bismarckia nobilis

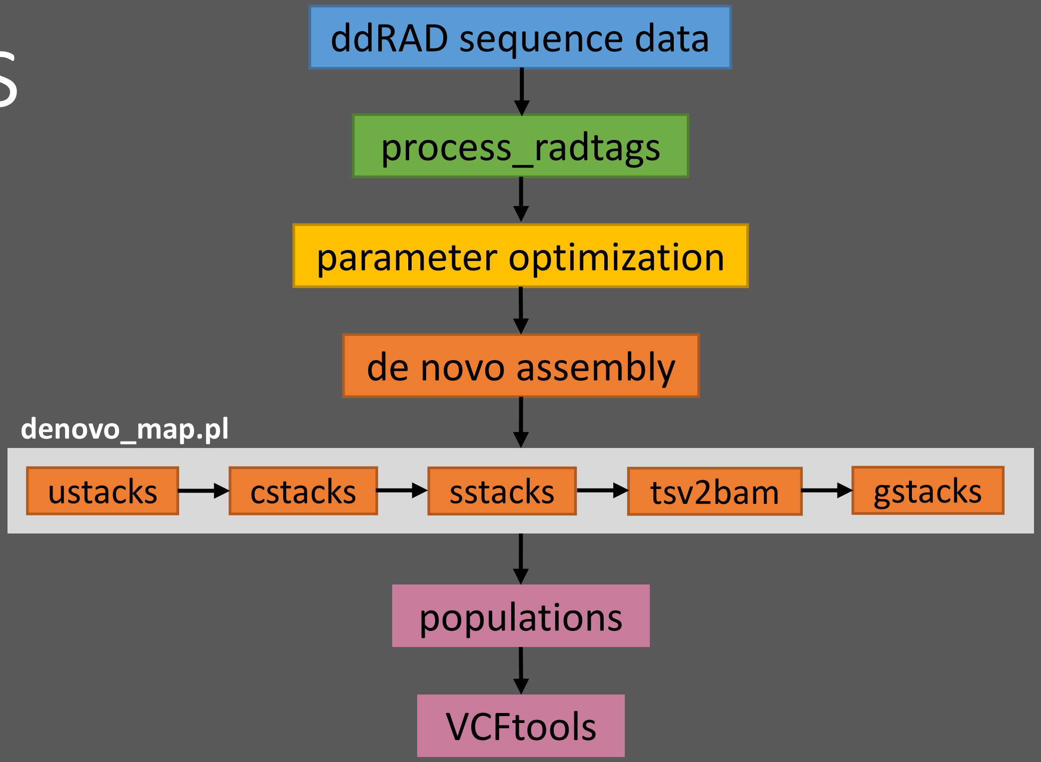
63 individuals


Variant mean depth

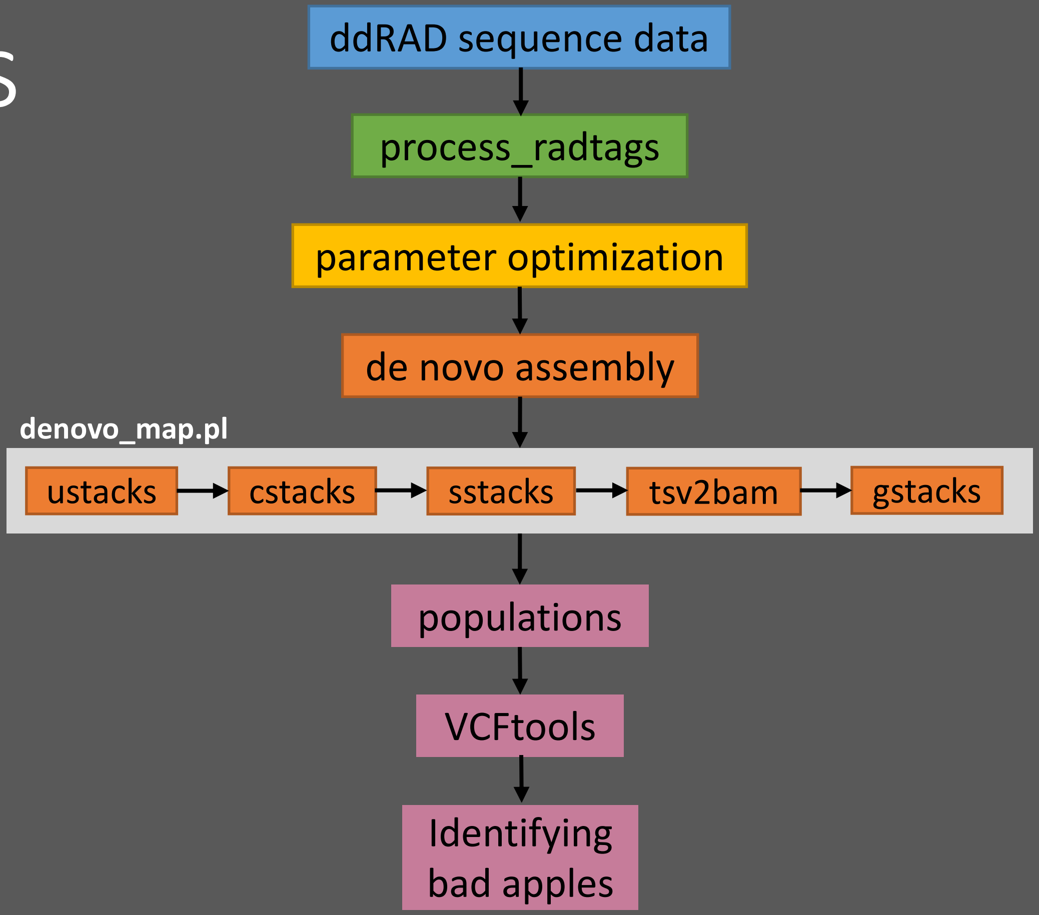

Mean depth per individual


Missing data per individual

Bioinformatics Workflow



Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

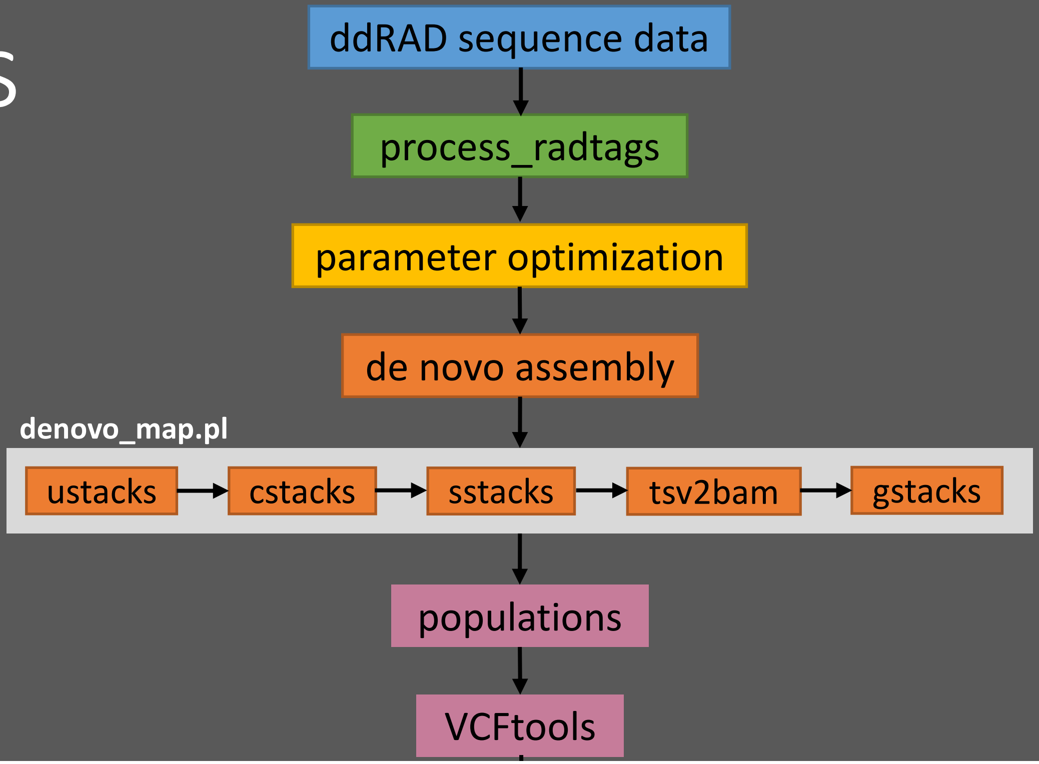
ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

VCFtools

Identifying bad apples

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

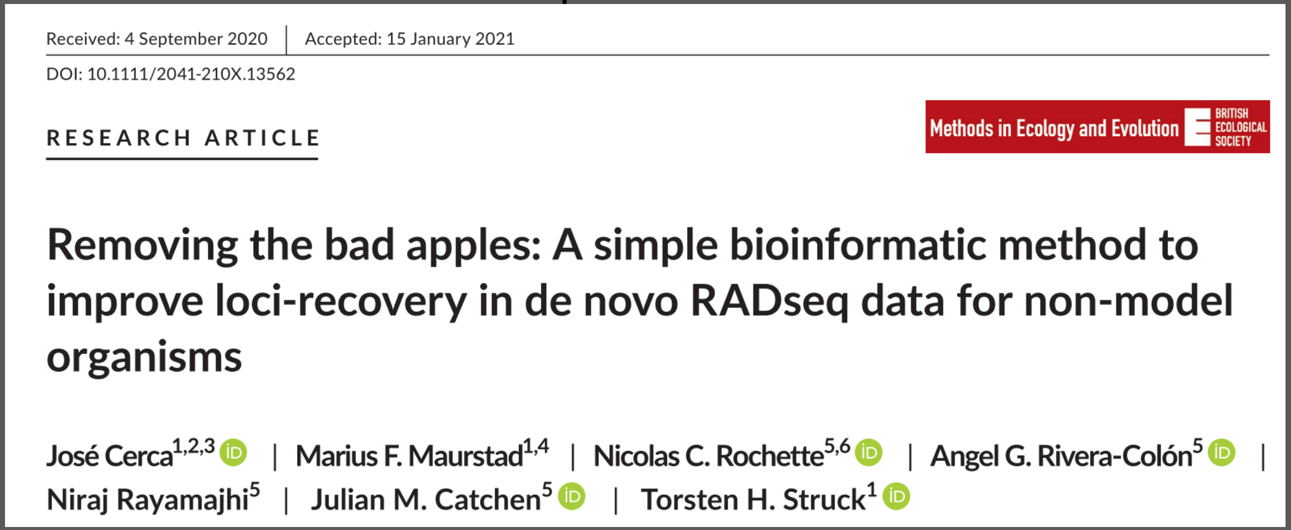
de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

