Implementation of nextflow pipelines for an efficient and flexible analysis of paired-end Illumina sequencing data using *Stacks*

Internship 1

October/ November 2019

Katharina Neuhaus

Katharina.Neuhaus@bioinfsys.uni-giessen.de

Systematic Botany

Justus-Liebig-University Giessen

Table of Contents

Quick start

1. Run `conda activate;` you should see `(base)` on the left side of the username in the terminal
2. Direct into the folder where nextflow is located `/data/nextflow`, then execute nextflow with `run`.
3. Navigate to code folder `/data/PE_Stacks_Pipelines/` and choose the script you want to execute
4. Specify the required inputs as **absolute paths** and run the code by pressing enter

Example: *Catananche lutea*

1-clean_reads.nf

```
/data/nextflow run /data/PE_Stacks_Pipelines/1-clean_reads.nf \
--inputDir /data/Israel-Projekt/Catanache_lutea_AdapterClipped/ \
--outputDir /data/Israel-Projekt/Analysis_Catananche_lutea/
```

1-prepare_reference_genome.nf

```
/data/nextflow run /data/PE_Stacks_Pipelines/1-prepare_reference_genome.nf \
--referenceDir /data/Israel-Projekt/genome_assemblies\ lettuce\ cv\ salinas/ \
--outputDir /data/Israel-Projekt/Reference_genomes_indices/lettuce_cv_salinas/
```

2-sort_control_and_map.nf

```
/data/nextflow run /data/PE_Stacks_Pipelines/2-sort_control_and_map.nf \
--workingDir /data/Israel-Projekt/Analysis_Catananche_lutea/ \
--referenceDir /data/Israel-Projekt/Reference_genomes_indices/lettuce_cv_salinas/ \
--dirName lettuce_cv_salinas
```

3-map_creation.nf

```
/data/nextflow run /data/PE_Stacks_Pipelines/3-map_creation.nf \
--workingDir /data/Israel-Projekt/Analysis_Catananche_lutea/ \
--populationMap /data/Israel-Projekt/Populationmap/Populationmap_Catananche.txt \
--dirName lettuce_cv_salinas
```

4-populations.nf

```
/data/nextflow run /data/PE_Stacks_Pipelines/4-populations.nf \
--inputDir /data/Israel-Projekt/Analysis_Catananche_lutea/denovo_map/ \
--outputDir /data/Israel-Projekt/Analysis_Catananche_lutea/denovo_map_populations/ \
--populationMap /data/Israel-Projekt/Populationmap/Populationmap_Catananche.txt
```

For your info: If an **error** occurred, you can resume the nextflow execution after fixing the bug specifying `-resume` behind the previous command. In case you get **funny errors** like the print-out of the usage of *Stack* tools run `conda deactivate` followed by `conda activate`. It is not possible to run s**everal nextflow applications** within different terminals at the same time but this should not be desired anyway regarding the computation power of the cloud.

Introduction

Study background and experimental setup

One of the most prominent problems nowadays are the major changes in biodiversity and increasing extinction rates (Reid 2005; Pitman et al. 2002; Dirzo and Raven 2003; Barbault 1995).Thus it is of major interest to gain a better understanding of the entity of evolutionary processes and the adaptation of ecosystems to changing circumstances (Müller et al. 2019). Invasion of the human into unaffected landscapes and the general change of climate result in truncated patches of land and/or diminished population sizes. In areas like this theory predicts that evolutional processes like genetic drift and the thereby caused increased homozygosity are much more induced when populations are isolated by patch fragmentation. This is especially the case when this fragments cause fundamental barriers to gene flow by pollen exchange and seed dispersal (Müller et al. 2017; Sexton et al. 2009; Fahrig 2003). Further, with regard to the evolutionary context, it was found that fragmented range edge populations promote selection of genotypes which are superior to others regarding adaptation to extreme and/or climatic conditions (Pannell and Fields 2014).

Investigating those effects the experimental setup of this project comprises three species of co-occurring annual plants at the boundary between the Mediterranean and arid climate zone in central Israel with an area of about 45 km x 30 km between Kiryat Gat and Beer Sheva (Müller et al. 2019). This area is characterised by a strong north-south gradient of precipitation ranging from an annual rainfall about 450 mm in the north to approximately 300 mm in the south (Ben-Gai et al. 1994; Goldreich 2003; Svoray et al. 2007; Yaacobi et al. 2007). After the mid-1950s the landscape was reshaped by intensive reclamation and general changes in the land resulting in a mosaic of scattered patches where natural vegetation embedded in agricultural fields still occurs (Giladi et al. 2011; Svoray et al. 2007; Yaacobi et al. 2007). This layout might act as a barriers for seed dispersal and pollen exchange between the populations within the natural islands (Poschlod and WallisDeVries 2002; May et al. 2013).

