

Population genomics with snpArcher



**HARVARD
INFORMATICS**

<https://informatics.fas.harvard.edu/>

Workshop December 2025



<https://snparcher.readthedocs.io/>

SNP calling is a cornerstone of population genomics

Individual 1



Individual 2



SNP calling is a cornerstone of population genomics

Individual 1



Where do they differ??

Individual 2



SNP calling is a cornerstone of population genomics

Individual 1



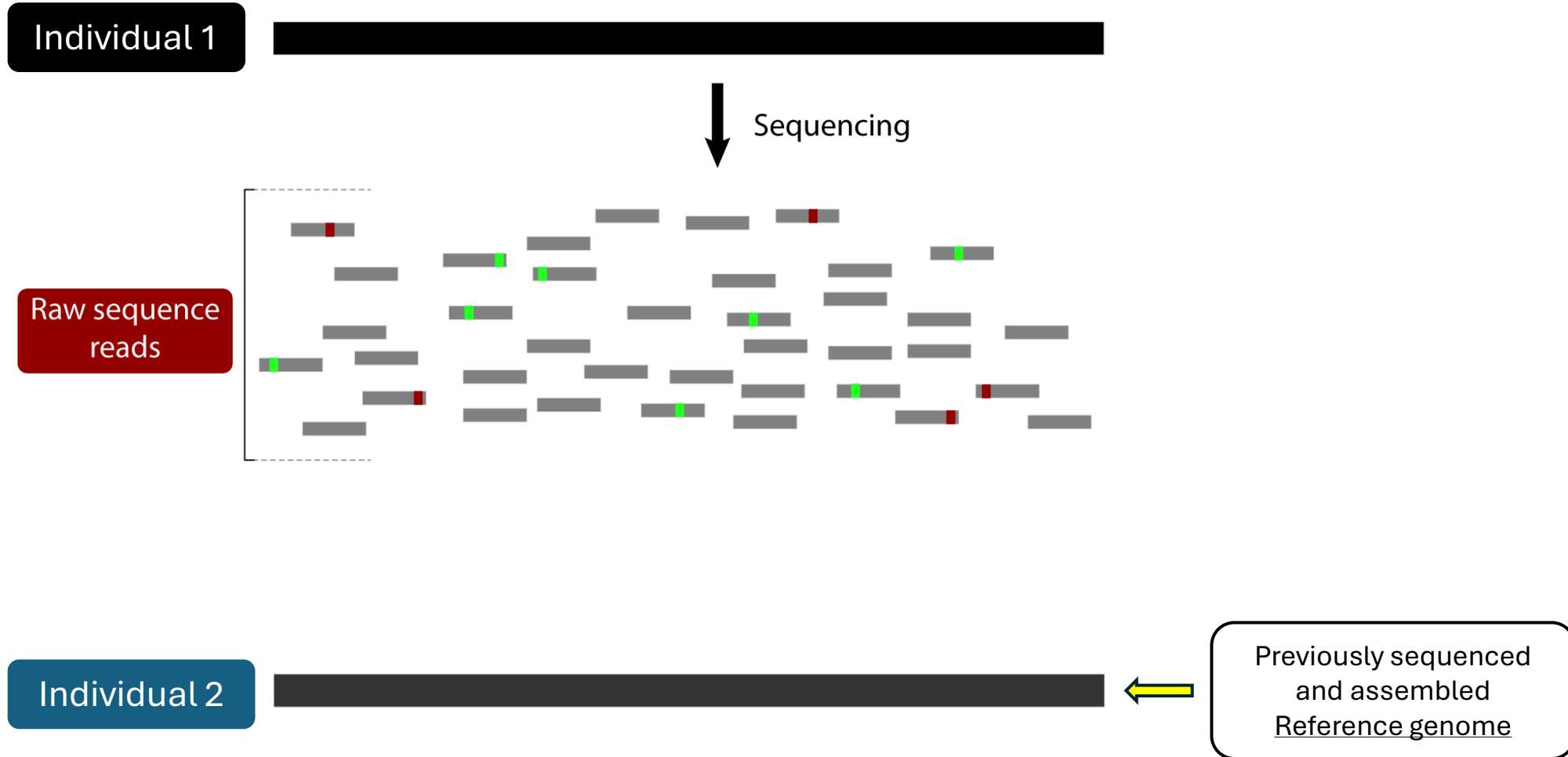
Individual 2



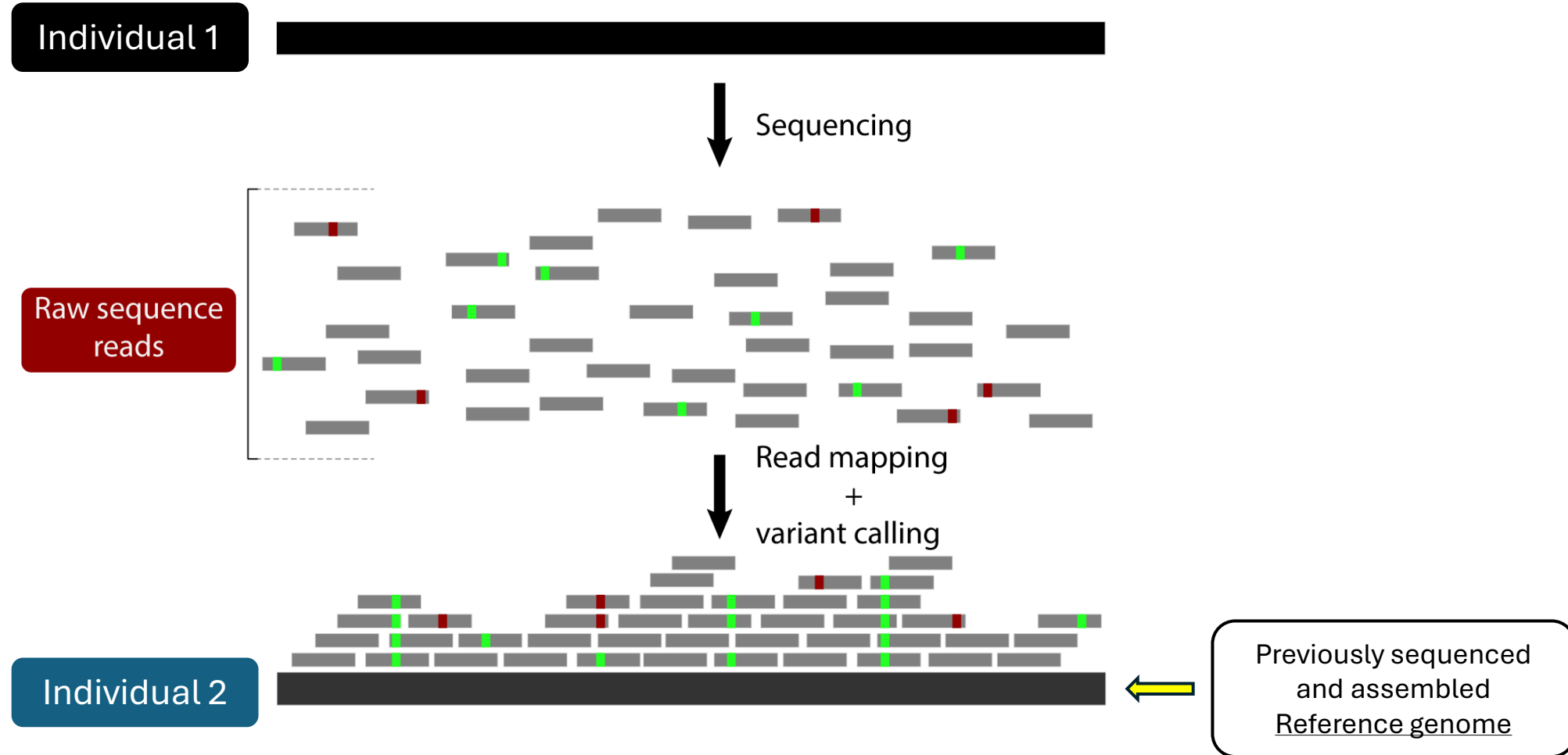
Previously sequenced
and assembled
Reference genome



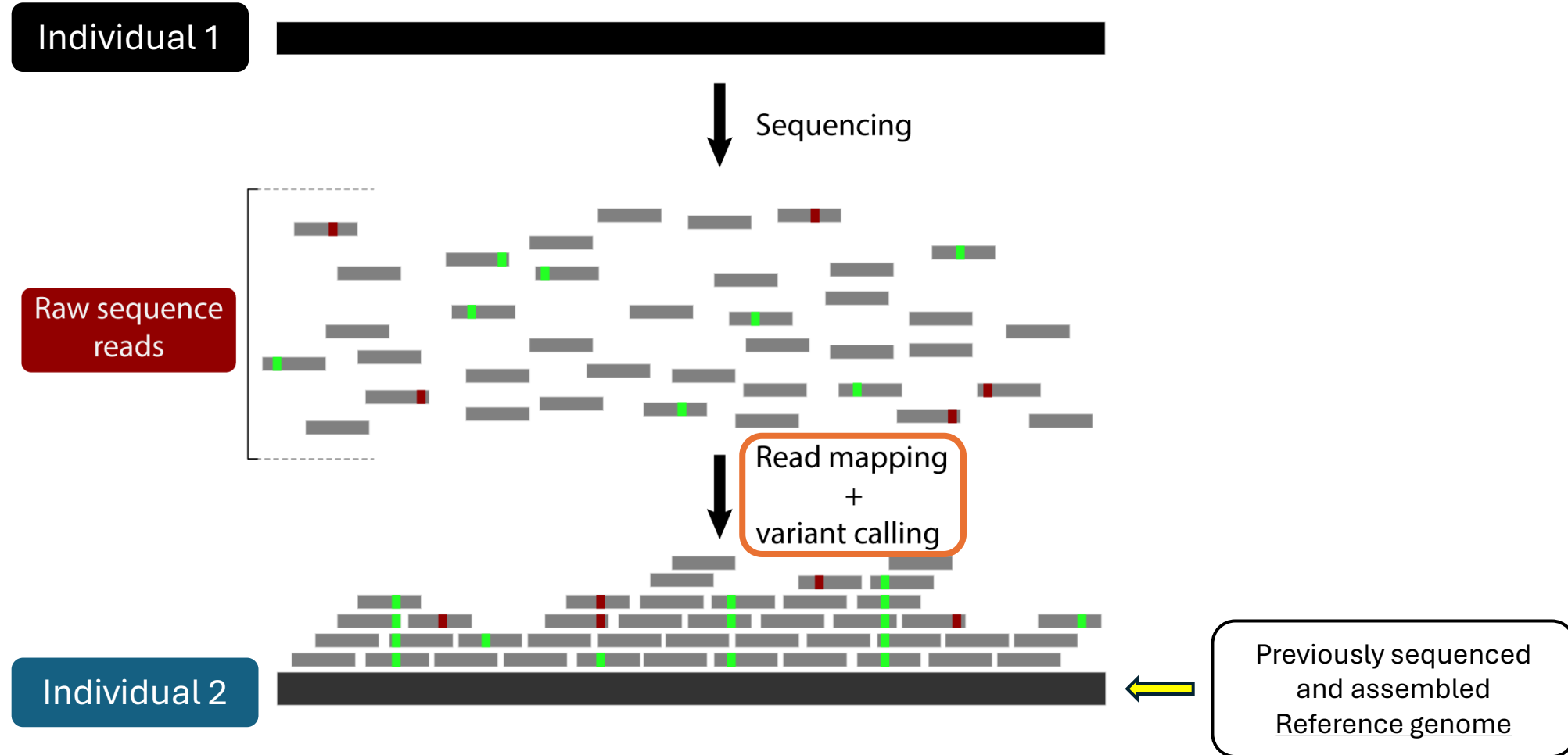
SNP calling is a cornerstone of population genomics