VCFtools



Received: 4 September 2020 | Accepted: 15 January 2021

DOI: 10.1111/2041-210X.13562

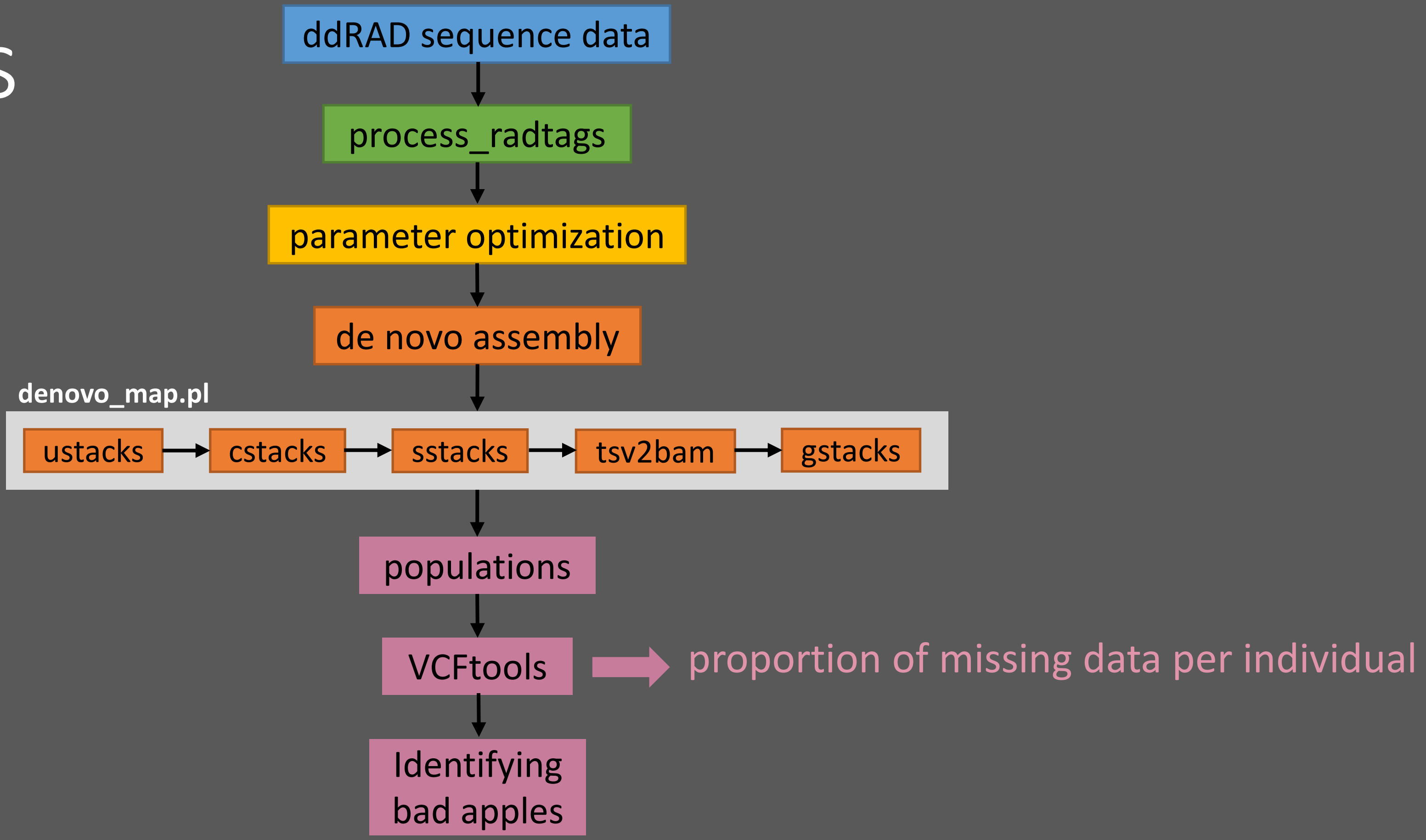
RESEARCH ARTICLE

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

Removing the bad apples: A simple bioinformatic method to improve loci-recovery in de novo RADseq data for non-model organisms

José Cerca[1,2,3] | Marius F. Maurstad[1,4] | Nicolas C. Rochette[5,6] | Angel G. Rivera-Colón[5] | Niraj Rayamajhi[5] | Julian M. Catchen[5] | Torsten H. Struck[1]

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

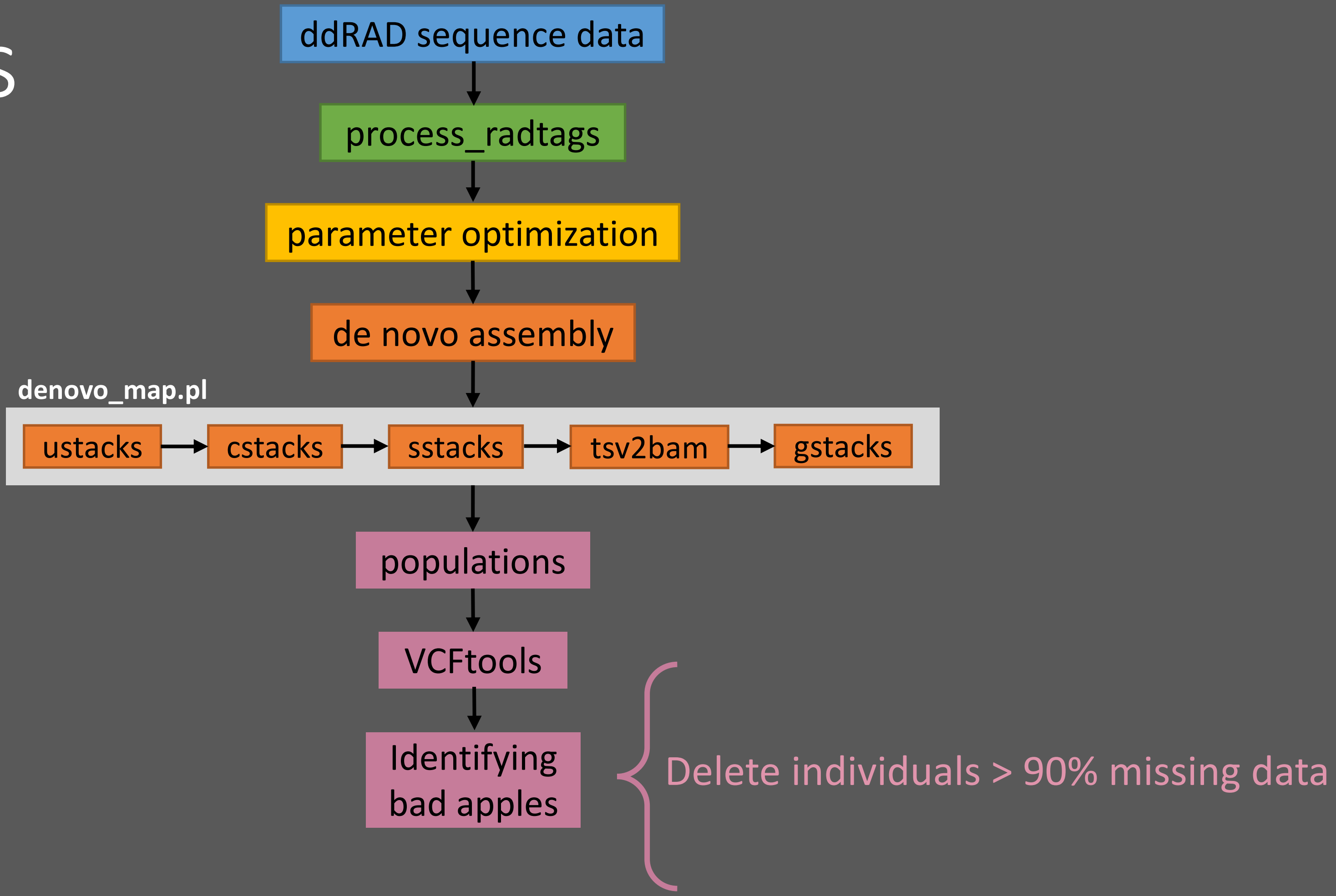
ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

VCFtools → proportion of missing data per individual

Identifying bad apples

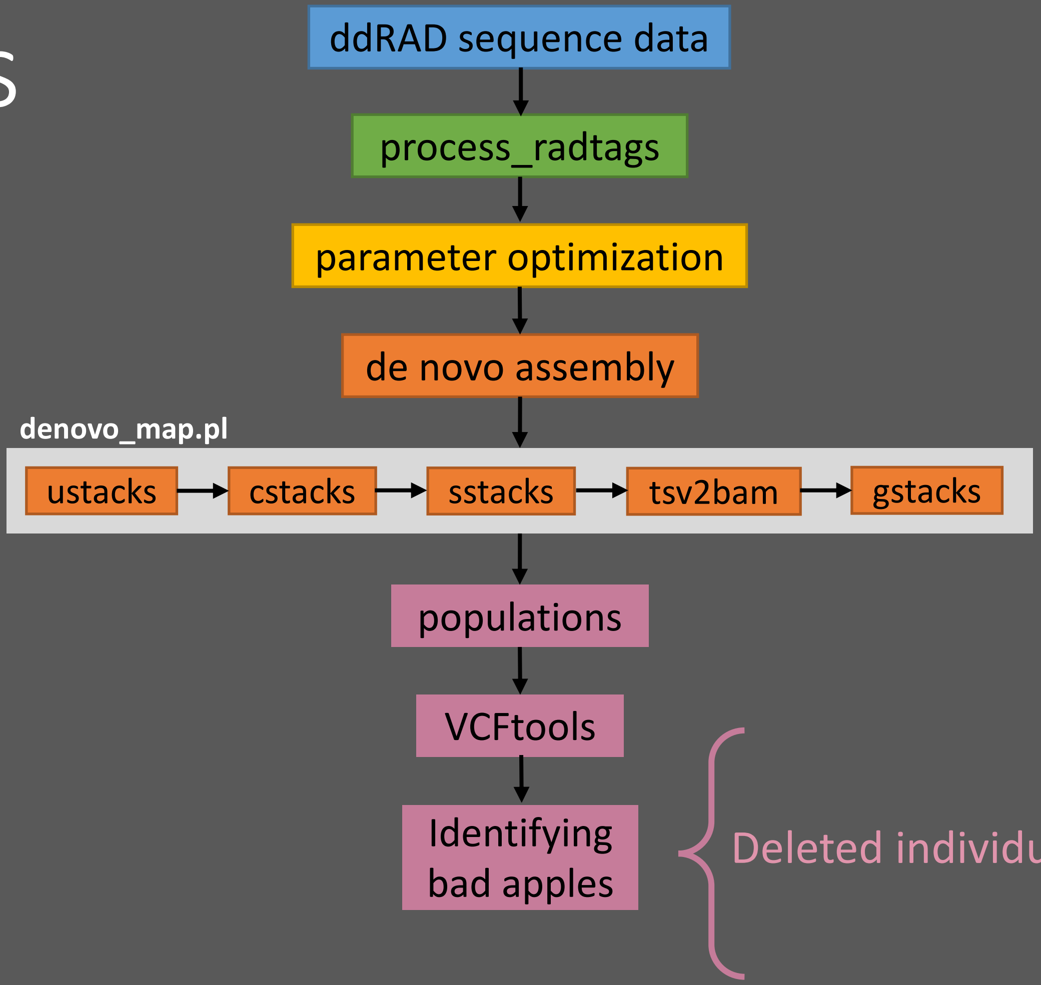
Bioinformatics Workflow



Delete individuals > 90% missing data

“helps recovering a higher number of loci”

