iDiv

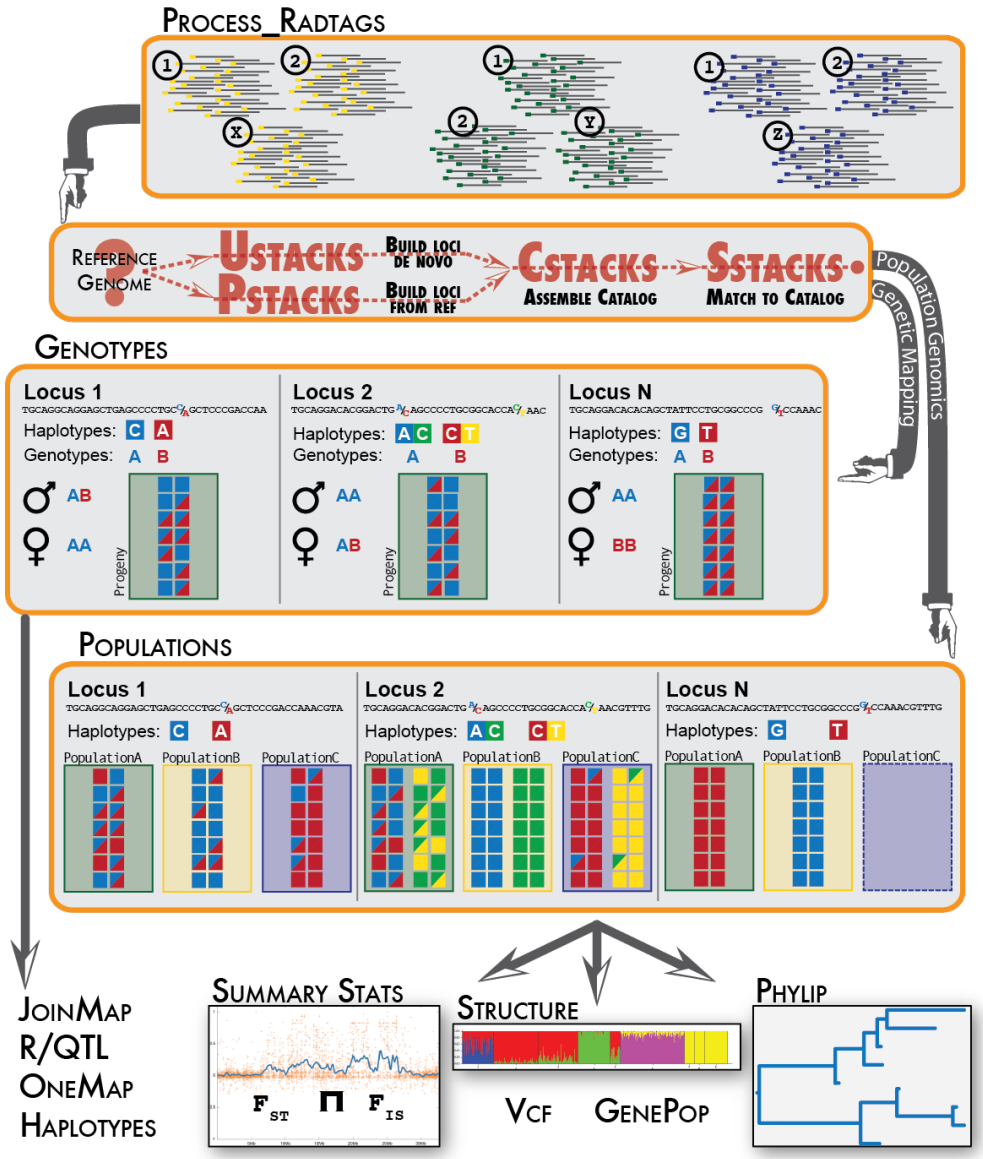
Estimating genetic diversity and population information from short read (ddRAD-seq) type data

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

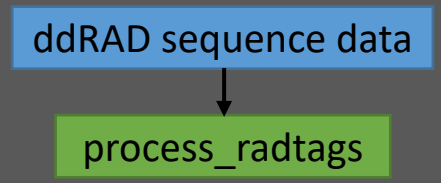
process_radtags — Clean and demultiplex the data

denovo_map.pl

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

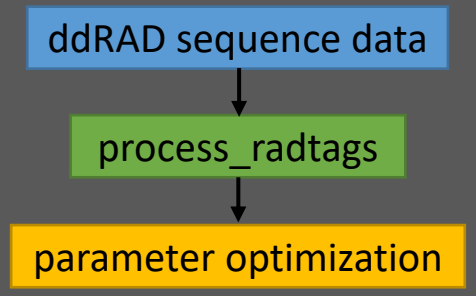
Output files for population structure analyses, phylogenetics, demographic history…

Bioinformatics Workflow

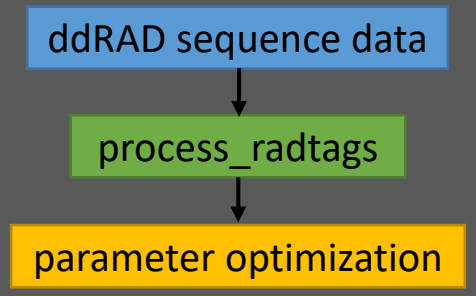


Bioinformatics Workflow

Bioinformatics Workflow



Bioinformatics Workflow



Methods in Ecology and Evolution

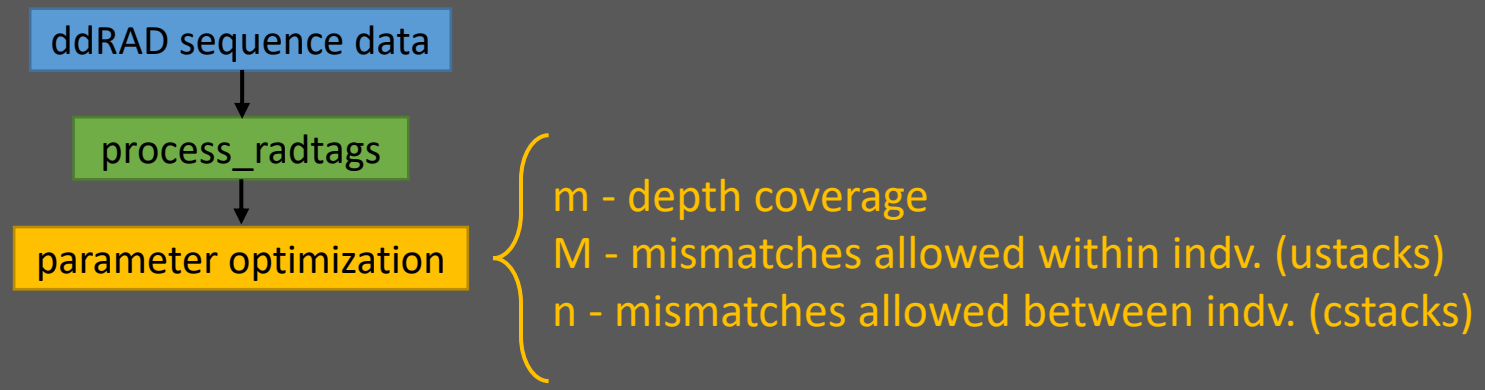
BRITISH ECOLOGICAL SOCIETY

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

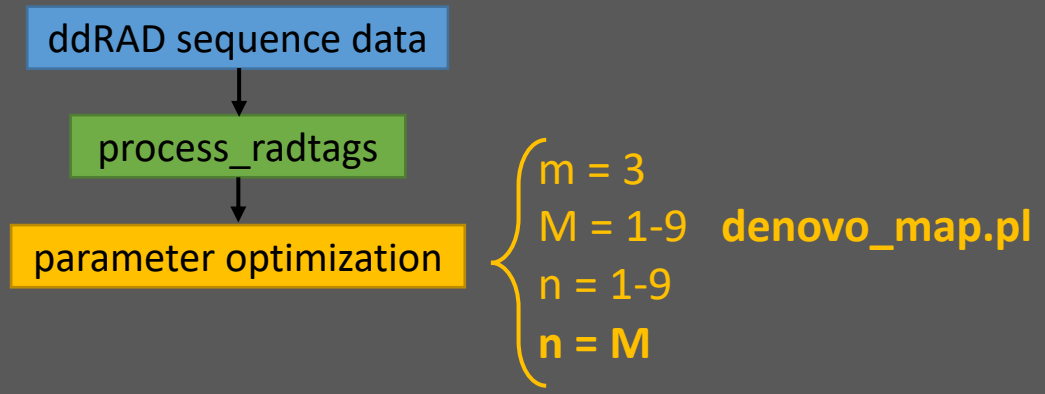
Lost in parameter space: a road map for STACKS

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

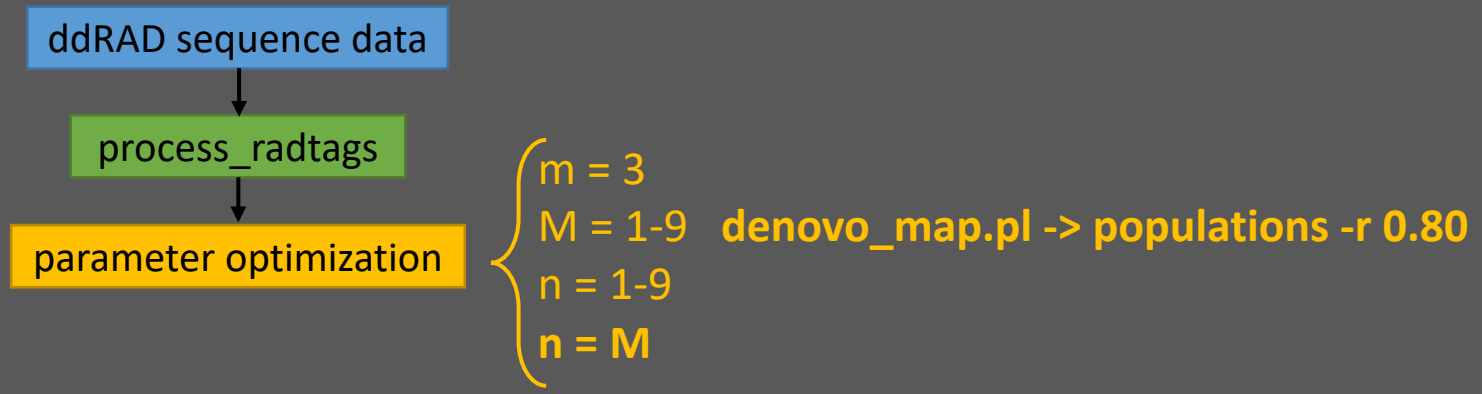
Bioinformatics Workflow



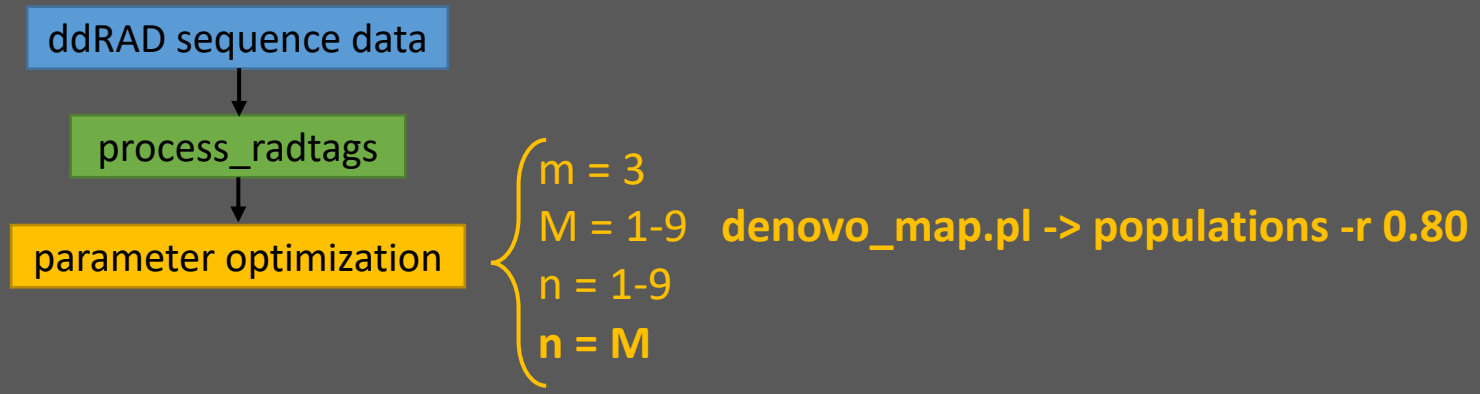
Bioinformatics Workflow



Bioinformatics Workflow

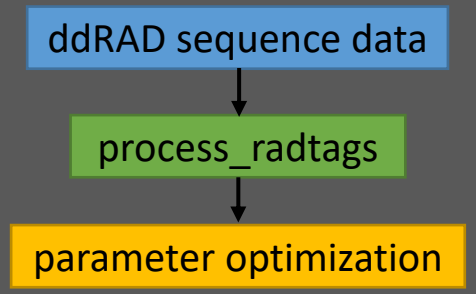


Bioinformatics Workflow



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

Bioinformatics Workflow



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

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
```

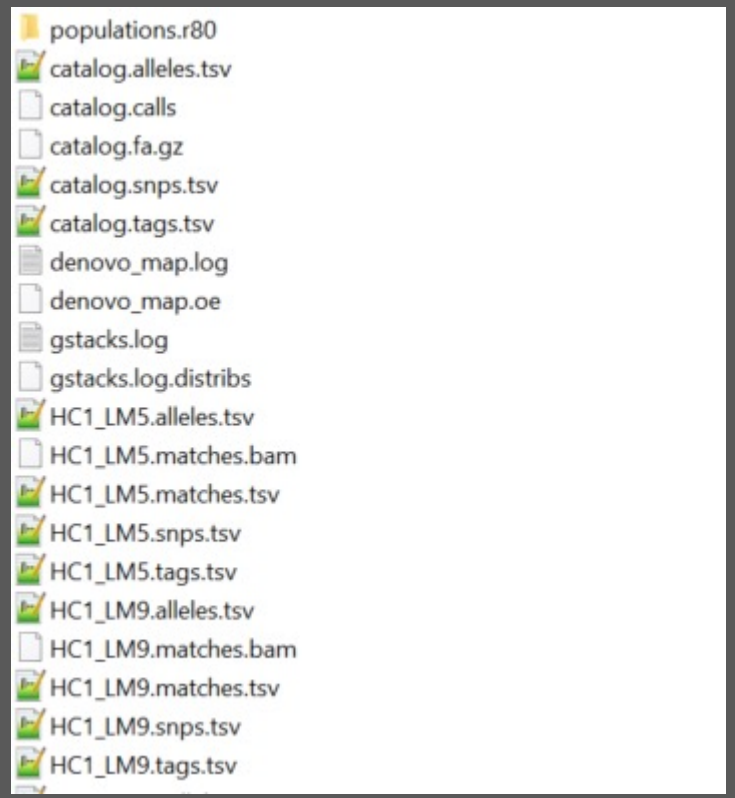
01_denovo_map_test.parameters

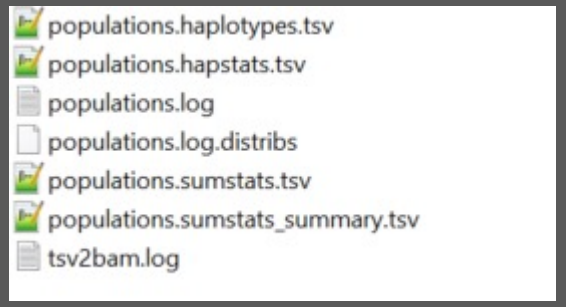
```
#!/bin/bash