As study species *Catananche lutea*, *Geropogon hybridus* and *Urospermum picroides* were chosen. All three are theropyhtes where the seed bearing plant dies off after fruiting. During unfavourable seasons the plants survive via seeds. The centre of distribution for the three species is in central and Eastern Mediterranean regions (Feinbrun-Dothan 1978). Dispersal strategies differ between the three species being either monocarpic, amphicarpic or heterocarpic. The fruits on a single plant also may feature developmentally determined differences. Due to those differences it creates the possibility for the plants to use varying dispersal strategies which can result in significant effects on gene flow among habitat patches (Müller et al. 2019; Müller et al. 2017; Gemeinholzer et al. 2012).

The whole study area was divided into three regions north, central and south. All of those cover 70 – 170 natural vegetation patches differing in sizes. For sampling, four patches were chosen within each region hosting the three study species of which 10 samples per patch were taken. A total of 120 samples for each of the study species were immediately dried and stored in *in silica* gel. For analysis, the samples were then sequenced using an Illumina HiSeq 2000 platform in one channel of 180 pooled samples. The resulting raw sequences were demultiplexed by their specific barcodes (Müller et al. 2019).

Why automation is required for sequencing data

The first technique of DNA sequencing was introduced in 1977 by Frederick Sanger and colleagues (Sanger et al. 1977). 52 years later the research looks back on a massive development within this field. With the development of the PCR (polymerase chain reaction) technique and the set up Human Genome Project sequencing needed to become faster and way cheaper: the field of next-generation sequencing was born. Multiple sequencing techniques can be used nowadays supporting different needs and experimental setups. Illumina, Ion Torrent or 454 Life Science sequencing are the platforms created for high-quality short-reads sequencing. On the other hand, less sensitive but long-read sequencing platforms like the MinION system are also available. All of these techniques generate huge amounts of raw data that need to be processed by bioinformatics workflows (Morganti et al. 2019). While the size of generated sequencing data is rapidly increasing, the implementation of computing power is not as fast as required for this amount of data. This assumption of technical development was made by Gordon E. Moore in 1965 where he predicted a doubling every year (Moore 2006a). This prediction was fulfilled within the first years of computational development but emerged after a few years. Thus, Moore's law was adapted to a doubling every two years that can still be observed until today (Moore 2006b). Given these circumstances the inventiveness of bioinformaticians is required for the development of new algorithms and parallelised computing for data analysis in a reasonable time and computational power. Pipeline languages like nextflow or Snakemake ease the way to high-throughput data analysis.

Pipeline implementation

As the experimental setup comprises three species, each consisting of 120 samples, a pipeline executing the single steps of data processing seemed to be useful. In the beginning the main idea was to set up one single pipeline performing all analyses at once. During implementation it became apparent that this approach might be handy but not suitable for this experimental setup. Mainly the mapping brought up the disadvantages of construction of one single pipeline: the lack of a reference genome for the three species.

The lack of reference genomes for the three observed species was known before the implementation of the pipeline started. Nonetheless, as the target species are closely related to *Lactuca sativa* they are most likely mappable against the reference genome of this species. Researches regarding the reference showed that multiple cultivars (cv.) are reported of which some are already sequenced and thus are available for mapping. However, a paper by Verwaaijen et al. 2018 showed significant differences between the genomes of the lettuce cv. Tizian and lettuce cv. Salinas. Only about 77 % of the reads of lettuce cv. Tizian were found in lettuce cv. Salinas with an average read identity of about 98 % (Verwaaijen et al. 2018). This finding proposed the execution of the pipeline with more than one reference genome and compare the results according to mapped reads.

In the end the execution of the data analysis was separated into five pipelines each executing one or multiple steps. The big advantage is that some of the initial pipelines can be applied once and the resulting files can be used by the subsequent pipelines multiple times without executing them again, this way, massively reducing computation time.

Overview

What's it all about?

