
EAGER Documentation

Release 1.92

Alexander Peltzer

Jan 26, 2018

Contents

1	Prerequisites for the installation of EAGER	3
1.1	Operating System Support	3
1.2	Software Requirements for EAGER	4
1.3	File Naming Scheme	4
2	Installation Instructions for the EAGER Pipeline	7
2.1	VirtualBox	7
2.2	Singularity	8
2.3	Manual Installation	9
3	General Usage of EAGER/EAGER-CLI	11
4	Module description	13
4.1	FastQC	13
4.2	Adapter RM / Merging	13
4.3	QualityFiltering	15
4.4	Mapping	15
4.5	Complexity Estimation	18
4.6	Remove Duplicates	18
4.7	Contamination Estimation	19
4.8	Coverage/Statistics Calculation	19
4.9	MapDamage Calculation	19
4.10	SNP Calling	20
4.11	SNP Filtering	21
4.12	VCF2Genome	22
4.13	CleanUp	23
4.14	Create Report	23
5	General Report Interpretation Guide	25
5.1	Sample Number	25
5.2	Sample name	25
5.3	# of raw reads after C&M prior mapping	25
5.4	# of merged reads	25
5.5	# Reads not attempted to map	26
5.6	% merged reads	26
5.7	# mapped reads prior RMDup	26
5.8	# mapped reads prior RMDup QF	26

5.9	# of Duplicates removed	26
5.10	Mapped Reads after RMDup	26
5.11	Endogenous DNA (%)	26
5.12	Cluster Factor	26
5.13	Mean Coverage	27
5.14	std. dev. Coverage	27
5.15	Coverage >= 1X	27
5.16	Coverage >= 2X	27
5.17	Coverage >= 3X	27
5.18	Coverage >= 4X	27
5.19	Coverage >= 5X	27
5.20	# SNPs	27
5.21	AVG Coverage on mitochondrium	27
5.22	Initial cont est	28
5.23	Initial cont est low	28
5.24	Initial cont est high	28
5.25	Final cont est	28
5.26	Final cont est low	28
5.27	Final cont est high	28
5.28	GC content	28
5.29	# of reads on mitochondrium	28
5.30	MT/NUC Ratio	28
5.31	DMG 1st Base 3'	29
5.32	DMG 2nd Base 3'	29
5.33	DMG 1st Base 5'	29
5.34	DMG 2nd Base 5'	29
5.35	average fragment length	29
5.36	median fragment length	29
6	Tutorials	31
6.1	Use case I: Mitochondrial analysis	31
6.2	Use case II: Bacterial analysis	37
6.3	Use Case III: Human (WGS) analysis	43
7	FAQ	51
7.1	I am missing Feature X for my analysis	51
7.2	I have some BAM files already preprocessed and don't want to map everything again	51
7.3	I am using EAGER to reconstruct several genomes simultaneously but it doesn't work	51
7.4	I have several samples from the same individual (e.g. pre-screening and a wgs dataset) and would like to combine these	52
7.5	I have an error and I don't know what to do	52
8	Licencing	53
8.1	Important Licencing Information	54
8.2	GATK documentation resources and support	54
9	Citations	55
9.1	Tools & Methods	55
10	Indices and tables	57



This is the main EAGER wiki, where all the information regarding installation, maintenance, updating and usage of the pipeline will be documented on several written wiki pages accompanied with tutorial material such as videos and short usage descriptions.

Contents:

Prerequisites for the installation of EAGER

1.1 Operating System Support

1.1.1 Linux

EAGER has been successfully tested on several types of operating systems, supporting the underlying tools and methods. These include several flavours of Linux based operating systems including Debian ‘Jessie’, Ubuntu 16.04 LTS, CentOS 7 and ArchLinux.

1.1.2 Mac OSX

The pipeline can be installed and configured on Mac OSX 10.x as well, however some of the tools used by EAGER are stating that they might be unstable on OSX. Therefore, we do not recommend to run the pipeline directly on OSX, but instead rely on a Linux workstation, cluster or the usage of our Docker based EAGER image instead, which is running perfectly fine on OSX as well.

1.1.3 Windows

Note: There are currently **no plans** to support Windows as a operating system.

Merely, this is a limitation posed not by EAGER itself, but rather of many of the underlying tools which are not running on Windows and have been developed for Linux based operating systems.

1.2 Software Requirements for EAGER

1.2.1 Docker Image Based Installation

You will only need to install Docker on your host system. There are several manuals available to install and run docker on your machine, depending on your operating system. Note that this is the only supported installation method on OSX and Windows host machines.

- For Linux, look up the installation manual [here](#)
- For OSX, look up the installation manual [here](#)
- For Windows, look up the installation manual [here](#)

Furthermore, you will need to install the [Docker compose framework](#) to make installation and maintenance of the Docker infrastructure easier. Furthermore, you will need [Git](#) on your system.

Note: The usage of Docker on both OSX and Windows machines is relying on virtualization technology. This means that you will experience a performance drawback of roughly 10-20% compared to a native installation on a similar Linux machine. This is not specific to EAGER, but to the Windows operating system and Docker connection between the host and guest system.

1.2.2 Direct Host Installation

You will need to have a running Linux machine (e.g. Ubuntu 16.04, ArchLinux, OpenSuSe, RedHat or CentOS), ideally in a 64bit flavor installed and running. Administrative rights are not required, but may make the installation of tools easier. Afterwards, follow the installation instructions in the advanced section.

1.2.3 VirtualBox Installation

If you would like to install EAGER as a VirtualBox image, to simply try out the pipeline without having to install many software packages, you will be required to install VirtualBox first on your operating system. To install the VirtualBox software, simply follow the instructions available [here](#).

For OSX users, we also created a video, describing the whole process on a OSX Yosemite client machine. Afterwards, you can follow the setup instructions on [VirtualBox Installation Guide for EAGER](#)

1.3 File Naming Scheme

EAGER relies on naming patterns that files should follow, to determine read pairs for example. For sample identification, the pipeline assumes that samples are sharing a same identifier and follow this kind of naming pattern:

```
SomeIdentifier_R1_LaneIdentifier.fq.gz
SomeIdentifier_R2_LaneIdentifier.fq.gz
```

If you select several samples like this, EAGER will automatically determine which ones belong to each other and process all of them in a single processing run.

Typically, depending on your local sequencing infrastructure or if you received samples from e.g. other labs, downloaded them from the SRA or other resources, you will receive several folders with each folder corresponding to a sample, e.g.:


```
Sample_XYZ/XYZ_R1_LaneIdentifier.fq.gz
           /XYZ_R1_LaneIdentifier.fq.gz
Sample_UVW/UVW_R1_LaneIdentifier.fq.gz
           /UVW_R1_LaneIdentifier.fq.gz
```

In this case you can simply select the parent folder of your input data containing the folders “Sample_XYZ” and “Sample_UVW” and EAGER will cope with the data itself.

Note: EAGER does not require your data to be uncompressed such as other pipelines do. All of the tools in the pipeline have been tuned to enable input to be compressed as *fq.gz*, so input from Illumina sequencers can directly processed without uncompressing the datasets first.

1.3.1 Reference Genomes

EAGER requires your reference genomes to be in FastA format. Generating an index for mapping is not required, as the pipeline determines whether the index needs to be determined automatically.

Note: If you have multiple reference genomes in a single folder, please generate folders for each of your references, otherwise index generation might run only once, creating indexes for only the first of your reference genomes.

Warning: Furthermore, EAGER (and some downstream tools) require your input FastA file to have a **.fasta* or **.fa* file ending and being encoded using UNIX newline characters. The tool ‘dos2unix’ can be used to convert your input to proper unix formatted FastA files. Rename your input reference to have a proper file ending to ensure the first constraint is met.

Installation Instructions for the EAGER Pipeline

We provide three kinds of installation instructions.

Note: We do not provide a docker image anymore, as the much more flexible Singularity method superseded the `deager` application and our Docker based approach.

2.1 VirtualBox

Warning: This should only be used for testing purposes as the image does not get updated at all and was only intended to try out the pipeline!

Note: This has some performance drawbacks due to virtualization techniques (typically ~20%).

We provide a VirtualBox based operating system image to end users that contains all the required software tools.

- Download the corresponding [VirtualBox Image](#)
- Unpack the image
- Load the image with VirtualBox, click on File, Open, Image and select the unpacked image file
- Click on Start in VirtualBox and wait a couple of seconds until you see a regular desktop environment in VirtualBox
- You may run the pipeline's two components now typing either `eager` or `eagercli`.

Two small videos illustrating the whole setup process can be found [online](#) and [here](#).

2.2 Singularity

Note: This is the default way to use EAGER in a containerized environment. Best user experience, minimum performance drawbacks.

In order to use this approach, you will need a running Linux operating system at hand (e.g. ArchLinux, Ubuntu > 14.04, CentOS 7 or similar).

Warning: In theory, this should work on OSX, but due to the nature of OSX using a Virtualization technique based on VirtualBox, you could instead use the VirtualBox image on such systems, too.

First of all, install Singularity on your machine that you would like to use for the setup. To do this, follow the instructions from the authors [here](#). There are installation instructions for OSX and Windows, too - but these will have some performance drawbacks. Once you have a working singularity installation, there is just three commands you will need to run for getting EAGER to work:

First of all, download the pipeline at a location where you want to run your analysis, e.g. `/home/<username>/Downloads`. Switch to that directory and type this in the commandline:

```
singularity pull shub://apeltzer/EAGER-GUI:master
```

2.2.1 Running the GUI

Now we can run the GUI for

```
singularity exec -B /path/to/your/data:/data /home/<username>/Downloads/apeltzer-  
↳EAGER-GUI-master.img eager  
#/path/to/your/data = Path where you store RAW sequencing data, a reference genome in,  
↳FastA format and the folder where you store your results in the end.  
#/home/<username>/Downloads/apeltzer-EAGER-GUI-master.img is the name of the,  
↳previously downloaded image file.
```

This will open the EAGER graphical user interface (GUI), that is required for configuring the pipeline. Make sure to remember this path, as you will need it for the pipeline execution later on. Within the GUI, you can find your data in `/data`. You can navigate there when opening input files, the reference genome or the results and should also not select any folders or files in other directories.

Note: The `path/to/your/data` can be any path accessible from your workstation, so for example a departments data storage in the network would work, too.

Warning: Please make sure, that you have a following `:/data` after entering the path to your data storage. Otherwise, you will not be able to run a configuration.

After you are done with configuring your data, please close the graphical user interface.

2.2.2 Running the Analysis

You can now run the actual analysis procedure with `eagercli` by issuing the following command.

```
singularity exec -B /path/to/your/data:/data /home/<username>/Downloads/apeltzer-
↳EAGER-GUI-master.img eagercli /data
#again, keep the same path to your data and specify the ".img" path as before.
```

This will run the analysis procedure on your machine using the `eagercli` application inside the container.

Note: The results will be stored in the folder you selected in the configuration procedure. A good practice would be to have a separate folder inside your `path/to/your/data` just for this purpose.

2.2.3 Reproducibility

An important feature of this Singularity based approach is, that you can rerun both configuration and analysis whenever you want it. Simply keep the downloaded “pulled” image file with your whole analysis and you can reproduce everything in the future. For your convenience, we even created a small script that can be used e.g. for a publication to state which versions of each tool were used to produce a result (!). You can see these by running

```
singularity exec -B /path/to/your/data:/data /home/<username>/Downloads/apeltzer-
↳EAGER-GUI-master.img eagerVersions utilized_versions.txt
```

This will produce a textfile, containing information of the used tools within the selected image that were used to produce a result. Version tags of all the tools are then available in that specific textfile, too.