#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

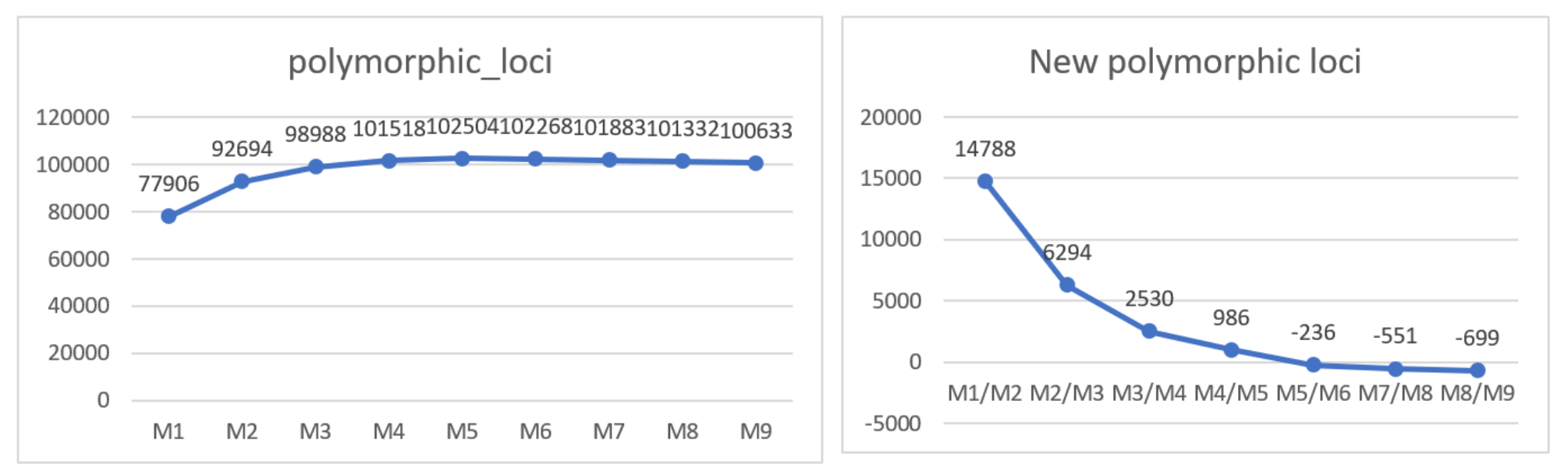
- Number of r80 polymorphic loci
- Number of new polymorphic found for each iteration of M

