MosiacHunter User Guide

Version 1.0

July, 21th, 2015

http://mosaichunter.cbi.pku.edu.cn/

mosaichunter@mail.cbi.pku.edu.cn

August Yue Huang, Adam Yongxin Ye, Zheng Zhang, Yanmei Dou

Center for Bioinformatics, Peking University

Contents

I Program

MosaicHunter is a bioinformatic command-line tool designed to call post-zygotic single nucleotide mosaicism (SNM) sites from mapped whole-genome or whole-exome sequencing data of unpaired, trio, or paired samples. It implements and wraps a Bayesian genotyper and several stringent filters in a java framework.

MosaicHunter is developed and implemented by August Yue Huang, Adam Yongxin Ye, Zheng Zhang, Yanmei Dou in the Center for Bioinformatics at Peking University, with helps from many other colleagues.

I.1 Getting Started

I.1.a Installation & quick-start

Prerequired tools

java (>=1.7) (https://java.com/en/)

UCSC blat (http://genome.ucsc.edu/FAQ/FAQblat.html#blat3)

Note you will need to include the directory containing runnable blat in your PATH. (or set the parameter `blat_path` for MisalignedReadsFilter)

Suggested tools

git (https://git-scm.com/downloads)

ant (https://ant.apache.org/bindownload.cgi)

samtools (http://sourceforge.net/projects/samtools/files/)

Picard (http://broadinstitute.github.io/picard/)

GATK (https://www.broadinstitute.org/gatk/download/auth?package=GATK)

Download MosaicHunter program and required resource files

Clone the MosaicHunter repository

```
cd your_path
```

```
git clone https://github.com/zzhang526/MosaicHunter.git
```

Some large required resource files are uploaded to our server, and can be accessed by URL:

http://soms.nibs.ac.cn:6235/glk/human_g1k_v37.fasta

http://soms.nibs.ac.cn:6235/dbsnp/dbsnp_137.b37.tsv

You also need to download these files, and move them into `your_path/resources/` directory.

Quick usage of a pre-compiled binary-code release

In order to make it easy to install MosaicHunter, we provide the pre-compiled release, which is in the unpacked `build/` directory. To use it, simply type

```
cd your_path/MosaicHunter
```

```
java -jar build/mosaichunter.jar
```

Build MosaicHunter from source

We also provide the source code of MosaicHunter. If you want to build it from source, you need to install ant first (http://ant.apache.org/bindownload.cgi), then type

```
cd your_path/MosaicHunter
```

```
ant
```

The compiled `mosaichunter.jar` file will overwrite the pre-compiled one in `build/` directory, and you can run MosaicHunter by

```
java -jar build/mosaichunter.jar
```

I.1.b Preparing your reference

MosaicHunter just requires a fasta file (.fasta, .fa) for your reference. It is better to make sure that the config order in the reference file is the same to the order in your .bam file(s) and also .bed file(s).

Reads mapped to any contigs not appeared in the reference file will be ignored.

When running MosaicHunter, you need to set the top parameter `reference_file`.

I.1.c Preparing your reads

MosaicHunter currently accepts mapped DNA-seq reads. We recommend bwa for mapping and GATK for further processing (Picard-SortSam, Picard-MarkDuplicates, GATK-IndelRealignment, GATK-BaseRecalibration). These preprocessing steps are important to reduce the false positives in mosaic calling.

MosaicHunter uses the sorted and indexed bam file(s) to call mosaic sites, as set by the top parameter `input_file` to the path of your major .bam file.

For trio mode, you also need to specify `father_bam_file`, `mother_bam_file`; and for paired mode, `control_bam_file` are required to be specified.

In addition, we suggest to only keep proper-mapped reads (for paired-end reads; with flag 0x2), and drop reads with NM > 4, to make the input cleaner. The command to achieve it can be like

```
samtools view -h -f 0x2 input.bam | perl -ne 'print if (/^@/||(/NM:i:(\d+)/&&$1<=4))' | samtools view -Sb - >cleaner.bam
```

I.1.d Prepare sample-specific files

To remove artifacts due to mapping problem near indels and CNVs, we suggest you to generate a list of genomic regions involving such variations. We recommend to identify and filter indels and CNVs from the processed .bam file following the pipelines of GATK and CNVnator, respectively. The regions of called indels (with +/-5bp flanks) and CNVs should be converted to the .bed format and specify the corresponding parameter `indel_region_filter.bed_file` when running MosaicHunter.