Data processing performed by the five pipelines (do not get confused, the first two pipelines both start with 1 as they are the two initial pipelines; those are not dependent on each other, thus the same numbering). Running all pipelines, it all starts with adapter clipped compressed sequencing data in `.bz2` format and ends with all possible output files generated by the *populations* tool of *Stacks*. Splitting up the single processes was necessary as it was not possible to execute multiple tools of *Stacks* in a row. In addition, this partition saves computation time in case parts of the data analysis needs to be re-run.

The first two pipelines, both with starting with *1*, are the pre-processing parts of the whole data analysis. The first re-zips the files from `.bz2` to `.gz`, performs quality control with *fastQC* and cleans the data with *process_radtags*, the cleaning tool from *Stacks*. The other one takes a reference genome file in `.fna` format and indexes the file as this is required for later mapping with *bowtie2*. Both of these pipelines must only be performed once, then the index files of the reference and the cleaned data are available for all other pipelines and can be used several times.

The second pipeline, *2*, reorganises the output file, maps the cleaned data (only if a reference file is submitted, otherwise no mapping is executed) and performs quality control on the cleaned reads. Here, the mapping pipeline can be executed without re-doing the quality control but only the mapping by specifying a flag. After execution the file system is all set up for map creation.

In third pipeline, *3*, the maps are created using the pipelines provided by *Stacks*. Here another flag was implemented for the execution of *ref_map* only, which is much faster than the *denovo_map* creation. As denovo it is not depending on a reference genome mapping must only be performed once. Thus it would not be time efficient to re-do it every time the *ref_map* pipeline is executed.

The final pipeline, *4*, performs the *populations* program of *Stacks* in which some of the possible flags are already pre-set. This mainly comprises the output files and the statistics performed on the data. The filtering steps can be entered by the user when the nextflow pipeline is executed. Every filter setting can be saved in a different output file.

Limitations

There are several limitations regarding the pipeline setup. First and foremost, the whole execution does only work with **paired-end reads** which are **adapter clipped** (with the extension `_clipped`) and zipped in **`.bz2` or `.gz` format**. In addition, the parameters for *bowtie2* mapping are fixed within the pipeline and is not accessible for the user while executing the nextflow script. If changes are necessary those must be made within the pipeline itself.

Also it is not possible to execute all five pipelines automatically one after another. All of them need to be initiated by the user individually. Same holds for multiple datasets which must be executed one after the other. This might not be the best possible solution but in regard to the computation power of the cloud it might be best anyways to execute each species on its own.

Tools

Nextflow

Nextflow is a domain specific language (DSL) allowing to combine various tasks and processes within one computational pipeline. Thereby it is portable and supports parallelisation. Continuous checkpoints makes is possible to resume the pipeline when parts are already run which reduces computation time and makes bug fixes or implementation of extensions easier (Di Tommaso et al.).

Bowtie2

Bowtie2 mapping was chosen for this pipeline as it is a memory-efficient tool working with the FM index. Here, the reference is saved in space efficient arrays that allow easy look-ups for sequence substrings within the genome. By this the mapping of reads against the reference genome is much more efficient and less time and space consuming later on. In addition the mapping can be split up on multiple processors for speeding up the mapping. Here, the option for large reference genomes is chosen and the *bowtie2* indexer generates index files in `.bt2l` format. The mapping algorithm works best for reads of a length from 50 up to 100 bp against a long reference genome. Output files are coming in `.sam` format. (Langmead and Salzberg 2012; Langmead et al. 2019).

SAMtools

As *bowtie2* mapping outputs files in `.sam` format it is necessary to write them to `.bam` files as this file format is required for further analyses. Useful for operations on `.sam` and `.bam` files is *SAMtools* with which it is possible to manipulate this format. Options like sorting, merging, indexing and also generation of alignments are part of *SAMtools* (Li et al. 2009; Li 2011). Here it is used to convert the `.sam` files to `.bam` files followed by sorting of the reads. This sorting is required as map creation in the later steps using the tools of *Stacks* relies on pre-sorted reads which eases the assignment of reads to the stacks.

FastQC

FastQC is a tool for sequence quality control as it is important to check for problems and biases before the data analysis. Multiple quality checks are performed next to the basic statistics such as e.g. *per base sequence quality* or *overrepresented k-mers*. All results of in total eleven analyses are gathered and visualised with plots. As *fastQC* provides a high-throughput approach it is possible to execute *fastQC* within an automated pipeline. Especially the difference in quality between raw and cleaned reads is of interest as the settings of read cleaning can be optimised by comparing both outcomes (Andrews et al.).