SNP calling is a cornerstone of population genomics

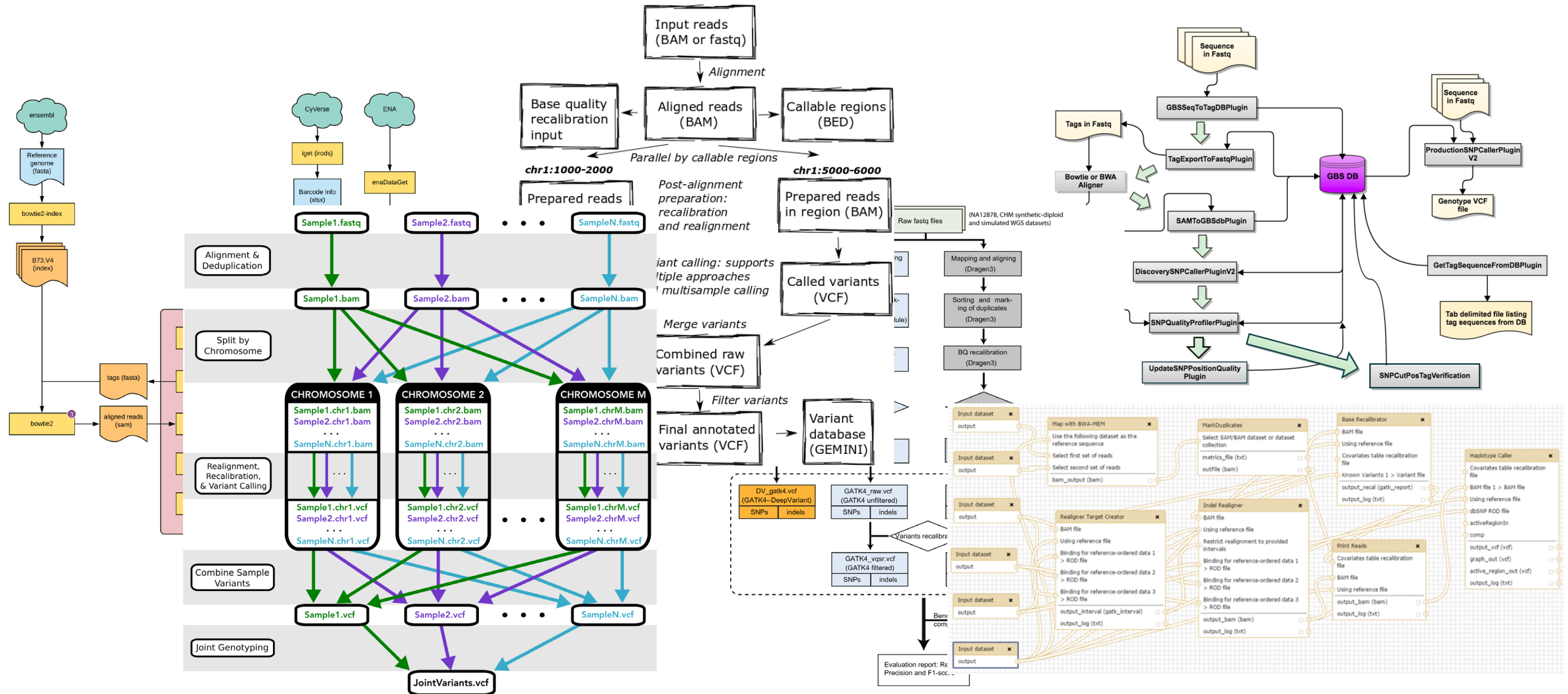


SNP calling is a cornerstone of population genomics

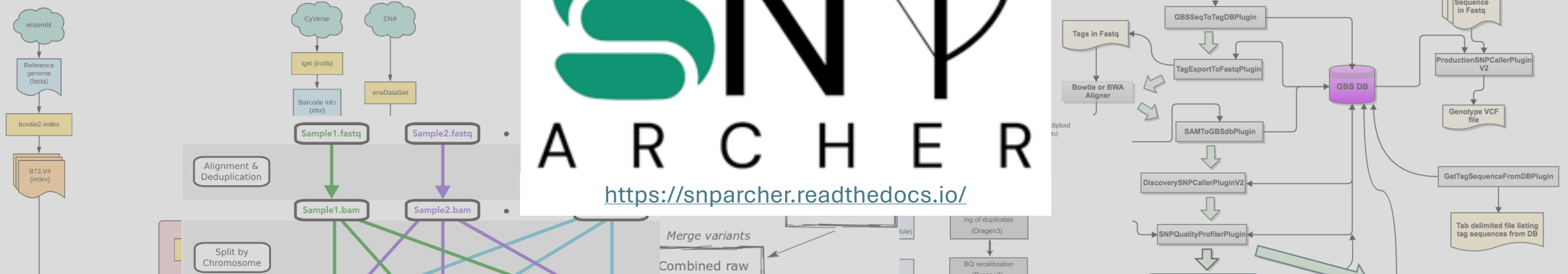


SNP calling is technically challenging

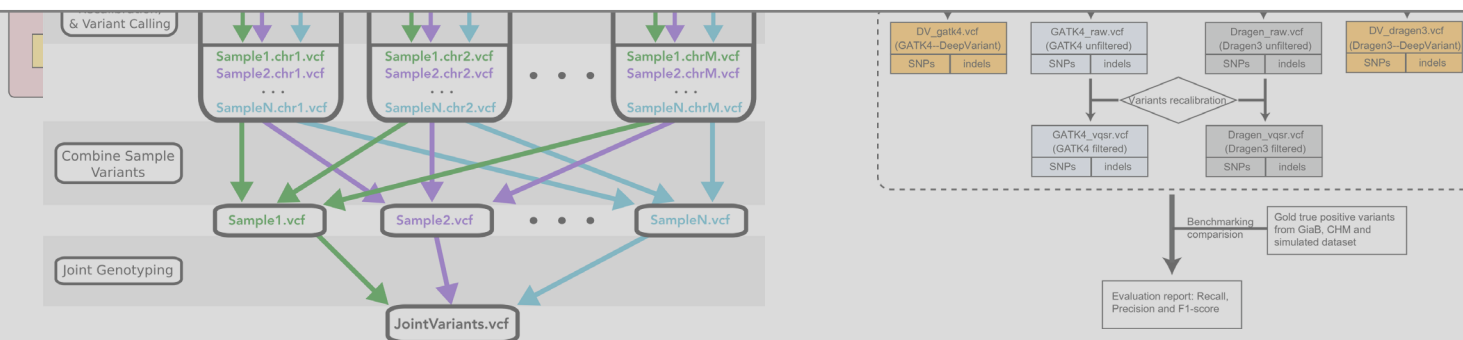
Variant calling overview



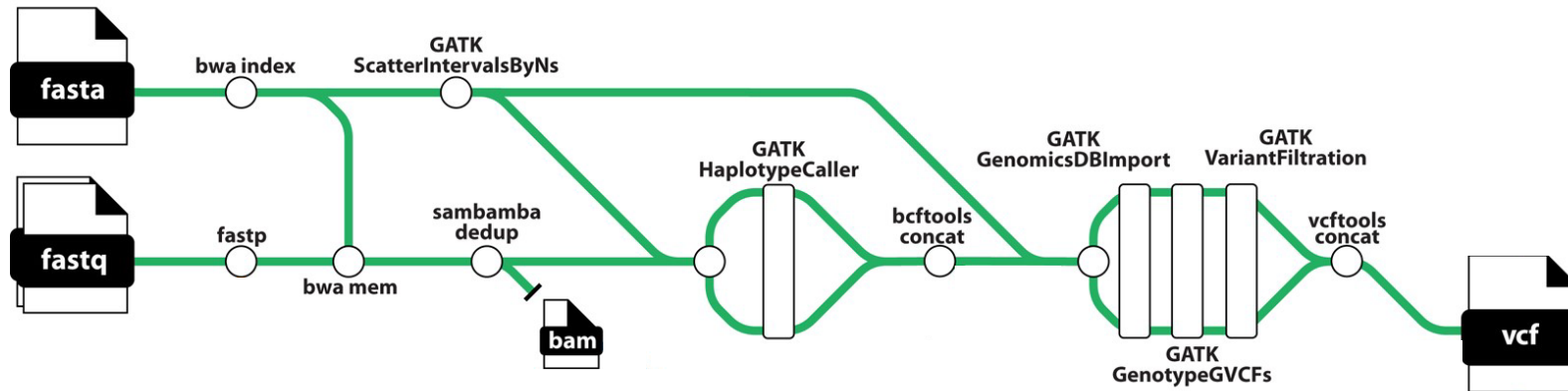
SNP calling is technically challenging



snpArcher is a Snakemake workflow that handles every step of the mapping and variant calling process for multiple samples



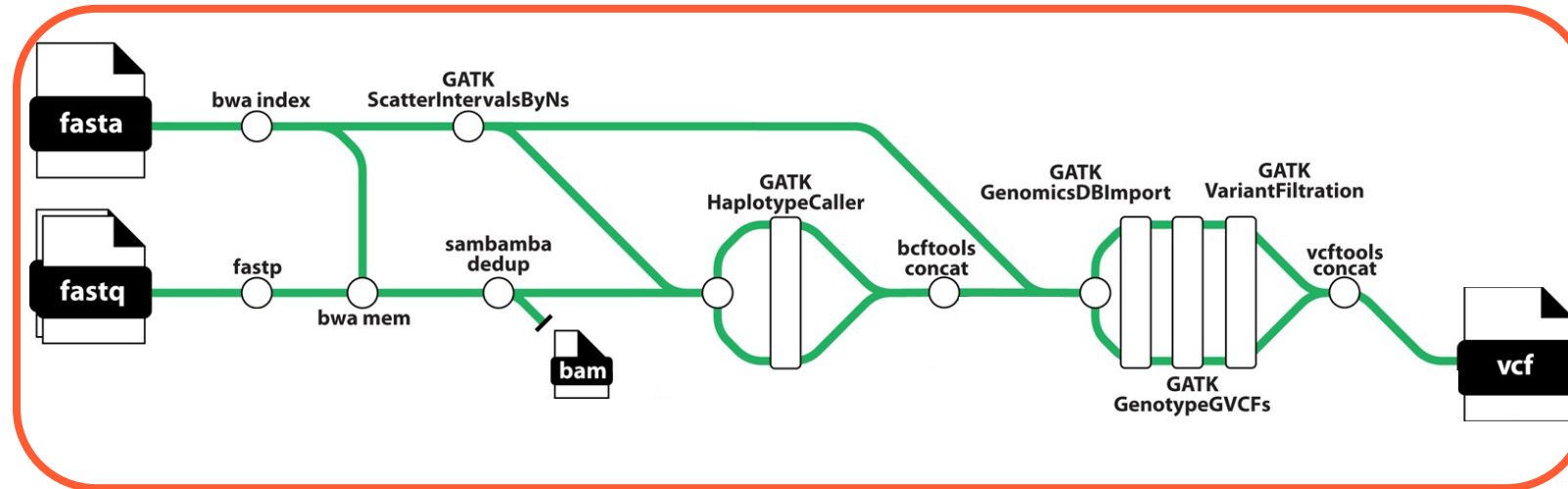
YAVCWI: Yet another variant calling workflow image