I.1.e Running MosaicHunter

The standard command-line usage of MosaicHunter is

```
java -jar your_path/build/mosaichunter.jar <predefined_configuration> -P param_1=value_1 [-P param_2=value_2 [...]]
```

(Note: `predefined_configuration` could be "genome", "exome", or "exome_parameters")

or

```
java -jar your_path/build/mosaichunter.jar -C <config.properties> -P param_1=value_1 [-P param_2=value_2 [...]]
```

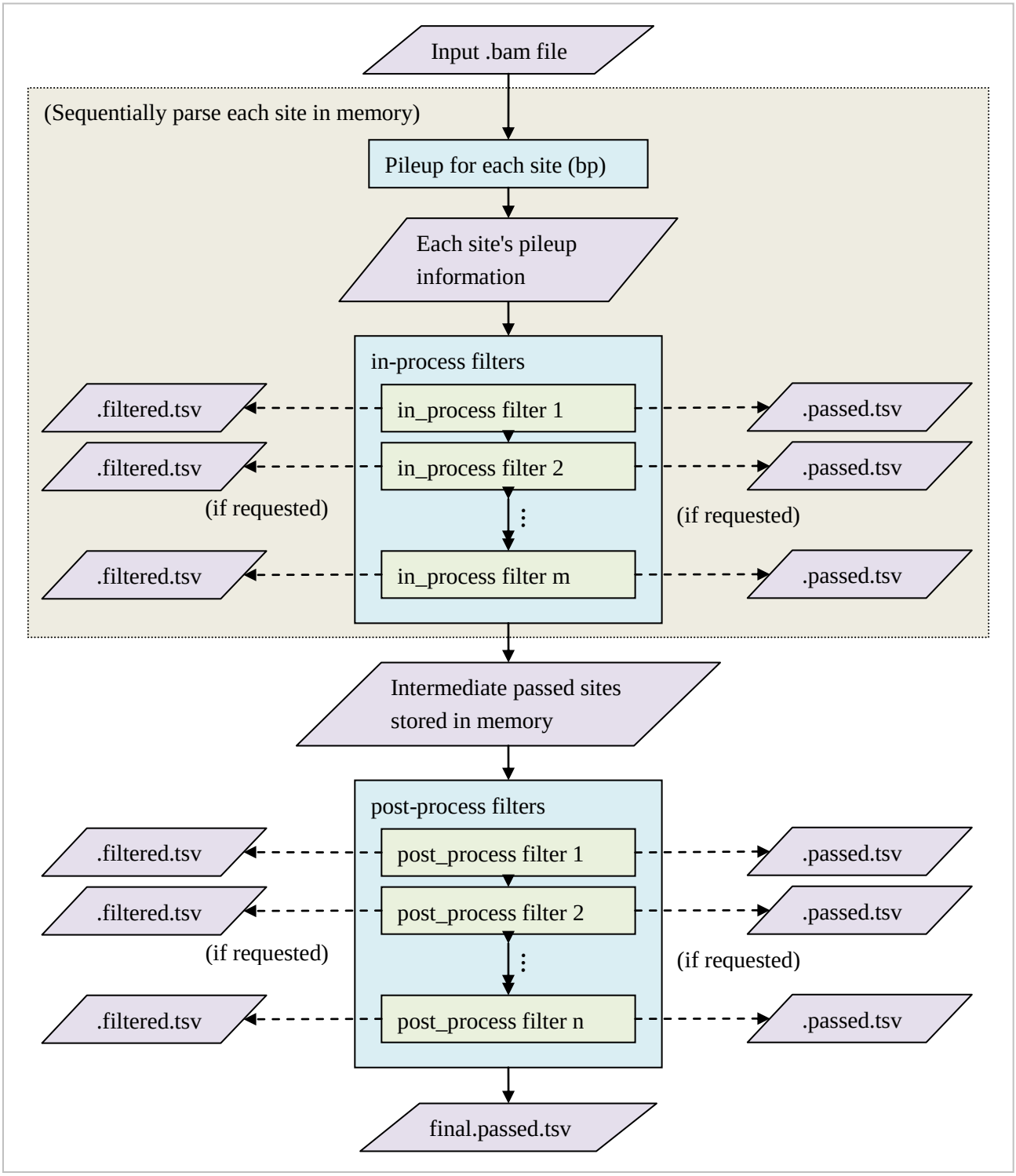
The output files will be put into the path set by the top parameter `output_dir`.