2.3 Manual Installation

Note: This is the native installation of the EAGER pipeline. It requires you to download tools manually, compile them and set paths accordingly in order for the pipeline to work on your operating system.

The manual installation on an infrastructure without access to a docker container is a bit more complex than installing the docker image, as all the requirements and subsequent tools for EAGER need to be linked correctly on the system running the pipeline in the end. This has certain requirements:

- Java 8 Environment, preferably the Oracle JDK8
- GNU Bash

After this, the following tools need to be installed by the user, ideally system wide or (if this is not possible due to access rights), by manually compiling them. In parentheses you can find the version(s) EAGER has been tested with.

Note: The EAGER-GUI, EAGER-CLI and all other components developed within the EAGER pipeline can be downloaded from Bintray as pre-compiled JAR files. You don’t need to re-compile these applications manually. In case you prefer to, please use [IntelliJ IDE](#) to do so.

List of Tools tested with EAGER:

- [ANGSD\(v0.910\)](#)
- [AdapterRemoval \(v2.2.1\)](#)

- BAM2TDF(v14)
- BGZip (depending on your linux distribution, you have this already installed)
- Bowtie 2(v2+)
- BWA (v0.7.15+)
- CircularMapper(latest)
- Clip & Merge(latest)
- Schmutzi (latest)
- DeDup (latest)
- EAGER (latest)
- EAGER-CLI (latest)
- FastX-Tools (v0.0.13)
- FastQC (v0.11.4)
- GATK (v3.7+)
- LibraryComplexityPlotter (latest)
- mapDamage (v2.0+)
- MTNucRatioCalculator (latest)
- Picard-Tools (v2+)
- Preseq (v2.0+)
- QualiMap (v2.3)
- ReportTable (latest)
- Samtools (v1.4.0+)
- Stampy (current)
- Tabix (v1.3.0)
- VCF2Genome (latest)

In order to make installation more easy, I provide [installation files for linking](#) the tools correctly. You will have to adjust in each file (open with a text editor) the correct location to the executables. Once you've done this and installed all the tools required for EAGER, you can simply add the location of these scripts to your path, e.g.

```
PATH=/data/eager-links/:$PATH
```

This will *add* links to the respective tools in order to allow EAGER to find the corresponding tools. If you for example already have working installations of *BWA*, *samtools* or similar, you will only need to install the missing tools of course. Please make sure, that you have the proper versions of the tools installed that EAGER needs or otherwise you might have to define these in your path as well.

Now you can check by e.g. entering *eager* whether you get a message about running EAGER. If you set EAGER up on a cluster infrastructure, you may need to have X11 forwarding enabled there to run the pipeline. For windows clients, there is a howto available [here](#). For Linux client machines, you'd probably only have to run:

```
ssh you@yourheadnode.yourcluster -Y
```

If you are uncertain on how to run X11 forwarded applications on your local infrastructure, your IT department should be able to set this up for you or will help you in achieving this.

General Usage of EAGER/EAGER-CLI

The second part of EAGER is referred to as *eagercli*, also called the command line version of EAGER. In order for running this part of the pipeline, it is crucial, that you followed the installation instructions and have a working installation of EAGER at your hand.

Once you have created configuration files for your analysis, you may run and execute the configuration files using this command line component of EAGER. Depending on your installation, you can simply execute:

```
eagercli /path/to/configuration/files/
```

The CLI should then find all the generated configuration files (which are stored in a XML format), then subsequently execute them and run the required analysis steps as you configured them. You may call individual XML files directly, too:

```
eagercli /data/real_analysis/Sample_XYZ/2016-03-07-EAGER.xml
```

If you have multiple samples, we however recommend executing these by selecting a folder/directory one level up and EAGER would find all XML files in the subdirectories of this folder:

```
eagercli /data/real_analysis/
```

Note: This will run all samples in the folder **real_analysis** for your convenience.

Module description

EAGER comes with lots of different modules for different use cases, thus enabling the user to configure the pipeline in a fine granular way. This section describes the different modules in more detail than e.g. the user tutorials that are offered here in the documentation too.

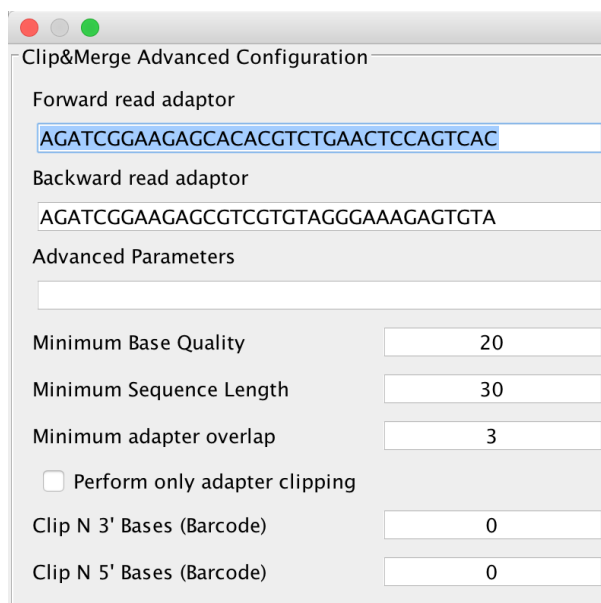
4.1 FastQC

This module can not be configured and is utilized to gain first insight into a raw sequencing dataset to determine important basic statistics, such as for example GC content and average read lengths prior to modifying the data. This should be used whenever you want to analyze data coming from sequencing without having an idea if even sequencing was successful as it creates basic plots showing whether the data is suitable for further downstream analysis.

4.2 Adapter RM / Merging

Enables you to select either our in-house adapter clipping and read merging application Clip&Merge or AdapterRemoval v2.2+ for adapter removal and read merging.

Clicking on the **Advanced** button next to this, you will get either the advanced configuration for Clip&Merge as depicted below:



Clip&Merge Advanced Configuration

Forward read adaptor
AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC

Backward read adaptor
AGATCGGAAGAGCGTCGTAGGAAAGAGTGTA

Advanced Parameters

Minimum Base Quality 20

Minimum Sequence Length 30

Minimum adapter overlap 3

☐ Perform only adapter clipping

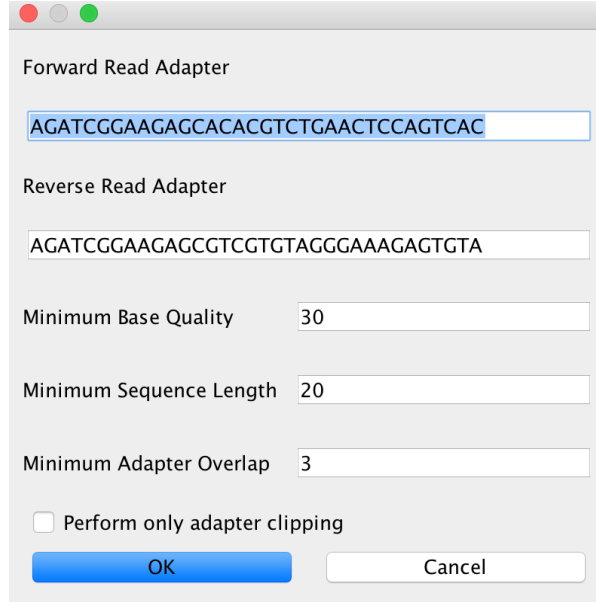
Clip N 3' Bases (Barcode) 0

Clip N 5' Bases (Barcode) 0

You can select forward and reverse read adapters here that are then subsequently clipped off your sequencing reads during analysis. Furthermore, Clip&Merge performs a base quality trimming of unmerged reads, filters out sequences falling below a certain length. Also, you can specify the minimum adapter overlap length you require (the number of bases of your specified adapter sequence required to overlap with your sequencing read). If you don't want to merge your reads, you can also specify to only clip adapters without merging reads afterwards. Barcodes are supported too, you can specify to trim bases from both 3' and 5' ends as well in the application. Another feature is available to only include merged reads into downstream analysis. Unmerged reads are kept in separate files, one for forward and one for reverse reads, stored in the same folder than the merged ones but unused for further downstream analysis.

Warning: Specifying wrong adapters, trimming too many bases here will result in poor analysis performance, so make sure beforehand which adapters to use in your analysis. For single ended data, it is advisable to use AdapterRemoval v2, as it is more sensitive to very small adapter fragments.

In case you selected AdapterRemoval, you will be able to select basically the same criteria than for Clip&Merge. The **Advanced** setting will look like this in that case:



Forward Read Adapter

AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC

Reverse Read Adapter

AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTCTA

Minimum Base Quality 30

Minimum Sequence Length 20

Minimum Adapter Overlap 3

☐ Perform only adapter clipping

OK Cancel

Note: It is completely up to you which adapter removal and merging procedure you'd like to use in your analysis. Recommendation: Use Clip&Merge for paired-end data with subsequent merging and AdapterRemoval for single-end data and paired-end data without read merging.

4.3 QualityFiltering

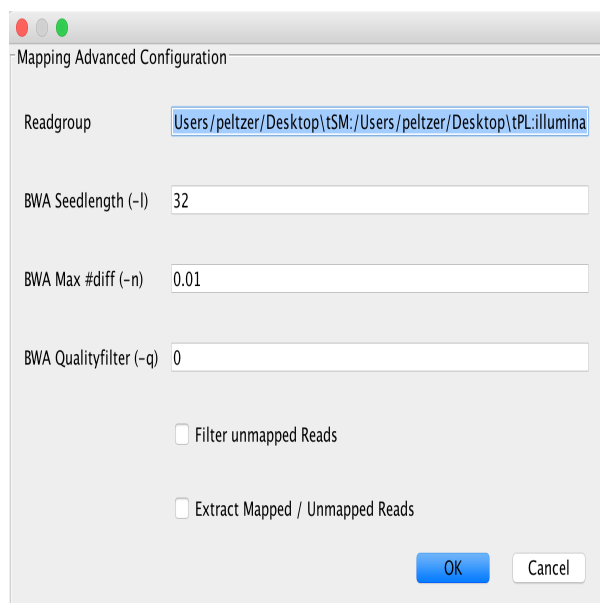
For downward compatibility reasons, we have the possibility to filter sequences based on quality in the pipeline, too. This tool has been replaced by Clip and Merge and is therefore deactivated by default when Clip and Merge is selected.

4.4 Mapping

These modules configure the read mapping process. EAGER currently features four mapping algorithms, which can be used. *BWA*, *CircularMapper* and *BWAMem* have been tested intensively, *Bowtie2* works well too, but can not be configured in detail as of now. *Stampy* is currently to be seen as experimental and may not work in all conditions.

4.4.1 BWA

This is the default mapping algorithm, largely used for mapping reads to ancient genomes and has been used in many ancient sequencing projects. If you're not sure what to use, use this algorithm.