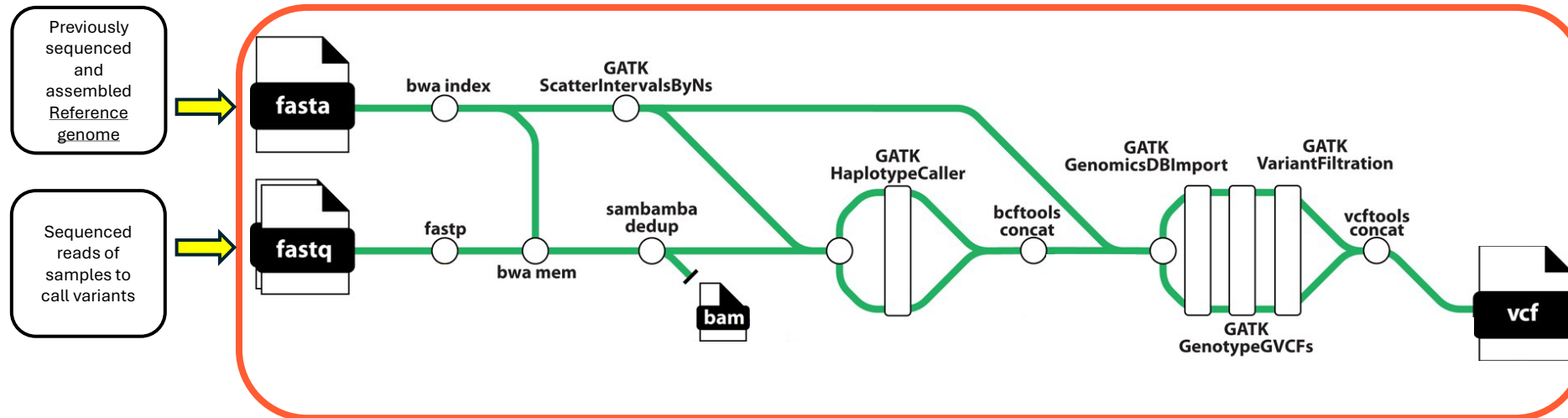
<https://snparcher.readthedocs.io/>

A single command!



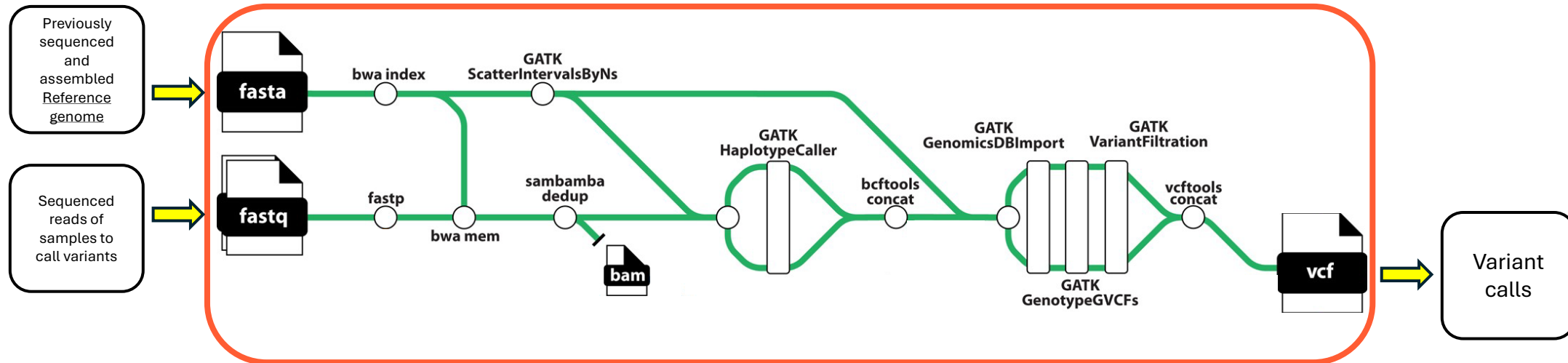
A reference genome and samples with sequenced reads are the only inputs

A single command!



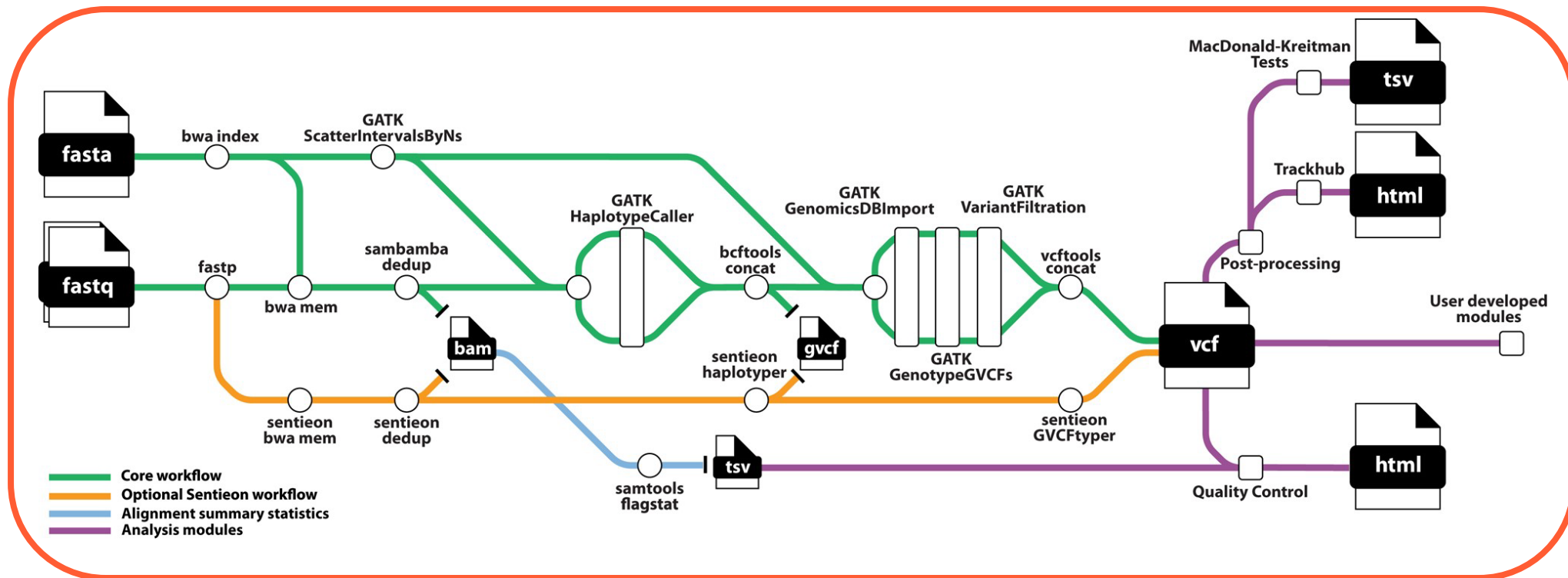
A variant call format (VCF) file is the main output

A single command!



snArcher has a range of other features

A single command!

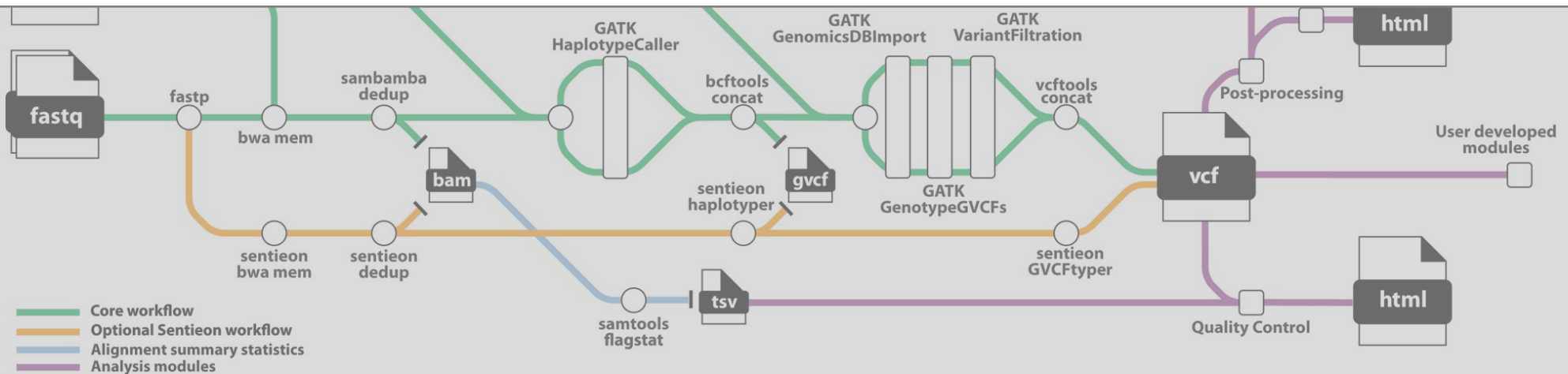




<https://snparcher.readthedocs.io/>

MacDonald-Kreitman
Tests
tsv

What is Snakemake?



Snakemake is a Python-based workflow management language

- Snakemake is based on rules – each rule is a step in the workflow (e.g. read mapping or variant filtering)
 - The output of one rule is the input for the next rule in the workflow
- Wildcards allow rules to be run on multiple files
- Snakemake integrates with SLURM and automatically submits each step in a rule as a single job

Installing Snakemake



Installing Snakemake

1. Install the package manager mamba

<https://github.com/conda-forge/miniforge>

If you already have mamba installed (type **mamba** to check) you should skip this step.

Installing Snakemake

1. Install the package manager mamba

<https://github.com/conda-forge/miniforge>

If you already have mamba installed (type `mamba` to check) you should skip this step.

If you have conda installed or want to use the system module (`module load python`) and don't wish to mess with your setup, you should skip this step. Just use the conda command instead whenever you see mamba.

Installing Snakemake

1. Install the package manager mamba

<https://github.com/conda-forge/miniforge>

1. Scroll to the Install section and copy and paste these commands into your shell
2. Follow the onscreen prompts to accept the license agreement and choose an install location (\$HOME, by default)
3. When prompted to initialize mamba, say yes. You will have to reconnect.

Install

Unix-like platforms (macOS & Linux)

Download the installer using curl or wget or your favorite program and run the script. For eg:

```
curl -L -O "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-$(uname)$(uname -m).sh"
bash Miniforge3-$(uname)$(uname -m).sh
```

or

```
wget "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-$(uname)$(uname -m).sh"
bash Miniforge3-$(uname)$(uname -m).sh
```

Uninstallation



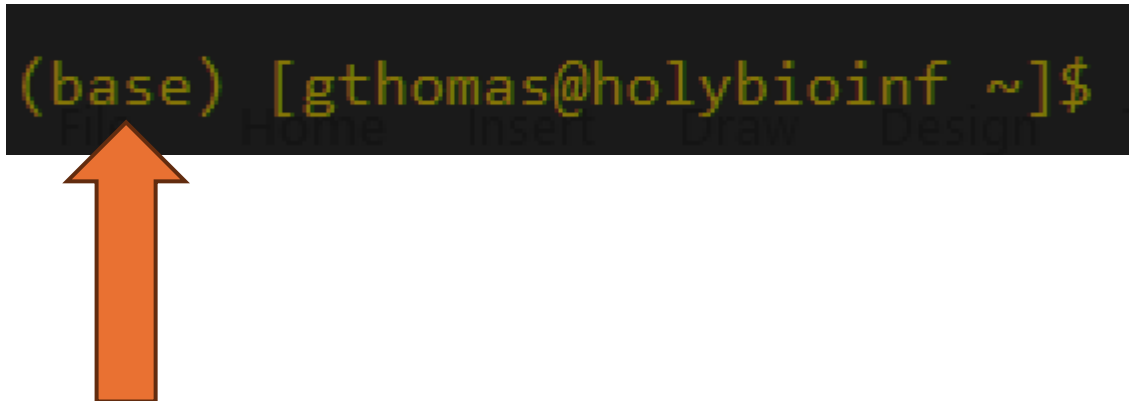
Installing Snakemake

1. Install the package manager mamba

<https://github.com/conda-forge/miniforge>

If the install was successful, you should now see **(base)** appended to your command prompt.