MultiQC

As *fastQC* provides output files for each individual sample tools like *multiQC* become handy if large data sets are checked for quality. The `.zip` files resulting from *fastQC* are all gathered and summarized in one single `.html` file which breaks down the visualisation in a browser. By hovering over the data, the single read files can still be differentiated and a targeted analysis remains possible (Ewels et al. 2016).

Stacks: Process Radtags

The program *process_radtags* is the cleaning tool of *Stacks*. It examines the reads retrieved by Illumina sequencing. It can perform barcode checks and proof whether RAD cut-sites are intact. Multiplexed data can be demultiplexed. For the actual quality control a sliding window checks the reads for their average quality score and drops reads which are below a 90 % threshold (a raw phred score of 10). The program is able to process paired-end reads, discarding the corresponding read in the other file if quality was not found to be sufficient (Catchen et al.).

Stacks: Denovo Map

Easing up the analysis of the data *Stacks* provided a pipeline executing the single tools in a row with an initial specification of all needed parameters. For this, files in FASTA format must be provided as well as a population map. Starting with *ustacks*, every sample is processed building *de novo* loci for each. This is followed by *cstacks* where a *catalog* of all loci which are matched up by sequence similarity across the population is generated. After *catalog* creation every sample is matched against the *catalog*. Following this the *tsv2bam* program executes a transposition: the data which is still organised by sample gets reorganised by RAD locus. Due to this the downstream analyses are able to perform a meta-population analysis of all data. After reorganisation of the data *gstacks* is performed. Paired-end reads are assembled to contigs and these are overlapped with the single-end locus. This is followed by SNP calling and haplotype phasing at each locus. Finally *populations* is executed that runs a population-level summary statistics and exports data in various formats for further analyses (Catchen et al.).

Stacks: Reference Map

For map creation using a reference genome another pipeline was provided by *Stacks*. Here, the input files are sorted and mapped reads in `.bam` format as well as a population map. The pipeline first executes *gstacks* on each sample specified which assembles the loci according to their alignment position which is provided with each read. For each sample SNPs are called. After this *populations* is executed which runs a population-level summary statistics (Catchen et al.).

Stacks: Populations

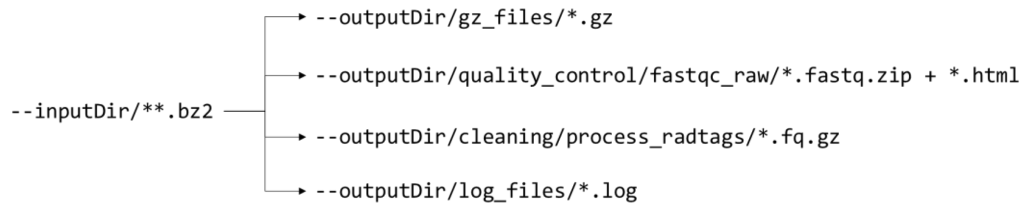
After the pipelines are executed, *populations* can be re-run as often as desired. The program analyses a population of individual samples and computes a variety of genetics statistics such as expected/ observed heterozygosity, π and F_{IS} at each nucleotide position. For the calculation for F_{ST} populations are compared pairwise. Further, haplotype-based population genetic statistics such as haplotype diversity can be applied. In addition *populations* comes along with multiple filters which can narrow the resulting data. Also *whitelists* and *blacklists* can be provided if specific loci should or should not be included into the analysis. All results can be exported in various file formats (Catchen et al.).

Pipelines

1-clean_reads.nf

The initial pipeline converts the reads from `.bz2` to `.gz` (here, the reads can be located in sub-directories of the `--inputDir`), performs quality control by *fastQC* and cleans reads using the *process_radtags* tool of *Stacks*. All output files are saved in their respective output file in the `--outputDir`.

File structure


--inputDir/**.bz2
--outputDir/gz_files/*.gz
--outputDir/quality_control/fastqc_raw/*.fastq.zip + *.html
--outputDir/cleaning/process_radtags/*.fq.gz
--outputDir/log_files/*.log

Input parameters and description

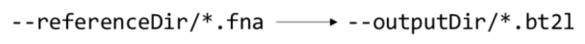
Flag	Description	Format	Type
`--inputDir`	Path to directory where `.bz2` input files are located	abs. path	mandatory
`--outputDir`	Path to output directory; file will be created	abs. path	mandatory
`--t`	truncate final read length to this value; default = `100`	integer	optional
`--e1`	specify restriction enzyme; default = `mslI`	name	optional
`--e2`	specify 2nd restriction enzyme in case of double digest	name	optional
`--gz`	specify if input files are in `.gz` format not `.bz2`	flag	optional
`--help`	Flag that can be specified to view pipeline options	flag	optional

* abs. path = absolute path

1-prepare_reference_genome.nf

This pipeline creates the index files within approximately 30 minutes by specifying the .fna reference genome file. The index files are all located in the specified `--outputDir`.