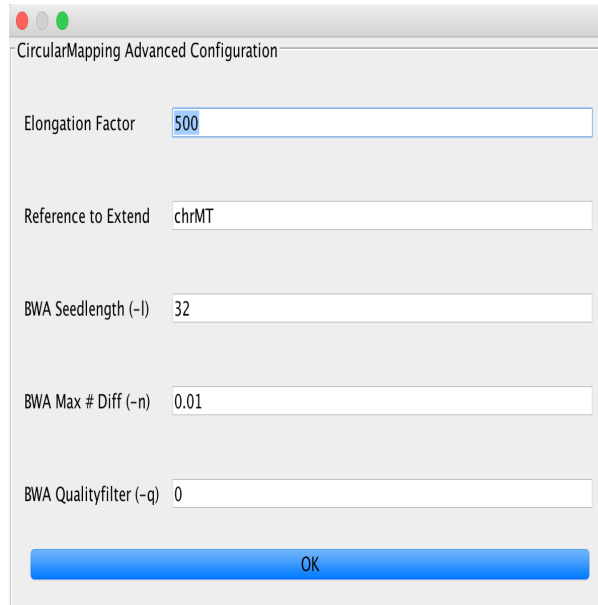
Note: If you're not sure which parameter you should be using for `-n`, use this web service to determine an optimal parameter for your data using an interactive choice [tool](#)

Note: In many ancient DNA sequencing projects, analysts turn off the seeding factor `-l` by setting it to a value significantly larger than the read length is done to gain better mapping rates for damaged ancient fragments. In case you receive bad mapping results, consider disabling seeding.

4.4.2 CircularMapper

This relies on the BWA mapper, but utilizes some tricks to obtain better mapping results on circular genomes. You can set the elongation factor to longer values in case you have data that includes longer reads. The *Reference to extend* value needs to describe the FastA entry that is used by the mapper for extension, e.g. if you have multiple chromosomes in your FastA reference, you need to specify one (or more, separated by a ;) chromosome to be extended by the algorithm.

Note: Make sure that you use the first part of your reference identifier, for example until the first space is reached as identifier. Something like `gi|123445|` works, whether our matching method doesn't work with `gi|34425|12345`. Don't worry about the identifier containing pipe symbols, this is taken care of.



A macOS-style dialog box titled "CircularMapping Advanced Configuration". It contains five text input fields with the following labels and values: "Elongation Factor" with "500", "Reference to Extend" with "chrMT", "BWA Seedlength (-l)" with "32", "BWA Max # Diff (-n)" with "0.01", and "BWA Qualityfilter (-q)" with "0". At the bottom is a blue "OK" button.

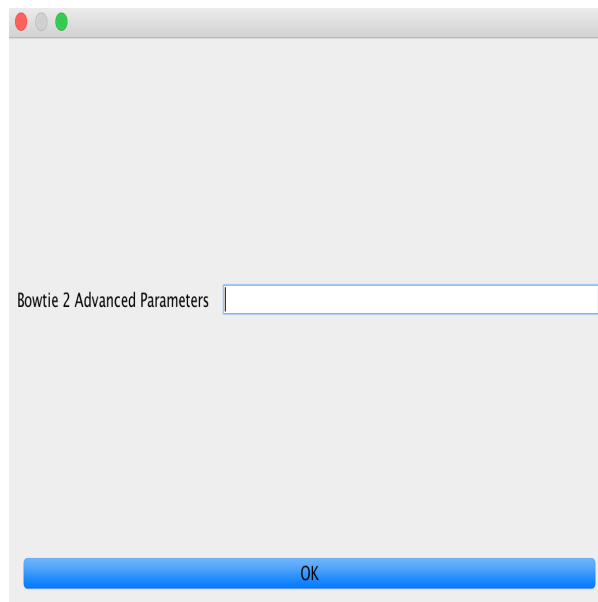
You can further adjust the BWA mapping parameters here, too.

4.4.3 BWAMem

BWAMem can not be configured in the pipeline and is executed with default values if you select this algorithm. We will add more parameters in an upcoming version of EAGER.

4.4.4 Bowtie2

You can specify parameters for Bowtie 2 here. These will be simply passed through to the mapping algorithm.



A macOS-style dialog box titled "Bowtie 2 Advanced Parameters". It features a single large text input field for specifying parameters. At the bottom is a blue "OK" button.

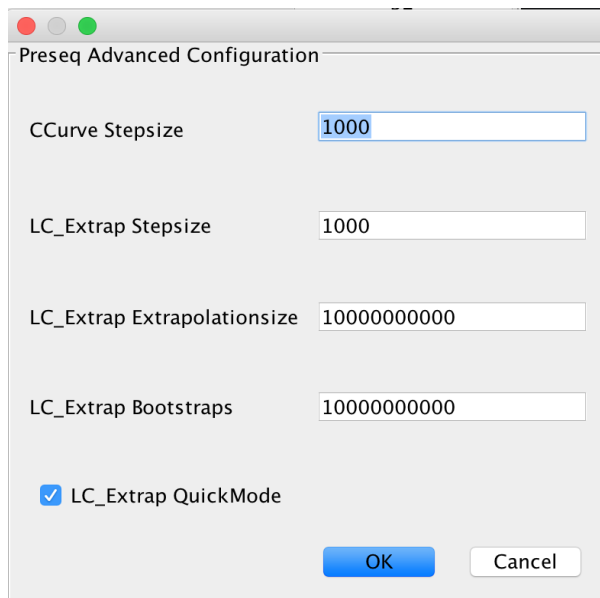
Warning: If you specify parameters that are either non-existent or incorrect for the mapper, your analysis will fail subsequently.

4.4.5 Stampy

Stampy can not be configured in the pipeline and is executed with default values if you select this algorithm. We will add more parameters in an upcoming version of EAGER.

4.5 Complexity Estimation

The complexity estimation is done using Preseq, running both components `c_curve` and `lc_extrap` after each other to determine the library complexity. Enable this module if you are testing a new sequencing library for complexity, to determine whether further deeper sequencing is justifiable.

A screenshot of a macOS-style dialog box titled "Preseq Advanced Configuration". It contains four text input fields: "CCurve Stepsize" with the value "1000", "LC_Extrap Stepsize" with the value "1000", "LC_Extrap Extrapolationsize" with the value "10000000000", and "LC_Extrap Bootstraps" with the value "10000000000". Below these fields is a checked checkbox labeled "LC_Extrap QuickMode". At the bottom right are "OK" and "Cancel" buttons.

CCurve Stepsize	1000
LC_Extrap Stepsize	1000
LC_Extrap Extrapolationsize	10000000000
LC_Extrap Bootstraps	10000000000
<input checked="" type="checkbox"/> LC_Extrap QuickMode	
<div>OK Cancel</div>	

4.6 Remove Duplicates

EAGER provides two different duplicate removal procedures: The *DeDup* and the *MarkDuplicates* method (provided by Picard).

4.6.1 DeDup

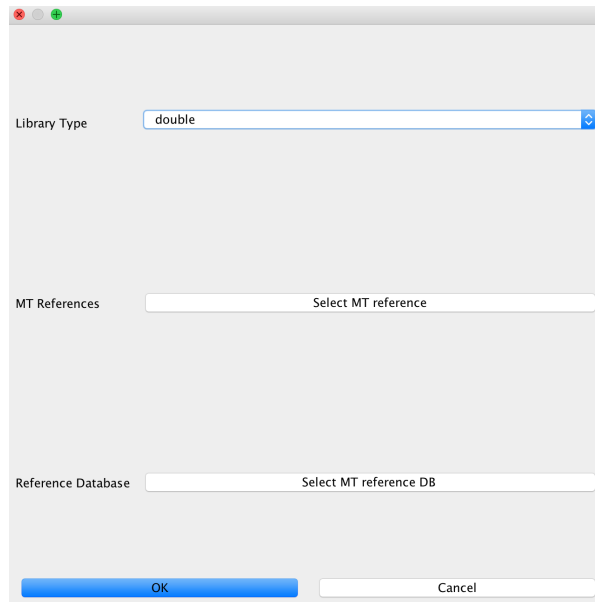
Use this if you're working with merged reads, single ended reads or a mixture of merged and remaining single ended reads that could not have been merged previously. This produces increased coverages as merged reads are treated correctly by looking at both ends of the merged reads instead of only considering start positions of these reads.

4.6.2 MarkDuplicates

Use this if you're working with paired end data, that has **not been merged**.

4.7 Contamination Estimation

This module is used to configure contamination estimation using `schmutzi`. In order to make this work, you will need to specify whether you have single stranded or double stranded libraries sequenced. Afterwards, you will need to specify the mitochondrial genome you would like to test against (usually of your human genome). Finally, select the folder with frequency data of putative mitochondrial sequences.

A screenshot of a software window titled "Contamination Estimation". The window has a light gray background and standard macOS window controls (red, yellow, green buttons) in the top-left corner. It contains three main configuration sections: "Library Type" with a dropdown menu set to "double"; "MT References" with a text field containing "Select MT reference"; and "Reference Database" with a text field containing "Select MT reference DB". At the bottom of the window are two buttons: a blue "OK" button and a white "Cancel" button with a gray border.

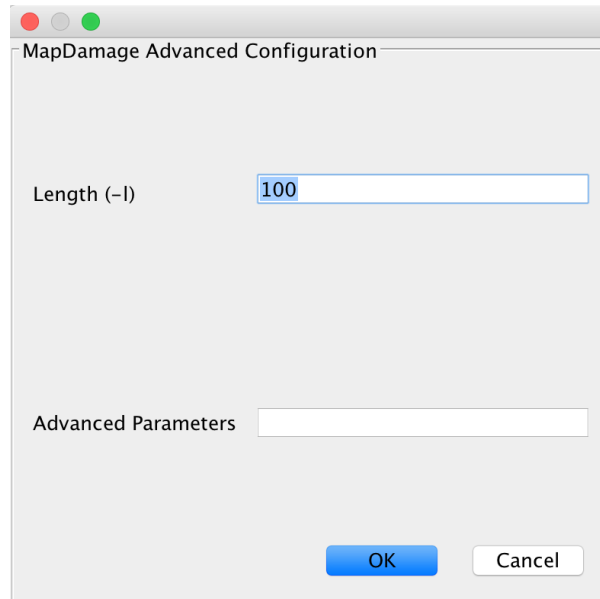
Note: If you are not working on *mitochondrial* data and did not select this, you may only specify the library type without configuring the other options. You don't need to specify these for bacterial data, too as the mitochondrial test can only be performed with a library of putative mitochondrial reference genomes.

4.8 Coverage/Statistics Calculation

This module handles coverage and other statistics calculation using QualiMap. This is enabled by default and can not be turned off at all.

4.9 MapDamage Calculation

This module handles calculation of DNA damage, which is used for authentication of samples. You will get a plot and damage statistics telling you whether you truly see ancient fragments in your dataset or not. You may specify more advanced parameters here, too.



4.10 SNP Calling

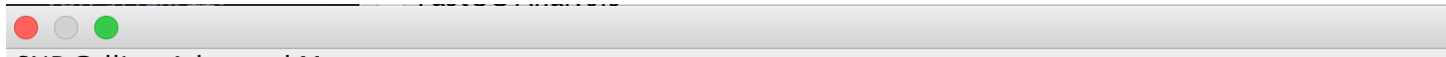
This section is used to specify methods for genotyping your mapped datasets. Note that these depend on your mapping results, meaning that samples containing very few reads will not result in good genotyping results either.

4.10.1 UnifiedGenotyper

You can set parameters for genotyping using the UnifiedGenotyper here. In case you have a reference database of known variants in VCF format for your respective organism (e.g. dbSNP for humans), you may specify this here, too. Refer to the [GATK documentation](#) to receive up to date information about the parameters offered here in EAGER.