Bioinformatics Workflow



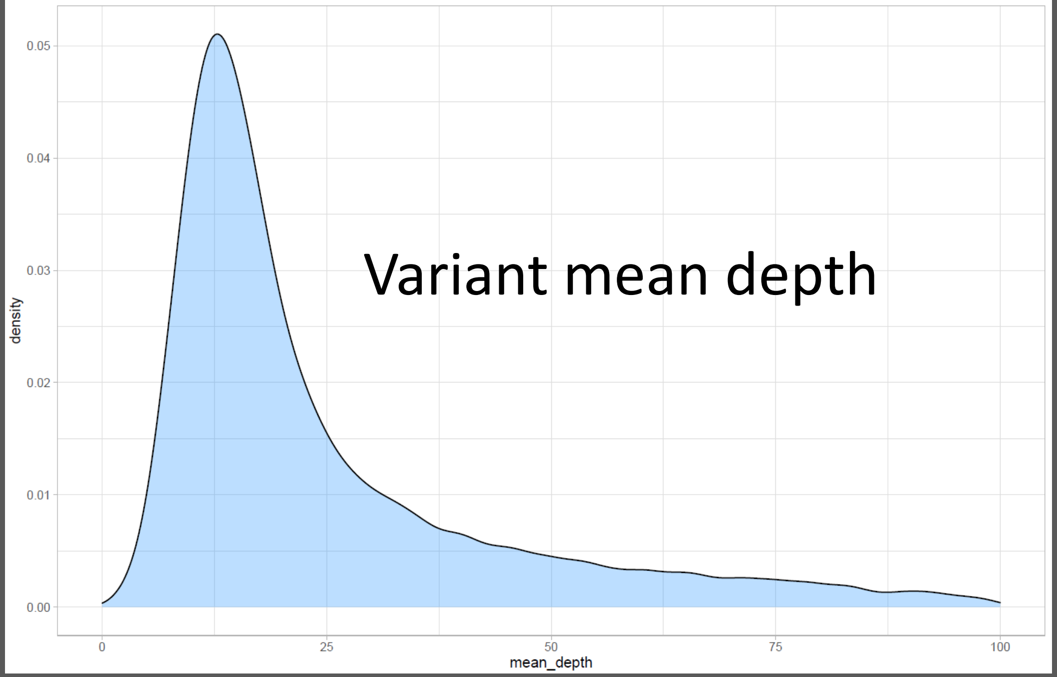
Deleted individuals > 90% missing data

New population map -> popmap_90

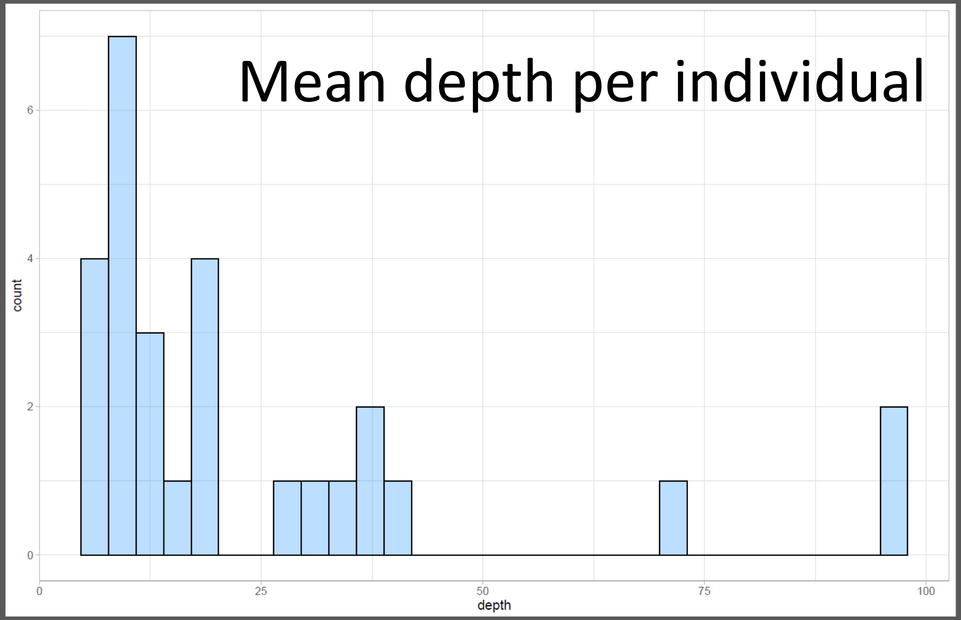
Bioinformatics

Bismarckia nobilis

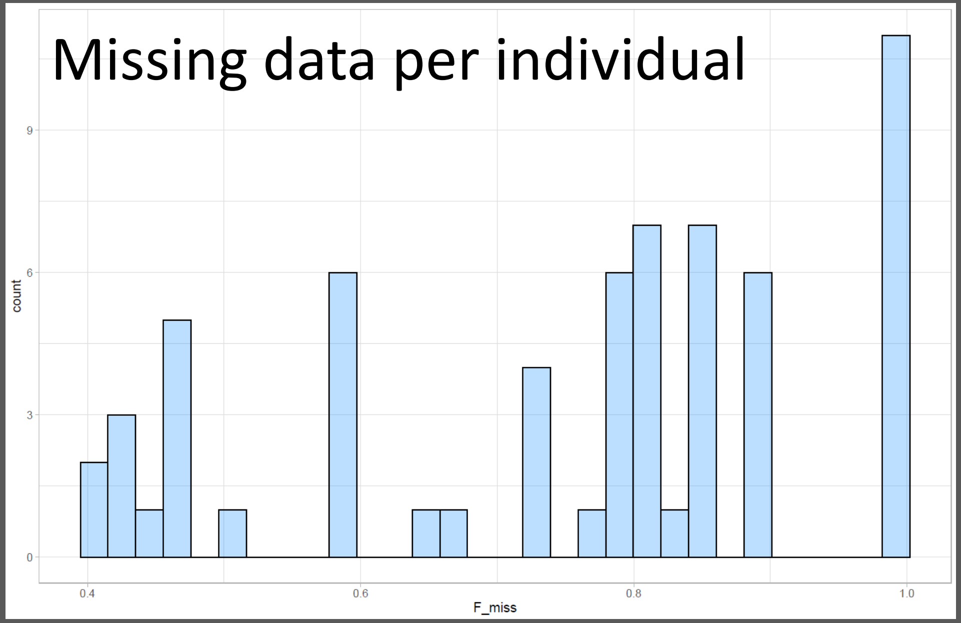
63 individuals



Variant mean depth

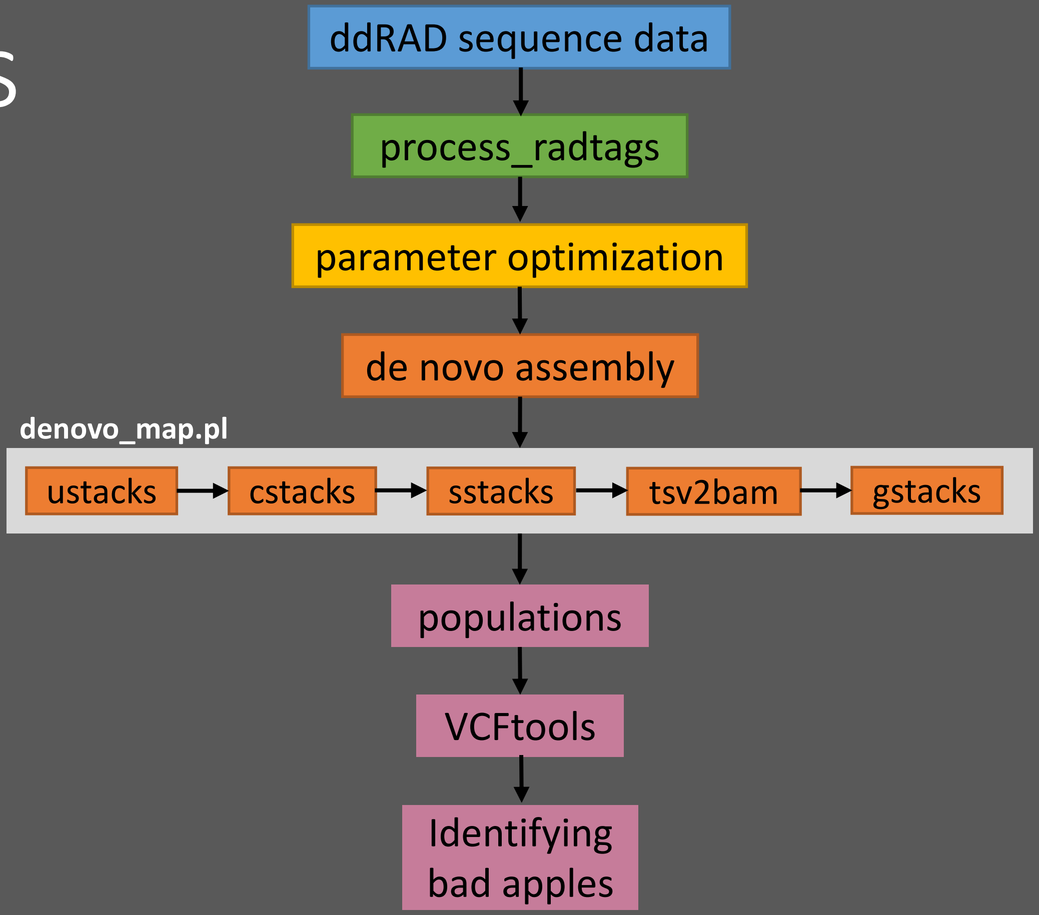


Mean depth per individual

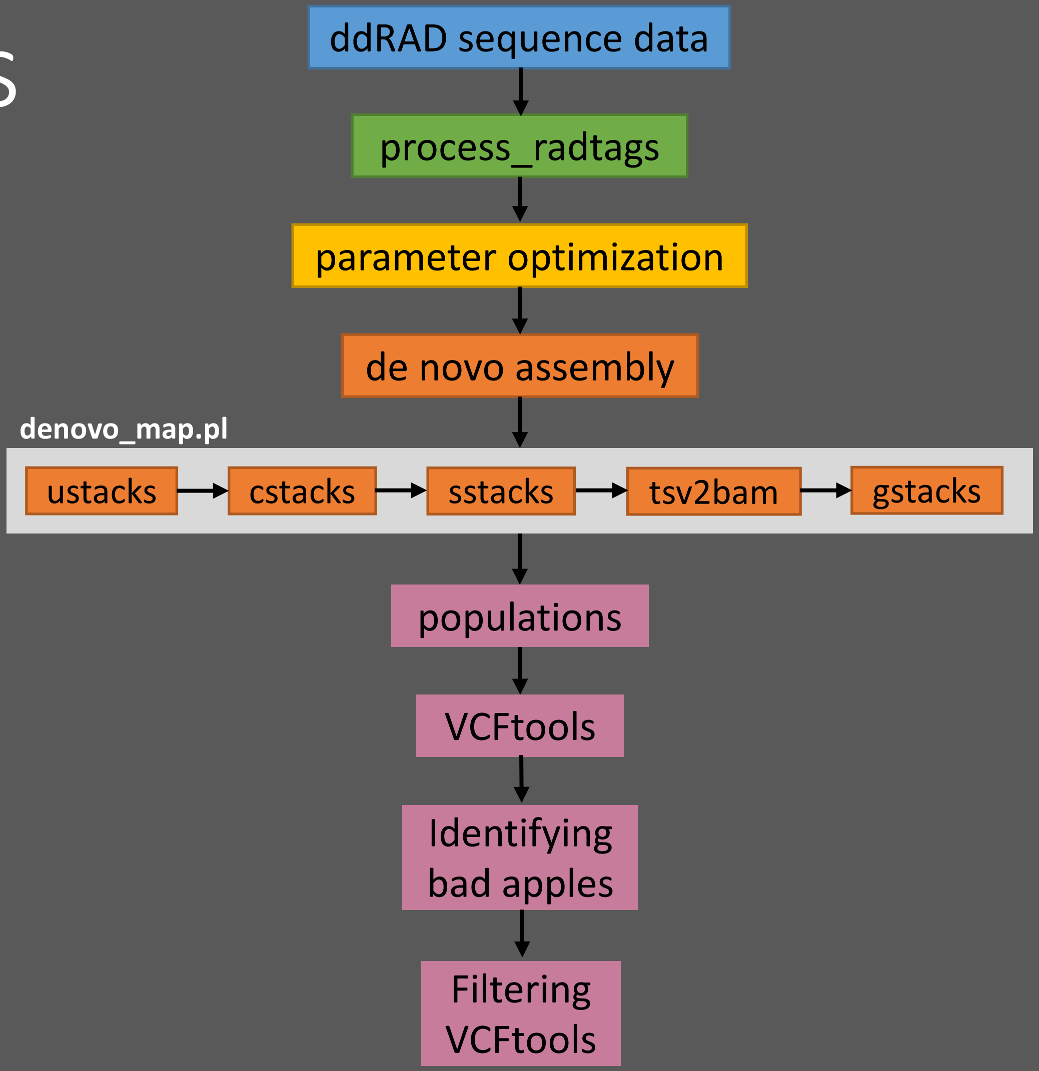


Missing data per individual

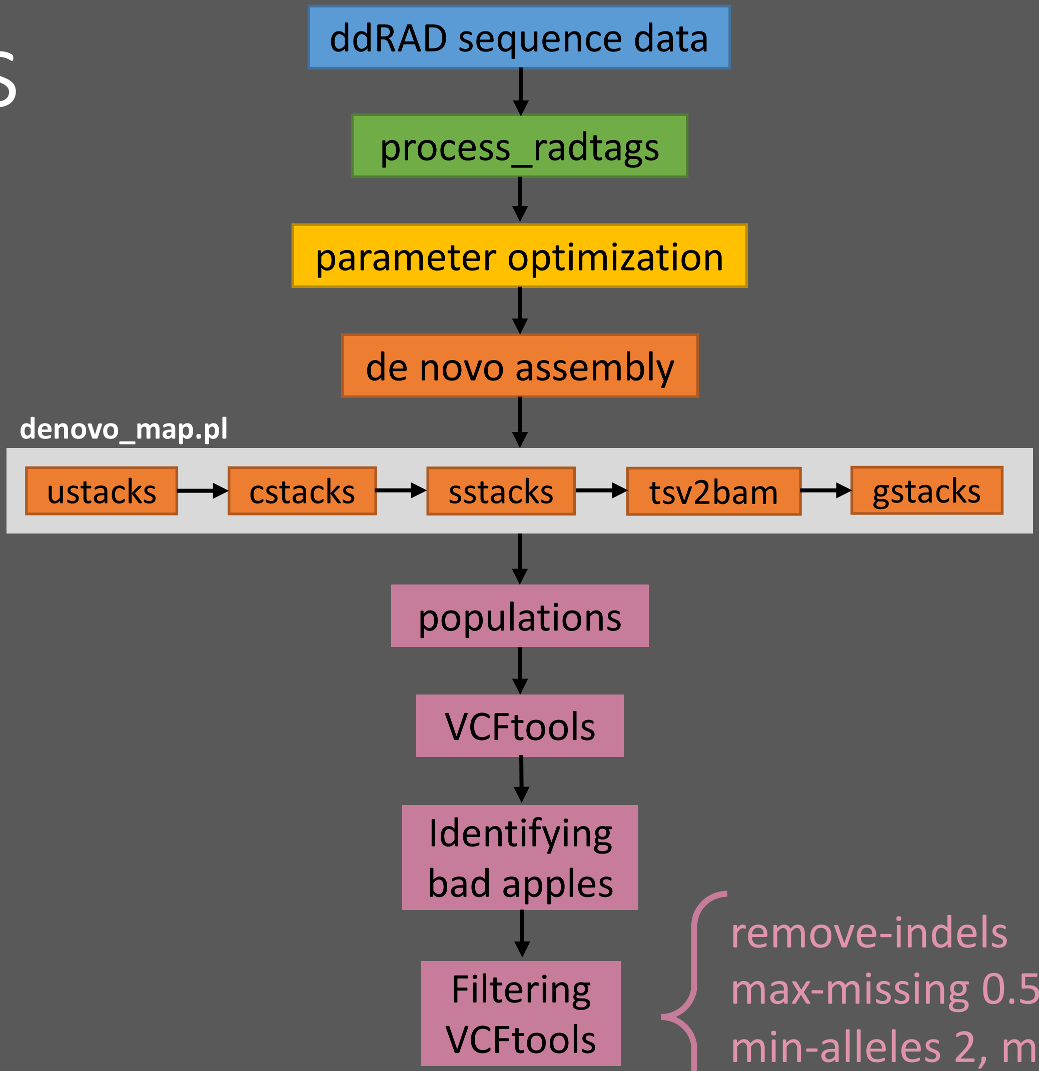
Bioinformatics Workflow



Bioinformatics Workflow



Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

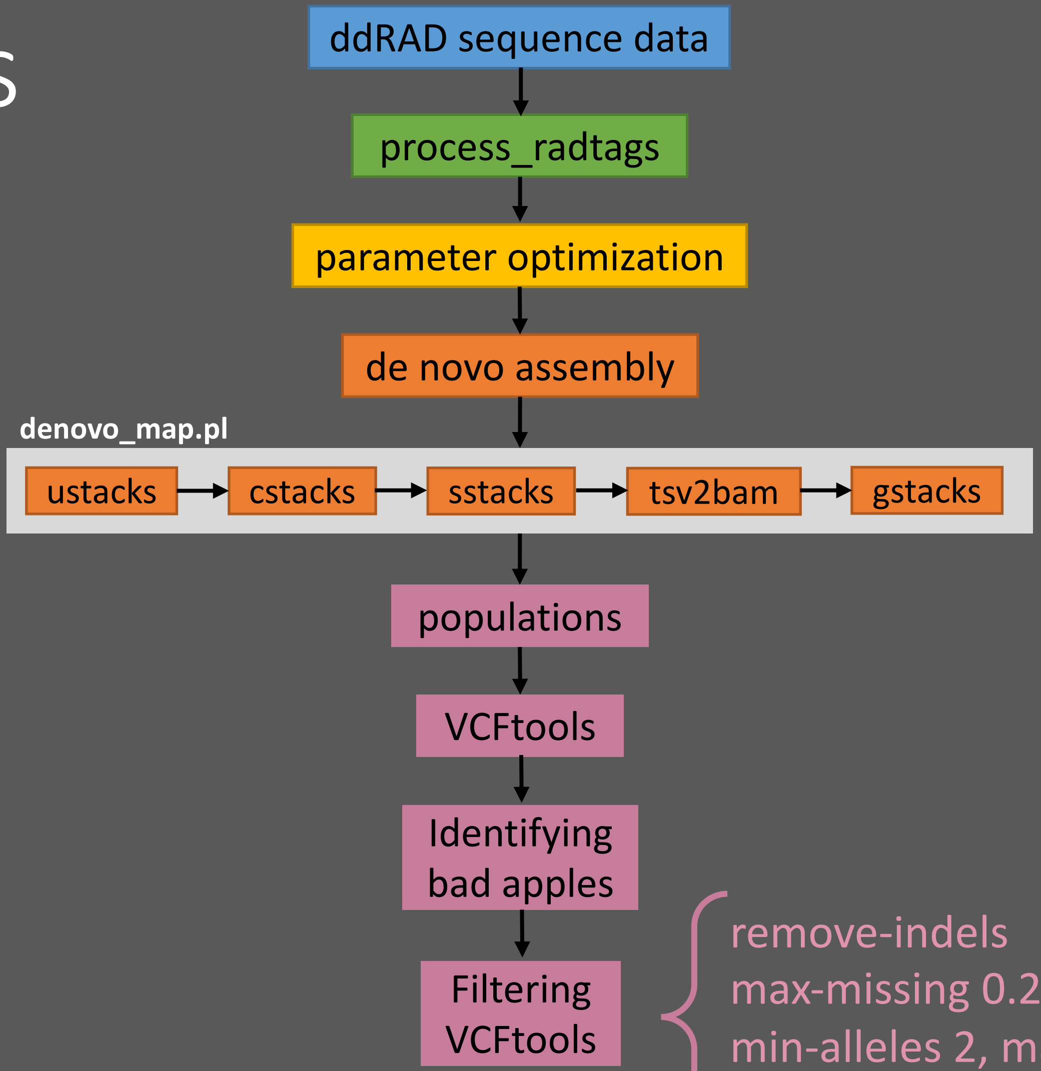
VCFtools

Identifying bad apples

Filtering VCFtools

remove-indels
max-missing 0.50 (max of 50% missing data)
min-alleles 2, max-alleles 2 (only biallelic)
minDP 5, maxDP 100

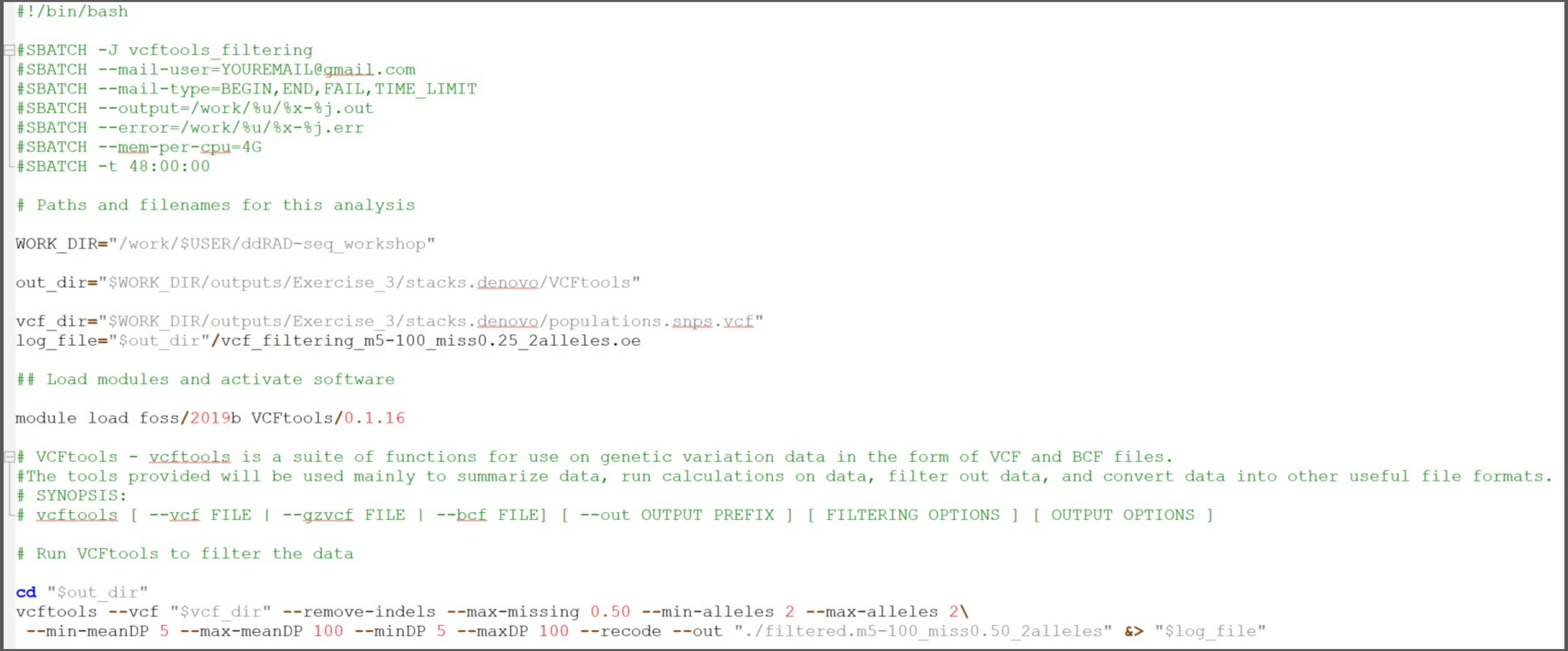
Bioinformatics Workflow



To control for false-positive loci

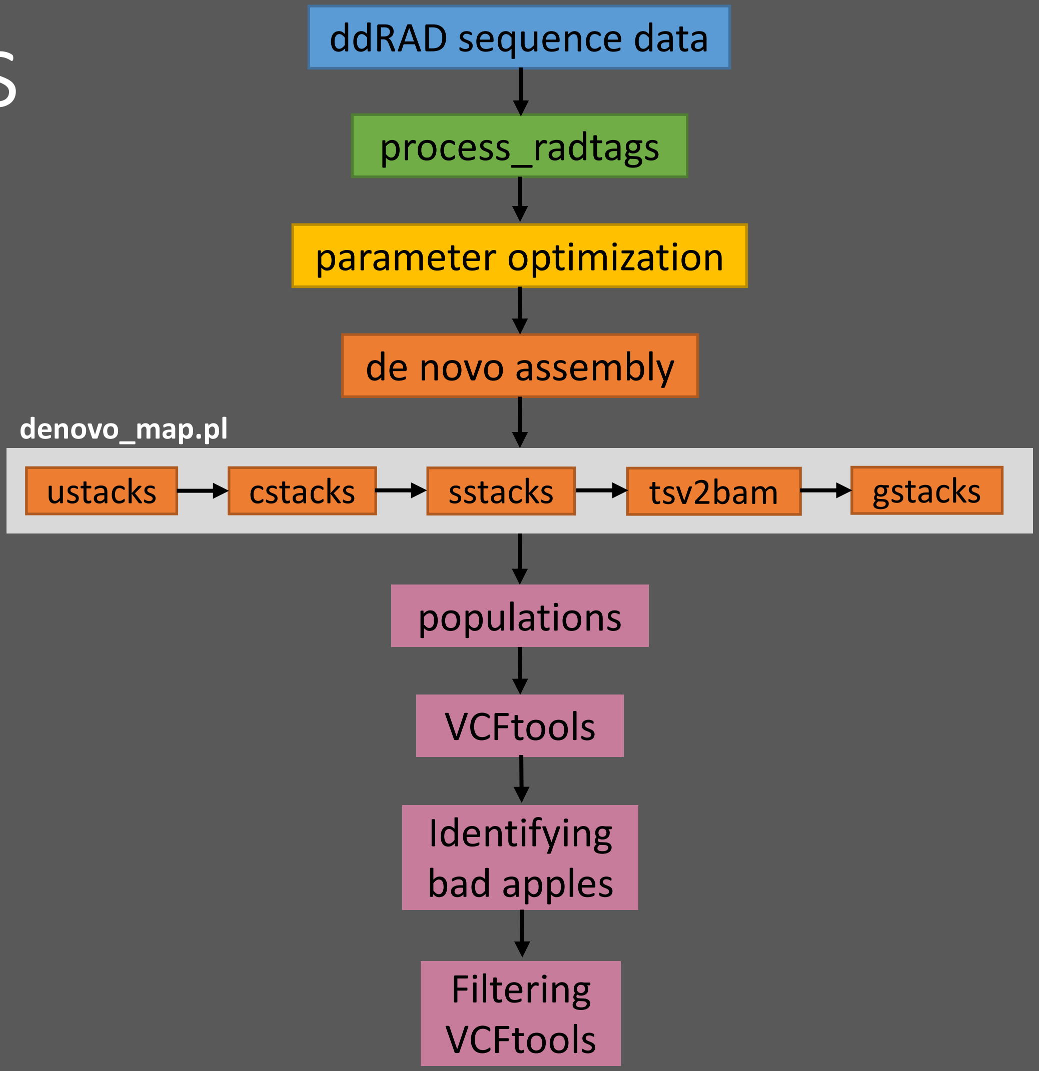
remove-indels
max-missing 0.25 (max of 75% missing data)
min-alleles 2, max-alleles 2 (only biallelic)
minDP 5, maxDP 100