(Details see also I.2 and II.3)

I.2 Framework and Best Practice

I.2.a Program framework and work flow

The main framework and work flow of MosaicHunter is demonstrated in the figure below.



For running efficiency, by default, we do not output the pileup information of each site to the disk,

but filter out sites as many as possible in memory (in_process filters), store the intermediate result in memory, and then run post_process filters sequentially. Any filters that work on one site independently to other sites can be in_process filters, while some filters that work on adjacent sites (clustered_filter) can only be post_process filters. The much slower filter (misaligned_reads_filter, which calls blat) is recommended to be put as the later post_process filter.

Details about filters see also II.2.

The pipeline of filters with their parameters (and the 'top parameters') are usually specified in the config.properties file (`-C <path>` option), with a few command-line overwritten parameters (`-P param=value` option). Note: if the value is a string with any space character, quotes (' or ") are not needed in the config.properties file, but may be required in the command-line, because bash will separate options by space character.

I.2.b Best practice pipelines

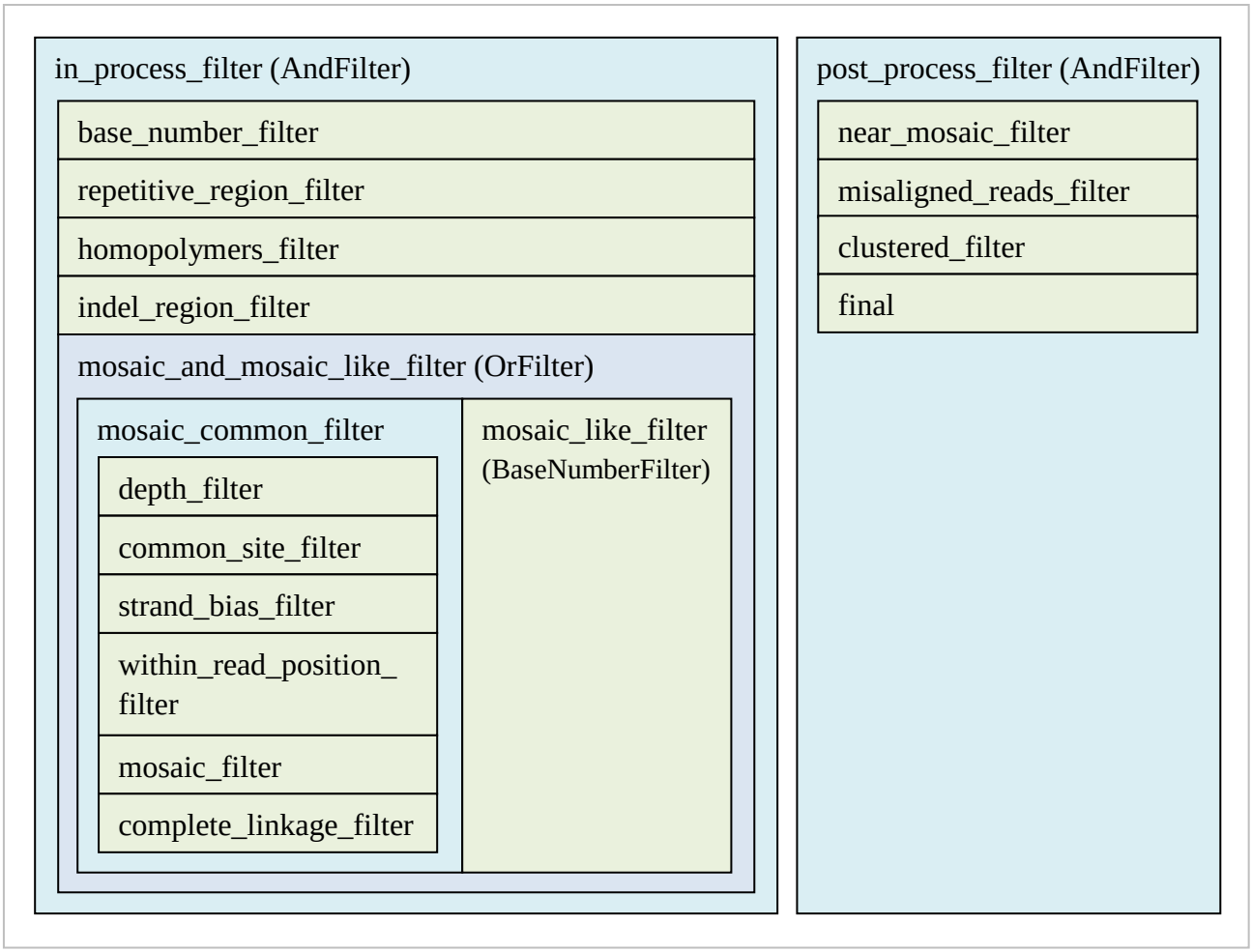
Here we listed some suggested config properties for WGS, WES of normal samples, or WGS of cancer samples. You can also find them in the conf/ folder (default.properties; exome_parameters.properties, exome.properties; cancer.properties).