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.tsv
cat  results/*.polymorphicLOCI_summary.tsv > results/all.polymorphicLOCI_summary.tsv
done
```

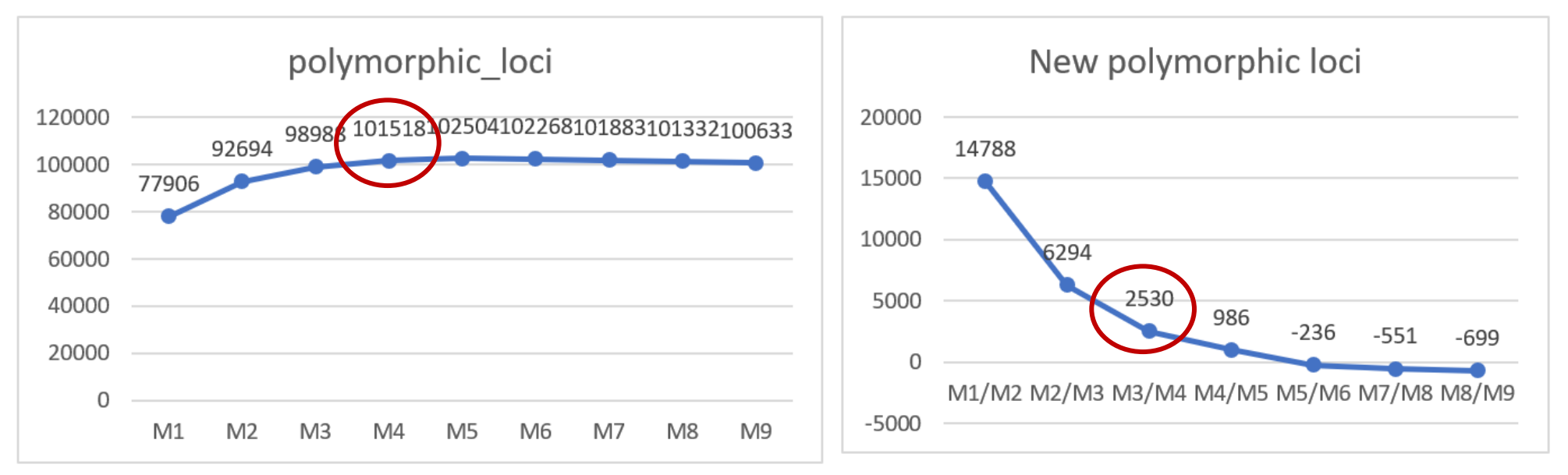
Bioinformatics

Bismarckia nobilis



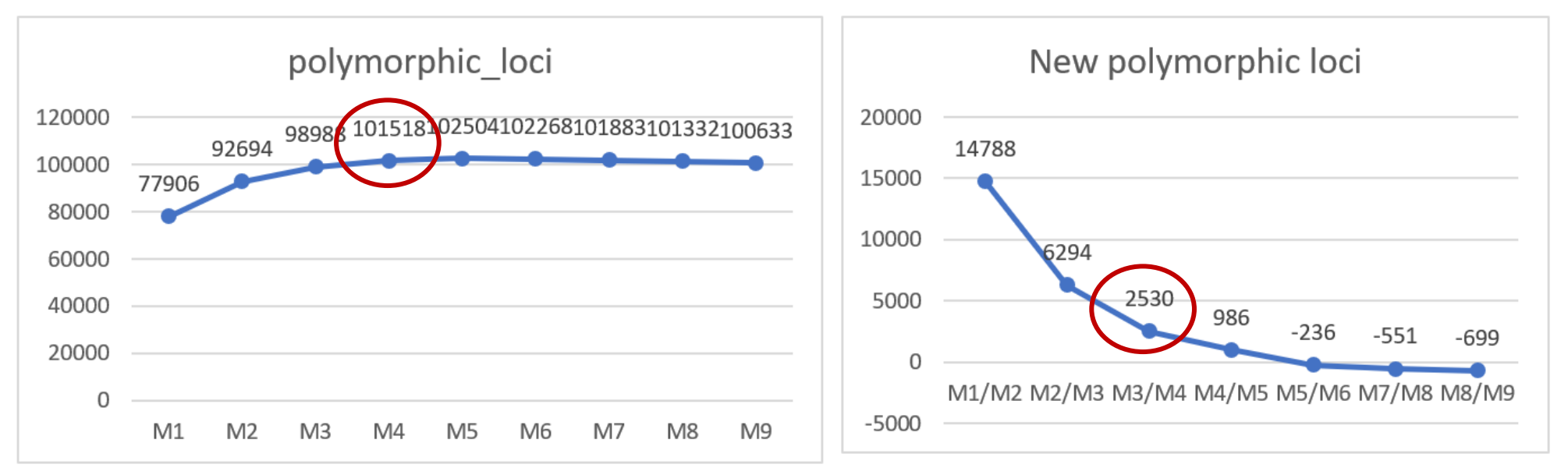
Bioinformatics

Bismarckia nobilis



Bioinformatics

Bismarckia nobilis

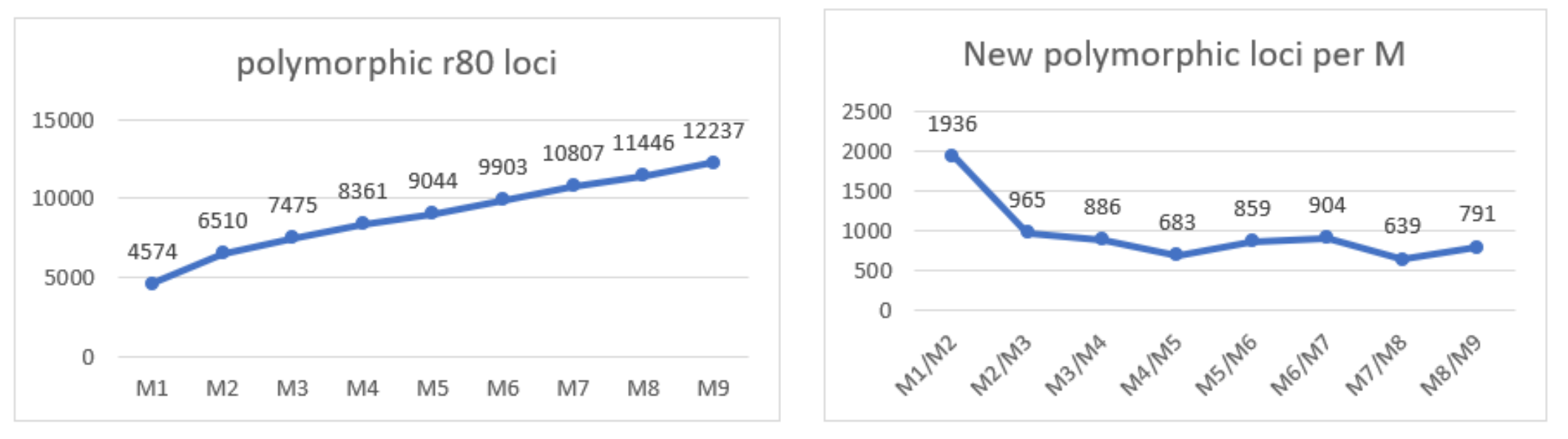


M = 4

M = n

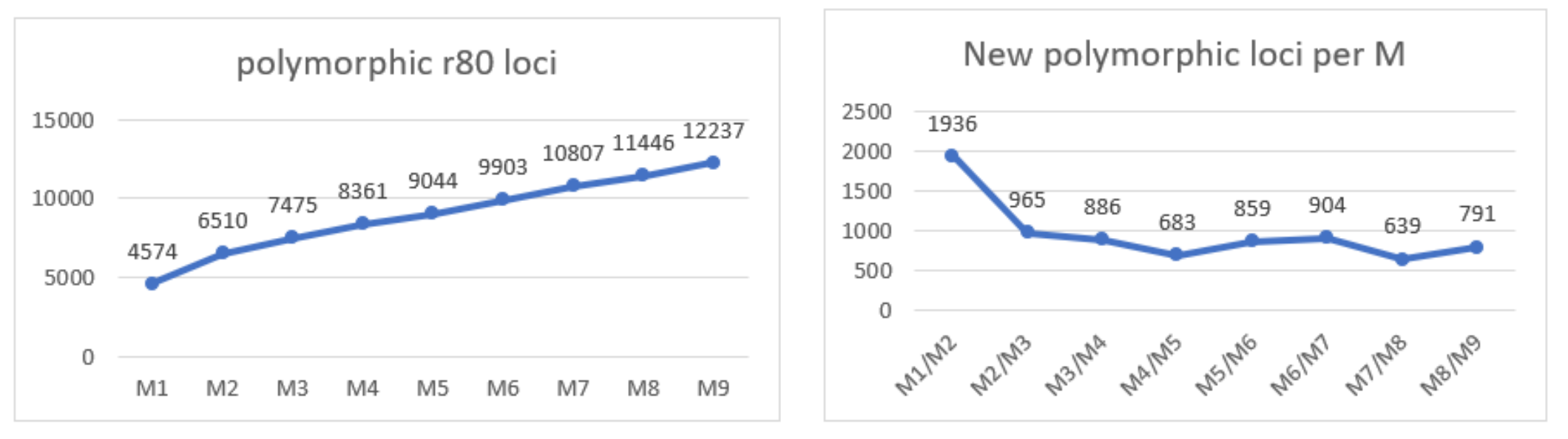
Bioinformatics

Dypsis pinnatifrons



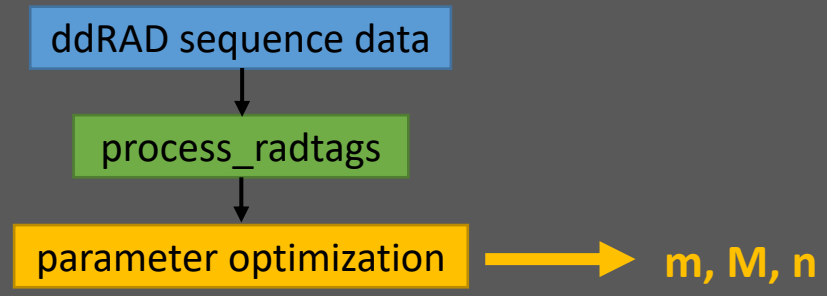
Bioinformatics

Dypsis pinnatifrons

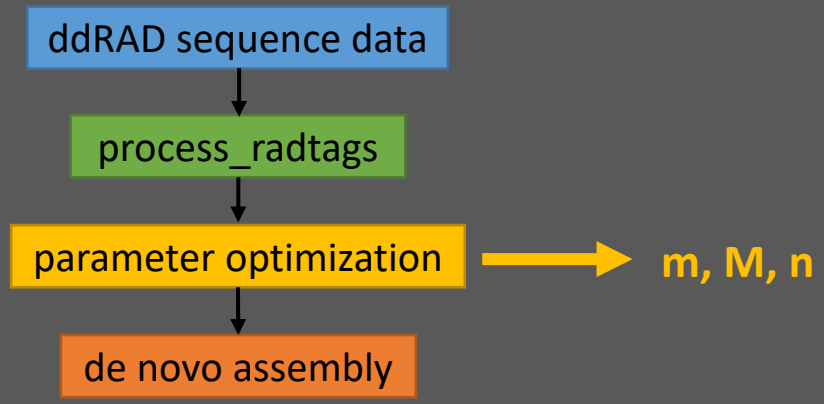


M = 5
M = n

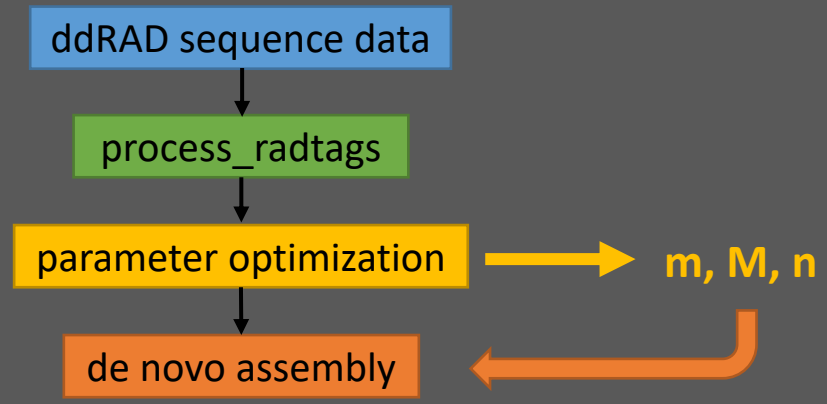
Bioinformatics Workflow



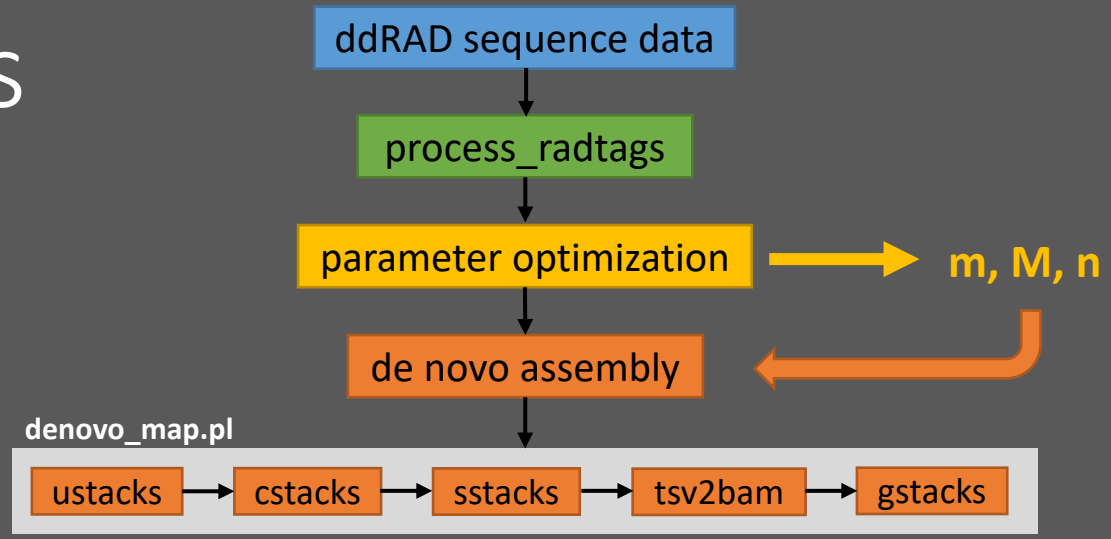