05_VCFtools_filtering



```
#!/bin/bash

#SBATCH -J vcftools_filtering
#SBATCH --mail-user=YOUREMAIL@gmail.com
#SBATCH --mail-type=BEGIN,END,FAIL,TIME_LIMIT
#SBATCH --output=/work/%u/%x-%j.out
#SBATCH --error=/work/%u/%x-%j.err
#SBATCH --mem-per-cpu=4G
#SBATCH -t 48:00:00

# Paths and filenames for this analysis

WORK_DIR="/work/$USER/ddRAD-seq_workshop"

out_dir="$WORK_DIR/outputs/Exercise_3/stacks.denovo/VCFtools"

vcf_dir="$WORK_DIR/outputs/Exercise_3/stacks.denovo/populations.snps.vcf"
log_file="$out_dir"/vcf_filtering_m5-100_miss0.25_2alleles.oe

## Load modules and activate software

module load foss/2019b VCFtools/0.1.16

# VCFtools - vcftools is a suite of functions for use on genetic variation data in the form of VCF and BCF files.
#The tools provided will be used mainly to summarize data, run calculations on data, filter out data, and convert data into other useful file formats.
# SYNOPSIS:
# vcftools [ --vcf FILE | --gzvcf FILE | --bcf FILE] [ --out OUTPUT PREFIX ] [ FILTERING OPTIONS ] [ OUTPUT OPTIONS ]

# Run VCFtools to filter the data

cd "$out_dir"
vcftools --vcf "$vcf_dir" --remove-indels --max-missing 0.50 --min-alleles 2 --max-alleles 2\
 --min-meanDP 5 --max-meanDP 100 --minDP 5 --maxDP 100 --recode --out "./filtered.m5-100_miss0.50_2alleles" &> "$log_file"
```

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

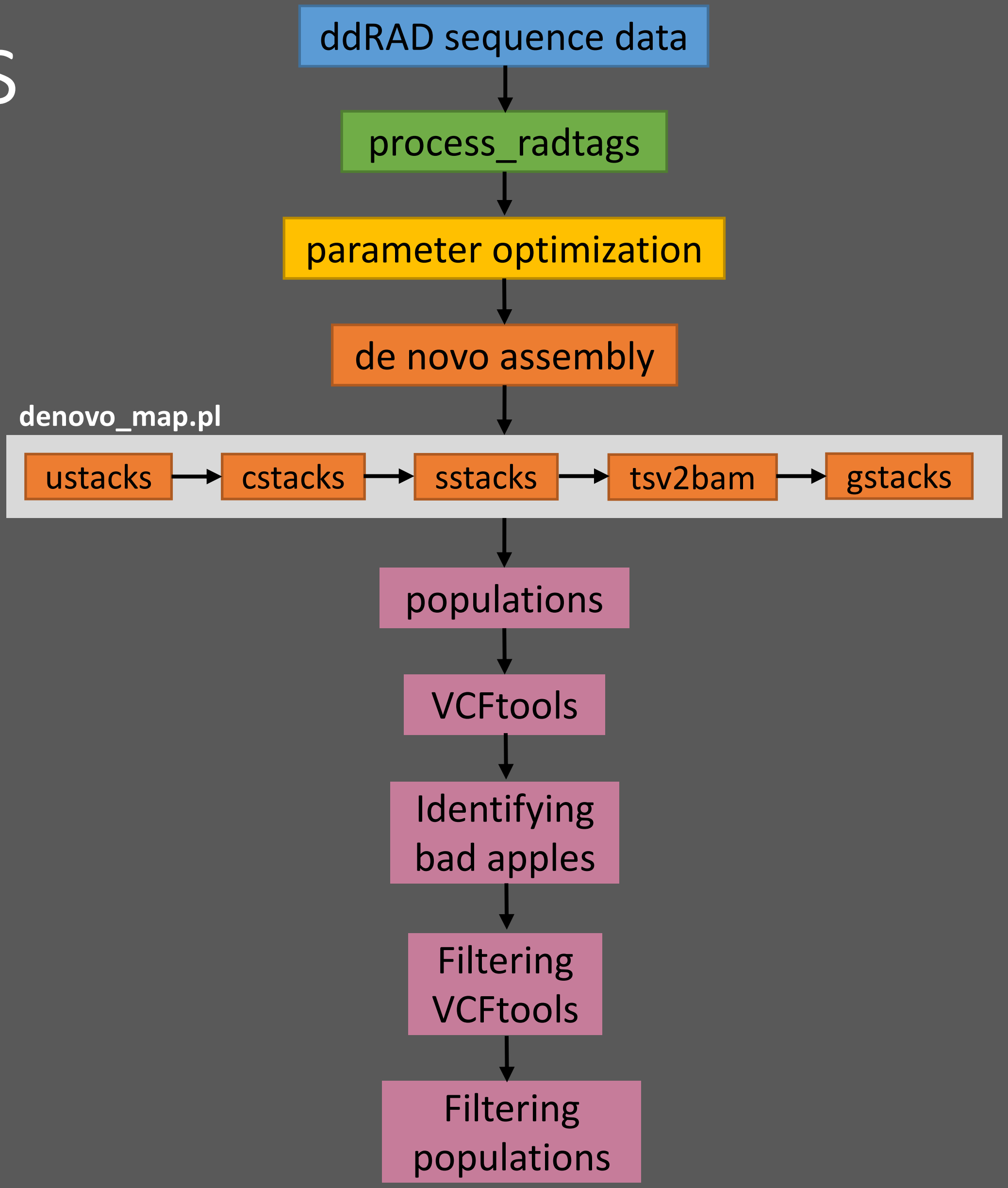
populations

VCFtools

Identifying bad apples

Filtering VCFtools

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

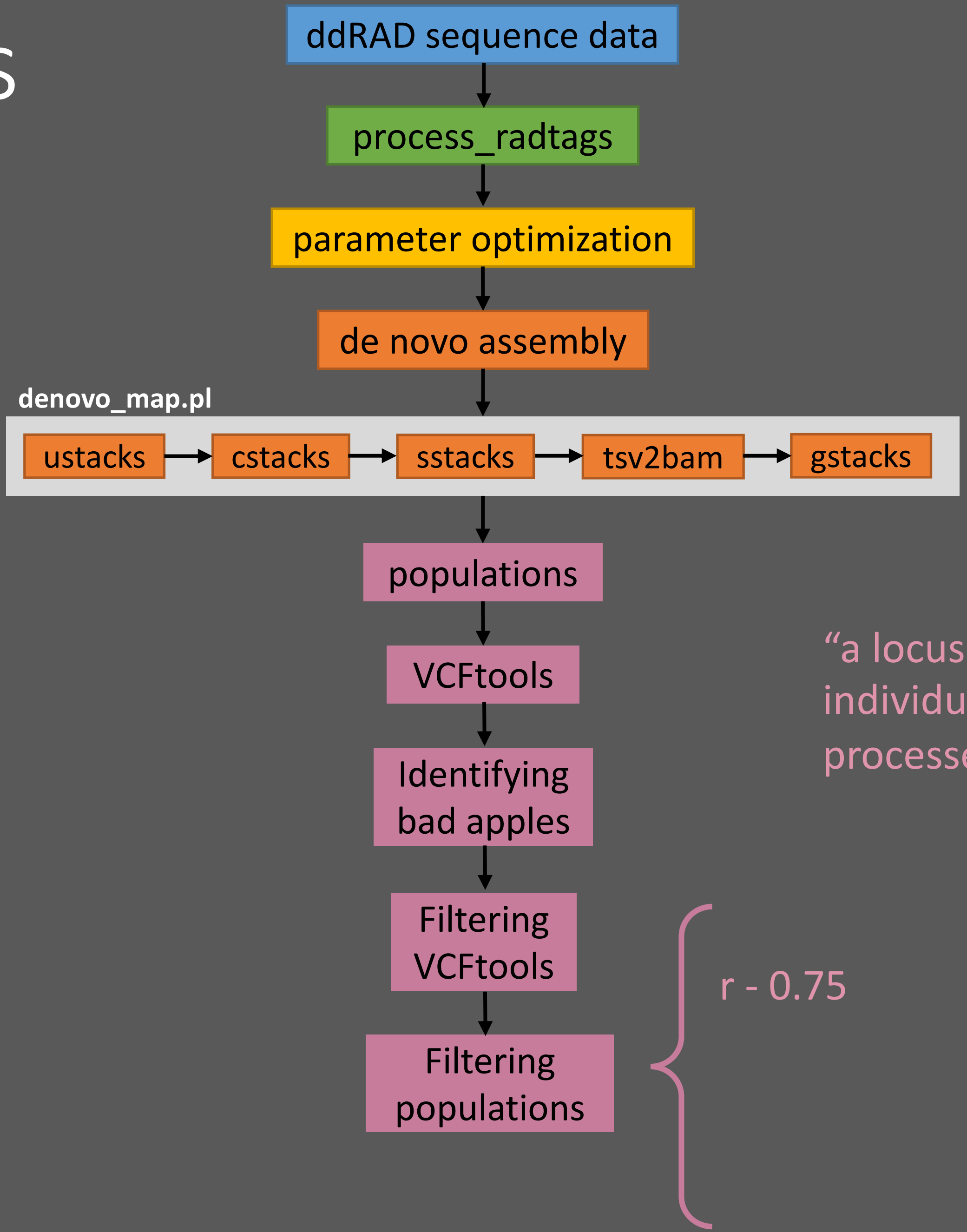
VCFtools

Identifying bad apples

Filtering VCFtools

Filtering populations

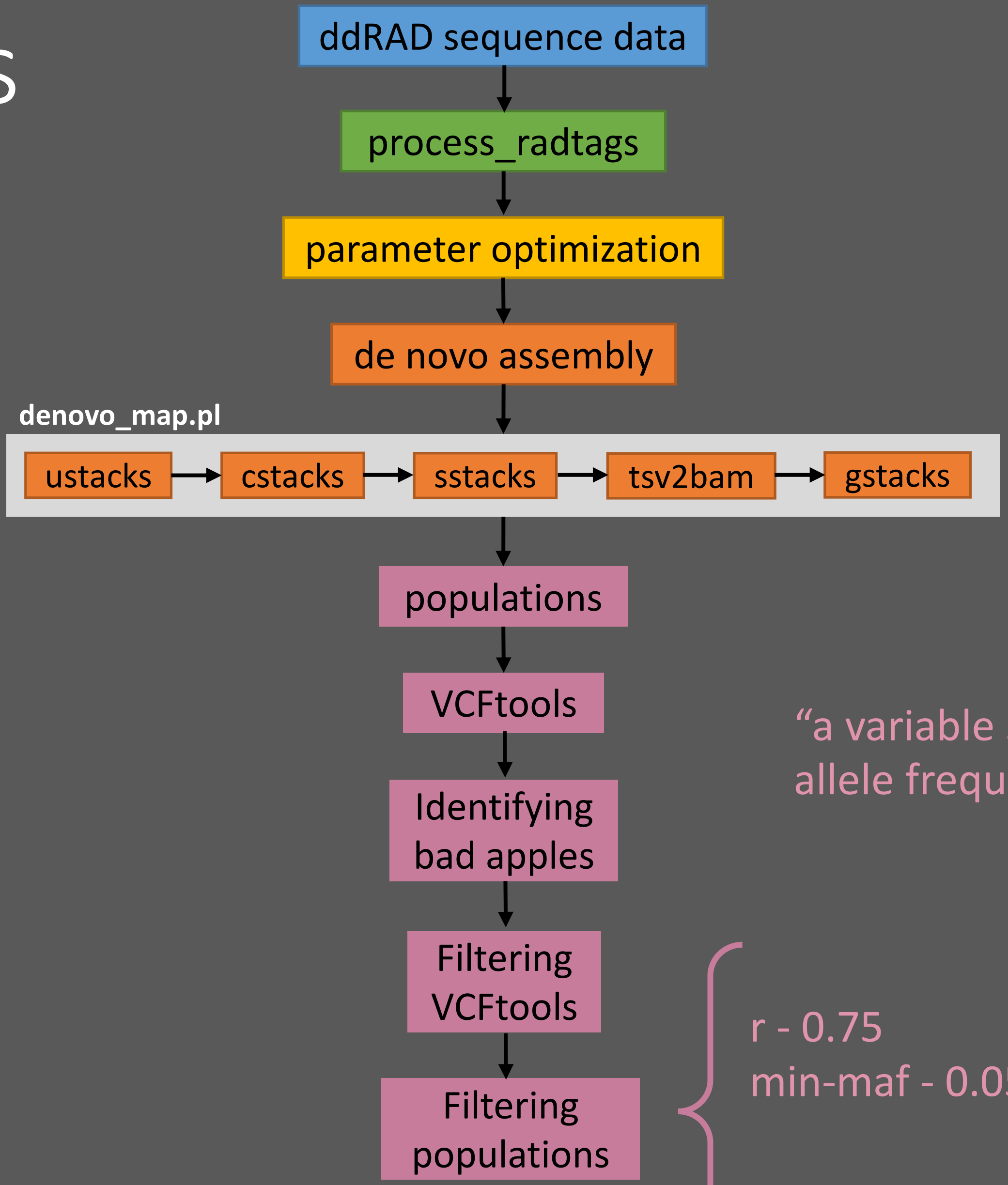
Bioinformatics Workflow



r - 0.75

"a locus must be found in 75% of individuals of a single population to be processed"