File structure



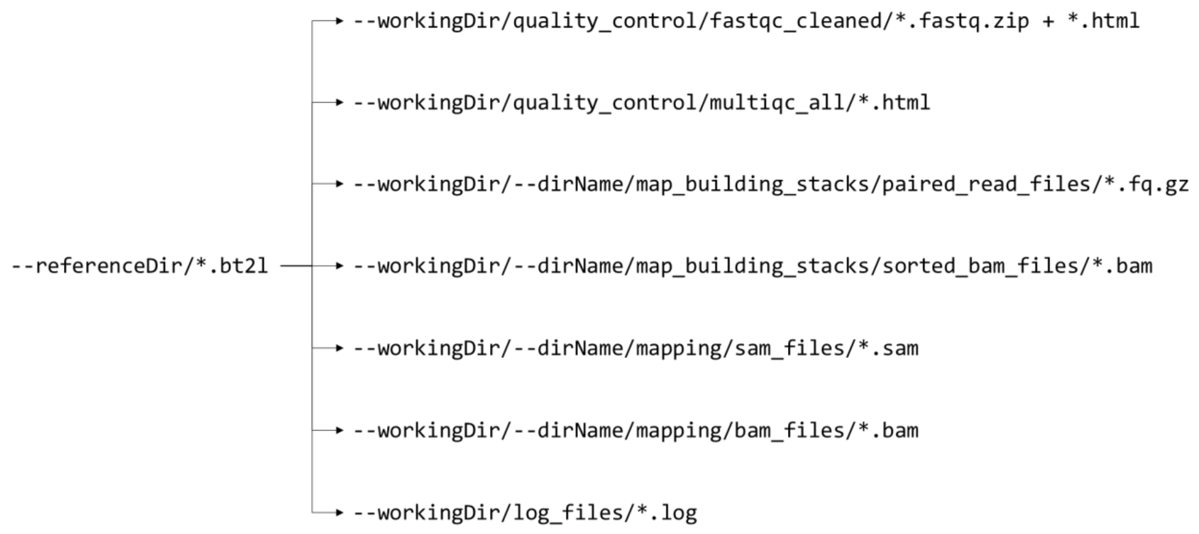
Input parameters and description

Flag	Description	Format	Type
`--referenceDir`	Path to directory with reference file in `.fna` format	abs. path	mandatory
`--outputDir`	Path to output directory; file will be created	abs. path	mandatory
`--help`	Flag that can be specified to view pipeline options	flag	optional

2-sort_control_and_map.nf

Within this pipeline, the cleaned reads are controlled for quality using *fastQC* again. The clipped and cleaned *fastQC* files are then summarized using *multiQC*. This quality control is optional: if the flag `--mappingOnly` is specified all those steps are not performed. The cleaned reads are also moved to another directory and are renamed to the format required for the execution of *denovo_map*. In addition, if a `--referenceDir` is submitted, mapping is performed using *bowtie2*. Here, the output files in `.sam` and `.bam` formats are kept but the `.bam` files are also copied and renamed in another directory as required for the *ref_map* creation.

File structure


--referenceDir/*.bt2l
--workingDir/quality_control/fastqc_cleaned/*.fastq.zip + *.html
--workingDir/quality_control/multiqc_all/*.html
--workingDir/--dirName/map_building_stacks/paired_read_files/*.fq.gz
--workingDir/--dirName/map_building_stacks/sorted_bam_files/*.bam
--workingDir/--dirName/mapping/sam_files/*.sam
--workingDir/--dirName/mapping/bam_files/*.bam
--workingDir/log_files/*.log