1) Whole genome sequencing for normal samples

single mode as an example

(see also conf/default.properties for the list of parameters with default values)

The flowchart and structure of filters applied to whole genome sequencing (WGS) data is shown in this figure. (see also I.2.a)



In brief, we filter out fallible regions (repetitive_region_filter, homopolymers_filter, indel_region_filter), and find out mosaic candidate sites (mosaic_common_filter) and any sites with strange allele fraction 'mosaic_like' sites (mosaic_like_filter). We use near_mosaic_filter to drop 'mosaic_like' sites far away from mosaic candidate sites, then check misalignment by calling blat, check and remove clustered mosaic candidate sites (probable artifacts).

Note: mosaic_and_mosaic_like_filter, mosaic_like_filter and near_mosaic_filter are designed to work coupling with clustered_filter, in which we focused on 'mosaic' candidate sites, with consideration of some auxilary nearby 'mosaic_like' sites (see also II.2).They make the structure of pipeline much more complicated, and all of them are removed in the whole exome sequencing analysis.

You can run this pipeline by:

```
java -jar mosaichunter.jar genome [-P param=value [-P ...]]
```

or

```
java -jar mosaichunter.jar -C conf/genome.properties [-P param=value [-P ...]]
```

Here are some required parameters:

`input_file=<path>`

`reference_file=<path>`

`output_dir=<path>`

`mosaic_filter.sex=<M|F>`

`mosaic_filter.mode=<single|trio|paired_naive|paired_fisher>`

Some path parameters of resource files:

`mosaic_filter.dbsnp_file=<path>`

`repetitive_region_filter.bed_file=<path>`

`indel_region_filter.bed_file=<path>`

`common_site_filter.bed_file=<path>`

Some parameters we suggest to modify according to your data:

`max_depth=500`

`depth_filter.min_depth=25`

`depth_filter.max_depth=150`

We suggest to set depth_filter.min_depth and depth_filter.max_depth to be Q10 (10% quantile) and Q90 (90% quantile) of depth in your data. Actually, mosaic candidate sites with depth < 25 may be unreliable. And sites with much higher depth may also enrich artifacts.

The final output will be `output_dir`/final.passed.tsv.

See also II.3 for interpretation of the format.