4.10.2 HaplotypeCaller

You can set parameters for genotyping using the HaplotypeCaller here. In case you have a reference database of known variants in VCF format for your respective organism (e.g. dbSNP for humans), you may specify this here, too. Refer to the [GATK documentation](#) to receive up to date information about the parameters offered here in EAGER.



SNP Calling Advanced Menu

SNP Reference (e.g. dbSNP)

Caller Configuration

Ploidy of Organism

Standard Call Confidence

Downsampling

Advanced Parameters

☐ Emit All Sites / BP_RES Mode HC ☐ Emit Conf Sites /

ANGSD Configuration

Warning: Selecting the `EMIT All Sites?` option should only be done on small reference genomes. For a human genome, this produces uncompressed VCF files in the size of up to 90GB/sample. For some purposes, it might still be required but in most cases its not advisable to turn this on.

4.10.3 ANGSD

This can be used to configure the ANGSD method for genotyping low coverage genomes using genotype likelihoods. You can specify the likelihood model to use, the output format you want to generate and method to make a call at a certain position. Furthermore, you can specify whether you'd like to generate a FastA sequence of your calls in the end.

ANGSD Configuration

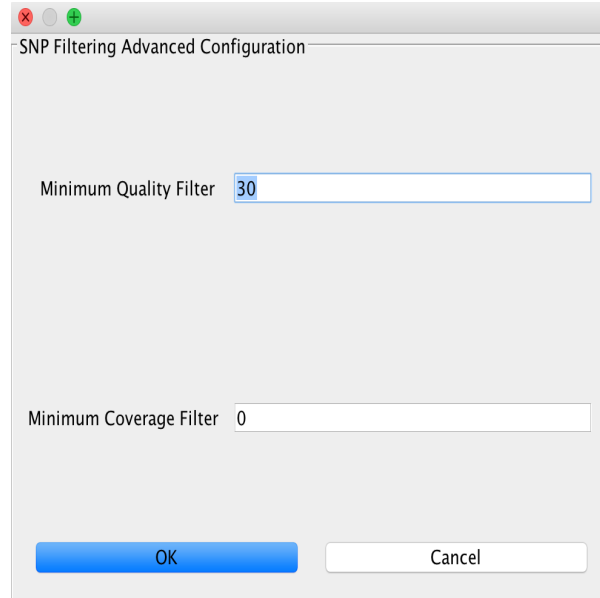
Genotype Likelihood Model

Genotype Likelihood Outformat

☐ Create FastA file?

4.11 SNP Filtering

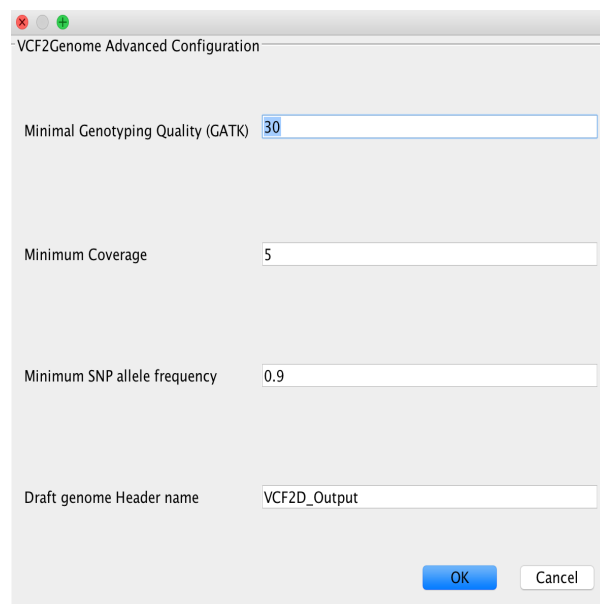
This can be used to filter variants based on minimum quality of a genotyping call and a minimum coverage using the GATK VariantFilter application.



Note: Note that this only has an effect on genotypes. If you used the ANGSD method producing genotype likelihoods as an output format, you will not be able to perform SNP filtering using this method.

4.12 VCF2Genome

This method can be used to generate FastA files incorporating called variants from a generated VCF file. Particularly useful for bacterial data, it allows the user to select minimum genotyping quality, coverage and SNP allele frequency to consider a call as true. For a more detailed description, see the paper citations.



4.13 CleanUp

This module is responsible for cleaning up intermediate results. Mainly, these are files generated during file conversion, e.g. SAM files and unsorted BAM files that have been converted to sorted BAM format already and can thus be safely deleted.

Note: This will only delete files that are redundant, e.g. from which there exist copies with the exact same content.

4.14 Create Report

This will generate a report of your whole analysis run. After each sample, the CSV file gets updated by EAGER automatically. This way, you can basically evaluate your results while waiting for other samples to finish.

General Report Interpretation Guide

This is meant to be a description of all the output statistics that are produced by the EAGER ReportTable module and will therefore be updated once new statistics are added in the upcoming future.

5.1 Sample Number

The number of the sample in this project run.

5.2 Sample name

The name or ID of the sample.

5.3 # of raw reads after C&M prior mapping

This is the number of raw reads after clipping (and merging if this has been applied). This included both unmerged and merged reads, if you selected to only use merged reads for downstream analysis, look in the column 'merged reads' for your number of reads that went into the downstream analysis.

5.4 # of merged reads

The number of merged reads, merged by Clip & Merge.

5.5 # Reads not attempted to map

Only stated when using AdapterRemoval v2 to remove adapters and perform read merging. Pairs that could not be merged (e.g. having no negative insert) are stated here, but only if both partners occur (fw/rv). Singletons are still taken into the analysis.

5.6 % merged reads

The percentage of merged reads, generated by applying

$$ofmergedreads/ofrawreadsintotal$$

5.7 # mapped reads prior RMDup

The number of reads that were mapping to the provided reference genome prior to applying duplication removal either using DeDup or MarkDuplicates.

5.8 # mapped reads prior RMDup QF

The number of reads that were mapping to the provided reference genome prior to applying duplication removal, also including a potentially applied quality filtering on BAM conversion.

5.9 # of Duplicates removed

The number of duplicates removed by the duplication removal tool.

5.10 Mapped Reads after RMDup

The number of mapped reads after duplicate removal was applied.

5.11 Endogenous DNA (%)

The % endogenous DNA content in the selected sample.

5.12 Cluster Factor

An informative measure to determine the complexity of the underlying sample. A good cluster factor of approximately 1.0 is a sign of a high number of unique reads in the respective sample, whereas a high cluster factor could be seen as a measure to not invest more in sequencing.

5.13 Mean Coverage

The average coverage on the provided reference genome.

5.14 std. dev. Coverage

The standard deviation of the average coverage on the provided reference genome.

5.15 Coverage $\geq 1X$

The % of the genome covered with greater or equal than 1x coverage.

5.16 Coverage $\geq 2X$

The % of the genome covered with greater or equal than 2x coverage.

5.17 Coverage $\geq 3X$

The % of the genome covered with greater or equal than 3x coverage.

5.18 Coverage $\geq 4X$

The % of the genome covered with greater or equal than 4x coverage.

5.19 Coverage $\geq 5X$

The % of the genome covered with greater or equal than 5x coverage.

5.20 # SNPs

The number of SNPs found in the finally generated consensus sequence using VCF2Genome, applying the filter criteria defined in this tools as well.

5.21 AVG Coverage on mitochondrium

The average coverage on the specified mitochondrial genome.

5.22 Initial cont est

The initial average contamination estimate provided by schmutzi's contDeam method.

5.23 Initial cont est low

The initial lower 95% CI estimate on schmutzis contdeam method.

5.24 Initial cont est high

The initial higher 95% CI estimate on schmutzis contdeam method.

5.25 Final cont est

The average contamination estimate based on the mtCont method in schmutzi.

5.26 Final cont est low

The lower 95% CI estimate on schmutzis mtCont method.

5.27 Final cont est high

The higher 95% CI estimate on schmutzis mtCont method.

5.28 GC content

The GC content of the respective sample.

5.29 # of reads on mitochondrium

The number of reads on the mitochondrial genome.

5.30 MT/NUC Ratio

The ratio between mt and autosomal reads, calculated as follows:

$$\text{avgcoverageonmitochondrium/averagecoverageonautosome}$$

5.31 DMG 1st Base 3'

The damage on the 1st base on the 3' end of the merged reads.

5.32 DMG 2nd Base 3'

The damage on the 2nd base on the 3' end of the merged reads.

5.33 DMG 1st Base 5'

The damage on the 1st base on the 5' end of the merged reads.

5.34 DMG 2nd Base 5'

The damage on the 2nd base on the 3' end of the merged reads.

5.35 average fragment length

The average fragment length of your samples reads.

5.36 median fragment length

The median fragment length of your samples reads.

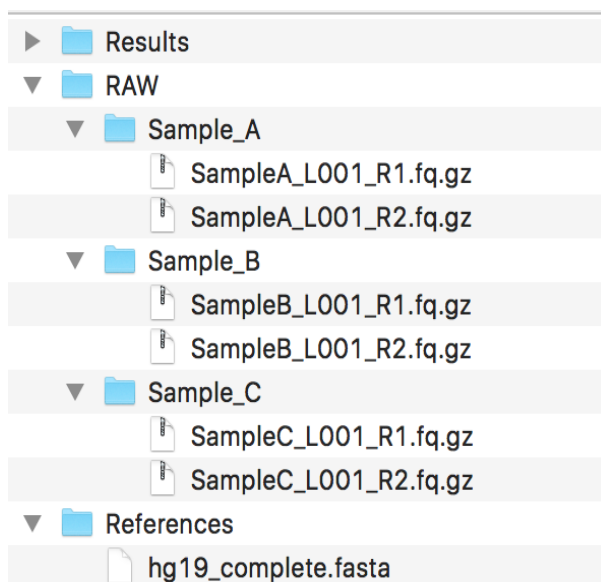
This section describes several potential use cases for EAGER: A complete analysis of a couple of mitochondrial captures, a bacterial genome analysis and an analysis of human whole genome shotgun data.

6.1 Use case I: Mitochondrial analysis

First, we will have to make sure that all data that we are using is there. This consists of three samples (Sample A-C) that have been created using a paired-end sequencing run on an Illumina sequencer. In order to run the analysis, we will utilize the EAGER pipeline and configure it step-by-step to run our analysis.

6.1.1 Step I: Data preparation

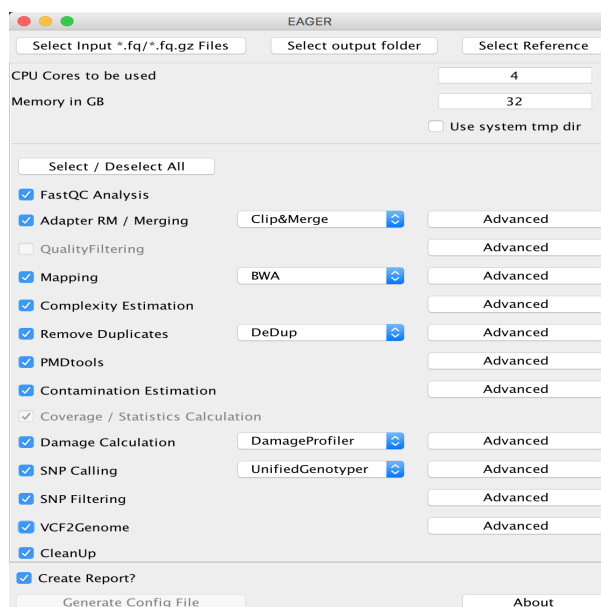
You should have a couple of folders set up, mainly containing the data, making sure the data follows the guidelines for *File Naming Scheme* . Ideally, your folder structure should look like:



Warning: If you need to perform genotyping, please ensure that your genome FastA file is ending with a *.fasta* file extension, or otherwise the GATK might complain about this.