(base) indicates you are in the base conda environment from which you create other environments.



```
(base) [gthomas@holybioinf ~]$
```

Installing Snakemake

2. Setup your channels for bioconda

<https://bioconda.github.io/>

```
conda config --add channels bioconda  
conda config --add channels conda-forge  
conda config --set channel_priority strict
```

Installing Snakemake

3. Create and activate an environment for Snakemake

Create:

```
mamba create -n snakemake-env
```

Activate:

```
mamba activate snakemake-env
```



```
(snakemake-env) [gthomas@holybioinf ~]$ |
```

Installing Snakemake

4. Install the snakemake-minimal and snakemake slurm packages

<https://anaconda.org/bioconda/snakemake-minimal>

<https://anaconda.org/bioconda/snakemake-executor-plugin-slurm>

snakemake-minimal:

```
mamba install bioconda::snakemake-minimal
```

SLURM snakemake plugin:

```
mamba install bioconda::snakemake-executor-plugin-slurm
```


Installing snpArcher



Installing snpArcher

- snpArcher is installed directly from github:

<https://github.com/harvardinformatics/snpArcher/>

1. Make a project folder

```
mkdir my-project/
```

Installing snpArcher

- snpArcher is installed directly from github:

<https://github.com/harvardinformatics/snpArcher/>

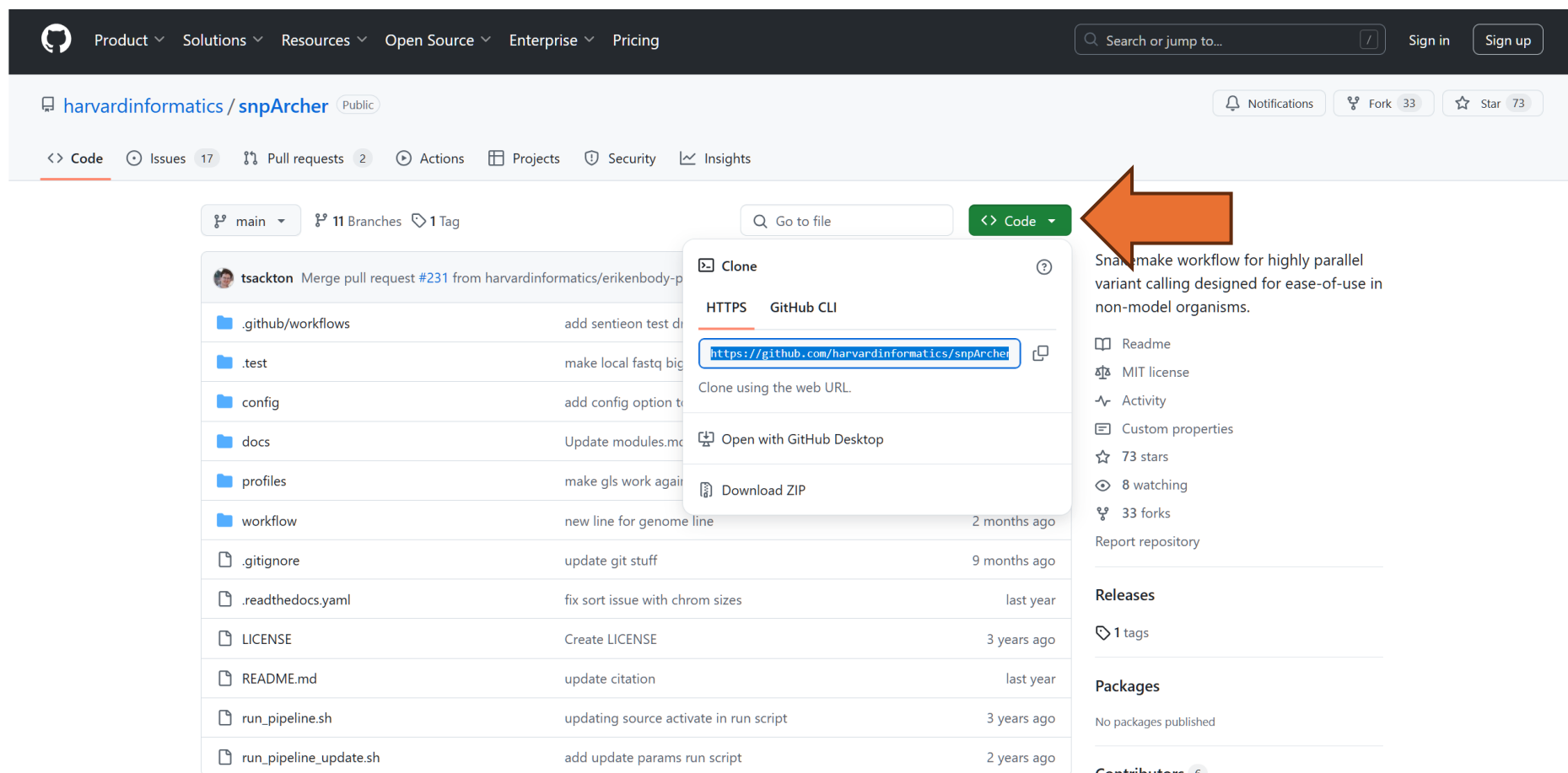
1. Make a project folder

```
mkdir my-project/
```

2. Copy the link from github

3. In your project folder type:

```
git clone <link>
```

A screenshot of the GitHub repository page for 'harvardinformatics/snpArcher'. The page shows the repository name, a 'Code' button, and a 'Clone' dropdown menu. The 'Clone' menu is open, showing options for cloning via HTTPS, GitHub CLI, or GitHub Desktop. The HTTPS URL 'https://github.com/harvardinformatics/snpArcher/' is highlighted. An orange arrow points to the 'Code' button. The repository page also shows a list of files and folders, including '.github/workflows', '.test', 'config', 'docs', 'profiles', 'workflow', '.gitignore', '.readthedocs.yaml', 'LICENSE', 'README.md', 'run_pipeline.sh', and 'run_pipeline_update.sh'. The right sidebar shows repository statistics like stars, forks, and watchers.

Preparing your data for snpArcher



Preparing your data for snpArcher

You will need:

1. A [sample sheet](#) (.csv file)
2. A Snakemake [config file](#) (template provided in github repo)
3. OPTIONAL: To adjust the resources in the Snakemake [profile](#)

Preparing your data for snpArcher

You will need:

1. A [sample sheet](#) (.csv file)
2. A Snakemake [config file](#) (template provided in github repo)
3. OPTIONAL: To adjust the resources in the Snakemake profile

Preparing your data for snpArcher: sample sheet



Data can be used locally or by automatically downloading from NCBI.
You will need:

	NCBI	Local
Reference genome	Assembly accession	Path to FASTA file
For each sample	The SRR run accession for the raw reads	Paths to FASTQ files

Preparing your data for snpArcher: sample sheet



Data can be used locally or by automatically downloading from NCBI.
You will need:

	NCBI	Local
Reference genome	Assembly accession	Path to FASTA file
For each sample	The SRR run accession for the raw reads	Paths to FASTQ files

Other optional information for each sample includes sample type,
and map coordinates (lat and lon).

Preparing your data for snpArcher: sample sheet



Compile your input data into a [sample sheet](#), a CSV file with one sample per row:

Save your samples.csv file
in your project folder

```
samples.csv X
agam-test > config > samples.csv
1 BioSample,LibraryName,refGenome,Run
2 AA0040-C,AA0040-C,GCF_943734735.2,ERR387821
3 AA0041-C,AA0041-C,GCF_943734735.2,ERR387920
4 AA0042-C,AA0042-C,GCF_943734735.2,ERR387921
5 AB0087-C,AB0087-C,GCF_943734735.2,ERR501782
6 AB0088-C,AB0088-C,GCF_943734735.2,ERR332013
7 AB0089-C,AB0089-C,GCF_943734735.2,ERR502048
```

Preparing your data for snpArcher: snakemake **config file**



You will also need to set-up the Snakemake **config file**. A template is provided in the repository you downloaded:

A screenshot of the GitHub web interface for the 'snparchiver/harvardinformatics/snpArcher' repository. The 'Files' tab is selected, showing a directory tree on the left with 'config' highlighted. The 'config.yaml' file is selected in the tree. The main area shows the file's commit history and the code content. A red arrow points to the file name 'config.yaml' in the breadcrumb navigation. The code content is a YAML file with various configuration options for Snakemake, including sample metadata, output prefixes, variant calling intervals, and reference genome paths.

```
1 #####
2 # Variables you need to change
3 #####
4
5 samples: "config/samples.csv" # path to the sample metadata CSV
6 final_prefix: "" # prefix for final output files
7 intervals: True #Set to True if you want to perform variant calling using interval approach.
8 sentieon: False #set to True if you want to use sentieon, False if you want GATK
9 sentieon_license: "" #set to path of sentieon license
10 remote_reads: False # Set True if reads are in a location separate from --default-remote-prefix.
11 remote_reads_prefix: "" # set to google bucket prefix where reads live. FOR SNAKEMAKE 7.X.X ONLY.
12 bigtmp: "" #Set to a path with lots of free space to use for commands that require large amounts of temp space; defaults to system tmpdir if empty
13 cov_filter: True #set to True if you want to include coverage thresholds in the callable sites bed file (default uses mappability only)
14 generate_trackhub: True #Set to true if you want to generate a Genome Browser Trackhub. Dependent on postprocessing module.
15 trackhub_email: ""
16 mark_duplicates: True
17 sort_reads: False
18 #####
19 # Variables you *might* need to change
20 #####
21
22 # Set reference genome here if you would like to you use the same reference genome for all samples in sample sheet. See docs for more info.
```