Bioinformatics Workflow



Bioinformatics Workflow



Bioinformatics Workflow



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

ddRAD sequencing

ddRADseq Data

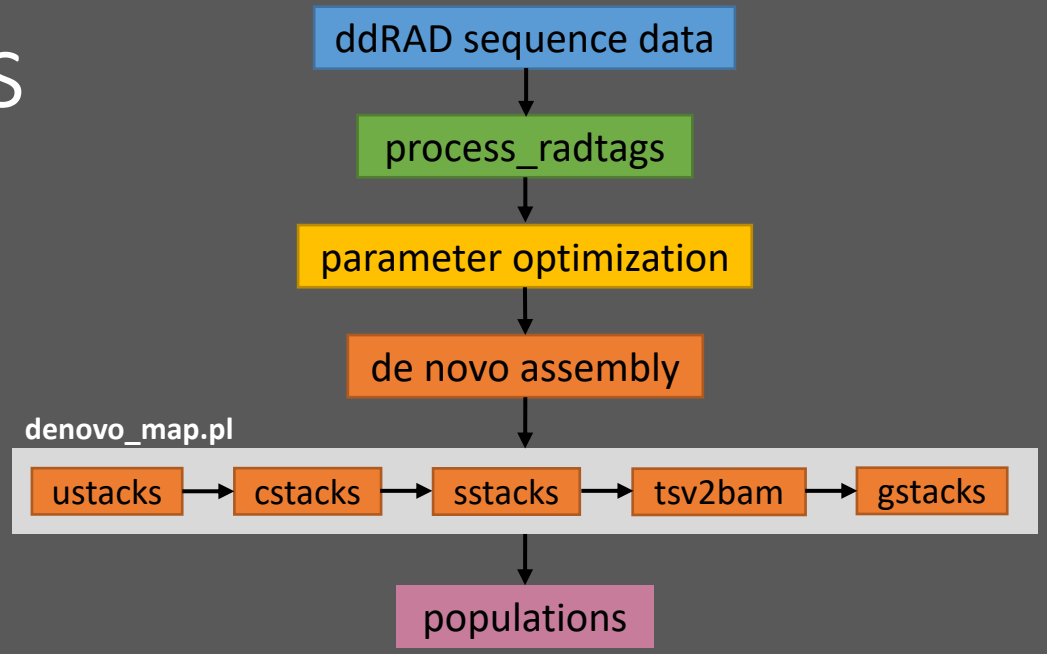
HPC cluster (EVE) - UFZ

Stacks v2.61

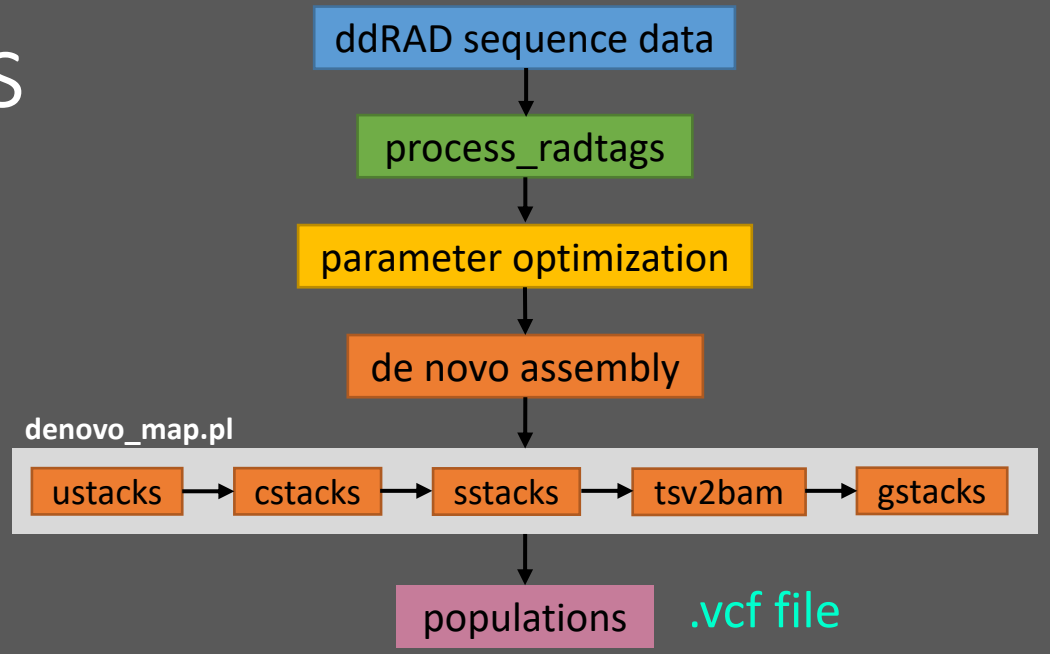
process_radtags	Clean and demultiplex the data
ustacks	Building loci *de novo* for each sample
cstacks	Creates a *catalog* of all loci across the populations according to sequence similarity
sstacks	Match each sample against the catalog
tsv2bam	Transpose the data to be organized by RAD locus. Paired-end 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…

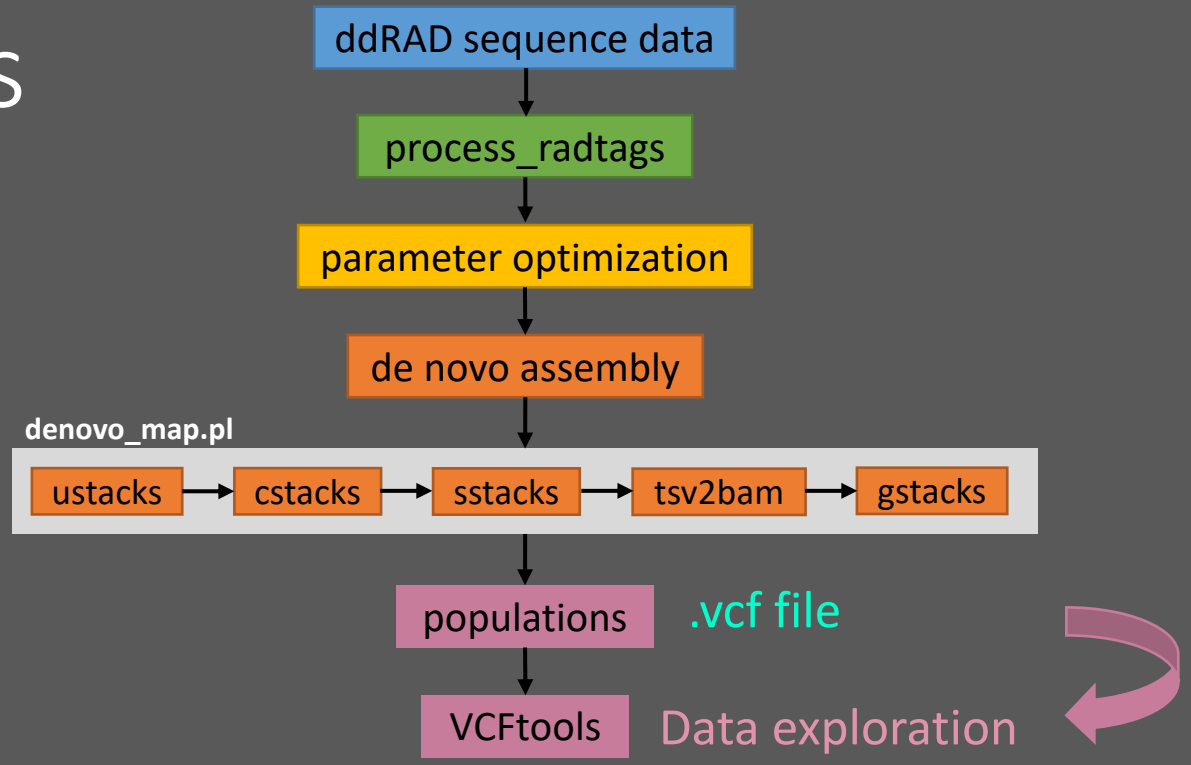