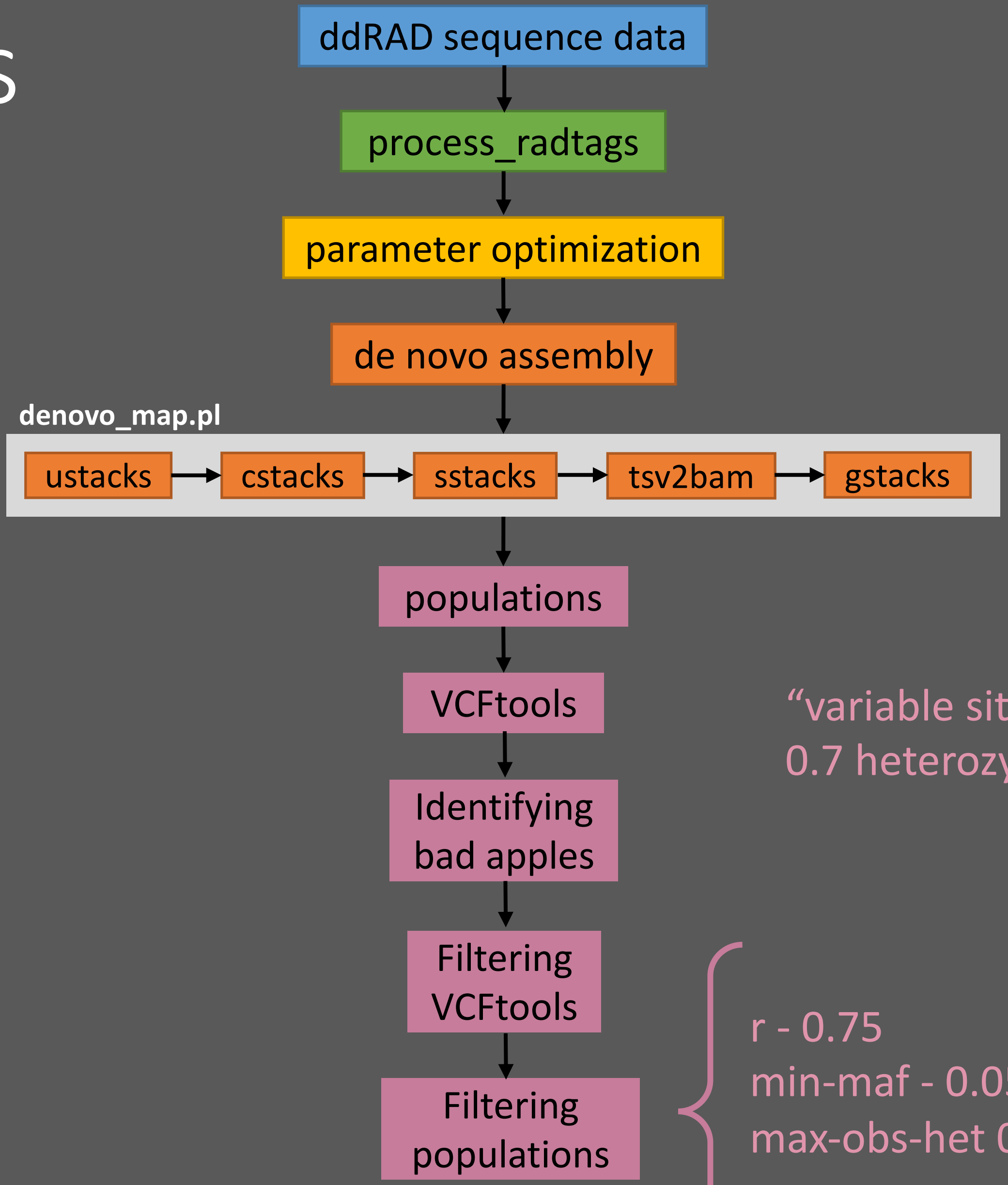
Bioinformatics Workflow



“a variable site must possess a minimum allele frequency of 5% to be included”

r - 0.75
min-maf - 0.05

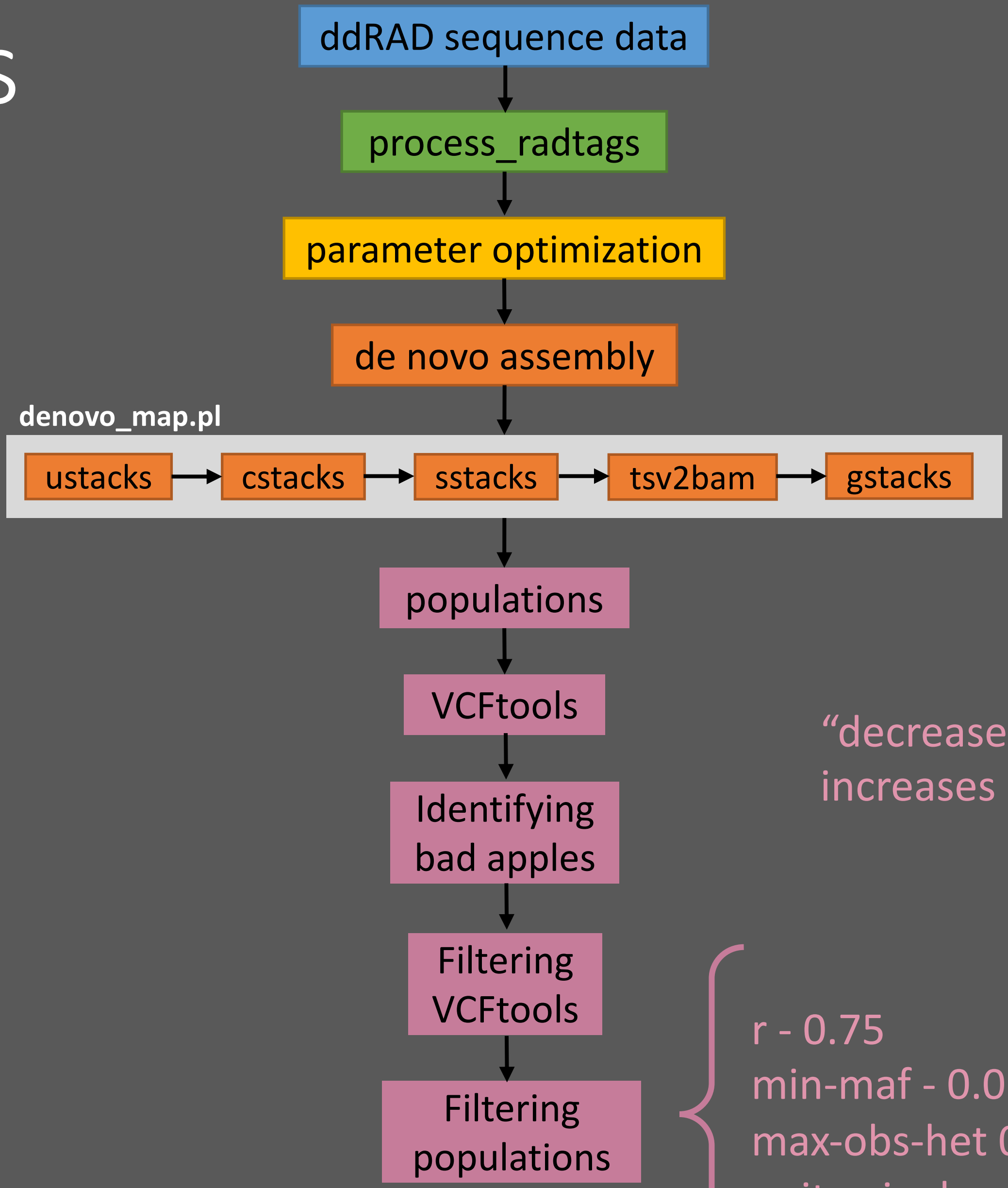
Bioinformatics Workflow



“variable sites with a maximum level of 0.7 heterozygosity will be included”

r - 0.75
min-maf - 0.05
max-obs-het 0.7

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

VCFtools

Identifying bad apples

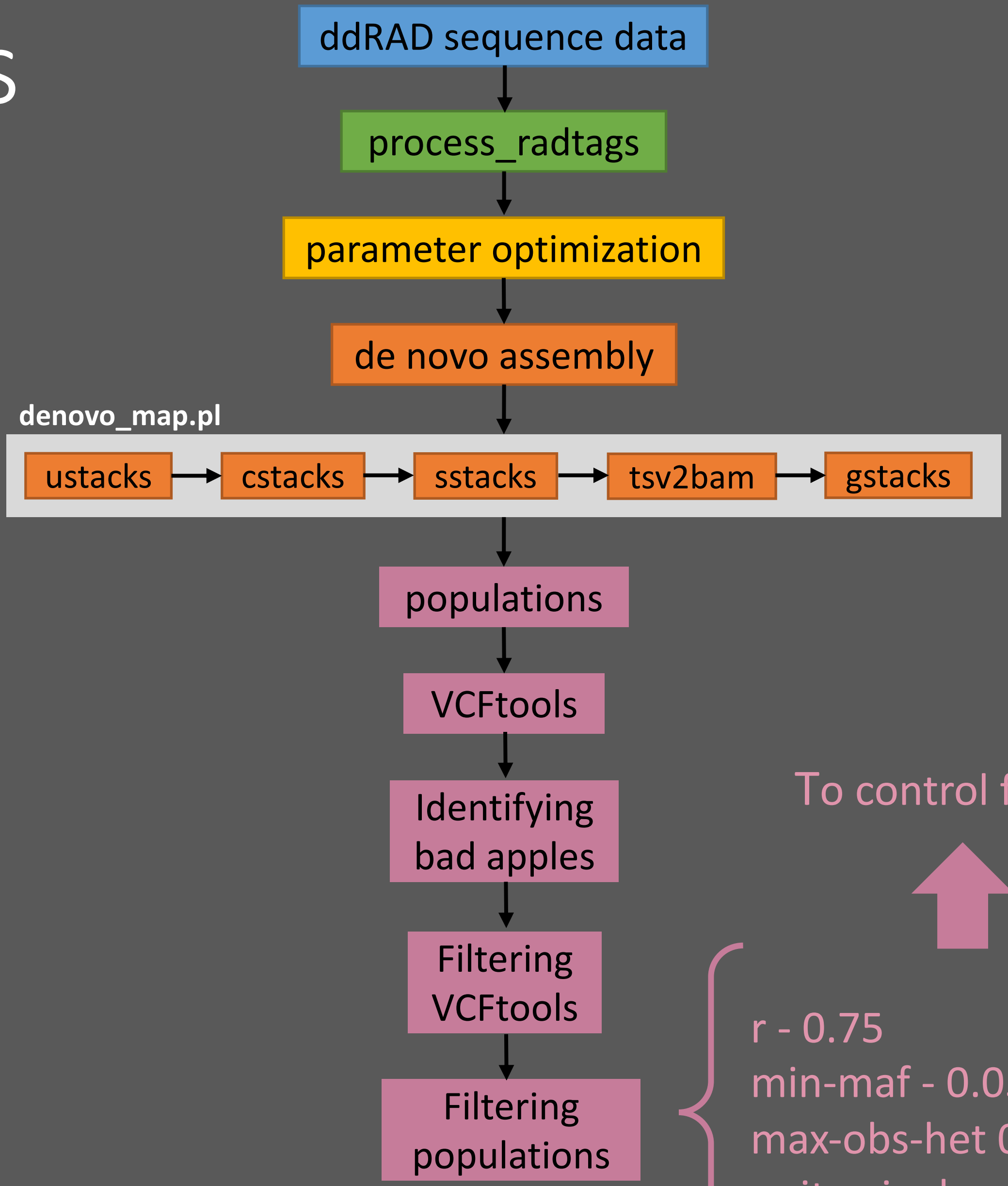
Filtering VCFtools

Filtering populations

“decreases linkage disequilibrium and increases reproducibility”