Preparing your data for snpArcher: snakemake **config file**



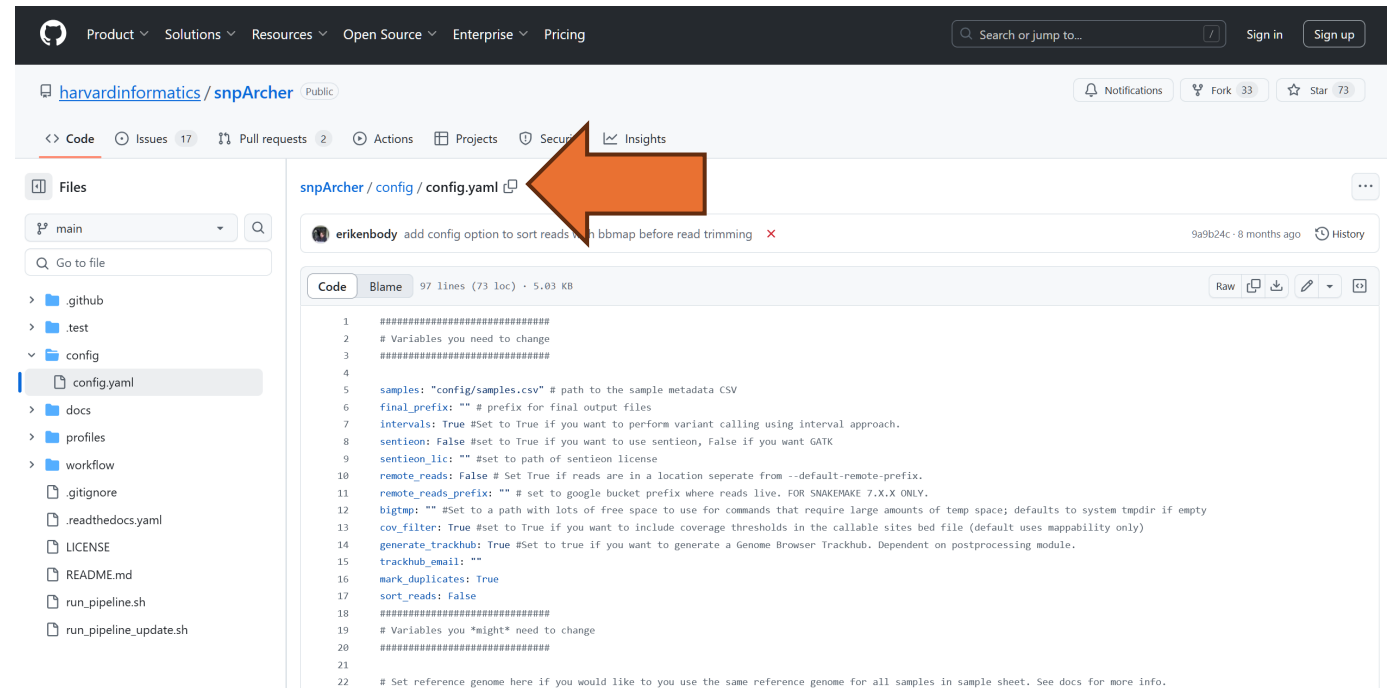
You will also need to set-up the Snakemake **config file**. A template is provided in the repository you downloaded:

1. In your project folder, create a sub-folder called configs:

```
mkdir configs/
```

2. Copy the template **config file**:

```
cp snpArcher/config/config.yaml configs/.
```



Preparing your data for snpArcher: snakemake **config file**



You will also need to set-up the Snakemake **config file**. A template is provided in the repository you downloaded:

1. In your project folder, create a sub-folder called configs:

```
mkdir configs/
```

2. Copy the template **config file**:

```
cp snpArcher/config/config.yaml configs/.
```

3. Edit the relevant parts of the copied **config file**

```
config.yaml X
agm-test > config > config.yaml
1 | #####
2 | # Variables you need to change
3 | #####
4 |
5 | samples: "config/samples.csv" # path to the sample metadata CSV
6 | final_prefix: "agm-test" # prefix for final output files
7 | intervals: True #Set to True if you want to perform variant calling using interval approach.
8 | sentieon: False #set to True if you want to use sentieon, False if you want GATK
9 | sentieon_lic: "" #set to path of sentieon license
10 | remote_reads: False # Set True if reads are in a location separate from --default-remote-prefix.
11 | remote_reads_prefix: "" # set to google bucket prefix where reads live. FOR SNAKEMAKE 7.X.X ONLY.
12 | bigtmp: "/n/holylfs05/LABS/informatics/Users/gthomas/tmp/" #Set to a path with lots of free space to use
13 | cov_filter: True #set to True if you want to include coverage thresholds in the callable sites bed file
14 | generate_trackhub: False #Set to true if you want to generate a Genome Browser Trackhub. Dependent on po
15 | trackhub_email: ""
16 | mark_duplicates: True
17 | sort_reads: False
18 | #####
19 | # Variables you *might* need to change
20 | #####
21 |
22 | # Set reference genome here if you would like to you use the same reference genome for all samples in sa
23 | #refGenome: GCF_000001215.4
24 | #refPath:
25 |
26 | # Interval approach options, only applicable if intervals is True
27 | minNmer: 500 # the minimum Nmer used to split up the genome; e.g. a value of 200 means only Nmers 200 or
28 | num_gvcf_intervals: 50 # The maximum number of intervals to create for GVCf generation. Note: the actual
29 | db_scatter_factor: 0.15 # Scatter factor for calculating number of intervals to create for genomics db g
30 | ploidy: 2 # Ploidy for HaplotypeCaller and Sentieon Haplotype
```

Preparing your data for snpArcher: snakemake **config file**



You will also need to set-up the Snakemake **config file**. A template is provided in the repository you downloaded:

1. In your project folder, create a sub-

```
config.yaml X
agam-test > config > config.yaml
1 | #####
2 | # Variables you need to change
3 | #####
```

You are now ready to run snpArcher!

2. Copy the template **config file**:

```
cp snpArcher/config/config.yaml configs/.
```

3. Edit the relevant parts of the copied **config file**

```
9  sentieon_license: "" #set to path of sentieon license
10 remote_reads: False # Set True if reads are in a location separate from --default-remote-prefix.
11 remote_reads_prefix: "" # set to google bucket prefix where reads live. FOR SNAKEMAKE 7.X.X ONLY.
12 bigtmp: "/n/holylfs05/LABS/informatics/Users/gthomas/tmp/" #Set to a path with lots of free space to use
13 cov_filter: True #set to True if you want to include coverage thresholds in the callable sites bed file
14 generate_trackhub: False #Set to true if you want to generate a Genome Browser Trackhub. Dependent on po
15 trackhub_email: ""
16 mark_duplicates: True
17 sort_reads: False
18 #####
19 # Variables you *might* need to change
20 #####
21
22 # Set reference genome here if you would like to you use the same reference genome for all samples in sa
23 #refGenome: GCF_000001215.4
24 #refPath:
25
26 # Interval approach options, only applicable if intervals is True
27 minNmer: 500 # the minimum Nmer used to split up the genome; e.g. a value of 200 means only Nmers 200 or
28 num_gvcf_intervals: 50 # The maximum number of intervals to create for GVCf generation. Note: the actual
29 db_scatter_factor: 0.15 # Scatter factor for calculating number of intervals to create for genomics db g
30 ploidy: 2 # Ploidy for HaplotypeCaller and Sentieon Haplotyper
```

Running snpArcher



Running snpArcher

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -s snpArcher/workflow/Snakefile --jobs 8 --use-conda --workflow-profile \
snpArcher/profiles/default/ --dryrun
```

Running snpArcher – local execution

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -s snpArcher/workflow/Snakefile --jobs 8 --use-conda --workflow-profile \
snpArcher/profiles/default/ --dryrun
```


Running snpArcher – on cluster

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -e slurm -s snpArcher/workflow/Snakefile --jobs 8 --use-conda \  
--workflow-profile snpArcher/profiles/default/ --dryrun
```

Running snpArcher – on cluster

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -e slurm -s snpArcher/workflow/Snakefile --jobs 8 --use-conda \  
--workflow-profile snpArcher/profiles/default/ --dryrun
```

Running snpArcher

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -e slurm -s snpArcher/workflow/Snakefile --jobs 8 --use-conda \  
--workflow-profile snpArcher/profiles/default/ --dryrun
```

Automatically finds your
config file in
configs/config.yaml

Running snpArcher

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -e slurm -s snpArcher/workflow/Snakefile --jobs 8 --use-conda \  
--workflow-profile snpArcher/profiles/default/ --dryrun
```

Automatically finds your
config file in
configs/config.yaml

Path can also be provided
with --config

Running snpArcher

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -s snpArcher/workflow/Snakefile --jobs 8 --use-conda --workflow-profile \
snpArcher/profiles/default/ --dryrun
```

```
[Fri Dec 6 11:29:32 2024]
rule all:
  input: results/GCF_943734735.2/genomics_db_import/DB_mapfile.txt, results/GCF_943734735.2/agam-test-all-sites_raw.vcf.gz, results/GCF_943734735.2/summary_stats/agam-test-all
34735.2/QC/agam-test-all-sites_qc.html
  jobid: 0
  reason: Input files updated by another job: results/GCF_943734735.2/QC/agam-test-all-sites_qc.html, results/GCF_943734735.2/agam-test-all-sites_raw.vcf.gz
  resources: mem_mb=<TBD>, disk_mb=<TBD>, tmpdir=<TBD>, mem_mb_reduced=<TBD>

Job stats:
job                count
-----
DB2vcf             44
all                1
filterVcfs         44
gvcf2DB            36
qc_admixture        1
qc_check_fai        1
qc_plink            1
qc_qc_plots         1
qc_setup_admixture  1
qc_subsample_snps   1
qc_vcftools_individuals 1
sort_gatherVcfs     1
total              133

Reasons:
(check individual jobs above for details)
input files updated by another job:
  DB2vcf, all, filterVcfs, qc_admixture, qc_check_fai, qc_plink, qc_qc_plots, qc_setup_admixture, qc_subsample_snps, qc_vcftools_individuals, sort_gatherVcfs
output files have to be generated:
  DB2vcf, filterVcfs, gvcf2DB, qc_admixture, qc_check_fai, qc_plink, qc_qc_plots, qc_setup_admixture, qc_subsample_snps, qc_vcftools_individuals, sort_gatherVcfs

This was a dry-run (flag -n). The order of jobs does not reflect the order of execution.
```

Running snpArcher

From your project folder and with your Snakemake environment activated, snpArcher can be run with a single command:

```
snakemake -p -s snpArcher/workflow/Snakefile --jobs 8 --use-conda --workflow-profile \
snpArcher/profiles/default/ --dryrun
```

```
[Fri Dec 6 11:29:32 2024]
rule all:
  input: results/GCF_943734735.2/genomics_db_import/DB_mapfile.txt, results/GCF_943734735.2/agam-test-all-sites_raw.vcf.gz, results/GCF_943734735.2/summary_stats/agam-test-all
34735.2/QC/agam-test-all-sites_qc.html
  jobid: 0
  reason: Input files updated by another job: results/GCF_943734735.2/QC/agam-test-all-sites_qc.html, results/GCF_943734735.2/agam-test-all-sites_raw.vcf.gz
  resources: mem_mb=<TBD>, disk_mb=<TBD>, tmpdir=<TBD>, mem_mb_reduced=<TBD>

Job stats:
job                count
-----
DB2vcf             44
all                1
filterVcfs         44
gvcf2DB            36
qc_admixture        1
qc_check_fai        1
qc_plink            1
qc_qc_plots         1
qc_setup_admixture  1
qc_subsample_snps   1
qc_vcftools_individuals 1
sort_gatherVcfs     1
total              133

Reasons:
(check individual jobs above for details)
input files updated by another job:
DB2vcf, all, filterVcfs, qc_admixture, qc_check_fai, qc_plink, qc_qc_plots, qc_setup_admixture, qc_subsample_snps, qc_vcftools_individuals, sort_gatherVcfs
output files have to be generated:
DB2vcf, filterVcfs, gvcf2DB, qc_admixture, qc_check_fai, qc_plink, qc_qc_plots, qc_setup_admixture, qc_subsample_snps, qc_vcftools_individuals, sort_gatherVcfs

This was a dry-run (flag -n). The order of jobs does not reflect the order of execution.
```

This didn't actually run anything!

Always do a --dryrun first!

Running snpArcher

When actually running jobs, you will need to ensure that the main snakemake process is persistent even if you disconnect from the server.

1. Submit the snakemake command itself as a SLURM job
2. Use nohup
3. Use a terminal multiplexer (e.g. screen or tmux)

Running snpArcher

When actually running jobs, you will need to ensure that the main snakemake process is persistent even if you disconnect from the server.

1. Submit the snakemake command itself as a SLURM job
2. Use `nohup`
3. Use a terminal multiplexer (e.g. `screen` or `tmux`)

Running snpArcher

When your dryrun completes without errors, you have a persistent connection, and you are ready to start submitting jobs:

```
snakemake -p -e slurm -s snpArcher/workflow/Snakefile --jobs 8 --use-conda \  
--workflow-profile snpArcher/profiles/default/
```

Running snpArcher


Test data is provided if you'd like to quickly ensure everything works before you run your own data:

```
Snakemake -e slurm -d .test/ecoli --jobs 1 --use-conda --workflow-profile /  
workflow-profiles/default/ --dryrun
```

Running snpArcher

Test data is provided if you'd like to quickly ensure everything works before you run your own data:

```
Snakemake -e slurm -d .test/ecoli --jobs 1 --use-conda --workflow-profile /  
workflow-profiles/default/ --dryrun
```

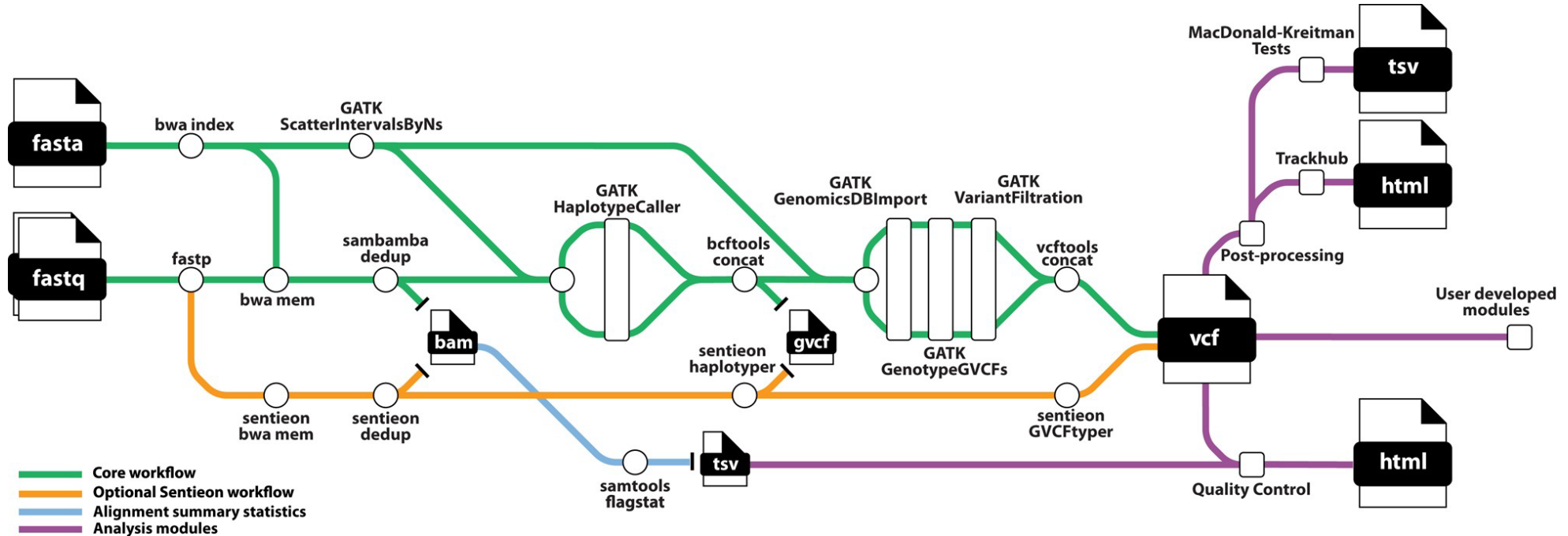
An orange arrow pointing upwards from the blue box to the command line.

Remember the --dryrun option! Remove it if you want to actually run the pipeline on the test data

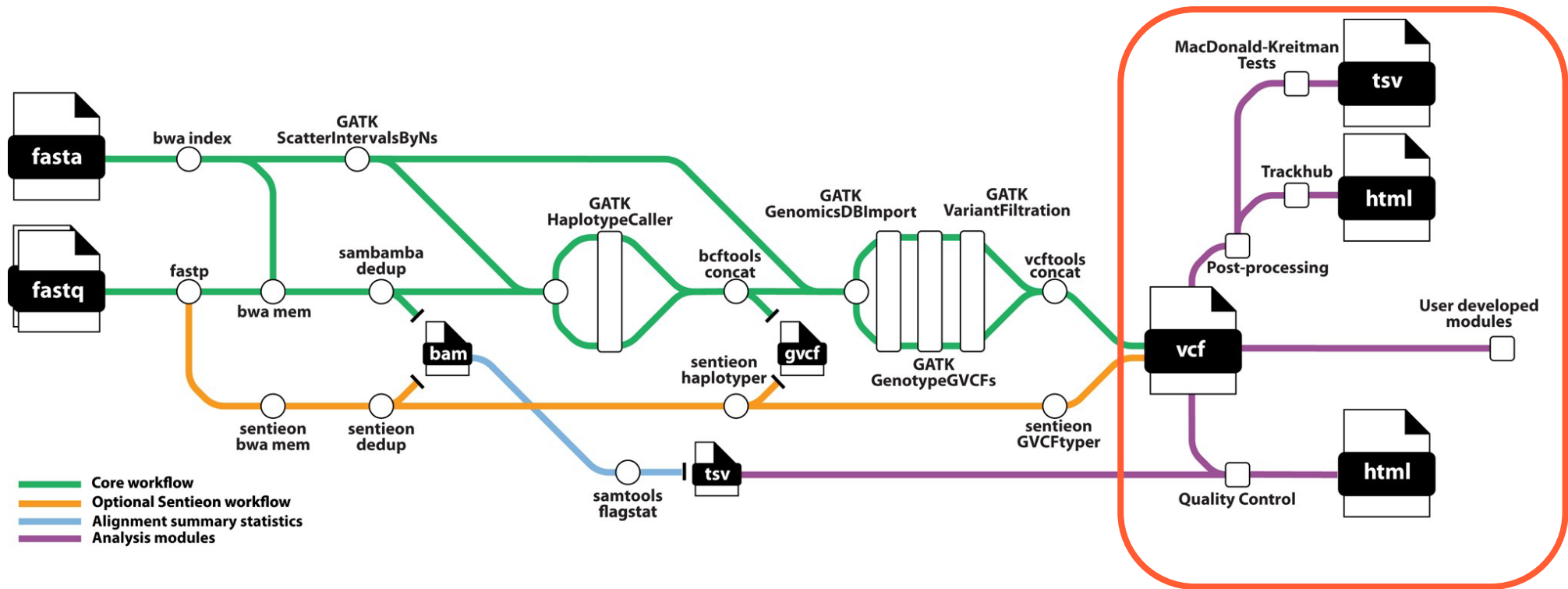
snpArcher output and modules



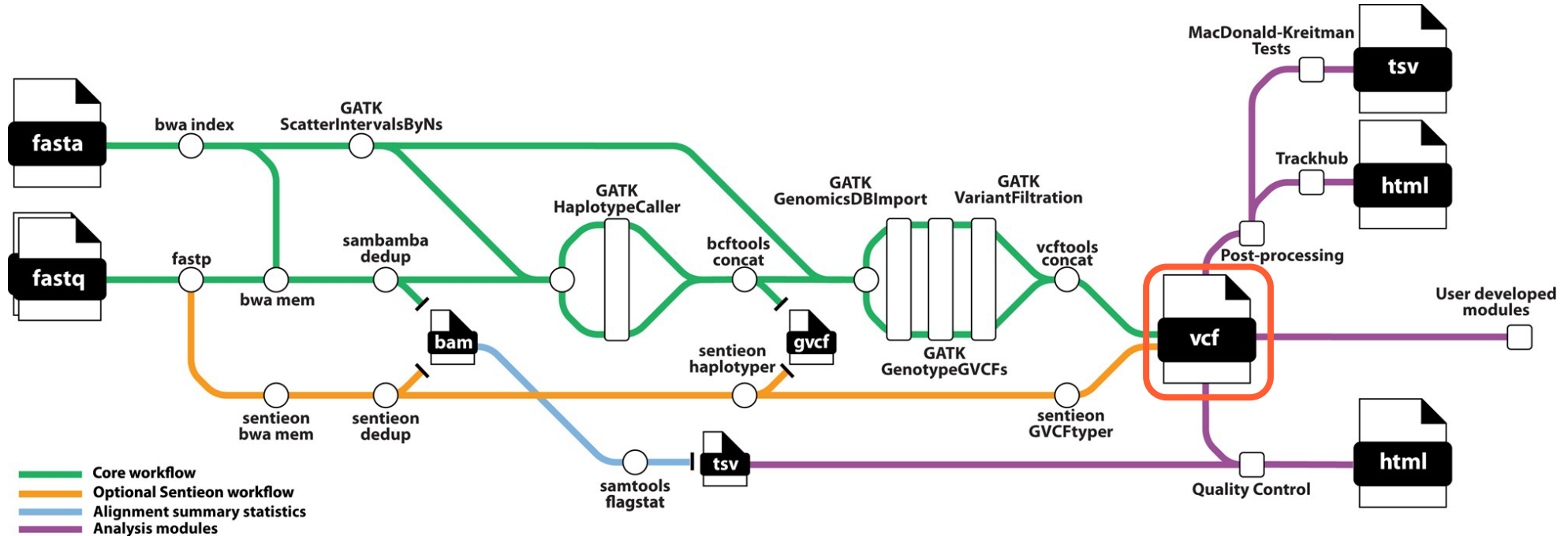
snpArcher has a range of other modules



snpArcher has a range of other modules



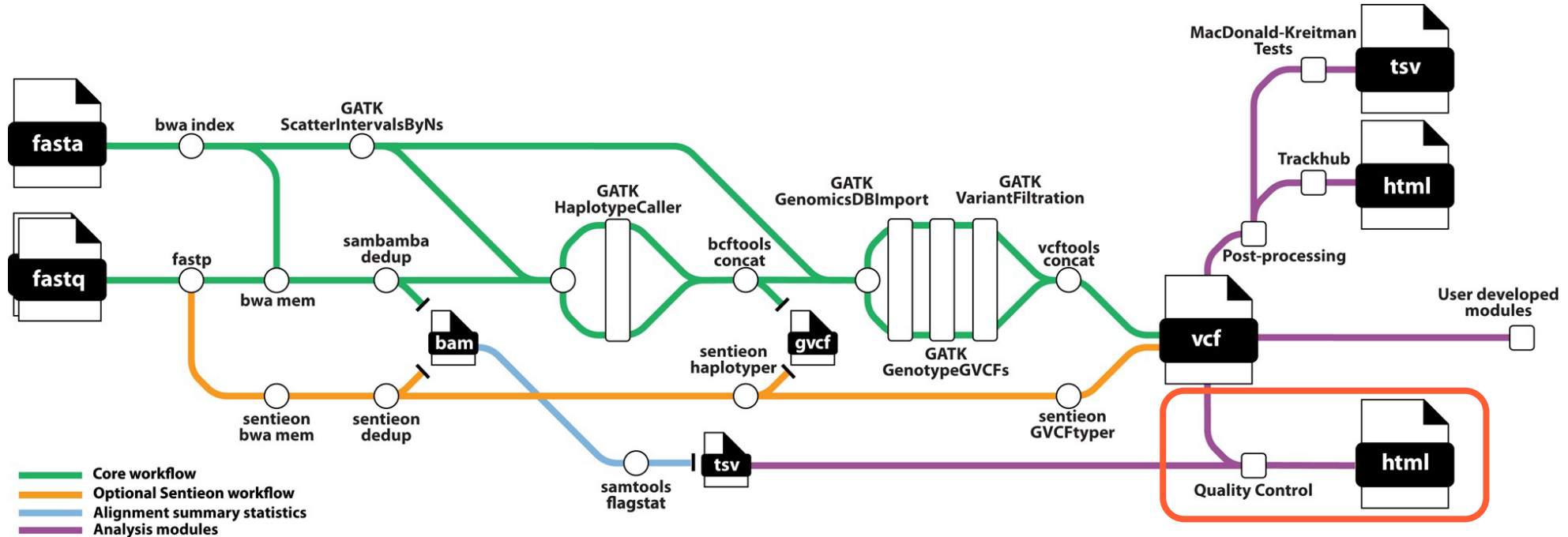
snpArcher's main output is a VCF file



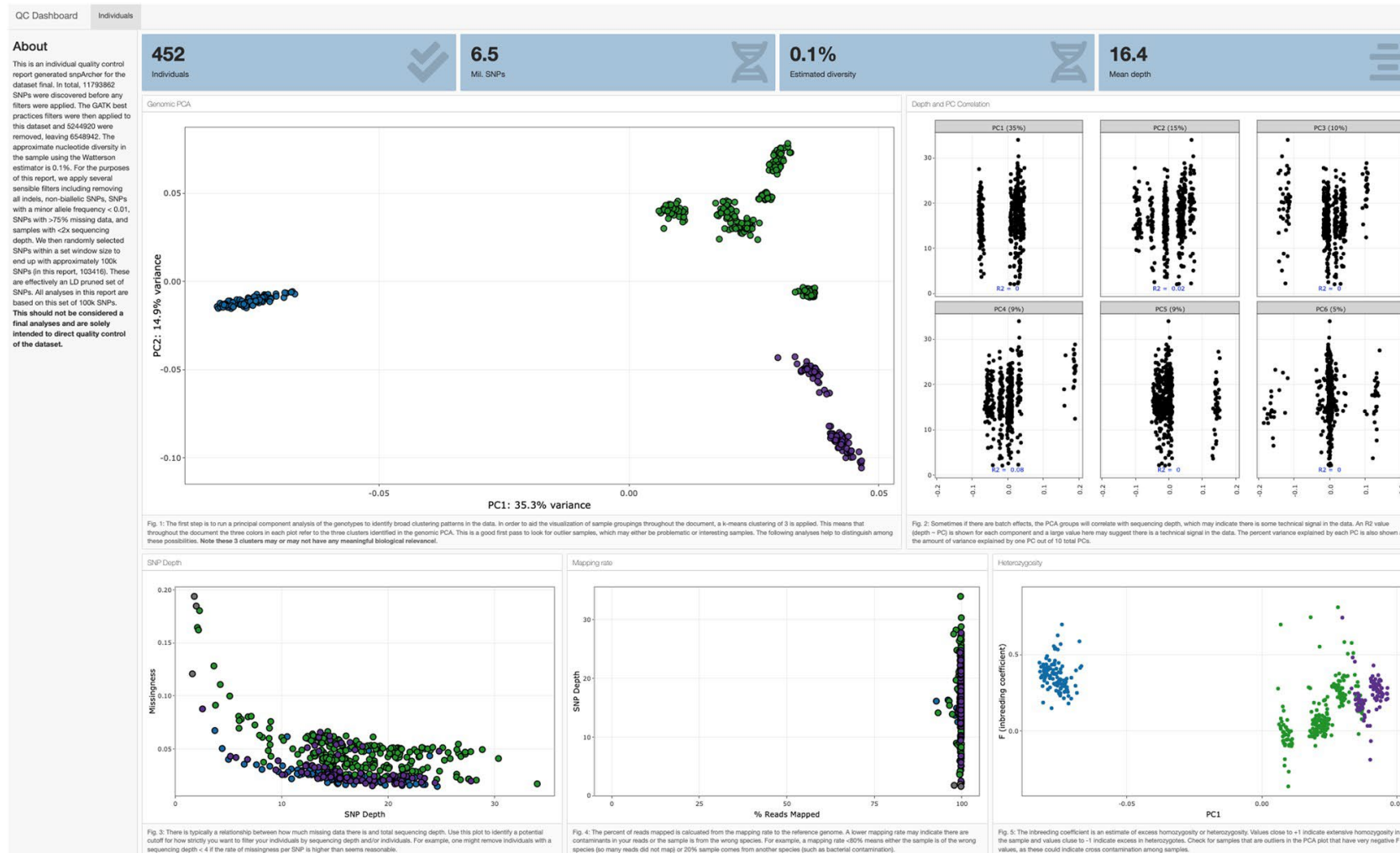
snpArcher's main output is a VCF file