Bioinformatics Workflow



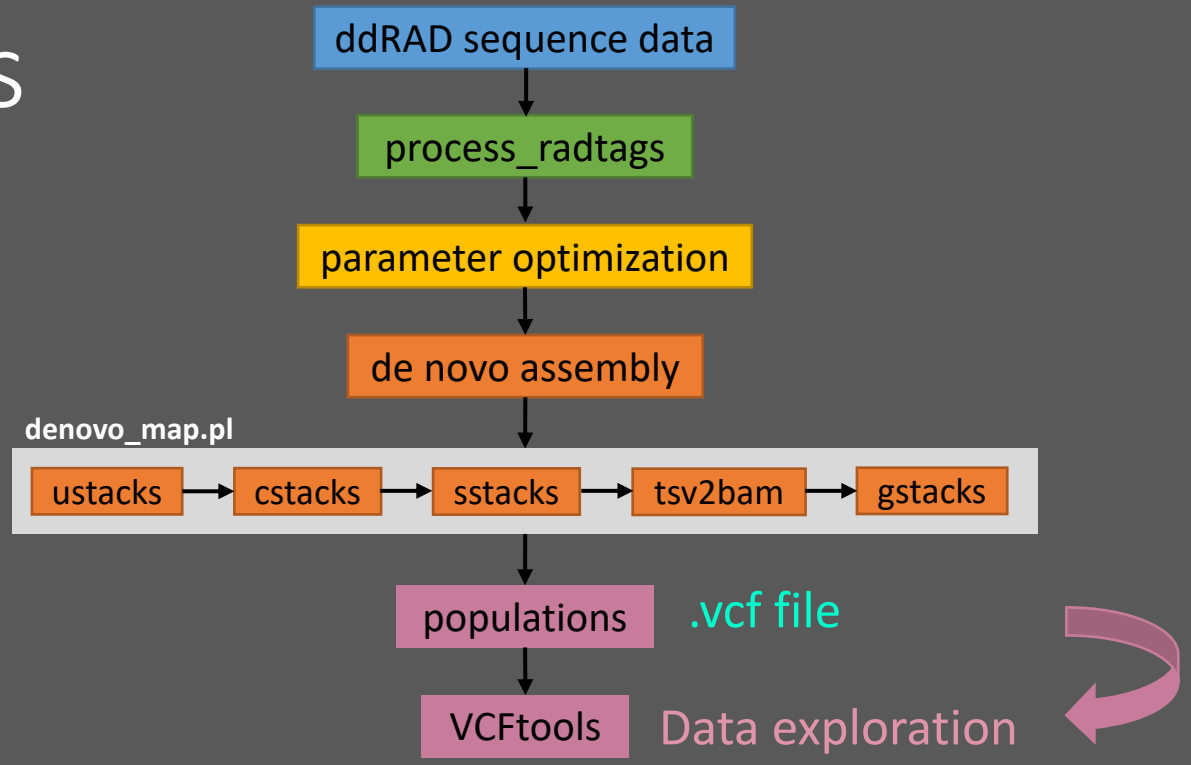
Bioinformatics Workflow



Bioinformatics Workflow



Bioinformatics Workflow



--freq2 = allele frequency

--depth = mean depth per individual

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

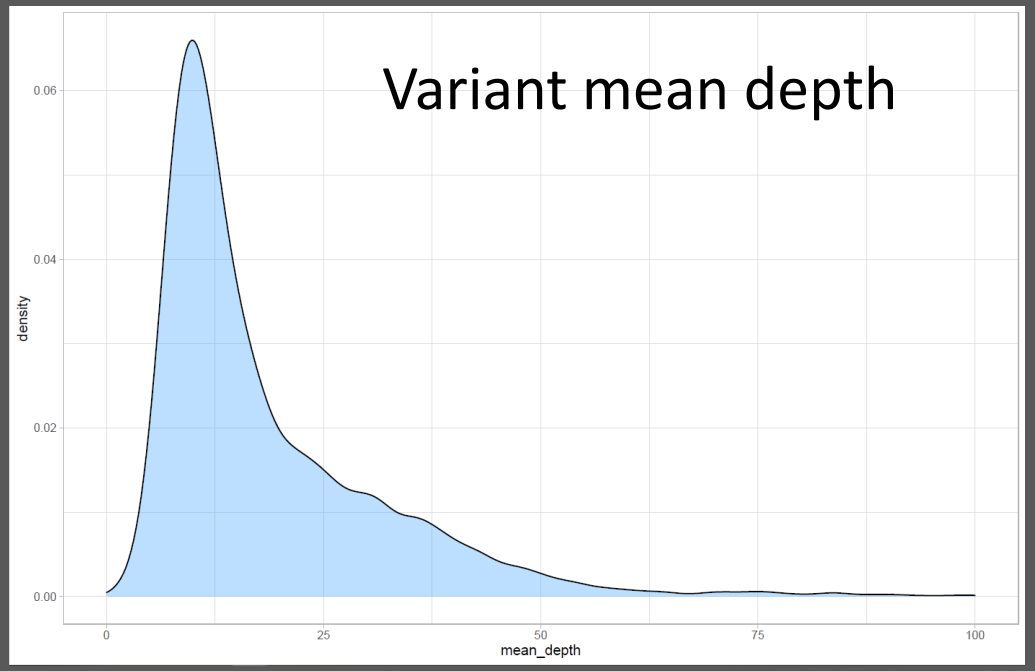
--missing-indv = proportion of missing data per individual

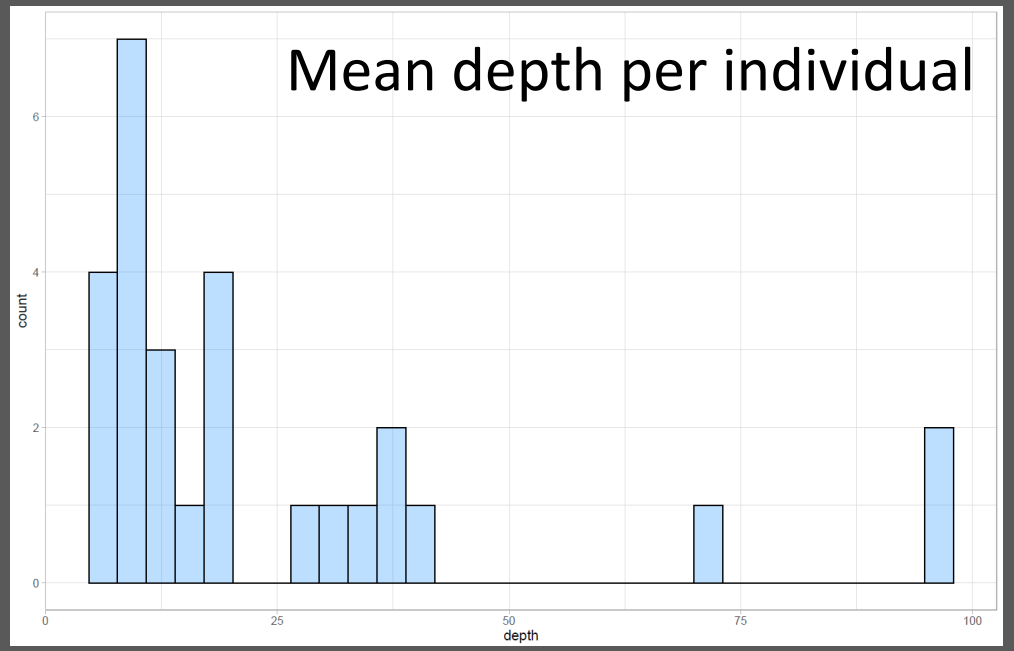
--missing-site = proportion of missing data per site

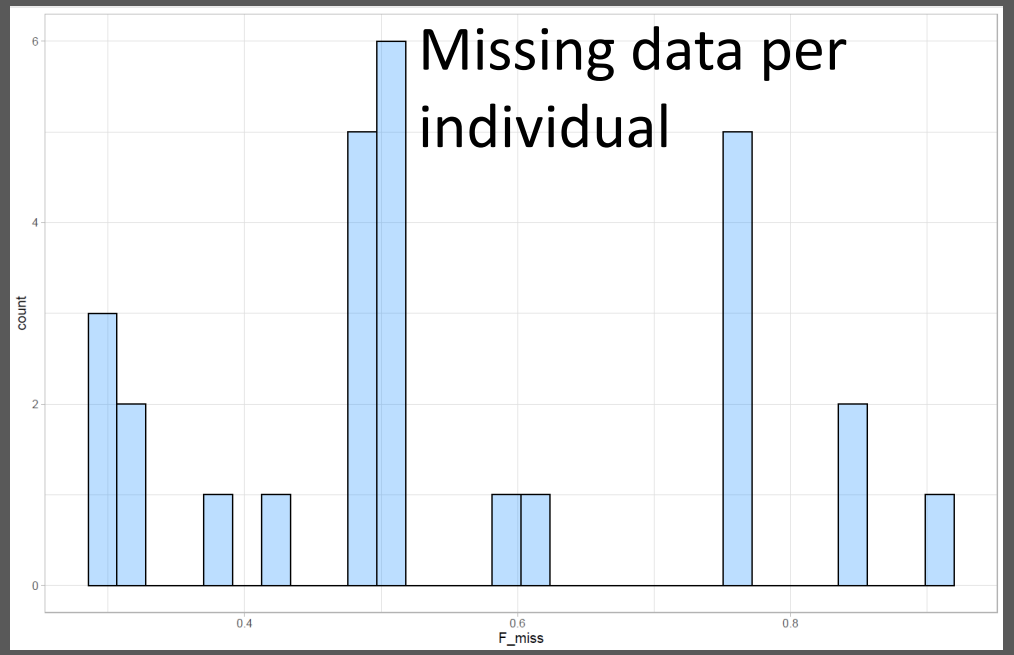
--het = heterozygosity and inbreeding coefficient per individual

Bioinformatics

Borassus madagascariensis
28 individuals



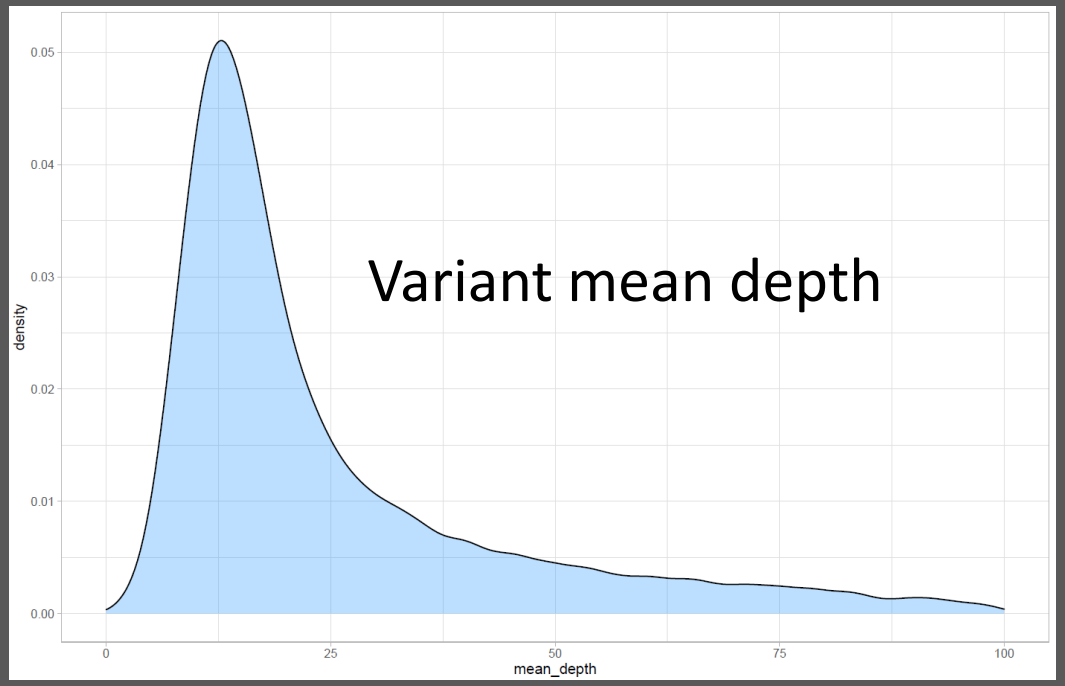


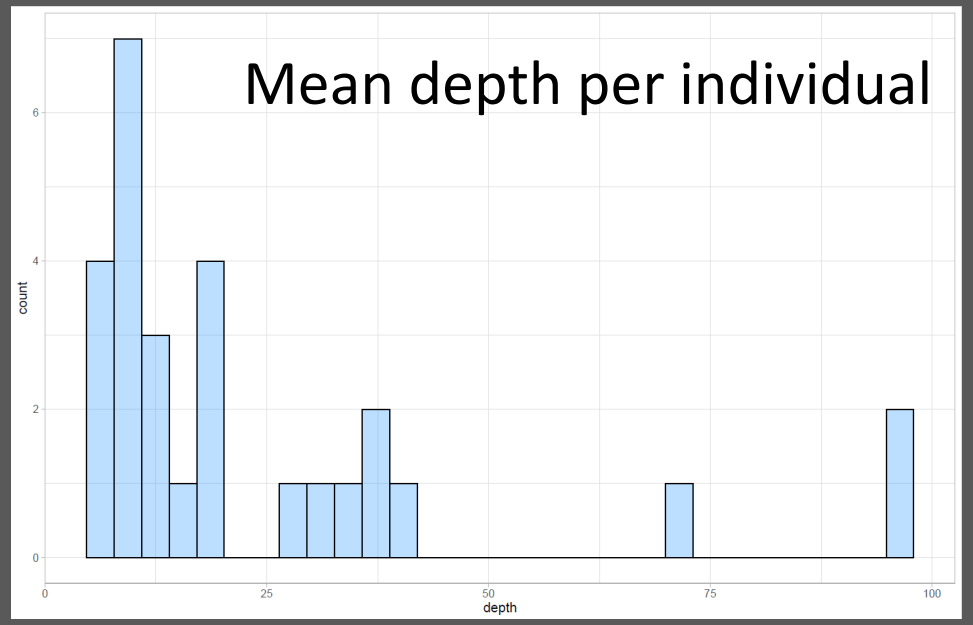