r - 0.75
min-maf - 0.05
max-obs-het 0.7
write-single-snp

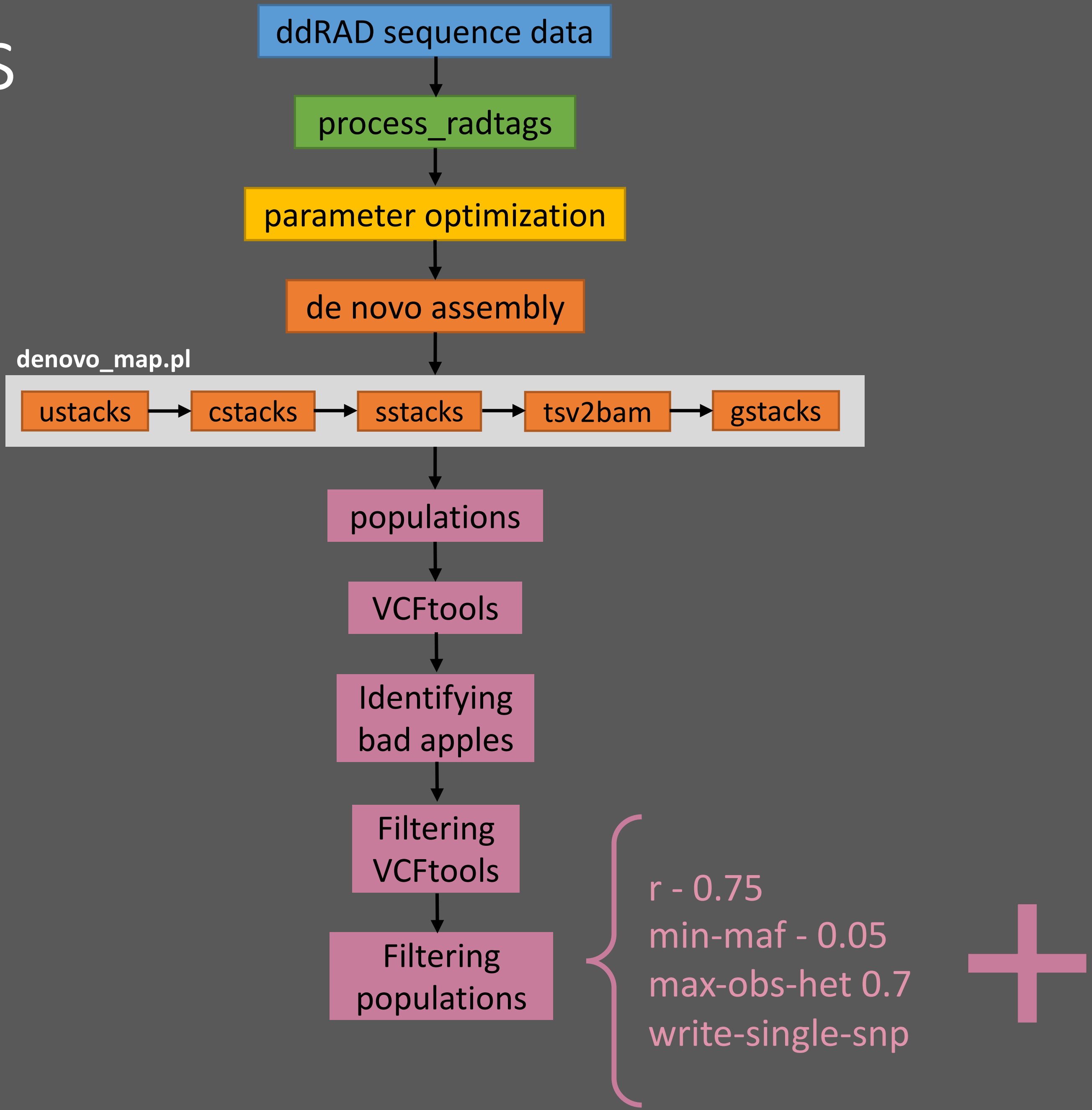
Bioinformatics Workflow



To control for false-positive loci

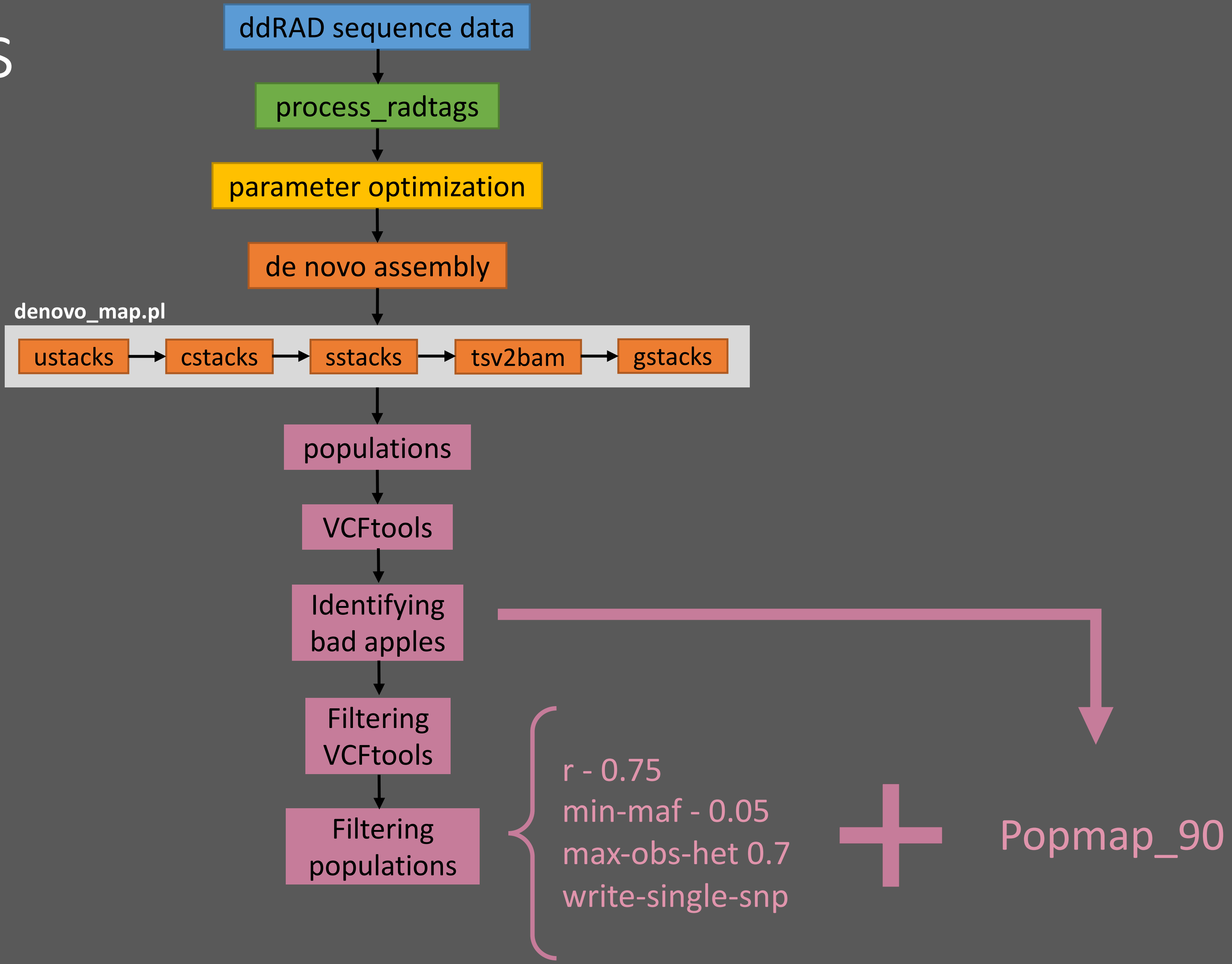
r - 0.75
min-maf - 0.05
max-obs-het 0.7
write-single-snp