```
##fileformat=VCFv4.2
##fileDate=20151002
##source=callMomV0.2
##reference=gi|251831106|ref|NC_012920.1| Homo sapiens mitochondrion, complete genome
##contig=<ID=MT,length=16569,assembly=b37>
##INFO=<ID=VT,Number=.,Type=String,Description="Alternate allele type. S=SNP, M=MNP, I=Indel">
##INFO=<ID=AC,Number=.,Type=Integer,Description="Alternate allele counts, comma delimited when multiple">
##FILTER=<ID=fa,Description="Genotypes called from fasta file">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT HG00096 HG00097 HG00099 HG00100 HG00101 HG00102 HG00103 HG00105 HG00106 HG00107
MT 10 . T C 100 fa VT=S;AC=3 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 16 . A T 100 fa VT=S;AC=3 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 26 . C T 100 fa VT=S;AC=3 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 35 . G A 100 fa VT=S;AC=2 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 40 . TC CT 100 fa VT=M;AC=1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 41 . C T 100 fa VT=S;AC=4 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 42 . TCC CCC,T 100 fa VT=S,I;AC=1,1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 46 . T C 100 fa VT=S;AC=1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 47 . G A 100 fa VT=S;AC=1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 52 . TGG CAA 100 fa VT=M;AC=1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 55 . TATTTT T,CATTTT,AATTTT,TTT,TTTTT 100 fa VT=I,S,S,I,I;AC=5,3,2,1,1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 57 . T C 100 fa VT=S;AC=3 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 58 . TTT T 100 fa VT=I;AC=4 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MT 59 . T A 100 fa VT=S;AC=1 GT 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

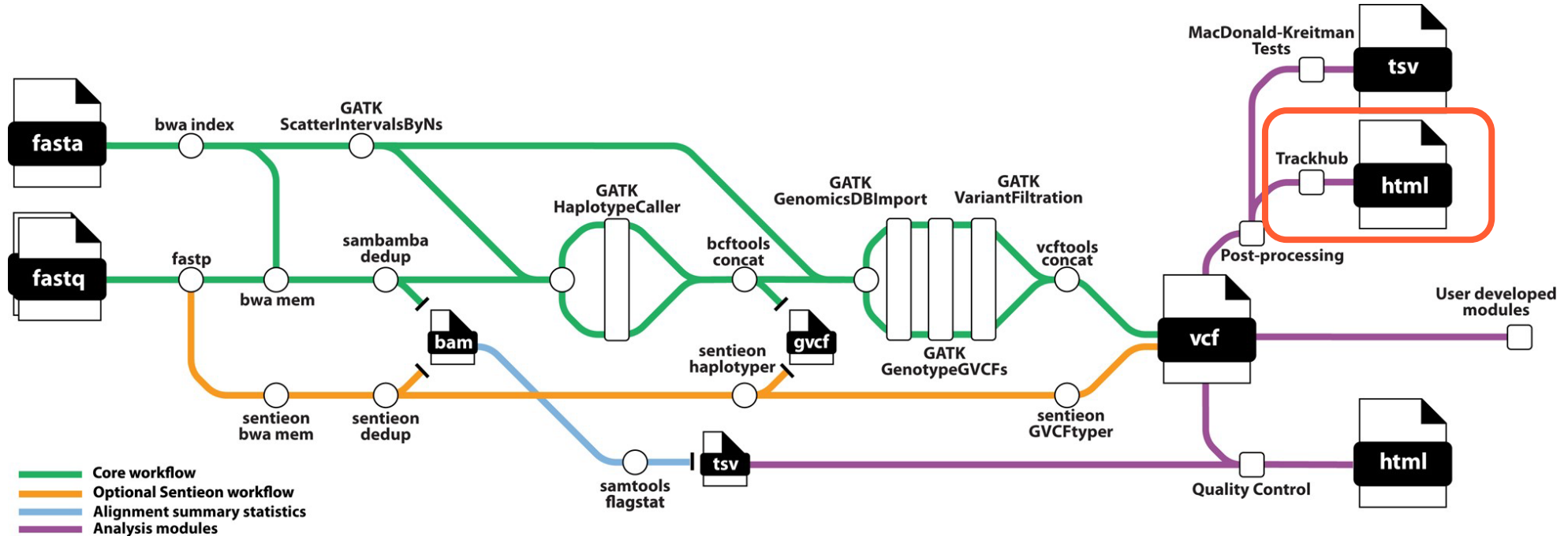

Summary statistics are provided as an HTML page



Summary statistics are provided as an HTML page



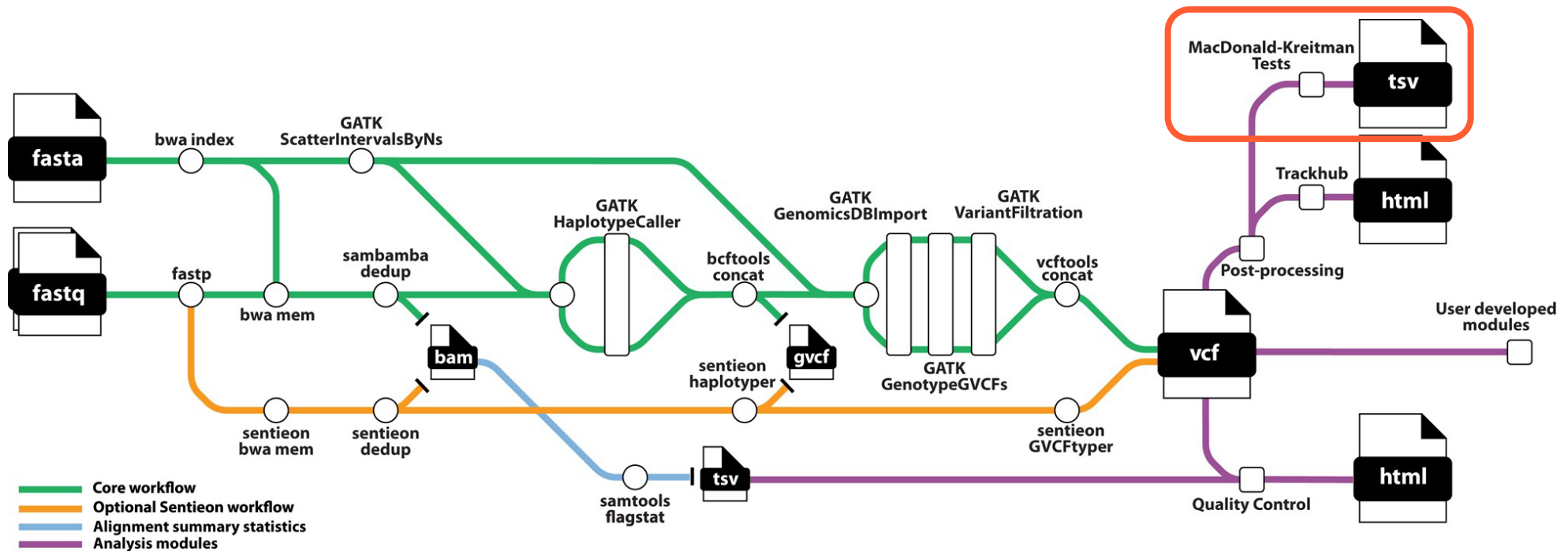
Trackhubs can be generated to visualize where SNPs occur