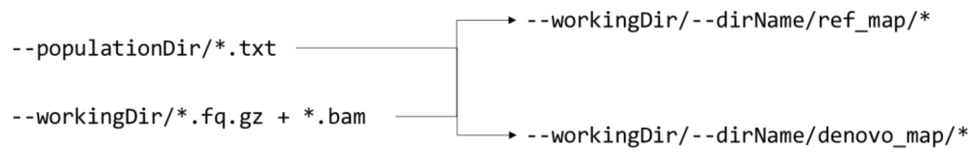
Input parameters and description

Flag	Description	Format	Type
`--workingDir`	Path to directory where the prepared reads of the preparation pipeline are located => former `--outputDir` of `1-clean_reads.nf`	abs. path	mandatory
`--referenceDir`	Path to directory where the index files of the prepared reference genome are located	abs. path	optional
`--dirName`	Name of the file generated within the `--workingDir` where the mapping files are saved to => preferably named after the reference genome used	name	mandatory
`--mappingOnly`	Flag that can be specified when only the mapping should be performed; if not submitted, quality control is also executed	flag	optional
`--help`	Flag that can be specified to view pipeline options	flag	optional

3-map_creation.nf

Here the map creation pipelines of *Stacks* are executed with the prepared and renamed reads from the previous pipeline. By specifying the current `--workingDir`, the pipeline takes the input files for both pipelines automatically and performs the map creation pipelines for denovo and for the reference map. `--dirName` must be specified as it gives the location of the files for the map creation (not an absolute path but only the name of the directory within the given `--workingDir`).

File structure


--populationDir/*.txt
--workingDir/*.fq.gz + *.bam
--workingDir/--dirName/ref_map/*
--workingDir/--dirName/denovo_map/*

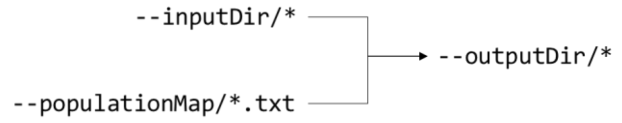
Input parameters and description

Flag	Description	Format	Type
`--workingDir`	Path to directory where the prepared reads of the preparation pipeline are located => former `--outputDir` of `1-clean_reads.nf`	abs. path	mandatory
`--populationMap`	Path to population map file	abs. path	mandatory
`--dirName`	Name of the file as specified in `2-sort_control_and_map.nf`	name	mandatory
`--refMapOnly`	Flag that can be specified when only `ref_map.pl` should be performed; saves massively computation time if `denovo_map.pl` was already executed	flag	optional
`--help`	Flag that can be specified to view pipeline options	flag	optional

4-populations.nf

The final pipeline executes the *Stacks* program *populations*. Some of the flags are already pre-set within the pipeline such as the output file formats (all possible), the f-statistics for SNP and haplotypes and the calculation of the divergence from Hardy-Weinberg-Equilibrium for each locus. All data filtering steps can be specified as the user desires.

File structure



Input parameters and description

Flag	Description	Format	Type
`--inputDir`	Path to directory where the *Stacks* data are located (either *denovo_map* or *ref_map* data)	abs. path	mandatory
`--outputDir`	Path to directory where output files are saved to	abs. path	mandatory
`--populationMap`	Path to the population map file	abs. path	mandatory
`--ref`	Flag to specify if data was mapped against reference genome (if ordered output is desired)	flag	optional
`--help`	Flag that can be specified to view pipeline options	flag	optional

Filter options

`--p`	Minimum number of populations a locus must be present in to process a locus	integer	optional
`--r`	Minimum percentage of individuals in a population required to process a locus for that population	float	optional
`--R`	Minimum percentage of individuals across populations required to process a locus	float	optional
`--H`	Apply filters above haplotype wise (unshared SNPs will be pruned to reduce haplotype-wise missing data)	flag	optional
`--minMaf`	Specify minimum minor allele frequency required to process a nucleotide site at a locus (0 < minMaf < 0.5)	float	optional
`--minMac`	Specify minimum minor allele count required to process a SNP	integer	optional
`--maxObsHet`	Specify a maximum observed heterozygosity required to process a nucleotide site at a locus	float	optional
`--writeSingleSnp`	Restrict data analysis to only the first SNP per locus	flag	optional
`--writeRandomSnp`	Restrict data analysis to one random SNP per locus	flag	optional
`--blacklist`	Path to file containing Blacklisted markers to be excluded from the export	abs. path	optional
`--whitelist`	Path to file containing Whitelisted markers to include in the export	abs. path	optional

Publication bibliography

Andrews, Simon; Krieger, Felix; Segonds-Pichon, Anne; Biggins, Laura; Krueger, Christel; Wingett, Steven; Montgomery, Jo: FastQC Manual. Available online at https://dnacore.missouri.edu/PDF/FastQC_Manual.pdf, checked on 11/7/2019.