6.1.2 Step II: Starting the GUI

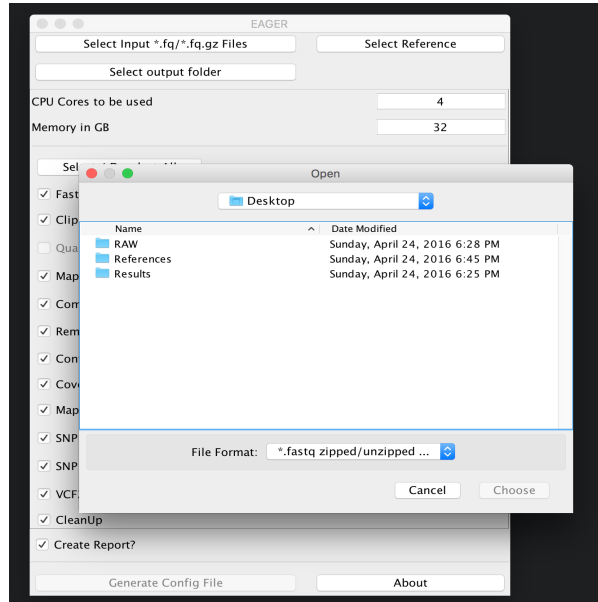
Depending on your installation type, you should have ensured a working graphical user interface of EAGER and start it. Once you have started the GUI, you will be welcomed by the main user interface of EAGER.



6.1.3 Step III: Selecting input

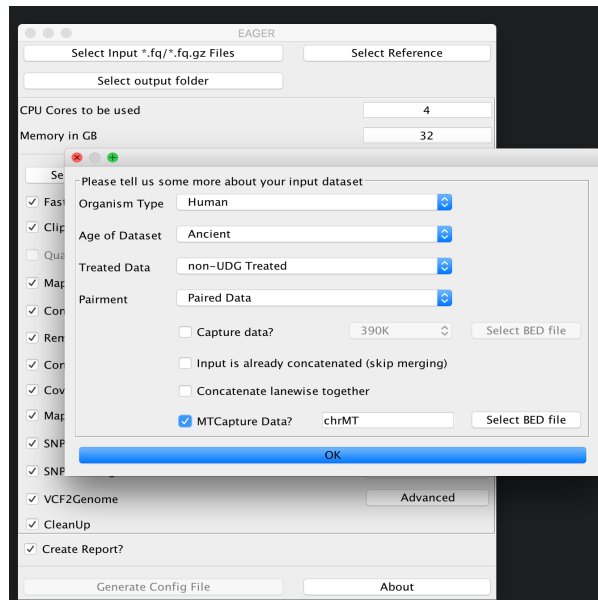
Selecting FastQ input

You can now click on *Select input *.fq/*.fq.gz Files* and navigate to your folder where the RAW sequencing input is stored on your network share or local hard drive.



Note: You may select **either** single/multiple FastQ files, **or** a folder containing subfolders with FastQ files. EAGER will pick up every FastQ file in all subfolders automatically.

In our case here, we simply select the folder *RAW* and click on *Choose*. A new window is opening up, asking you several questions to determine which kind of analysis should be performed on the selected data.



In our case here, we choose that our data has not been treated with UDG, we have paired-end sequencing data and

want to analyse a mitochondrial capture dataset.

Note: You have to specify a **BED** file for your reference genome if you want to analyse capture data in general. A typical BED file that could be used e.g. for HG19 mitochondrial analysis could look like this. Make sure to have tabulators between the six columns in your BED file. *NEW* BED files with solely 3-column format are supported now, too.

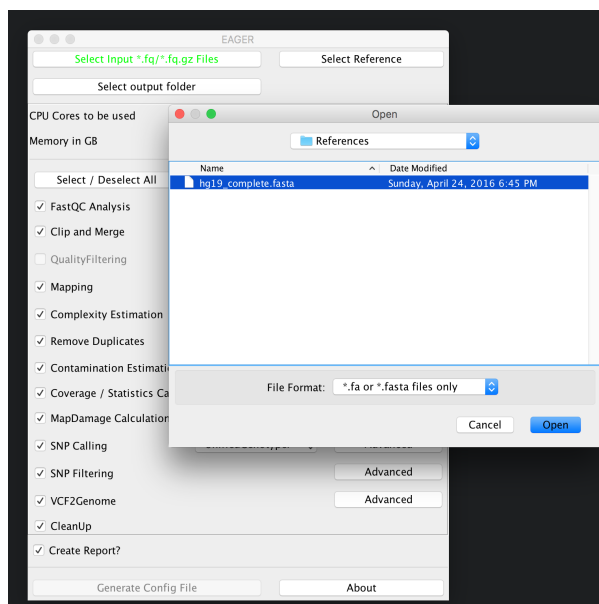
```
chrMT 1 16770 MT 1 +
```

Once you are done with selecting the appropriate BED file, you can click *ok* and the *Select input *.fq/*.fq.gz Files* button on top of the GUI should be green to display, that you have successfully selected your input dataset.

Selecting your reference genome

Continue now by clicking on *Select Reference* and select your reference genome in FastA format.

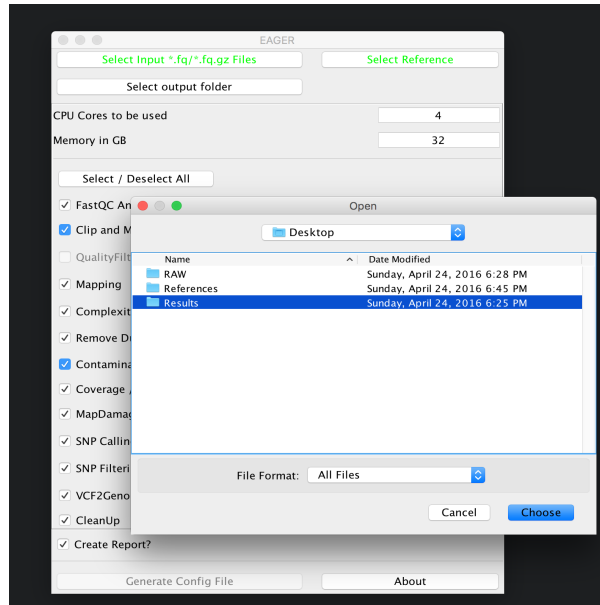
Note: You don't need to index any reference genomes manually. EAGER will take care of generating required indices on-the-fly when running the pipeline. If an index has been created, the pipeline will figure this out and no new one will be generated to save disk space and time.



Selecting your results folder

Note: EAGER uses a typical folder structure to store any produced output. This is called the results folder, in which EAGER creates subfolders on a per-sample basis, then populating these with the typical EAGER folder structure.

Simply click on the *Select output folder* button, then select a folder of your choice to store the analysis results in the end.



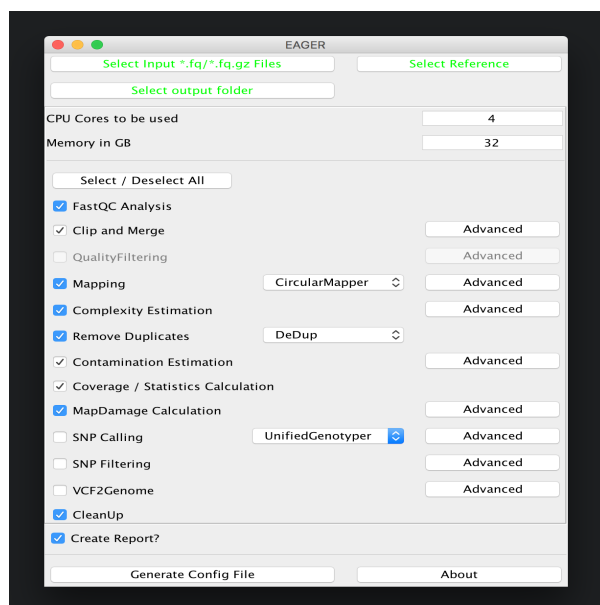
Warning: You have to ensure that you have proper access rights to the results folder and the reference genome FastA file or otherwise the analysis will fail.

6.1.4 Step IV: Configure your Analysis

Now that you have selected your input data, your reference genome and the corresponding output folder, you can configure the pipeline more in detail. Start by configuring how many CPU cores and how much random access memory (RAM) can be used by the pipeline for your analysis.

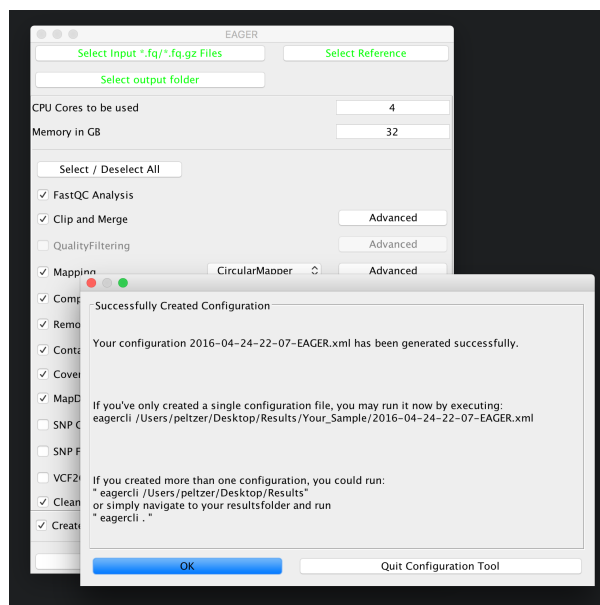
Warning: If you're unsure what to select for CPU cores and/or RAM consumption, you may want to look up your system configuration prior to starting an analysis here. Some processes can fail and make your system unstable when failing if you select too many CPU cores / use too much memory.

For a mitochondrial analysis, EAGER offers special features, e.g. a special mapping application called *CircularMapper* that produces improved mappings at both ends of your reference genome. In this case we basically keep most of the configuration at default settings, keeping initial FastQC analysis, Clip&Merge, Mapping with CircularMapper, Duplicate Removal, Contamination Estimation with Schmutzi, Coverage Calculation and MapDamage Calculation turned on but disabling the genotyping part of the pipeline. A final report in CSV format is also desirable in many applications, so we keep this turned on as well.



Note: The CleanUp module is removing *redundant* data, e.g. intermediate processing results, that are stored in different file formats to save disk space. In almost all cases you can safely keep this module turned on without compromising your analysis results.

After you are done with the configuration of the selected modules, e.g. by clicking on the *Advanced* buttons of the respective tools, you may click on *Generate Config File* on the bottom of the GUI to generate the required pipeline configuration files. A window should open up, telling you that your analysis run has been configured successfully.



6.1.5 Step V: Run the Analysis Pipeline

In order to execute the generated configuration files, the GUI is already giving you a little hint on how to run the execution part of the pipeline. Open up a Terminal application of your choice and then navigate to your folder(s)

containing the configuration files (your *result* folder) and run the *eagercli* command to execute the configuration file(s):

```
→ ~ cd /Users/peltzer/Desktop/Results/
→ Results ls -l
total 0
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_A
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_B
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_C
→ Results eagercli .
```

Note: You don't need to specify the full path to the generated configuration files, e.g. if you specify the *results* folder, EAGER will detect all configuration files automatically and run these sequentially after each other. For some purposes (e.g. a cluster system) you might want to schedule single jobs for each configuration file however, which can be done by specifying the path to the respective configuration files directly.