Bioinformatics

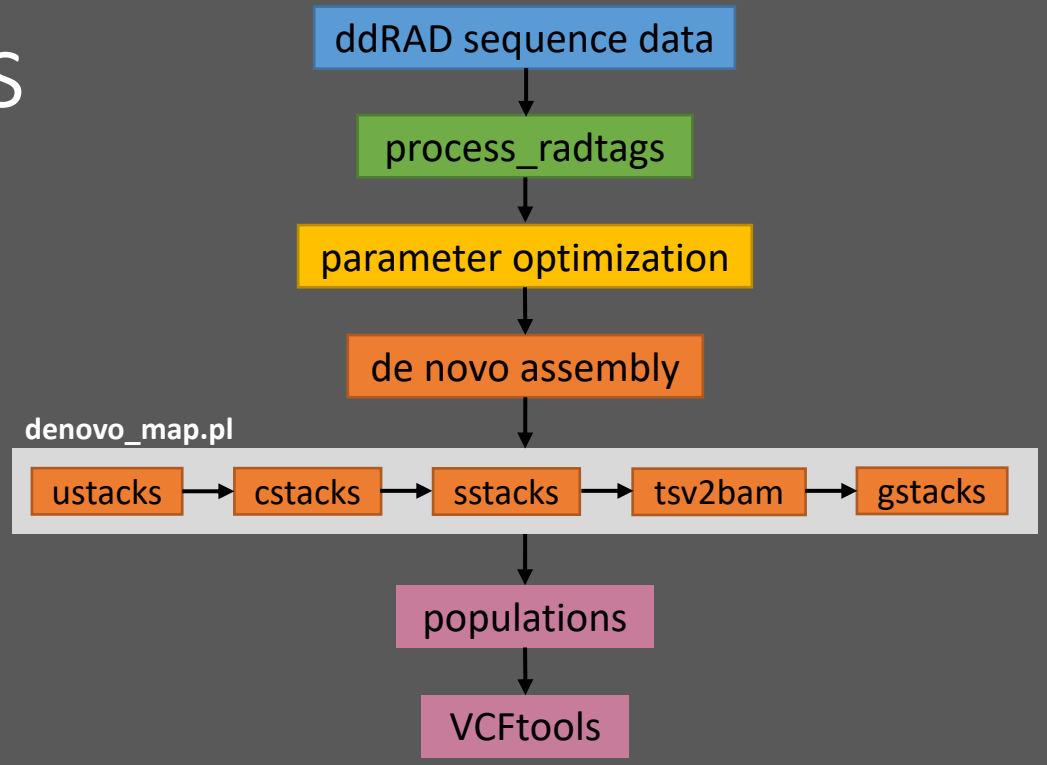
Bismarckia nobilis

63 individuals

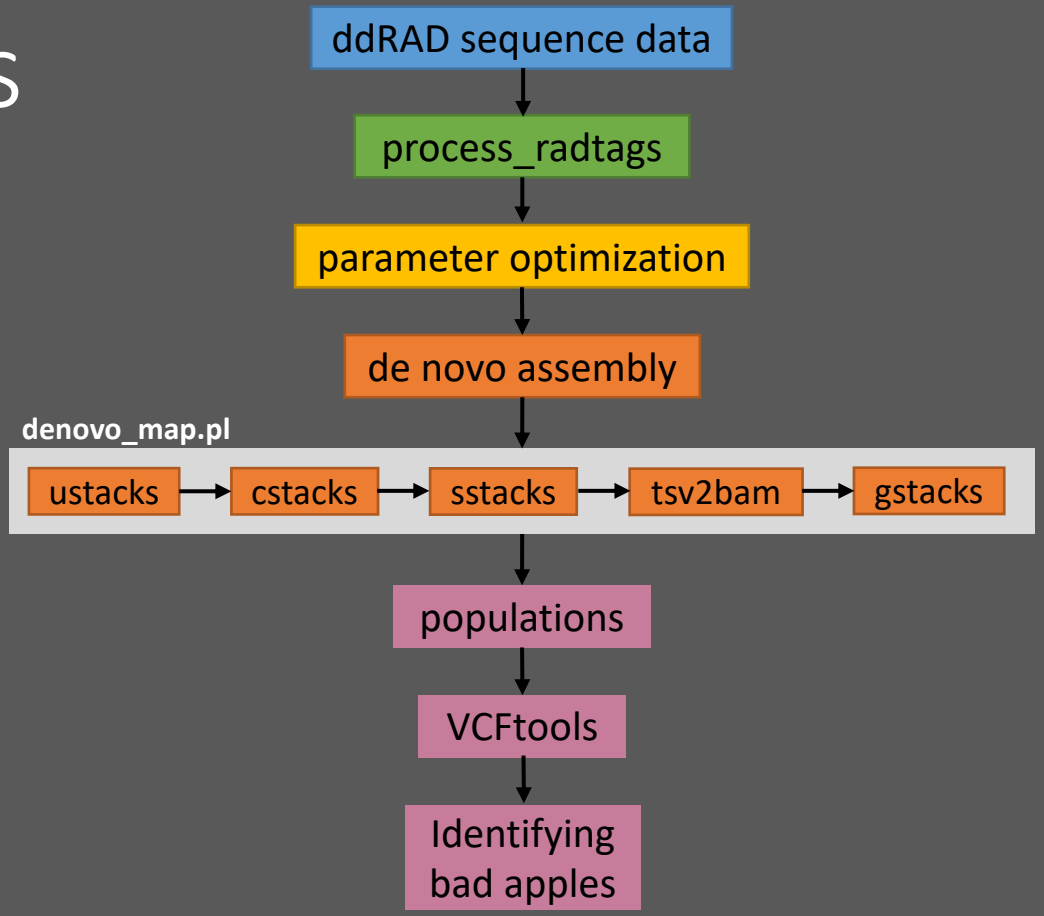




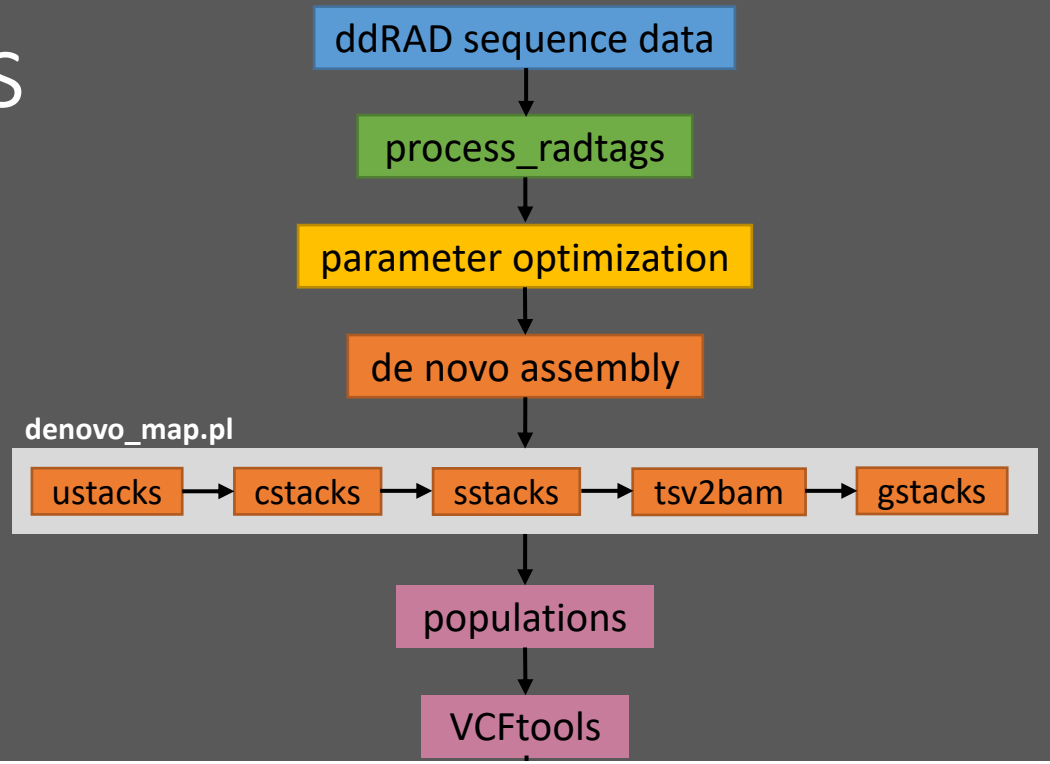
Bioinformatics Workflow



Bioinformatics Workflow



Bioinformatics Workflow



Received: 4 September 2020 | Accepted: 15 January 2021

DOI: 10.1111/2041-210X.13562

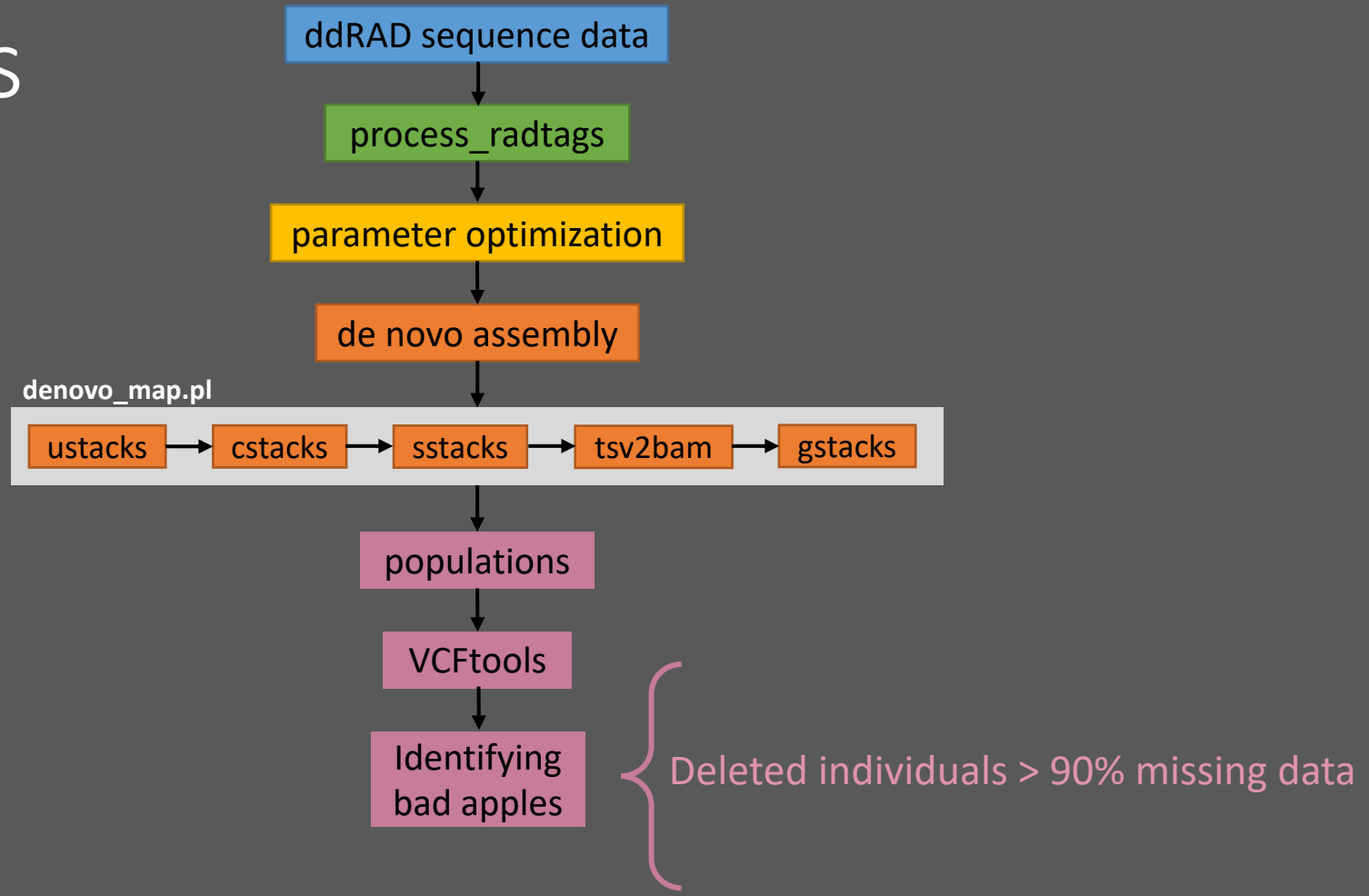
RESEARCH ARTICLE

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

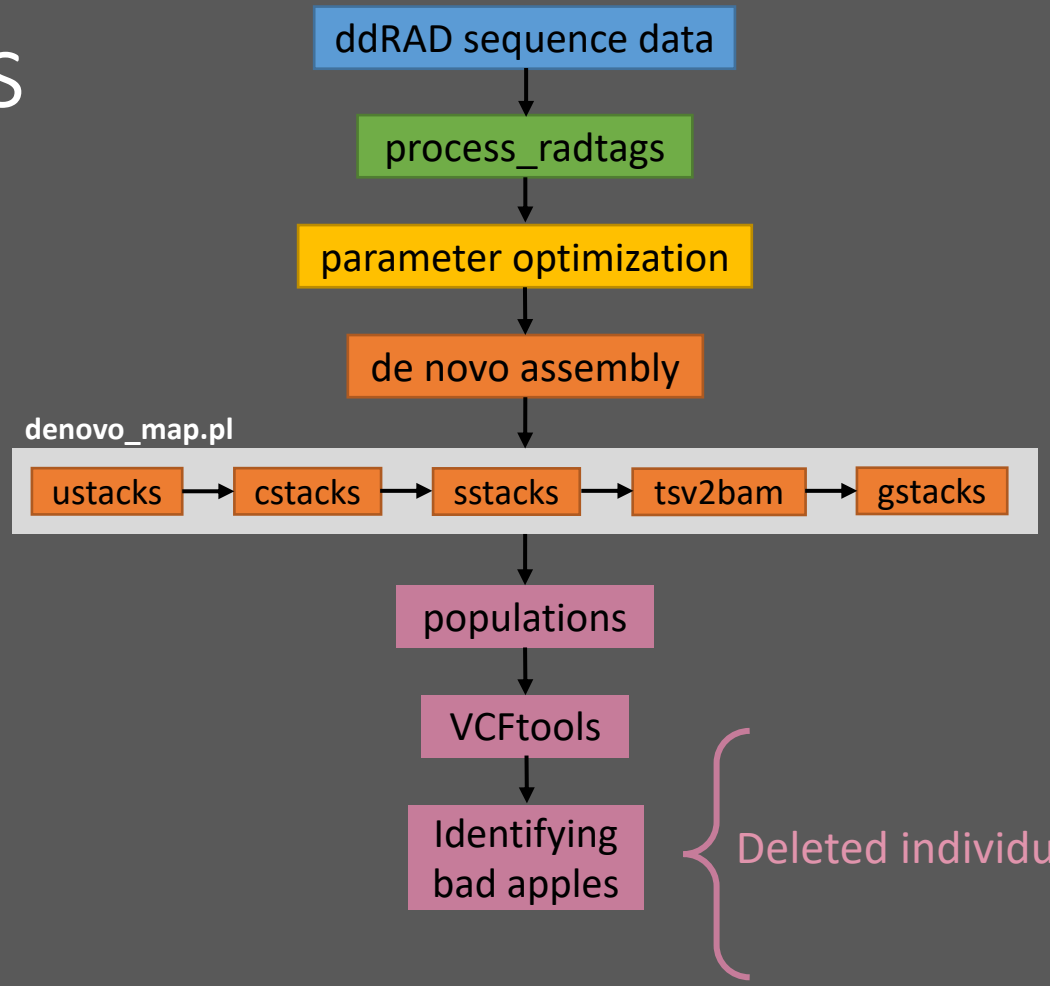
Removing the bad apples: A simple bioinformatic method to improve loci-recovery in de novo RADseq data for non-model organisms

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]