Barbault, Robert (1995): Biodiversity dynamics: from population and community ecology approaches to a landscape ecology point of view. In *Landscape and Urban Planning* 31 (1-3), pp. 89–98. DOI: 10.1016/0169-2046(94)01038-A.

Ben-Gai, T.; Bitan, A.; Manes, A.; Alpert, P. (1994): Long-term changes in annual rainfall patterns in southern Israel. In *Theor Appl Climatol* 49 (2), pp. 59–67. DOI: 10.1007/BF00868190.

Catchen, Julian; Rochette, Nicolas; Cresko, William A.; Hohenlohe, Paul A.; Amores, Angel; Bassham, Susan; Postlethwait, John: Stacks Manual. Available online at http://catchenlab.life.illinois.edu/stacks/manual/, checked on 11/7/2019.

Di Tommaso, Paulo; Floden, Evan; Garriga, Edgar; Notredame, Cedric: nexflow. Data-driven computational pipelines. Available online at https://www.nextflow.io/index.html#Features, checked on 11/14/2019.

Dirzo, Rodolfo; Raven, Peter H. (2003): Global State of Biodiversity and Loss. In *Annu. Rev. Environ. Resour.* 28 (1), pp. 137–167. DOI: 10.1146/annurev.energy.28.050302.105532.

Ewels, Philip; Magnusson, Måns; Lundin, Sverker; Käller, Max (2016): MultiQC: summarize analysis results for multiple tools and samples in a single report. In *Bioinformatics (Oxford, England)* 32 (19), pp. 3047–3048. DOI: 10.1093/bioinformatics/btw354.

Fahrig, Lenore (2003): Effects of Habitat Fragmentation on Biodiversity. In *Annu. Rev. Ecol. Evol. Syst.* 34 (1), pp. 487–515. DOI: 10.1146/annurev.ecolsys.34.011802.132419.

Feinbrun-Dothan, Naomi (1978): Flora Palaestina. Jerusalem: The Israel Acad. of Sciences and Humanities (Publications of the Israel Academy of Sciences and Humanities / Section of Sciences).

Gemeinholzer, B.; May, F.; Ristow, M.; Batsch, C.; Lauterbach, D. (2012): Strong genetic differentiation on a fragmentation gradient among populations of the heterocarpic annual Catananche lutea L. (Asteraceae). In *Plant Syst Evol* 298 (8), pp. 1585–1596. DOI: 10.1007/s00606-012-0661-1.

Giladi, Itamar; Ziv, Yaron; May, Felix; Jeltsch, Florian (2011): Scale-dependent determinants of plant species richness in a semi-arid fragmented agro-ecosystem. In *Journal of Vegetation Science* 22 (6), pp. 983–996. DOI: 10.1111/j.1654-1103.2011.01309.x.

Goldreich, Yair (2003): The Climate of Israel. Boston, MA: Springer US.

Langmead, Ben; Salzberg, Steven L. (2012): Fast gapped-read alignment with Bowtie 2. In *Nature methods* 9 (4), pp. 357–359. DOI: 10.1038/nmeth.1923.

Langmead, Ben; Wilks, Christopher; Antonescu, Valentin; Charles, Rone (2019): Scaling read aligners to hundreds of threads on general-purpose processors. In *Bioinformatics (Oxford, England)* 35 (3), pp. 421–432. DOI: 10.1093/bioinformatics/bty648.

Li, Heng (2011): A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. In *Bioinformatics (Oxford, England)* 27 (21), pp. 2987–2993. DOI: 10.1093/bioinformatics/btr509.

Li, Heng; Handsaker, Bob; Wysoker, Alec; Fennell, Tim; Ruan, Jue; Homer, Nils et al. (2009): The Sequence Alignment/Map format and SAMtools. In *Bioinformatics (Oxford, England)* 25 (16), pp. 2078–2079. DOI: 10.1093/bioinformatics/btp352.

May, Felix; Giladi, Itamar; Ristow, Michael; Ziv, Yaron; Jeltsch, Florian (2013): Metacommunity, mainland-island system or island communities? Assessing the regional dynamics of plant communities in a fragmented landscape. In *Ecography* 36 (7), pp. 842–853. DOI: 10.1111/j.1600-0587.2012.07793.x.

Moore, Gordon E. (2006a): Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff. In *IEEE Solid-State Circuits Soc. Newsl.* 11 (3), pp. 33–35. DOI: 10.1109/N-SSC.2006.4785860.