6.1.6 Step VI: Pick up results!

EAGER creates a CSV based report file in the results folder, which contains statistics for the analysis run. A typical results report looks like this:

Sample number	Sample Name	# of Merged Reads	% Merged Reads	# reads after C&M prior mapping	# mapped reads prior RMDup	# of Dupli
Sample 1/5	Sample_A	5437812	94.78 %	5737174	1023502	
Sample 2/5	Sample_B	12956100	80.5 %	16093580	2659178	
Sample 3/5	Sample_C	32041091	73.61 %	43528407	12782665	

All the output BAM files, VCF files and other important analysis results can be found in the sample specific folders in the results folder.

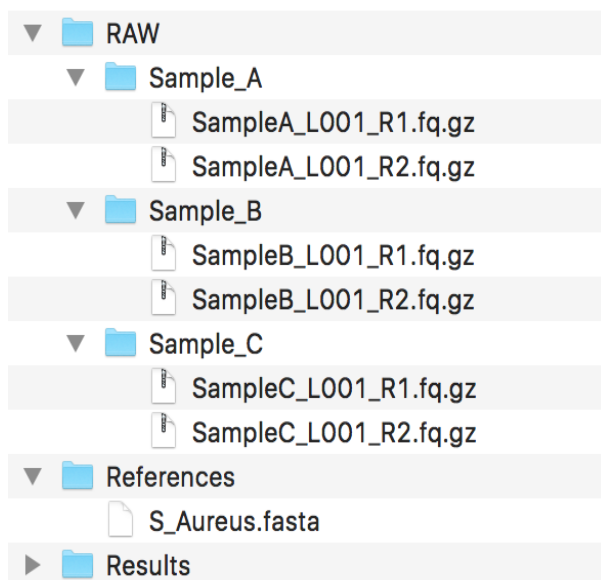
Note: You can import the results table in CSV format in any compatible sheet calculation software, LibreOffice for example works very well.

6.2 Use case II: Bacterial analysis

EAGER can be used to reconstruct ancient bacterial genomes in an efficient way, too. In order to perform such a bacterial genome reconstruction, we will be reconstructing three sample entries from ancient bacterial data in this tutorial using a *S. aureus* reference genome in FastA format.

6.2.1 Step I: Data preparation

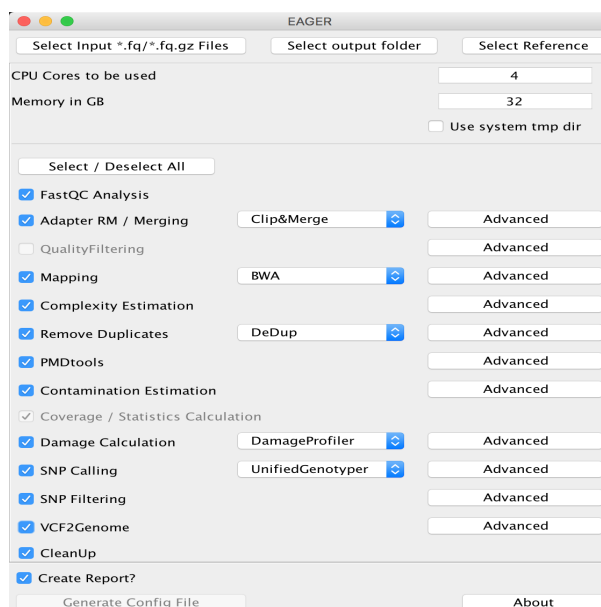
You should have a couple of folders set up, mainly containing the data, making sure the data follows the guidelines for *File Naming Scheme* . Ideally, your folder structure should look like:



Warning: As we do perform genotyping using the GATK, please ensure that your genome FastA file is ending with a *.fasta* file extension. If you don't do this, the pipeline can fail.

6.2.2 Step II: Starting the GUI

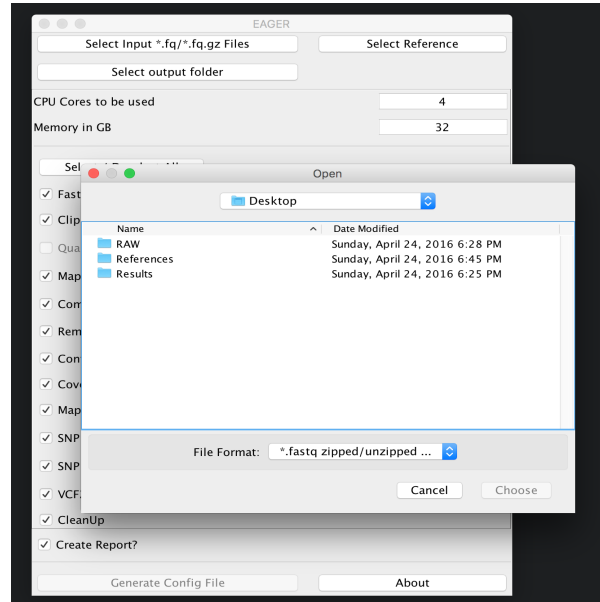
Depending on your installation type, you should have ensured a working graphical user interface of EAGER and start it. Once you have started the GUI, you will be welcomed by the main user interface of EAGER.



6.2.3 Step III: Selecting input

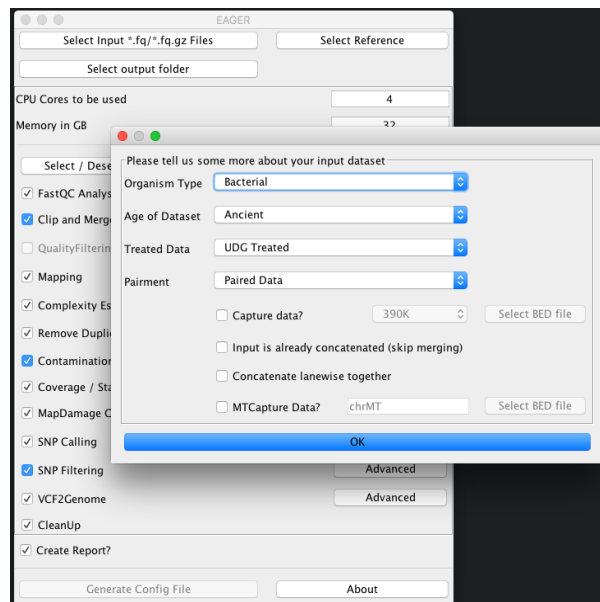
Selecting FastQ input

You can now click on *Select input *.fq/*.fq.gz Files* and navigate to your folder where the RAW sequencing input is stored on your network share or local hard drive.



Note: You may select **either** single/multiple FastQ files, **or** a folder containing subfolders with FastQ files. EAGER will pick up every FastQ file in all subfolders automatically.

In our case here, we simply select the folder *RAW* and click on *Choose*. A new window is opening up, asking you several questions to determine which kind of analysis should be performed on the selected data.



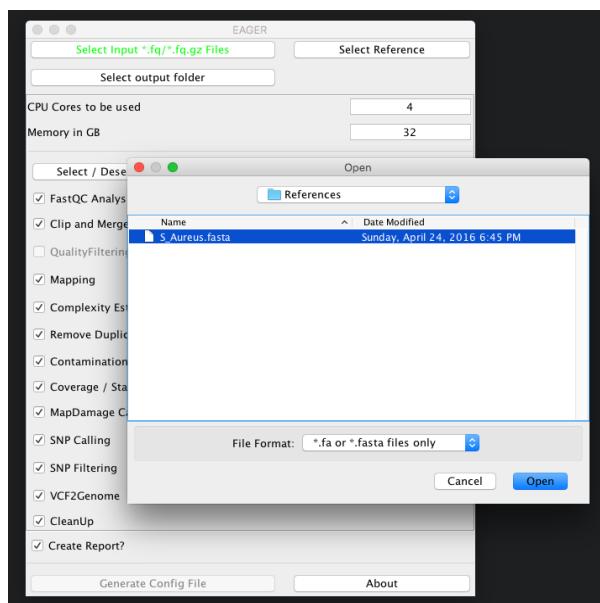
As we have UDG treated ancient bacterial data here, which has been sequenced in paired-end sequencing mode, we

simply select the appropriate types and click on *OK*. The *Select input *.fq/*.fq.gz Files* button on top of the GUI should be green to display, that you have successfully selected your input dataset.

Selecting your reference genome

Continue now by clicking on *Select Reference* and select your reference genome in FastA format.

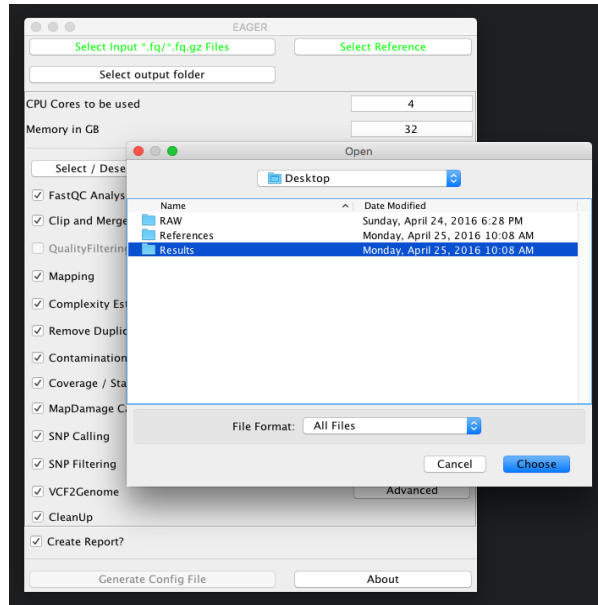
Note: You don't need to index any reference genomes manually. EAGER will take care of generating required indices on-the-fly when running the pipeline. If an index has been created, the pipeline will figure this out and no new one will be generated to save disk space and time.



Selecting your results folder

Note: EAGER uses a typical folder structure to store any produced output. This is called the results folder, in which EAGER creates subfolders on a per-sample basis, then populating these with the typical EAGER folder structure.

Simply click on the *Select output folder* button, then select a folder of your choice to store the analysis results in the end.



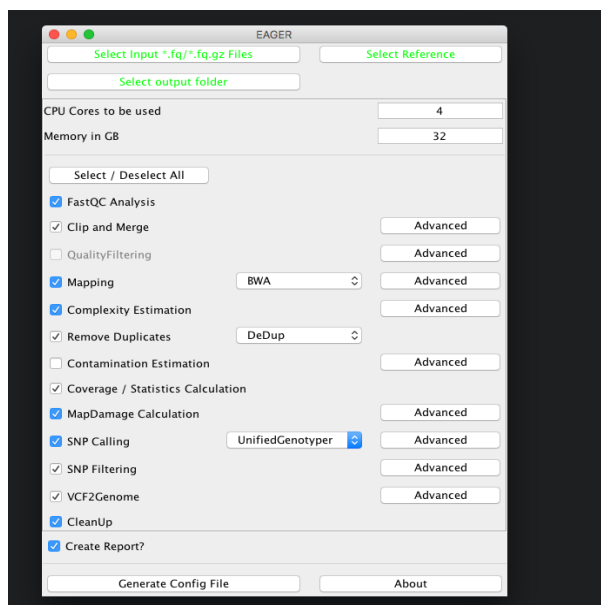
Warning: You have to ensure that you have proper access rights to the results folder and the reference genome FastA file or otherwise the analysis will fail.