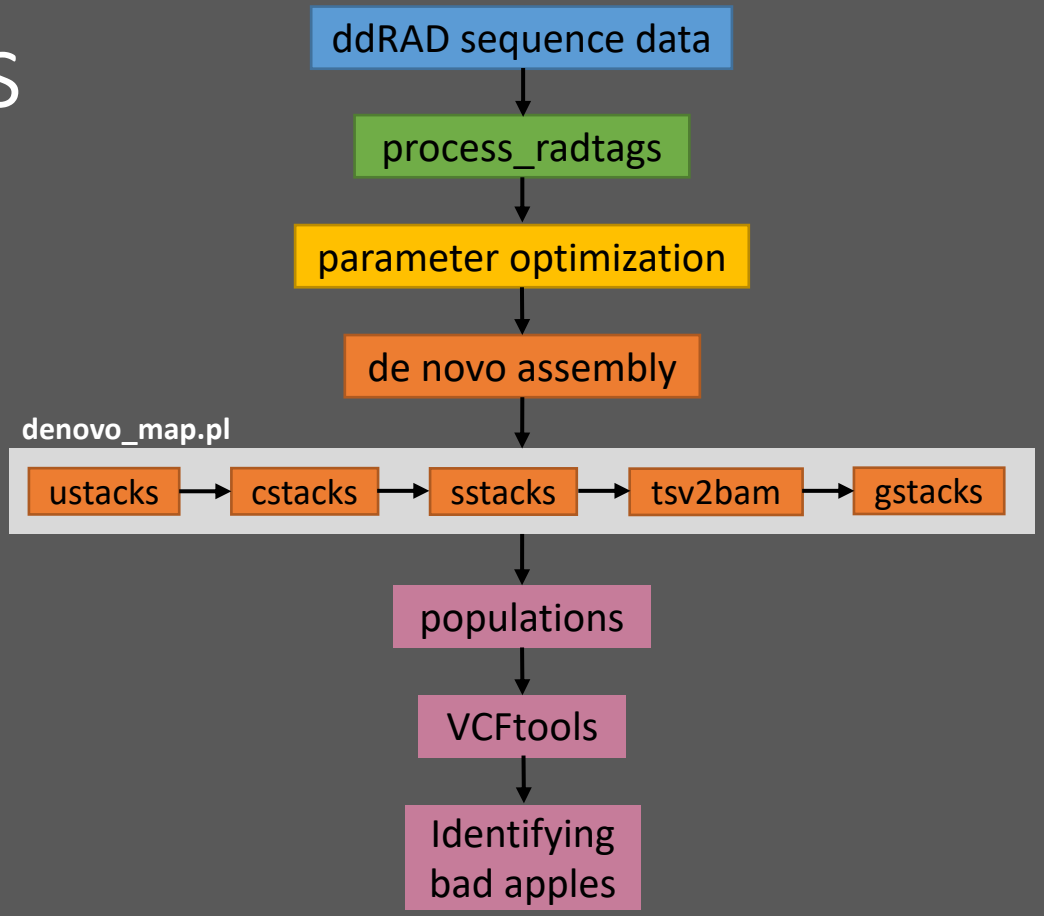
Bioinformatics Workflow



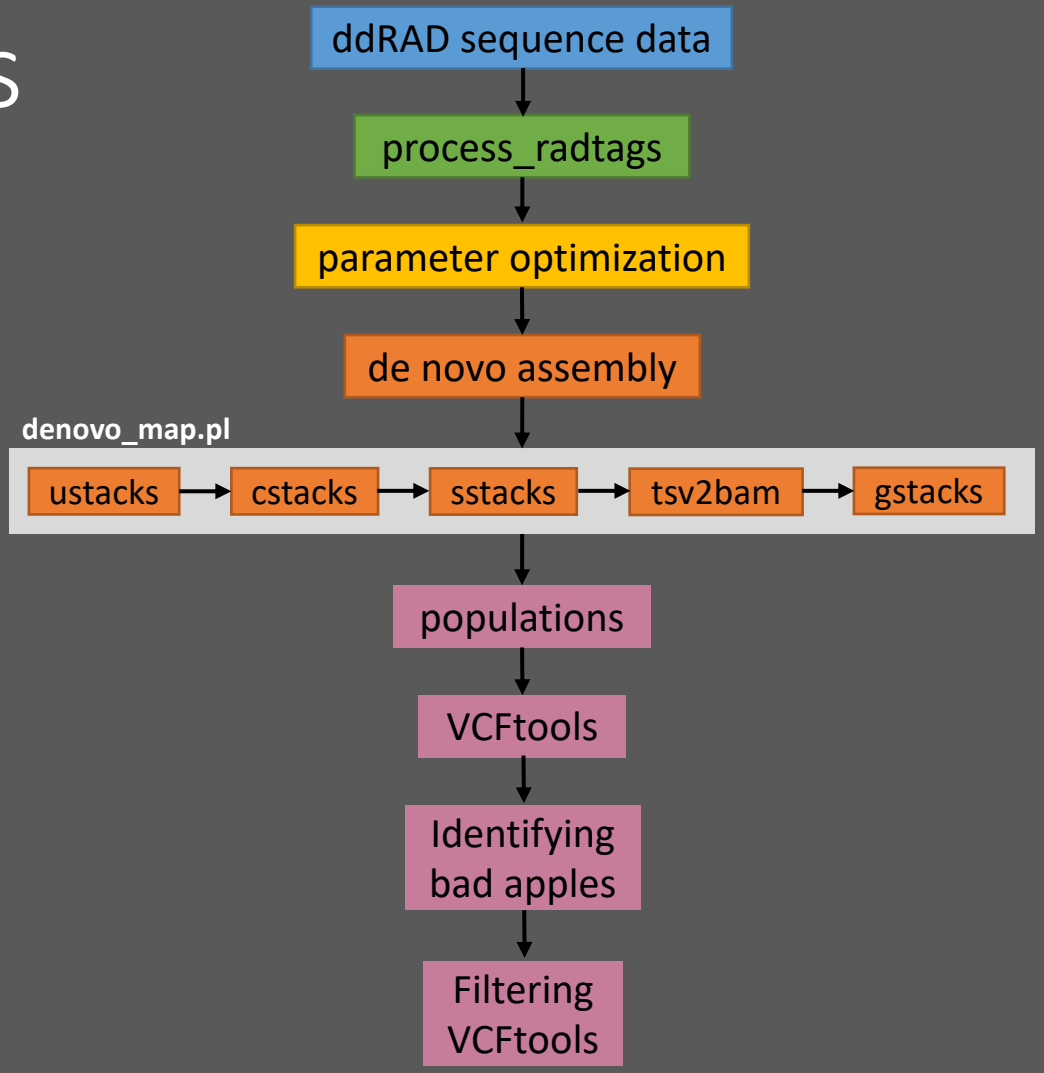
Bioinformatics Workflow



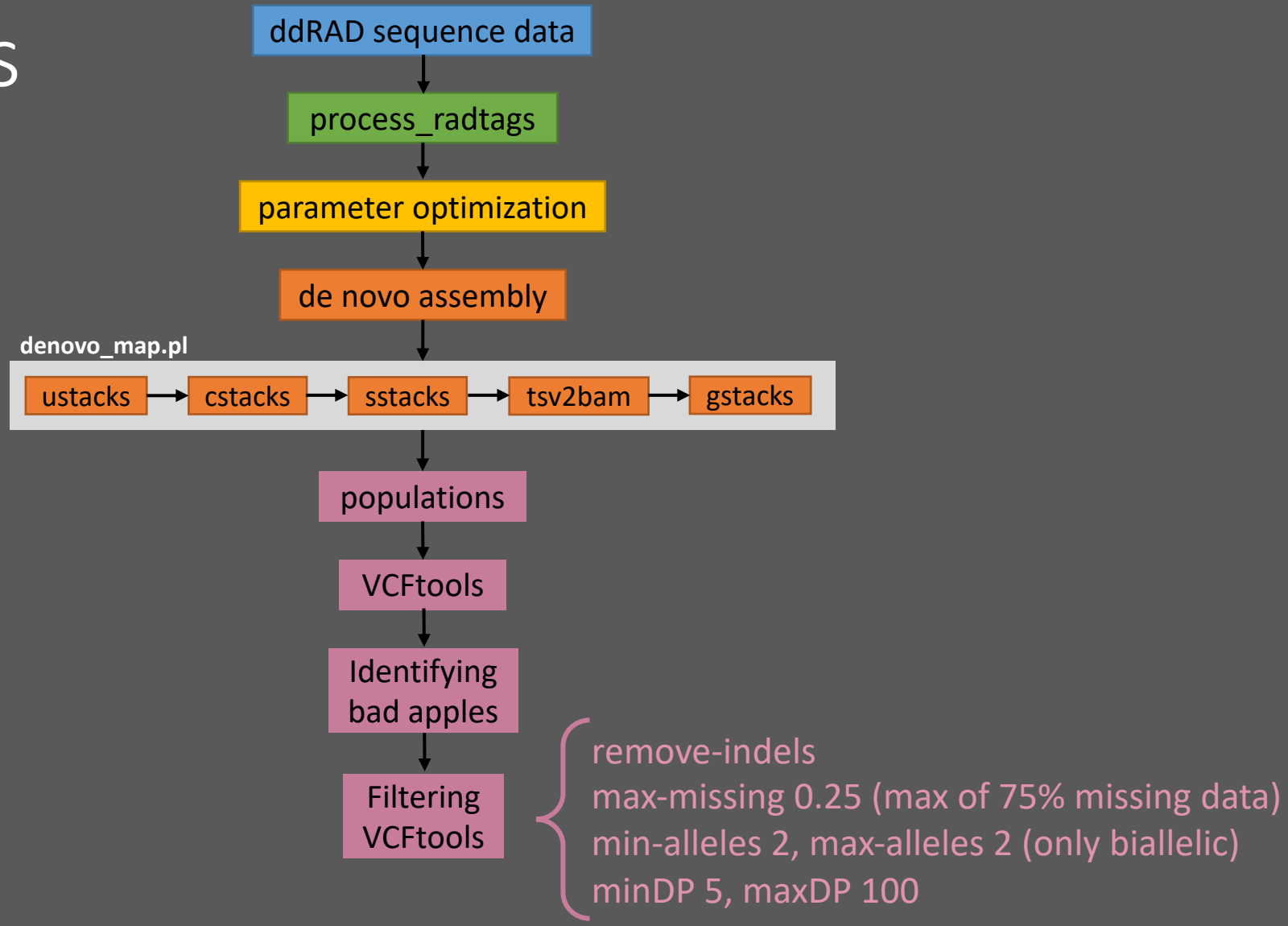
Bioinformatics Workflow



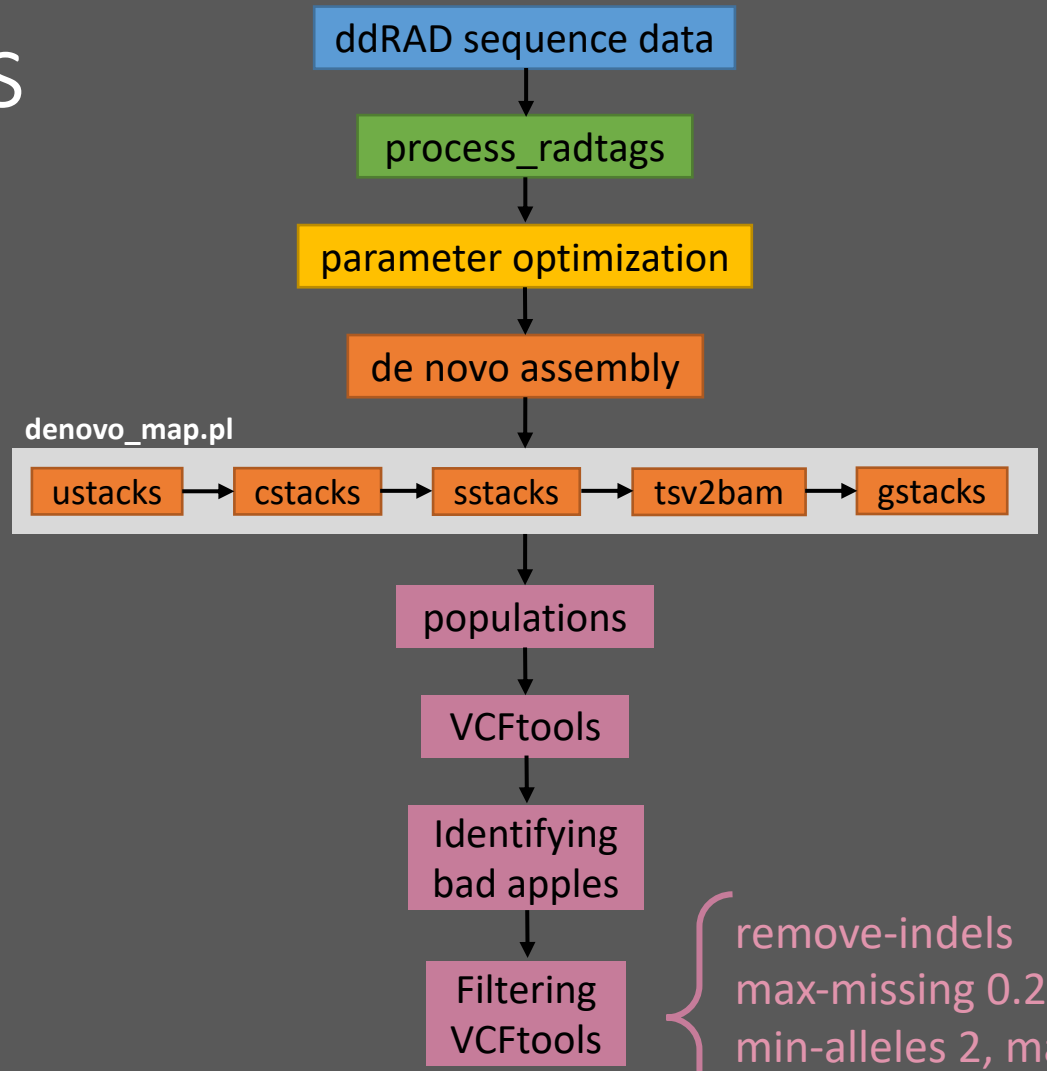
Bioinformatics Workflow



Bioinformatics Workflow



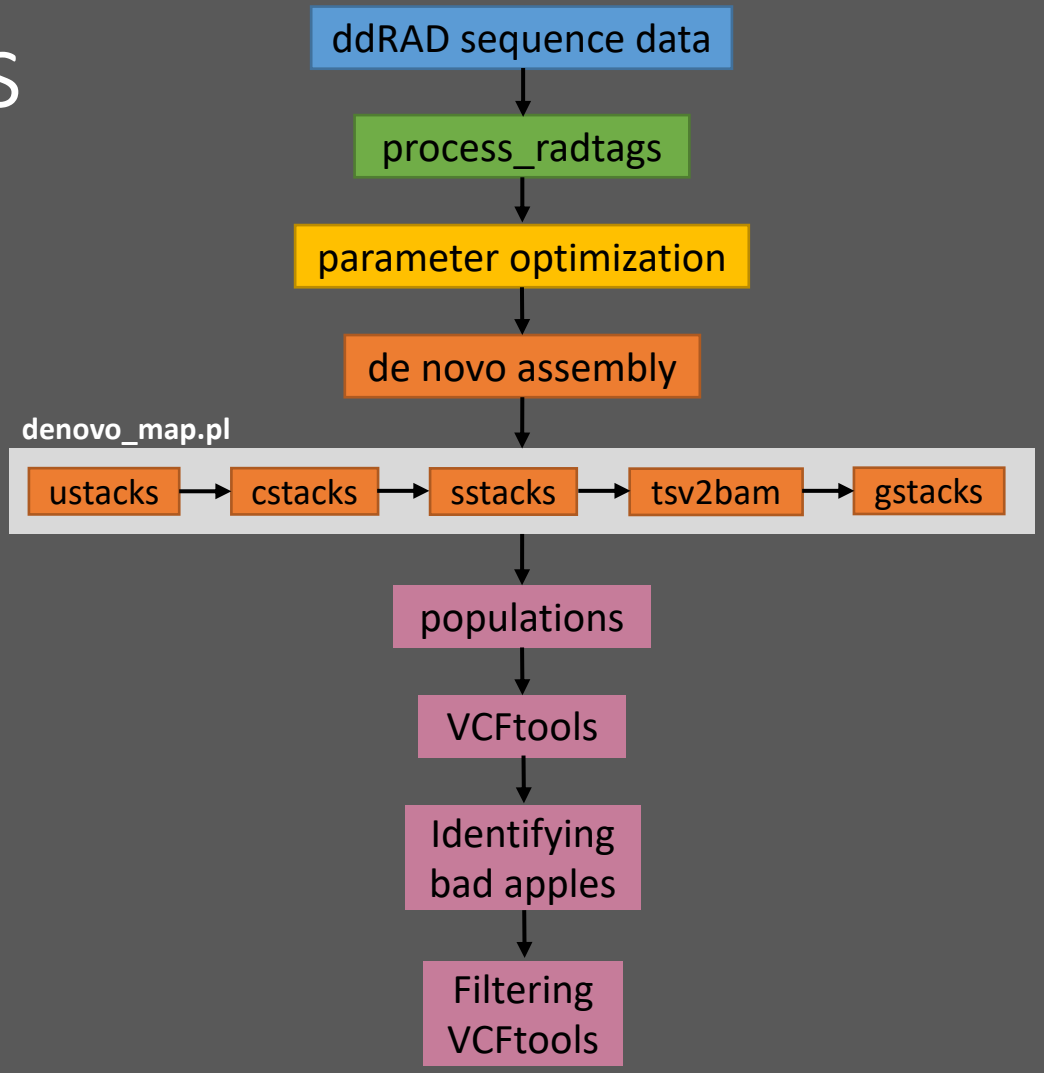
Bioinformatics Workflow



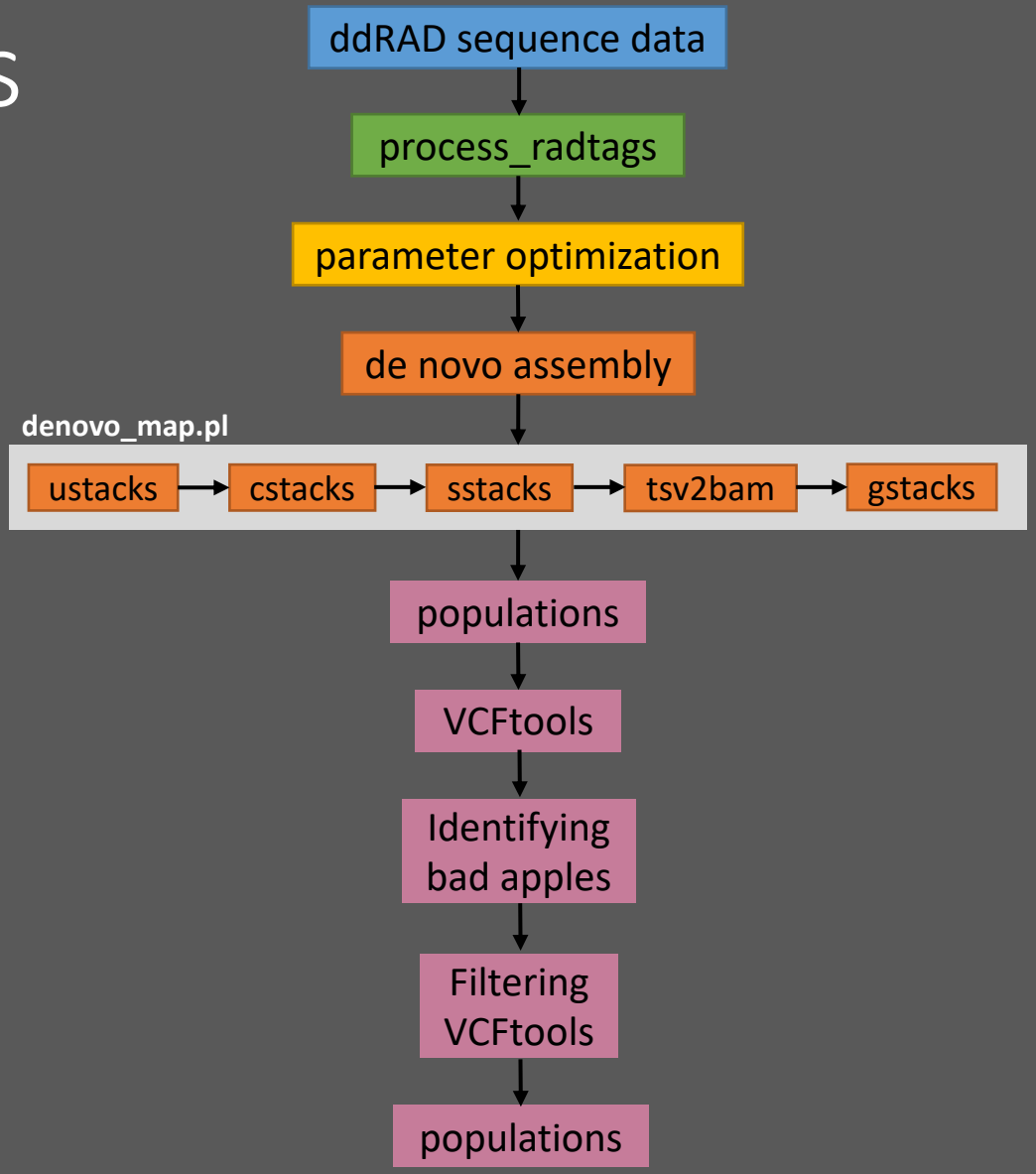
To control for false-positive loci

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

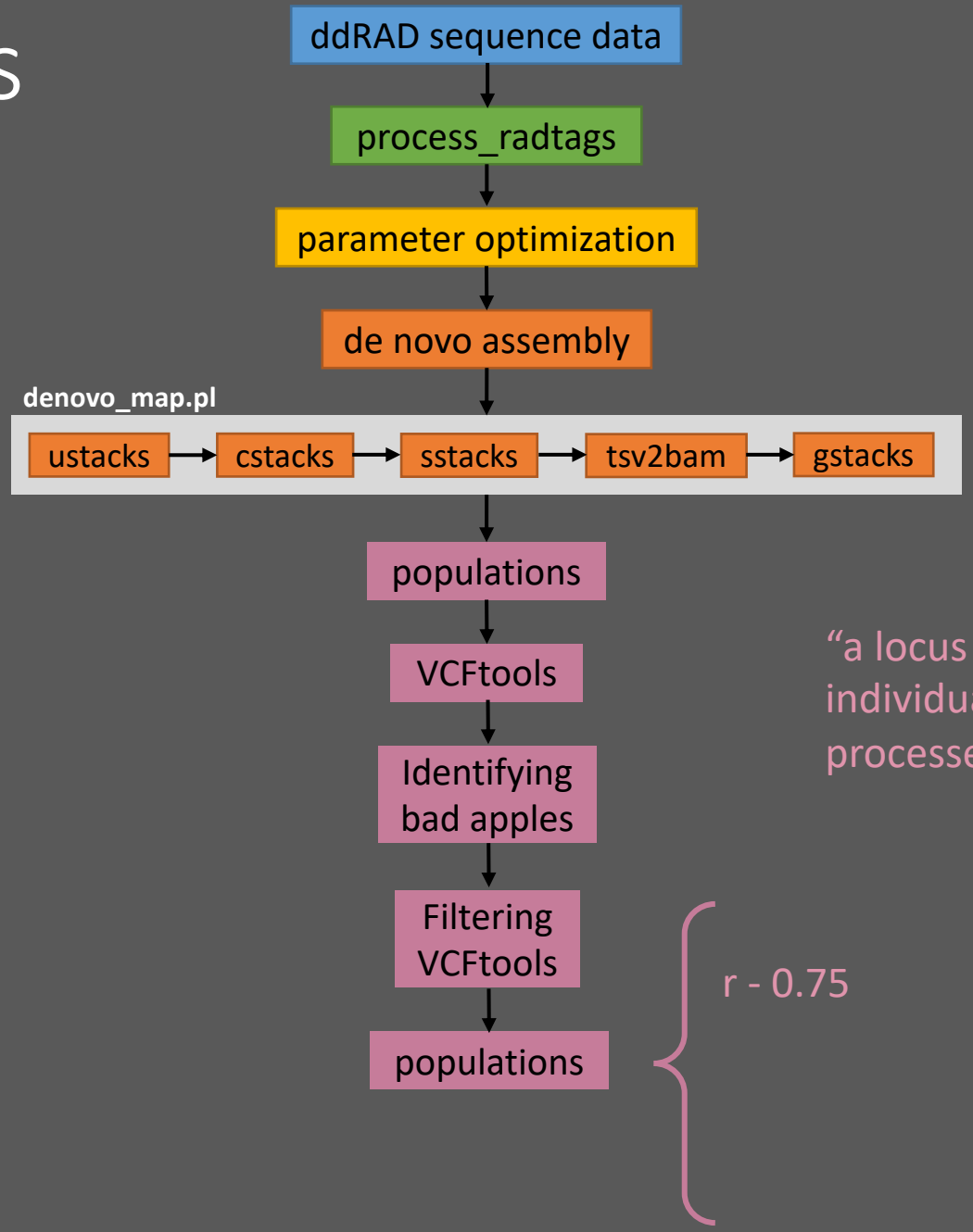
Bioinformatics Workflow



Bioinformatics Workflow

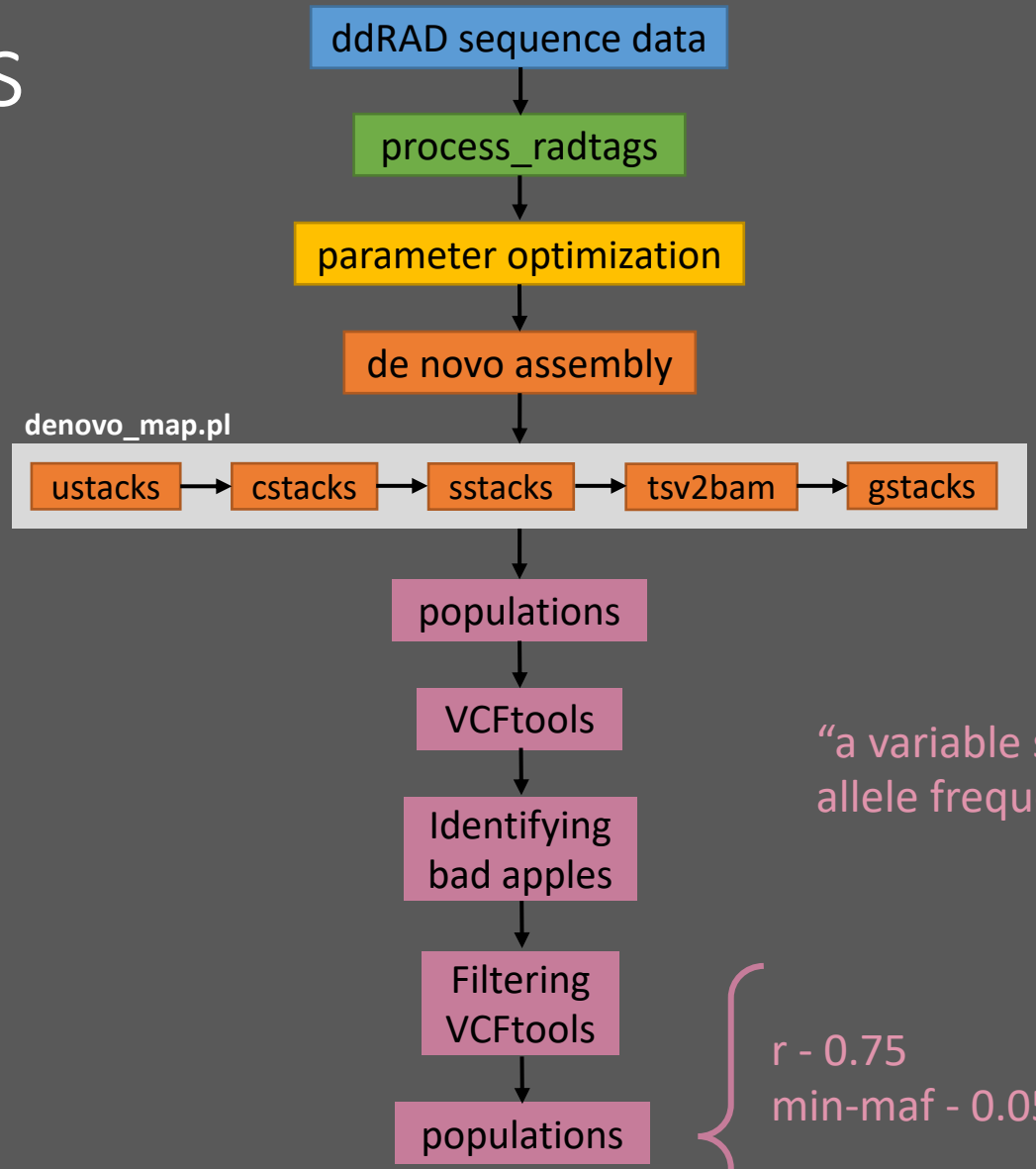


Bioinformatics Workflow



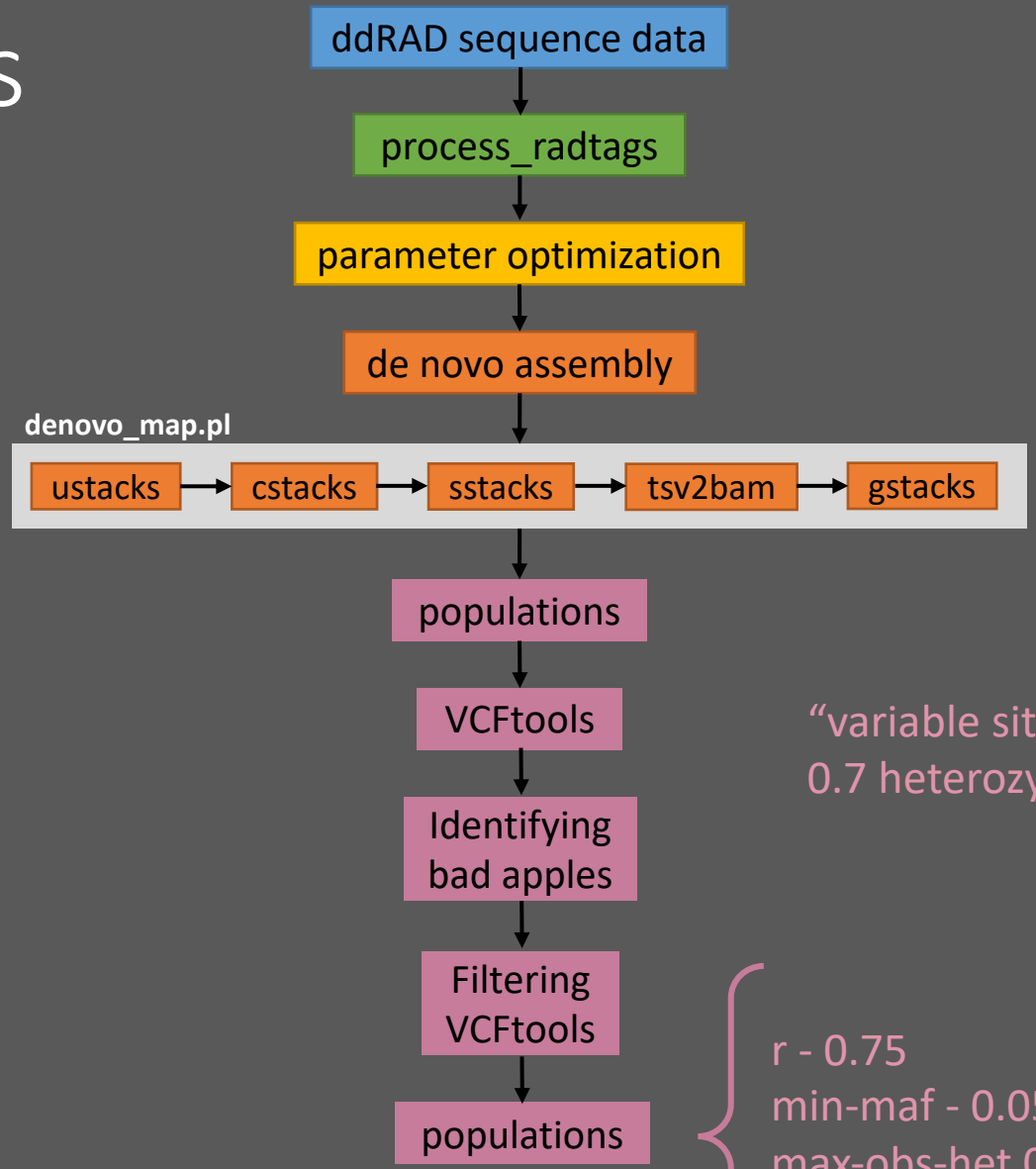