Bioinformatics Workflow



r - 0.75
min-maf - 0.05
max-obs-het 0.7
write-single-snp

Bioinformatics Workflow



ddRAD sequence data

process_radtags

parameter optimization

de novo assembly

denovo_map.pl

ustacks → cstacks → sstacks → tsv2bam → gstacks

populations

VCFtools

Identifying bad apples

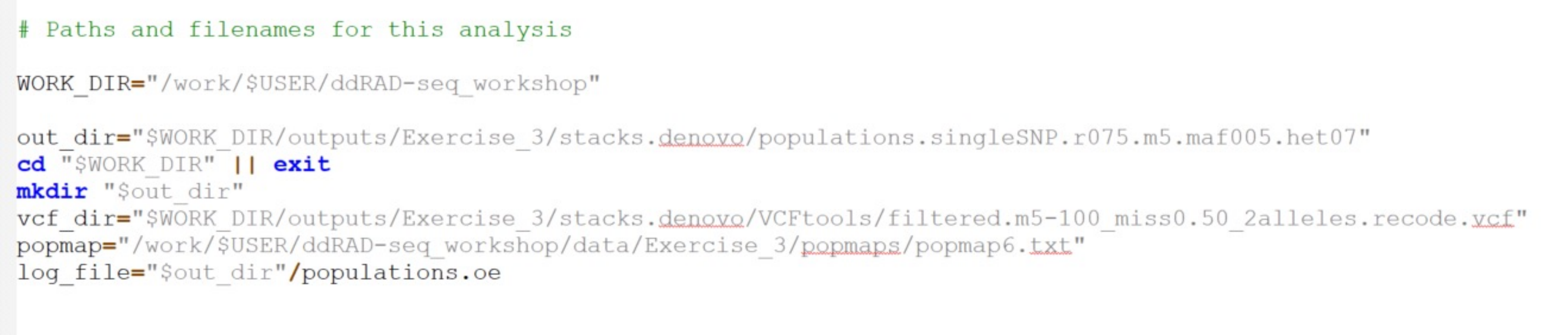
Filtering VCFtools

Filtering populations

r - 0.75
min-maf - 0.05
max-obs-het 0.7
write-single-snp

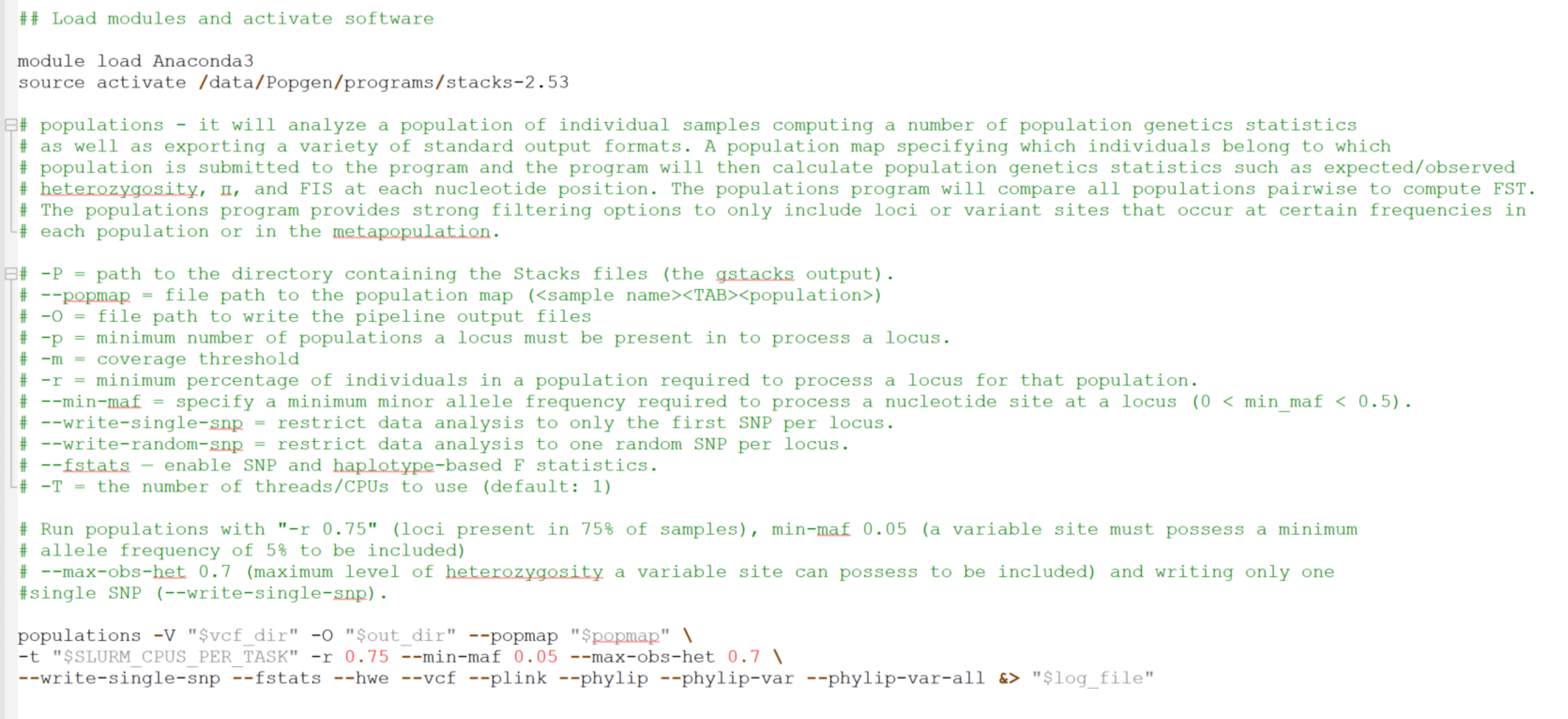
Popmap_90

06_populations_filtering



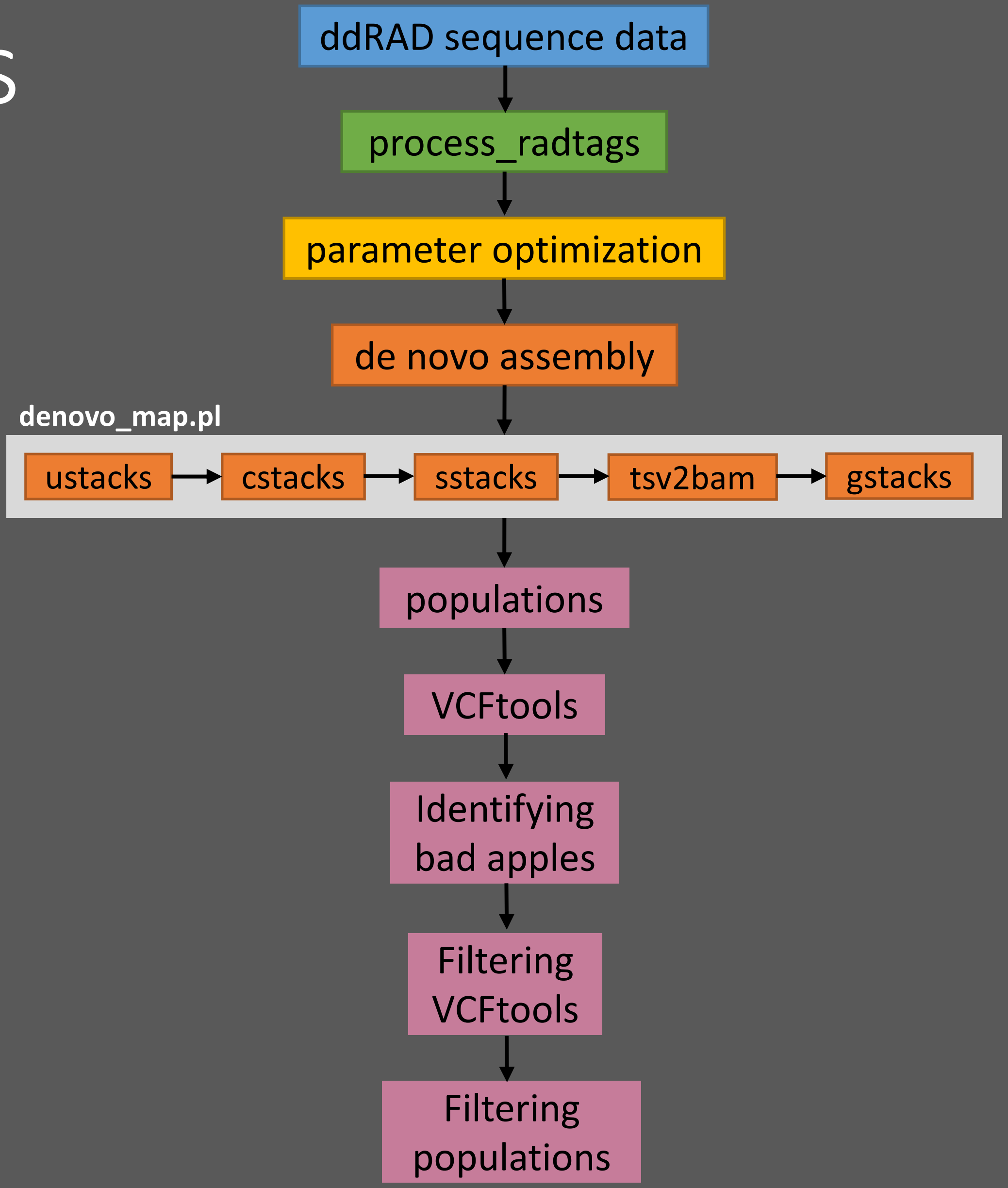
```
# Paths and filenames for this analysis

WORK_DIR="/work/$USER/ddRAD-seq_workshop"

out_dir="$WORK_DIR/outputs/Exercise_3/stacks.denovo/populations.singleSNP.r075.m5.maf005.het07"
cd "$WORK_DIR" || exit
mkdir "$out_dir"
vcf_dir="$WORK_DIR/outputs/Exercise_3/stacks.denovo/VCFtools/filtered.m5-100_miss0.50_2alleles.recode.vcf"
popmap="/work/$USER/ddRAD-seq_workshop/data/Exercise_3/popmaps/popmap6.txt"
log_file="$out_dir"/populations.oe
```

06_populations_filtering

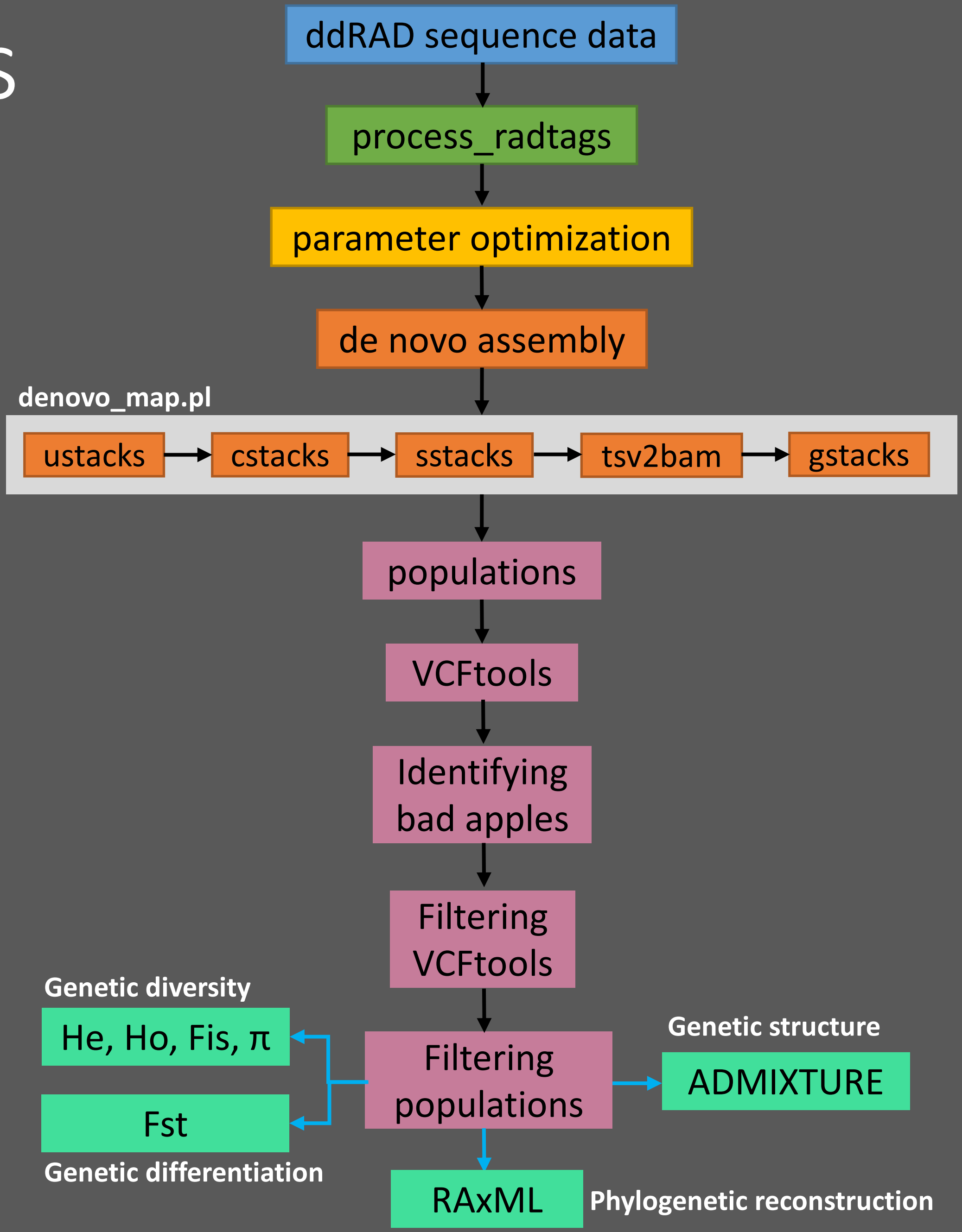


```
## Load modules and activate software

module load Anaconda3
source activate /data/Popgen/programs/stacks-2.53

# populations - it will analyze a population of individual samples computing a number of population genetics statistics
# as well as exporting a variety of standard output formats. A population map specifying which individuals belong to which
# population is submitted to the program and the program will then calculate population genetics statistics such as expected/observed
# heterozygosity, π, and FIS at each nucleotide position. The populations program will compare all populations pairwise to compute FST.
# The populations program provides strong filtering options to only include loci or variant sites that occur at certain frequencies in
# each population or in the metapopulation.

# -P = path to the directory containing the Stacks files (the gstacks output).
# --popmap = file path to the population map (<sample name><TAB><population>)
# -O = file path to write the pipeline output files
# -p = minimum number of populations a locus must be present in to process a locus.
# -m = coverage threshold
# -r = minimum percentage of individuals in a population required to process a locus for that population.
# --min-maf = specify a minimum minor allele frequency required to process a nucleotide site at a locus (0 < min_maf < 0.5).
# --write-single-snp = restrict data analysis to only the first SNP per locus.
# --write-random-snp = restrict data analysis to one random SNP per locus.
# --fstats — enable SNP and haplotype-based F statistics.
# -T = the number of threads/CPUs to use (default: 1)

# Run populations with "-r 0.75" (loci present in 75% of samples), min-maf 0.05 (a variable site must possess a minimum
# allele frequency of 5% to be included)
# --max-obs-het 0.7 (maximum level of heterozygosity a variable site can possess to be included) and writing only one
#single SNP (--write-single-snp).

populations -V "$vcf_dir" -O "$out_dir" --popmap "$popmap" \
-t "$SLURM_CPUS_PER_TASK" -r 0.75 --min-maf 0.05 --max-obs-het 0.7 \
--write-single-snp --fstats --hwe --vcf --plink --phylip --phylip-var --phylip-var-all &> "$log_file"
```

Bioinformatics Workflow



Bioinformatics Workflow


ddRAD sequence data
process_radtags
parameter optimization
de novo assembly
denovo_map.pl
ustacks
cstacks
sstacks
tsv2bam
gstacks
populations
VCFtools
Identifying bad apples
Filtering VCFtools
Genetic diversity
He, Ho, Fis, π
Fst
Genetic differentiation
Filtering populations
Genetic structure
ADMIXTURE
RAxML
Phylogenetic reconstruction