6.2.4 Step IV: Configure your Analysis

Now that you have selected your input data, your reference genome and the corresponding output folder, you can configure the pipeline more in detail. Start by configuring how many CPU cores and how much random access memory (RAM) can be used by the pipeline for your analysis.

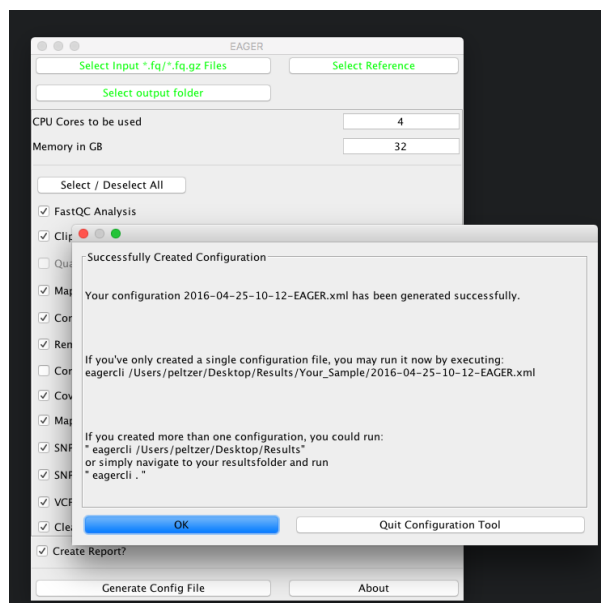
Warning: If you're unsure what to select for CPU cores and/or RAM consumption, you may want to look up your system configuration prior to starting an analysis here. Some processes can fail and make your system unstable when failing if you select too many CPU cores / use too much memory.

For bacterial data analysis, you may want to deselect the Contamination Estimation module, as it is tailored to mitochondrial contamination estimation and less suited for bacterial data. We would like to get a final FastA file with our called variants incorporated, so we keep the SNP calling, filtering and the VCF2Genome modules turned on in the pipeline.



Note: The CleanUp module is removing *redundant* data, e.g. intermediate processing results, that are stored in different file formats to save disk space. In almost all cases you can safely keep this module turned on without compromising your analysis results.

After you are done with the configuration of the selected modules, e.g. by clicking on the *Advanced* buttons of the respective tools, you may click on *Generate Config File* on the bottom of the GUI to generate the required pipeline configuration files. A window should open up, telling you that your analysis run has been configured successfully.



6.2.5 Step V: Run the Analysis Pipeline

In order to execute the generated configuration files, the GUI is already giving you a little hint on how to run the execution part of the pipeline. Open up a Terminal application of your choice and then navigate to your folder(s)

containing the configuration files (your *result* folder) and run the *eagercli* command to execute the configuration file(s):

```
→ ~ cd /Users/peltzer/Desktop/Results/
→ Results ls -l
total 0
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_A
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_B
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_C
→ Results eagercli .
```

Note: You don't need to specify the full path to the generated configuration files, e.g. if you specify the *results* folder, EAGER will detect all configuration files automatically and run these sequentially after each other. For some purposes (e.g. a cluster system) you might want to schedule single jobs for each configuration file however, which can be done by specifying the path to the respective configuration files directly.

6.2.6 Step VI: Pick up results!

EAGER creates a CSV based report file in the results folder, which contains statistics for the analysis run. A typical results report looks like this:

Sample number	Sample Name	# of Merged Reads	% Merged Reads	# reads after C&M prior mapping	# mapped reads prior RMDup	# of Dupli
Sample 1/5	Sample_A	5437812	94.78 %	5737174	1023502	
Sample 2/5	Sample_B	12956100	80.5 %	16093580	2659178	
Sample 3/5	Sample_C	32041091	73.61 %	43528407	12782665	

All the output BAM files, VCF files and other important analysis results can be found in the sample specific folders in the results folder.

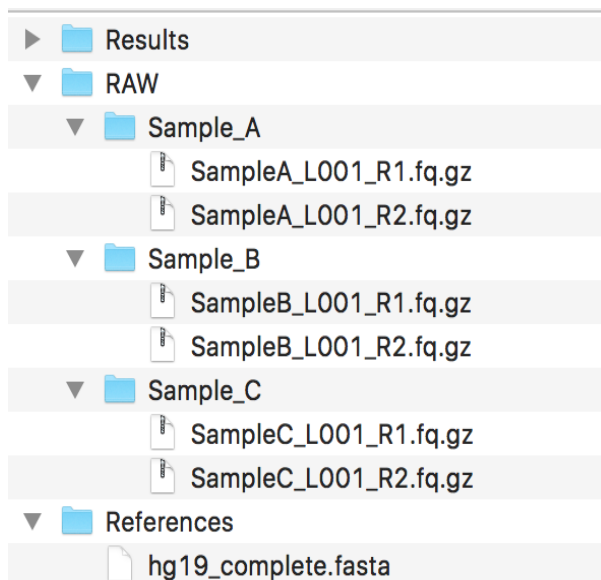
Note: You can import the results table in CSV format in any compatible sheet calculation software, LibreOffice for example works very well.

6.3 Use Case III: Human (WGS) analysis

First, we will have to make sure that all data that we are using is there. This consists of three samples (Sample A-C) that have been created using a paired-end sequencing run on an Illumina sequencer. In order to run the analysis, we will utilize the EAGER pipeline and configure it step-by-step to run our analysis.

6.3.1 Step I: Data preparation

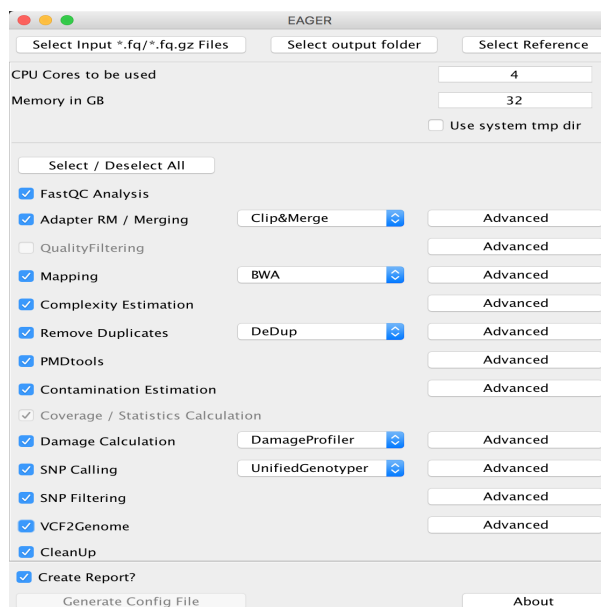
You should have a couple of folders set up, mainly containing the data, making sure the data follows the guidelines for *File Naming Scheme* . Ideally, your folder structure should look like:



If you need to perform genotyping, please ensure that your genome FastA file is ending with a *.fasta* file extension, or otherwise the GATK might complain about this.

6.3.2 Step II: Starting the GUI

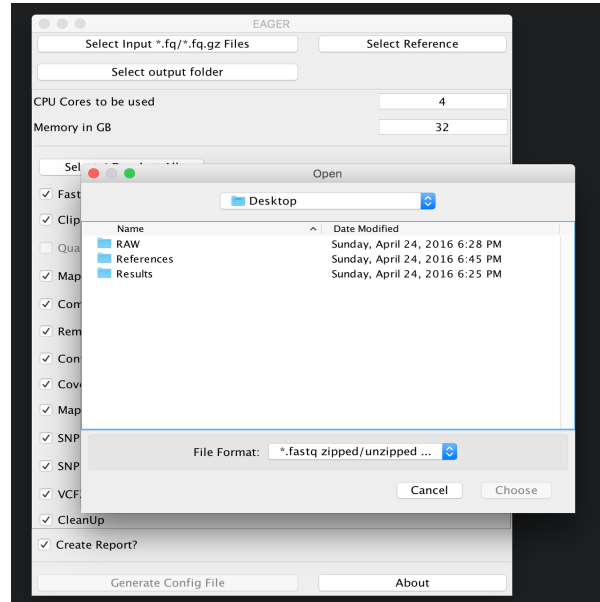
Depending on your installation type, you should have ensured a working graphical user interface of EAGER and start it. Once you have started the GUI, you will be welcomed by the main user interface of EAGER.



6.3.3 Step III: Selecting input

Selecting FastQ input

You can now click on *Select input *.fq/*.fq.gz Files* and navigate to your folder where the RAW sequencing input is stored on your network share or local hard drive.



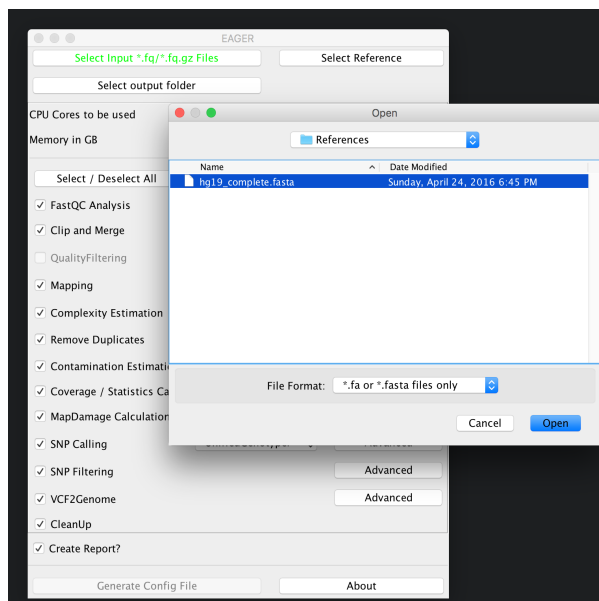
Note: You may select **either** single/multiple FastQ files, **or** a folder containing subfolders with FastQ files. EAGER will pick up every FastQ file in all subfolders automatically.

In our case here, we simply select the folder *RAW* and click on *Choose*. A new window is opening up, asking you several questions to determine which kind of analysis should be performed on the selected data.

Selecting your reference genome

Continue now by clicking on *Select Reference* and select your reference genome in FastA format.

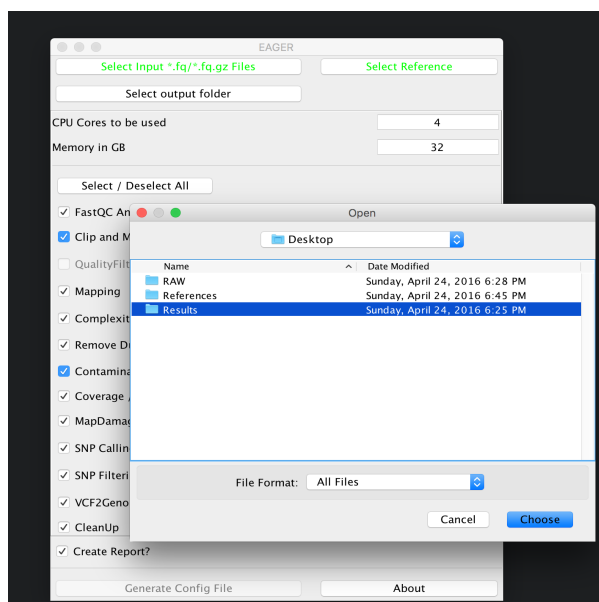
Note: You don't need to index any reference genomes manually. EAGER will take care of generating required indices on-the-fly when running the pipeline. If an index has been created, the pipeline will figure this out and no new one will be generated to save disk space and time.