Moore, Gordon E. (2006b): Progress in digital integrated electronics [Technical literaiture, Copyright 1975 IEEE. Reprinted, with permission. Technical Digest. International Electron Devices Meeting, IEEE, 1975, pp. 11-13.]. In *IEEE Solid-State Circuits Soc. Newsl.* 11 (3), pp. 36–37. DOI: 10.1109/N-SSC.2006.4804410.

Morganti, Stefania; Tarantino, Paolo; Ferraro, Emanuela; D'Amico, Paolo; Viale, Giulia; Trapani, Dario et al. (2019): Complexity of genome sequencing and reporting: Next generation sequencing (NGS) technologies and implementation of precision medicine in real life. In *Critical reviews in oncology/hematology* 133, pp. 171–182. DOI: 10.1016/j.critrevonc.2018.11.008.

Müller, Christina M.; Linke, Burkhard; Strickert, Marc; Ziv, Yaron; Giladi, Itamar; Gemeinholzer, Birgit (2019): Comparative Genomic Analysis of Three Co-occurring Annual Asteraceae Along Micro-geographic Fragmentation Scenarios. In *Perspectives in Plant Ecology, Evolution and Systematics*, p. 125486. DOI: 10.1016/j.ppees.2019.125486.

Müller, Christina M.; Schulz, Benjamin; Lauterbach, Daniel; Ristow, Michael; Wissemann, Volker; Gemeinholzer, Birgit (2017): Geropogon hybridus (L.) Sch.Bip. (Asteraceae) exhibits micro-geographic genetic divergence at ecological range limits along a steep precipitation gradient. In *Plant Syst Evol* 303 (1), pp. 91–104. DOI: 10.1007/s00606-016-1354-y.

Pannell, John R.; Fields, Peter D. (2014): Evolution in subdivided plant populations: concepts, recent advances and future directions. In *The New phytologist* 201 (2), pp. 417–432. DOI: 10.1111/nph.12495.

Pitman, Nigel C. A.; Jorgensen, Peter M.; Williams, Robert S. R.; Leon-Yanez, Susana; Valencia, Renato (2002): Extinction-Rate Estimates for a Modern Neotropical Flora. In *Conservation Biology* 16 (5), pp. 1427–1431. DOI: 10.1046/j.1523-1739.2002.01259.x.

Poschlod, Peter; WallisDeVries, Michiel F. (2002): The historical and socioeconomic perspective of calcareous grasslands—lessons from the distant and recent past. In *Biological Conservation $V 104* (3), pp. 361–376.

Reid, Walter V. (2005): Ecosystems and human well-being. Synthesis ; a report of the Millennium Ecosystem Assessment. Washington, DC: Island Press. Available online at http://www.loc.gov/catdir/enhancements/fy0666/2005010265-d.html.

Sanger, F.; Nicklen, S.; Coulson, A. R. (1977): DNA sequencing with chain-terminating inhibitors. In *Proceedings of the National Academy of Sciences of the United States of America* 74 (12), pp. 5463–5467. DOI: 10.1073/pnas.74.12.5463.

Sexton, Jason P.; McIntyre, Patrick J.; Angert, Amy L.; Rice, Kevin J. (2009): Evolution and Ecology of Species Range Limits. In *Annu. Rev. Ecol. Evol. Syst.* 40 (1), pp. 415–436. DOI: 10.1146/annurev.ecolsys.110308.120317.

Svoray, Tal; Mazor, Shira; Bar, Pua (2007): How is Shrub Cover Related to Soil Moisture and Patch Geometry in the Fragmented Landscape of the Northern Negev desert? In *Landscape Ecol* 22 (1), pp. 105–116. DOI: 10.1007/s10980-006-9004-3.

Verwaaijen, Bart; Wibberg, Daniel; Nelkner, Johanna; Gordin, Miriam; Rupp, Oliver; Winkler, Anika et al. (2018): Assembly of the Lactuca sativa, L. cv. Tizian draft genome sequence reveals differences within major resistance complex 1 as compared to the cv. Salinas reference genome. In *Journal of biotechnology* 267, pp. 12–18. DOI: 10.1016/j.jbiotec.2017.12.021.

Yaacobi, Gal; Ziv, Yaron; Rosenzweig, Michael L. (2007): Habitat fragmentation may not matter to species diversity. In *Proceedings. Biological sciences* 274 (1624), pp. 2409–2412. DOI: 10.1098/rspb.2007.0674.