"a locus must be found in 75% of individuals of a single population to be processed"

Bioinformatics Workflow



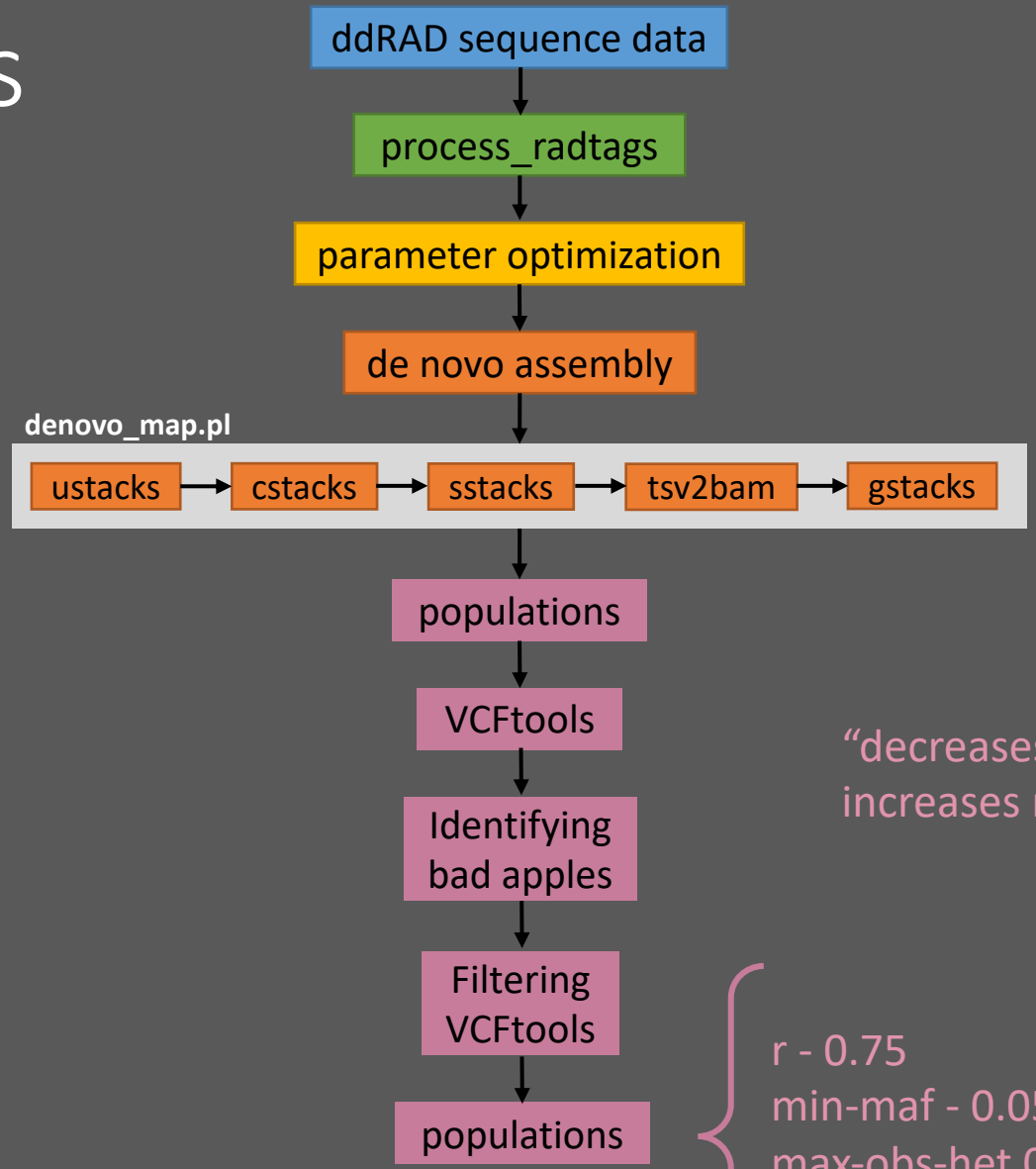
"a variable site must possess a minimum allele frequency of 5% to be included"

Bioinformatics Workflow



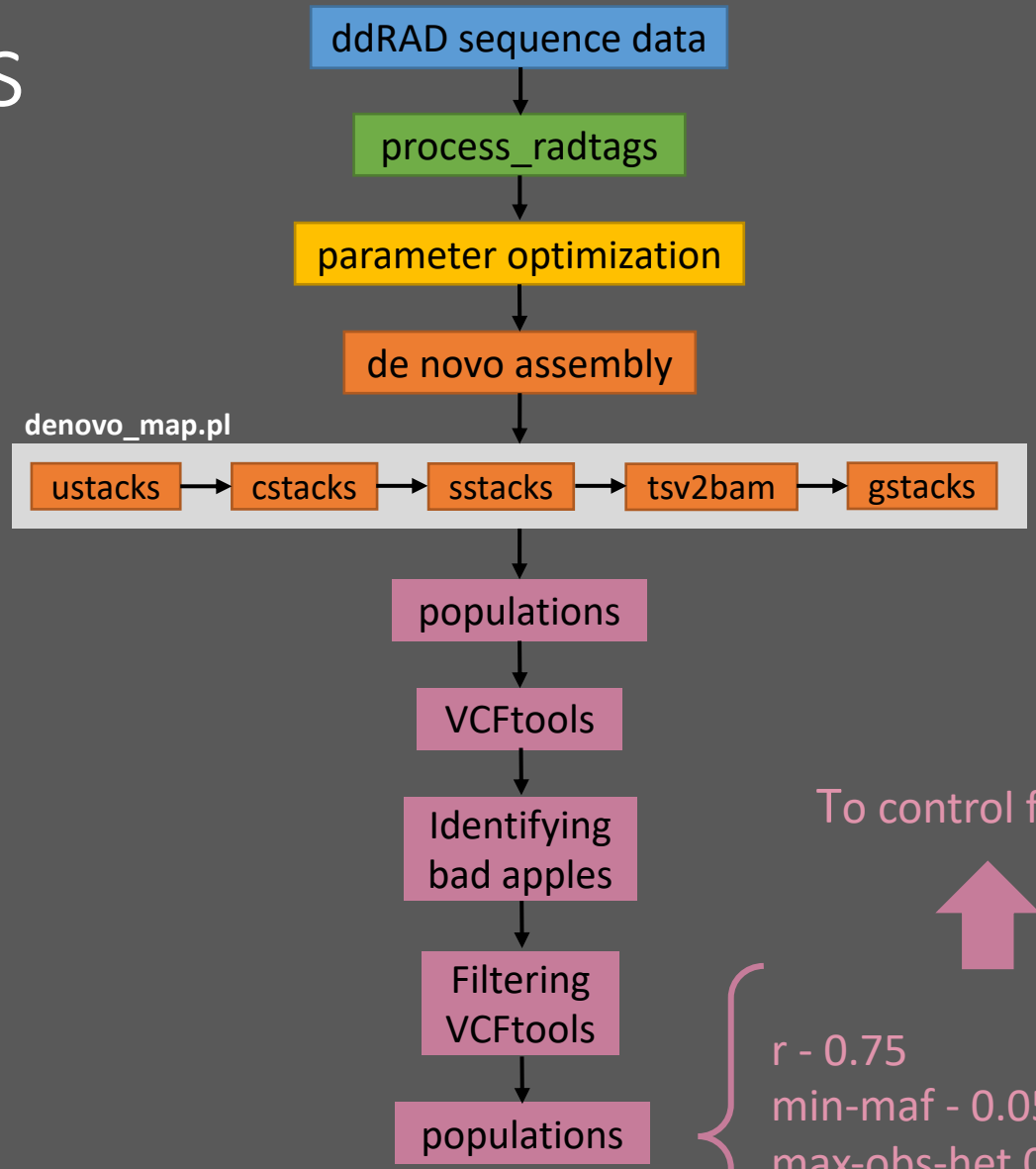
"variable sites with a maximum level of 0.7 heterozygosity will be included"

Bioinformatics Workflow

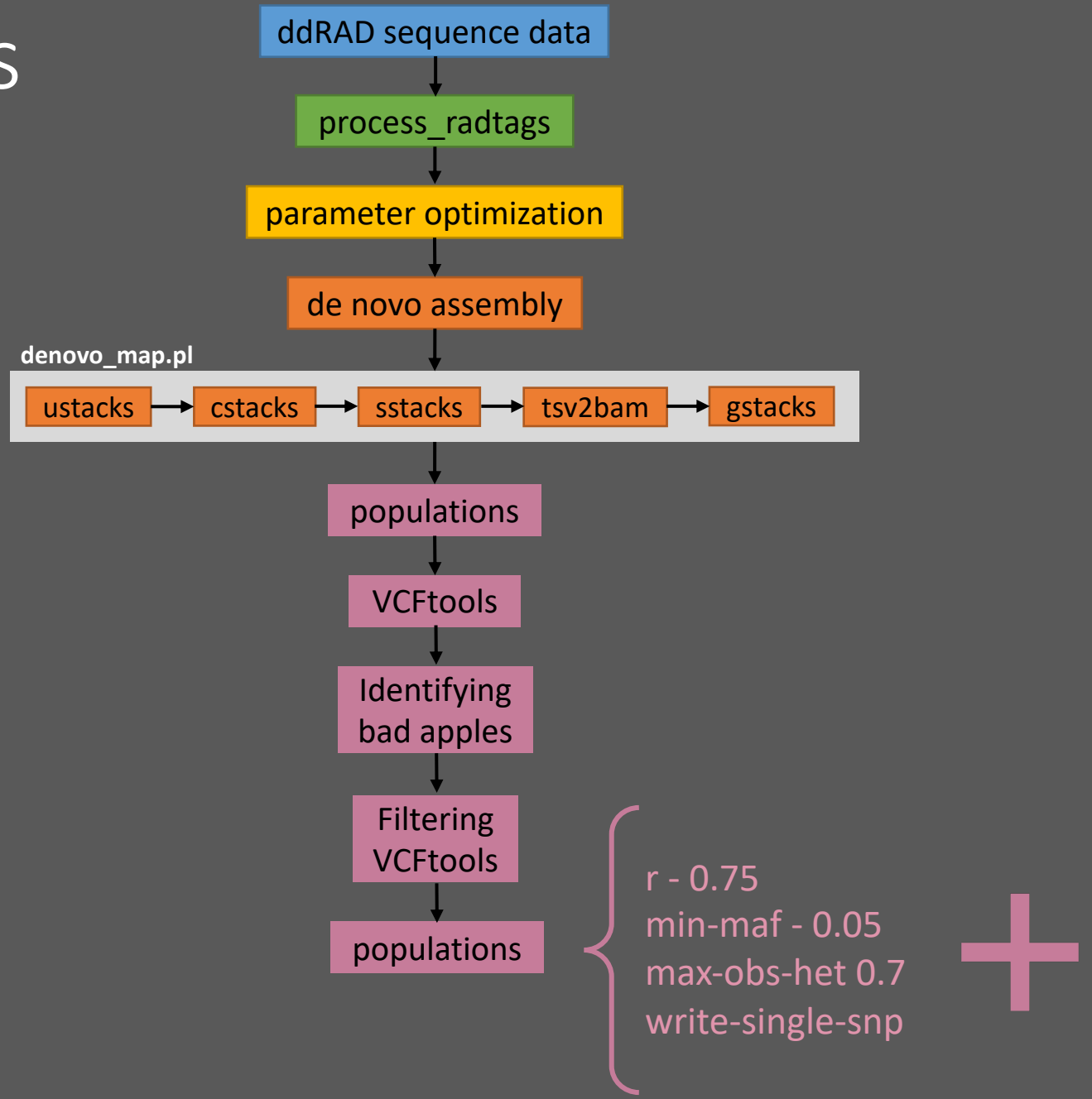


"decreases linkage disequilibrium and increases reproducibility"

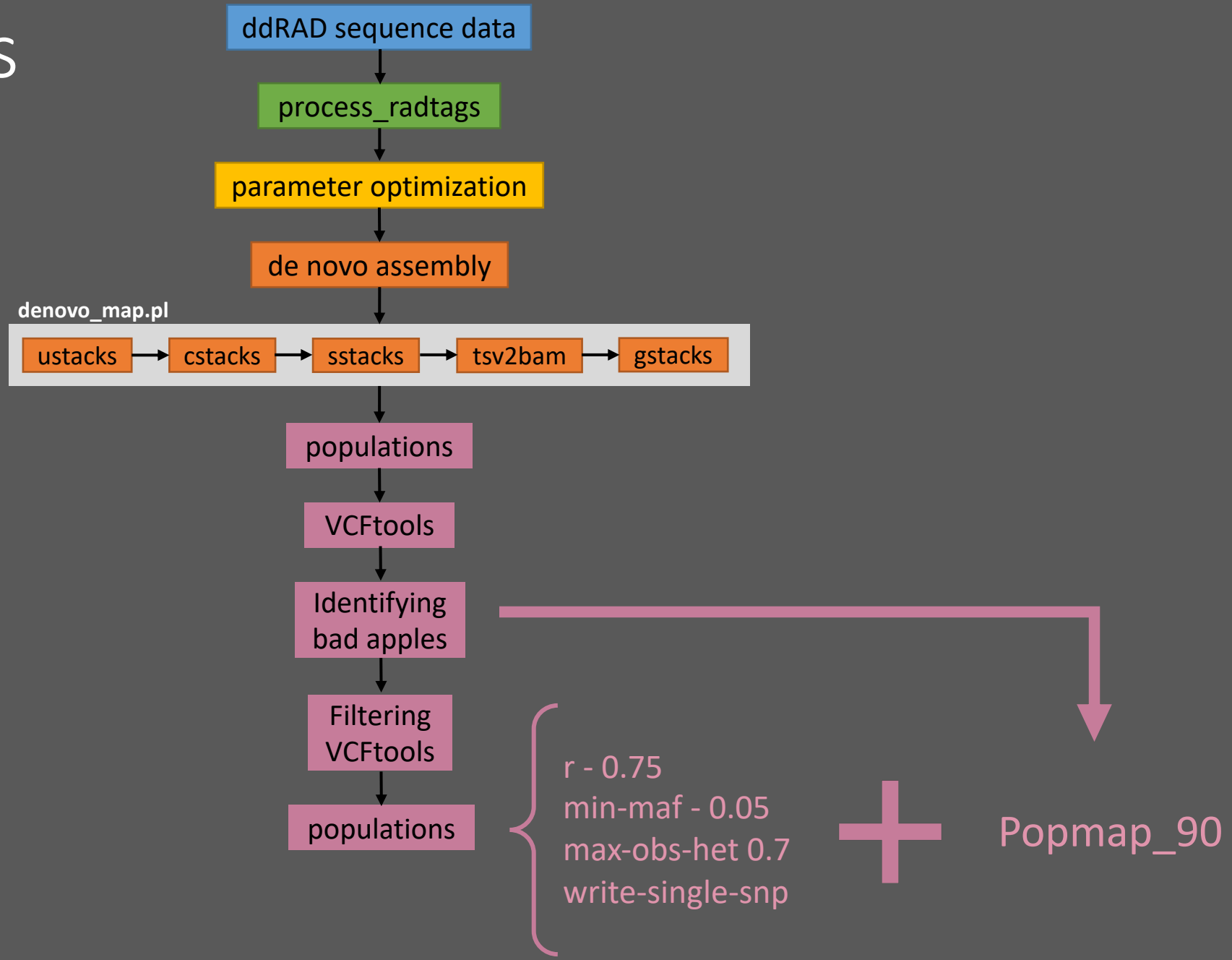
Bioinformatics Workflow



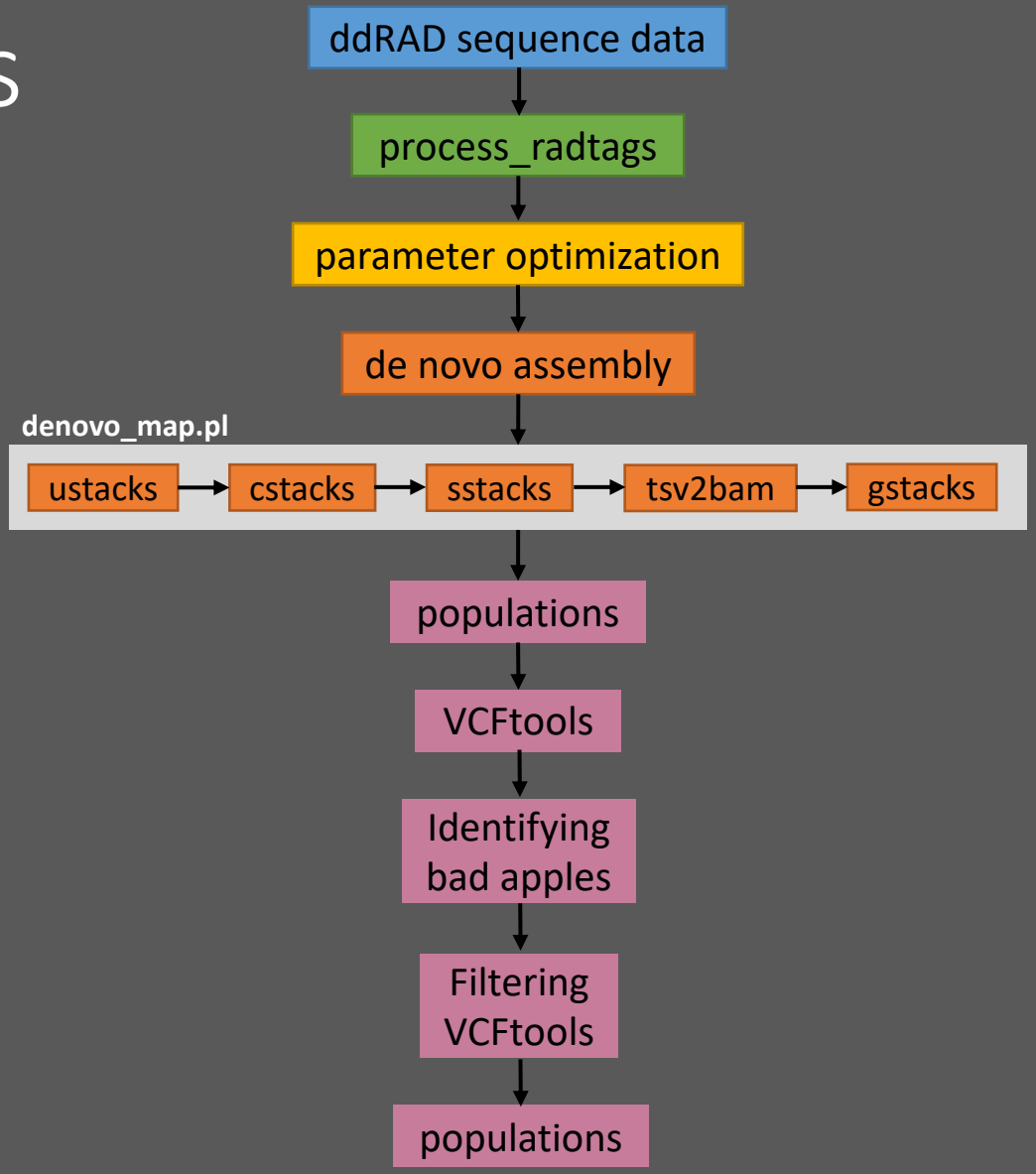
Bioinformatics Workflow



Bioinformatics Workflow



Bioinformatics Workflow



Bioinformatics Workflow