Selecting your results folder

Note: EAGER uses a typical folder structure to store any produced output. This is called the results folder, in which EAGER creates subfolders on a per-sample basis, then populating these with the typical EAGER folder structure.

Simply click on the *Select output folder* button, then select a folder of your choice to store the analysis results in the end.

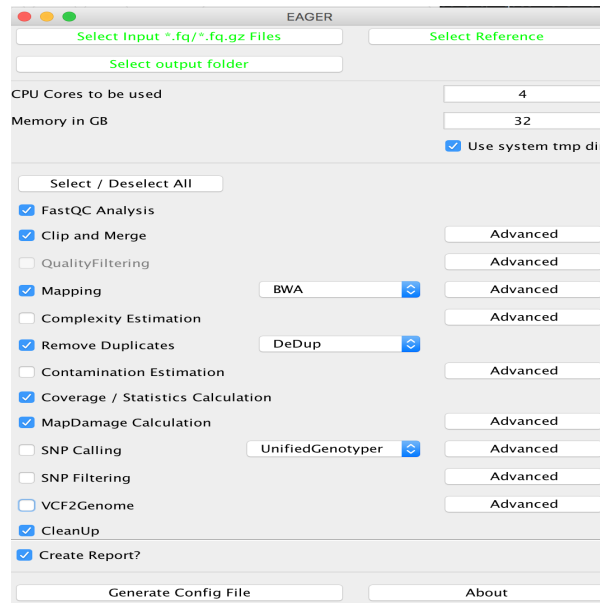


Warning: You have to ensure that you have proper access rights to the results folder and the reference genome FastA file or otherwise the analysis will fail.

6.3.4 Step IV: Configure your Analysis

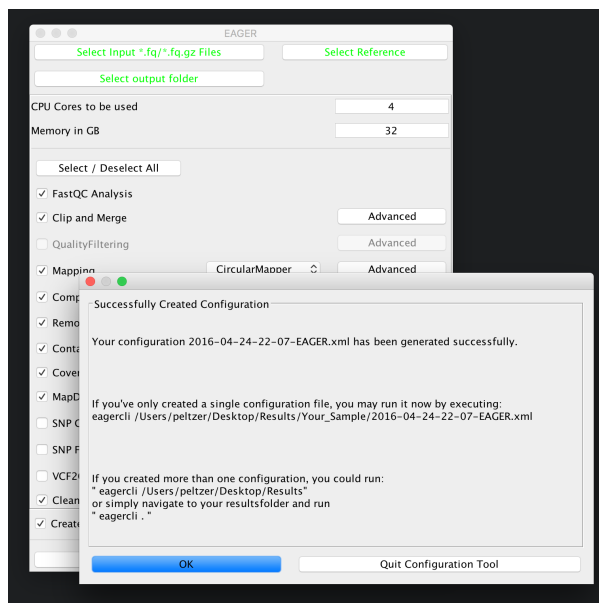
Now that you have selected your input data, your reference genome and the corresponding output folder, you can configure the pipeline more in detail. Start by configuring how many CPU cores and how much random access memory (RAM) can be used by the pipeline for your analysis.

Warning: If you're unsure what to select for CPU cores and/or RAM consumption, you may want to look up your system configuration prior to starting an analysis here. Some processes can fail and make your system unstable when failing if you select too many CPU cores / use too much memory.



Note: The CleanUp module is removing *redundant* data, e.g. intermediate processing results, that are stored in different file formats to save disk space. In almost all cases you can safely keep this module turned on without compromising your analysis results.

After you are done with the configuration of the selected modules, e.g. by clicking on the *Advanced* buttons of the respective tools, you may click on *Generate Config File* on the bottom of the GUI to generate the required pipeline configuration files. A window should open up, telling you that your analysis run has been configured successfully.



6.3.5 Step V: Run the Analysis Pipeline

In order to execute the generated configuration files, the GUI is already giving you a little hint on how to run the execution part of the pipeline. Open up a Terminal application of your choice and then navigate to your folder(s) containing the configuration files (your *result* folder) and run the *eagercli* command to execute the configuration file(s):

```
→ ~ cd /Users/peltzer/Desktop/Results/
→ Results ls -l
total 0
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_A
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_B
drwxr-xr-x  3 peltzer  staff  102 Apr 24 22:07 Sample_C
→ Results eagercli .
```

Note: You don't need to specify the full path to the generated configuration files, e.g. if you specify the *results* folder, EAGER will detect all configuration files automatically and run these sequentially after each other. For some purposes (e.g. a cluster system) you might want to schedule single jobs for each configuration file however, which can be done by specifying the path to the respective configuration files directly.

6.3.6 Step VI: Pick up results!

EAGER creates a CSV based report file in the results folder, which contains statistics for the analysis run. A typical results report looks like this:

Sample number	Sample Name	# of Merged Reads	% Merged Reads	# reads after C&M prior mapping	# mapped reads prior RMDup	# of Dupli
Sample 1/5	Sample_A	5437812	94.78 %	5737174	1023502	
Sample 2/5	Sample_B	12956100	80.5 %	16093580	2659178	
Sample 3/5	Sample_C	32041091	73.61 %	43528407	12782665	

All the output BAM files, VCF files and other important analysis results can be found in the sample specific folders in the results folder.

Note: You can import the results table in CSV format in any compatible sheet calculation software, LibreOffice for example works very well.

7.1 I am missing Feature X for my analysis

You can either contact Alexander Peltzer <alexander.peltzer@uni-tuebingen.de> directly or open a [ticket](#) on Github directly, de

Quick explanation: The image has been updated for example and thus the SSH fingerprint doesn't match anymore with what your local ssh "known_hosts" states. We remove this line and then the image is accepted again.

7.2 I have some BAM files already preprocessed and don't want to map everything again

You can also select BAM files in the EAGER pipeline! Just select the BAM files as input, set the `_same_` reference genome as you used for mapping, select an output folder and *deselect adapter clipping & mapping modules* and you're ready to go! You could for example use EAGER in these cases for the assessment of BAM files, genotyping and duplicate removal, without mapping your preprocessed BAM files again.

7.3 I am using EAGER to reconstruct several genomes simultaneously but it doesn't work

Make sure that you have your individual reference genomes in separate folders and don't use a single folder for all of your references. EAGER relies on generating execution files (named `DONE.<modulname>`) to figure out whether parts of the pipeline have been executed before. If you have two reference genomes in one folder, it will generate indexes for the first reference genome and then find these files, stopping to create an index for the second genome. In order to prevent this, please use different folders for different reference genomes, as mentioned in the file naming pattern section of this documentation.

7.4 I have several samples from the same individual (e.g. pre-screening and a wgs dataset) and would like to combine these

This is not directly supported in the pipeline but there is a possibility to achieve this.

1. Run your preprocessing and mapping steps as usual (up to deduplication) for all your individual samples.
2. Combine these using e.g.

```
samtools merge output.bam input1.bam input2.bam input3.bam ...
```

3. This will create a file called ‘output.bam’ that you can then subsequently load into EAGER (yes, select as input!).
4. Deselect everything up till “Duplicate Removal” as these modules can’t be run on BAM files.
5. Continue with e.g. Genotyping - you can also select to get the proper statistics.

Note: You need to select the same reference genome in the input selection to get proper statistics.

7.5 I have an error and I don’t know what to do

A good start would be having a look at your “EAGER.log” logfile. This contains information about the commands causing potential errors, runtime and other important information. It helps us to verify whats going on even if you contact us directly. In many cases you might already figure out what went wrong with just reading this file!

Note: If you send us an e-mail reporting an error/bug or just because you didn’t find out yourself what might have went wrong, please always include the “EAGER.log” logfile. This is important for us to understand what might have gone wrong.

The EAGER pipeline uses several tools and methods in order to process data. The following list contains all tools and methods with respective links to the corresponding webpages. For licencing information regarding these methods and tools, please see the respective web pages.

- FastQC
- FastXTools
- BWA/BWAMem
- Samtools
- Bowtie 2
- Stampy
- Preseq
- Picard-Tools
- QualiMap
- mapDamage
- Genome Analysis Toolkit (GATK)
- schmutzi
- ANGSD
- BAM2TDF

The EAGER pipeline is available free of charge for academic purposes. The pipeline is available under **GPLv3** (see source code on GitHub, too). For more information see the Docker Image Installation Instructions.

8.1 Important Licencing Information

The GATK is licensed by the Broad Institute and is made available to academic users of the EAGER pipeline described at http://it.inf.uni-tuebingen.de/?page_id=161 **for non-commercial research use only**. The full text of the GATK license is available at <https://www.broadinstitute.org/gatk/about/license.html>. For more information about GATK, please visit the GATK website at <https://www.broadinstitute.org>.

8.2 GATK documentation resources and support

General GATK documentation can be found at on the GATK website at <http://www.broadinstitute.org/gatk/guide/>. Users of this pipeline are welcome to ask GATK-related questions and report problems that are not specific to this pipeline in the GATK forum at <http://gatkforums.broadinstitute.org/gatk>.

If you use EAGER, please cite

- 1. Peltzer; G. Jäger; A. Herbig; S. Seitz; C. Kniep; J. Krause; K. Nieselt: EAGER: efficient ancient genome reconstruction (Genome Biology 2016, 17:60, doi:10.1186/s13059-016-0918-z)

The project URL is:

<https://github.com/apeltzer/eager-gui>

9.1 Tools & Methods

- Andrews, S. (2010). FastQC: A quality control tool for high throughput sequence data. Reference Source.
- Daley, T., & Smith, A. D. (2013). Predicting the molecular complexity of sequencing libraries. *Nature Methods*, 10(4), 325–7. doi:10.1038/nmeth.2375
- Ginolhac, A., Rasmussen, M., Gilbert, M. T. P., Willerslev, E., & Orlando, L. (2011). mapDamage: testing for damage patterns in ancient DNA sequences. *Bioinformatics (Oxford, England)*, 27(15), 2153–5. doi:10.1093/bioinformatics/btr347
- Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, 10(3), R25. doi:10.1186/gb-2009-10-3-r25
- Korneliussen, T. S., Albrechtsen, A., & Nielsen, R. (2014). ANGSD: Analysis of Next Generation Sequencing Data. *BMC Bioinformatics*, 15(1), 356. doi:10.1186/s12859-014-0356-4
- Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv Preprint arXiv:1303.3997*.
- Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 25(14), 1754–60. doi:10.1093/bioinformatics/btp324
- Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., ... Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16), 2078–9. doi:10.1093/bioinformatics/btp352

- Lunter, G., & Goodson, M. (2011). Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Research*, 21(6), 936–9. doi:10.1101/gr.111120.110
- McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., . . . DePristo, M. A. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9), 1297–303. doi:10.1101/gr.107524.110
- Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239), 2.
- Okonechnikov, K., Conesa, A., & García-Alcalde, F. (2016). Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics (Oxford, England)*, 32(2), 292–4. doi:10.1093/bioinformatics/btv566
- Renaud, G., Slon, V., Duggan, A. T., & Kelso, J. (2015). Schmutzi: estimation of contamination and endogenous mitochondrial consensus calling for ancient DNA. *Genome Biology*, 16(1), 224. doi:10.1186/s13059-015-0776-0

CHAPTER 10

Indices and tables

- `genindex`
- `modindex`
- `search`