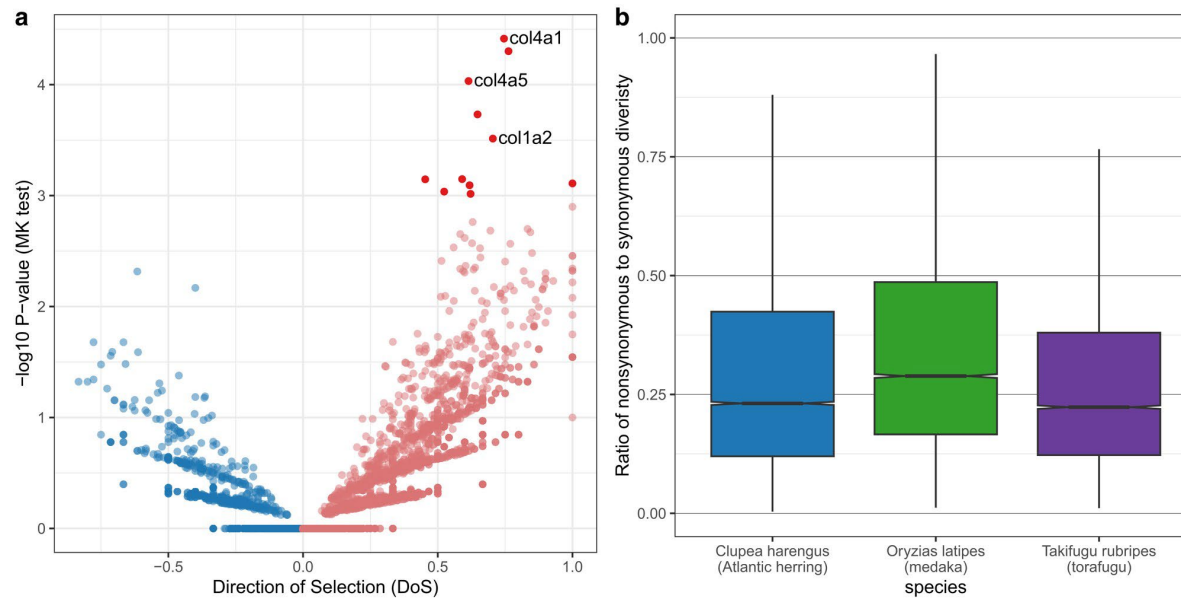
Trackhubs can be generated to visualize where SNPs occur



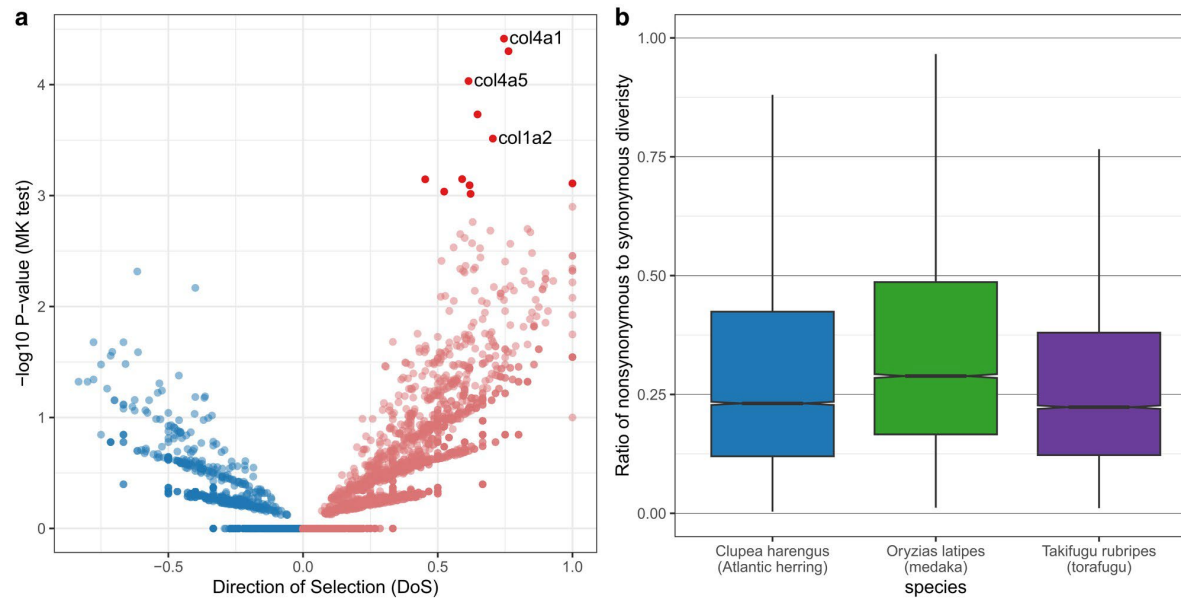
MK tests can be performed to test for selection



MK tests can be performed to test for selection



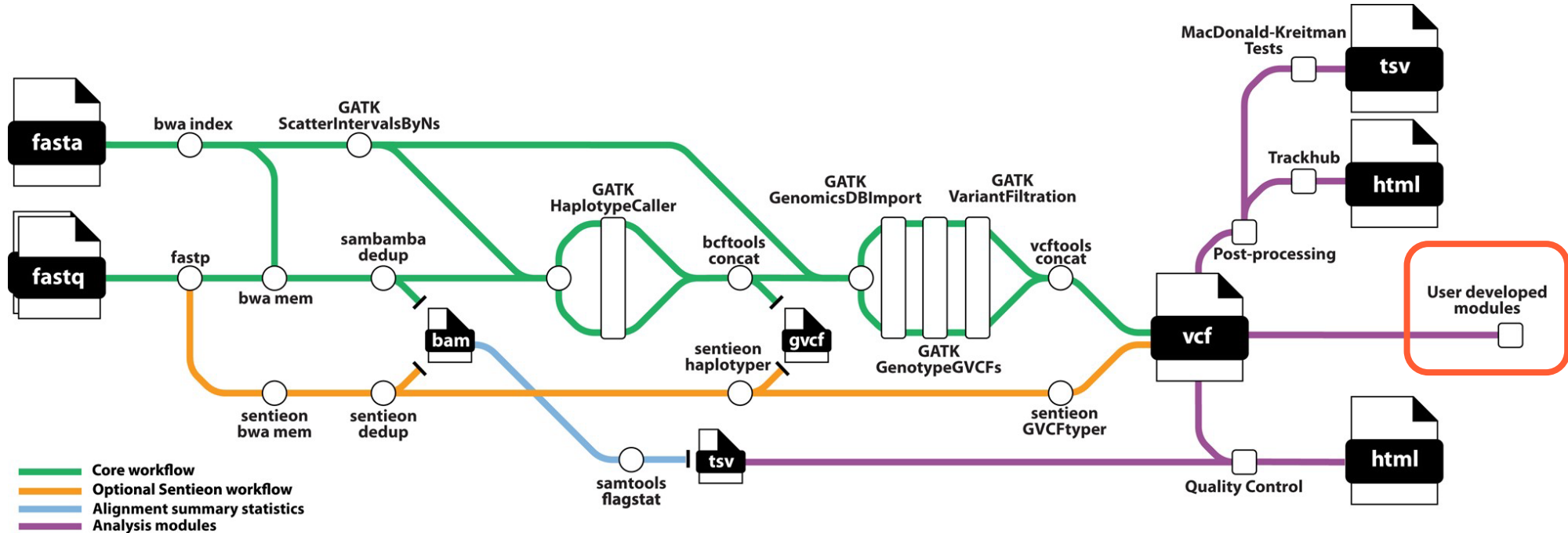
MK tests can be performed to test for selection



<https://github.com/harvardinformatics/degenotate>

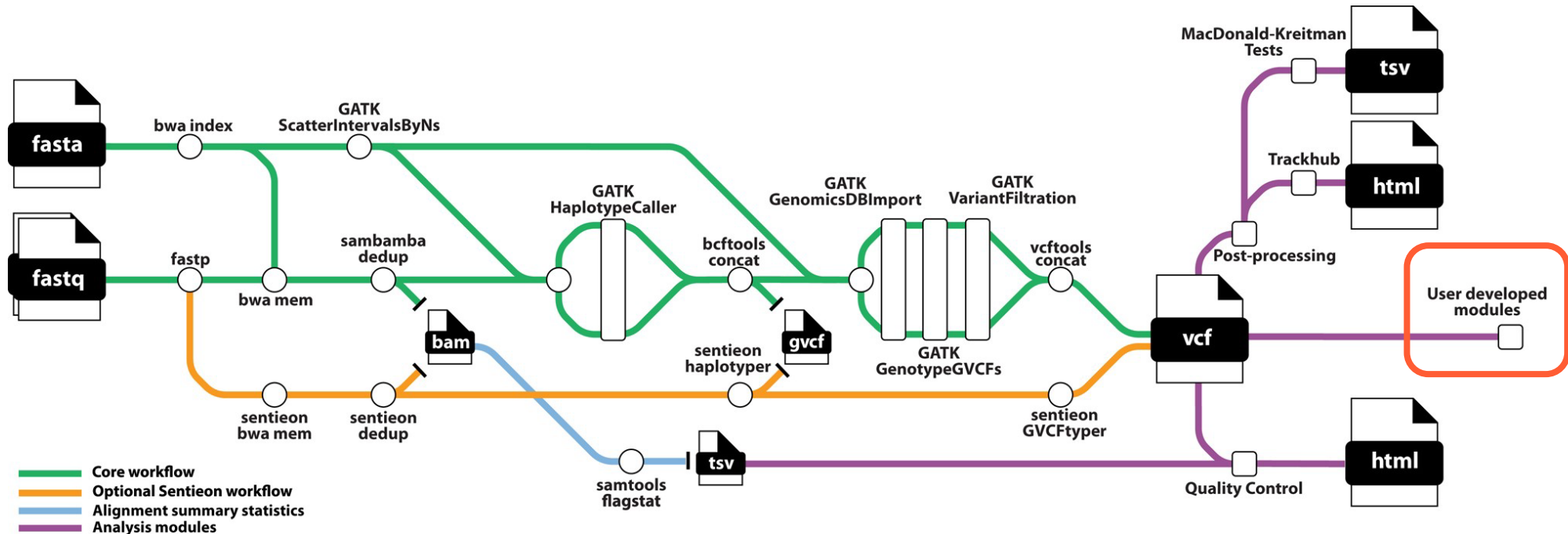
A (user-developed)
module!

Users can develop their own modules and integrate them into snpArcher



Users can develop their own modules and integrate them into snpArcher

Module to infer population size: <https://github.com/tforest/popsiz>



Thanks for your time!



Links:

snpArcher repository.....<https://github.com/harvardinformatics/snpArcher>

snpArcher documentation...<https://snparcher.readthedocs.io/>

snpArcher paper.....<https://doi.org/10.1093/molbev/msad270>

Informatics contact.....<https://informatics.fas.harvard.edu/contact/>