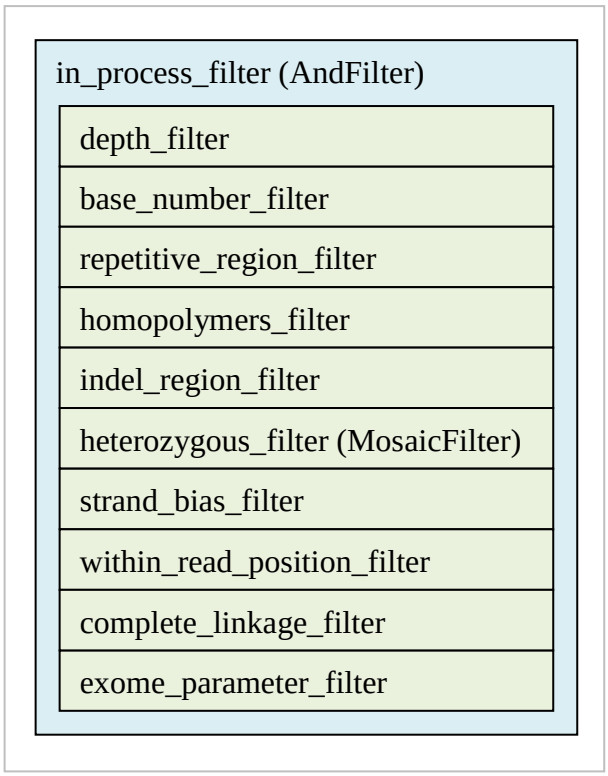
2) Whole exome sequencing for normal samples

single mode as an example

1st step – estimate the shape parameters (α, β) for the beta prior of heterozygous AF θ

(see also conf/exome_parameters.properties for the list of parameters with default values)

The flowchart and structure of filters applied to this step is shown in this figure. (see also I.3.a)



In brief, we filter out fallible regions (repetitive_region_filter, homopolymers_filter, indel_region_filter), and do statistics on heterozygous sites. The final output of the average depth and the estimated shape parameters (α, β) will be reported to `output_dir`/stdout_*.log.

You can run this step by:

```
java -jar mosaichunter.jar exome_parameters [-P param=value [-P ...]]
```

or

```
java -jar mosaichunter.jar -C conf/exome_parameters.properties [-P param=value [-P ...]]
```

Here are some required parameters:

`input_file=<path>`

`reference_file=<path>`

`output_dir=<path>`

`heterozygous_filter.sex=<M|F>`

Some path parameters of resource files:

`repetitive_region_filter.bed_file=<path>`

`indel_region_filter.bed_file=<path>`

`common_site_filter.bed_file=<path>`

`mosaic_filter.dbsnp_file=<path>`

Some parameters we suggest to modify according to your data:

`max_depth=5001`

`depth_filter.min_depth=25`

`depth_filter.max_depth=5000`

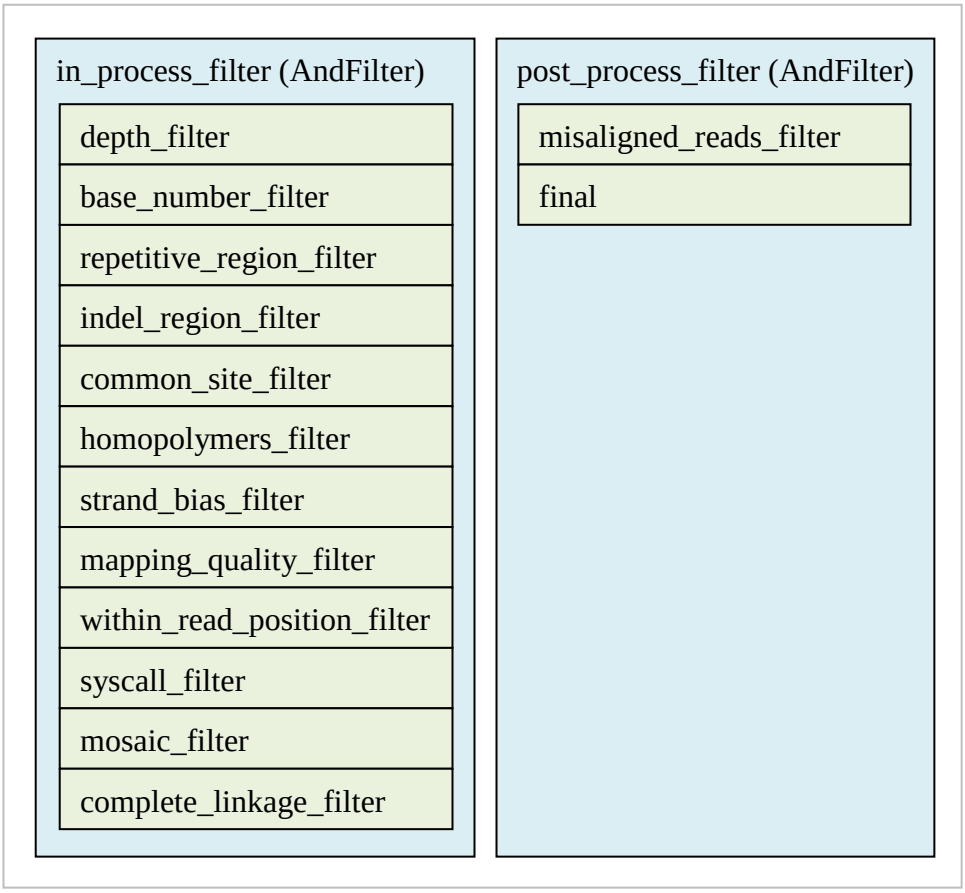
For efficiency issue, we suggest to limit the parameter `max_depth` <= 5000.

The final output will be `output_dir`/stdout_*.log.

2nd step – actually call pSNMs

(see also conf/exome.properties for the list of parameters with default values)

The flowchart and structure of filters applied to this step is shown in this figure. (see also I.3.a)



In brief, compared to the WGS flowchart, for WES we add mapping_quality_filter, syscall_filter, and remove clustered_filter and its related near_mosaic_filter and mosaic_like_filter. Remember to specify `mosaic_filter.alpha_param`, `mosaic_filter.beta_param` and `syscall_filter.depth` to α, β and the average depth, which are estimated in the 1st step (check the corresponding `output_dir`/stdout_*.log). The `output_dir` is suggested to be different from the one in the 1st step. Besides, we suggest to use the same parameters for the corresponding filters with the same name in the 1st step and in the 2nd step.

You can run this step by:

```
java -jar mosaichunter.jar exome [-P param=value [-P ...]]
```

or

```
java -jar mosaichunter.jar -C conf/exome.properties [-P param=value [-P ...]]
```

Here are some required parameters:

```
input_file=<path>
reference_file=<path>
output_dir=<path>
misaligned_reads_filter.reference_file=<path>
mosaic_filter.sex=<M|F>
mosaic_filter.alpha_param=<int>
mosaic_filter.beta_param=<int>
```

Some path parameters of resource files:

```
repetitive_region_filter.bed_file=<path>
indel_region_filter.bed_file=<path>
common_site_filter.bed_file=<path>
mosaic_filter.dbsnp_file=<path>
```

Some parameters we suggest to modify according to your data:

```
max_depth=500
depth_filter.min_depth=25
```

`depth_filter.max_depth=150`

`syscall_filter.depth=66`

The final output will be `output_dir`/final.passed.tsv.

See also II.3 for interpretation of the format.

3) Whole genome sequencing for cancer samples

single-sample mode as an example

(see also conf/cancer.properties for the list of parameters with default values)

The pipeline for cancer sample is very similar to the pipeline for normal sample, except that we modify mosaic_filter's parameter `mosaic_rate=1e-6` and reduce clustered_filter's parameters `inner_distance=2000` and `outer_distance=2000`, to increase its sensitivity.

You can run this pipeline by:

```
java –jar mosaichunter.jar –C conf/cancer.properties [-P param=value [-P ...]]
```

I.3 Bayesian Models and Program Modes

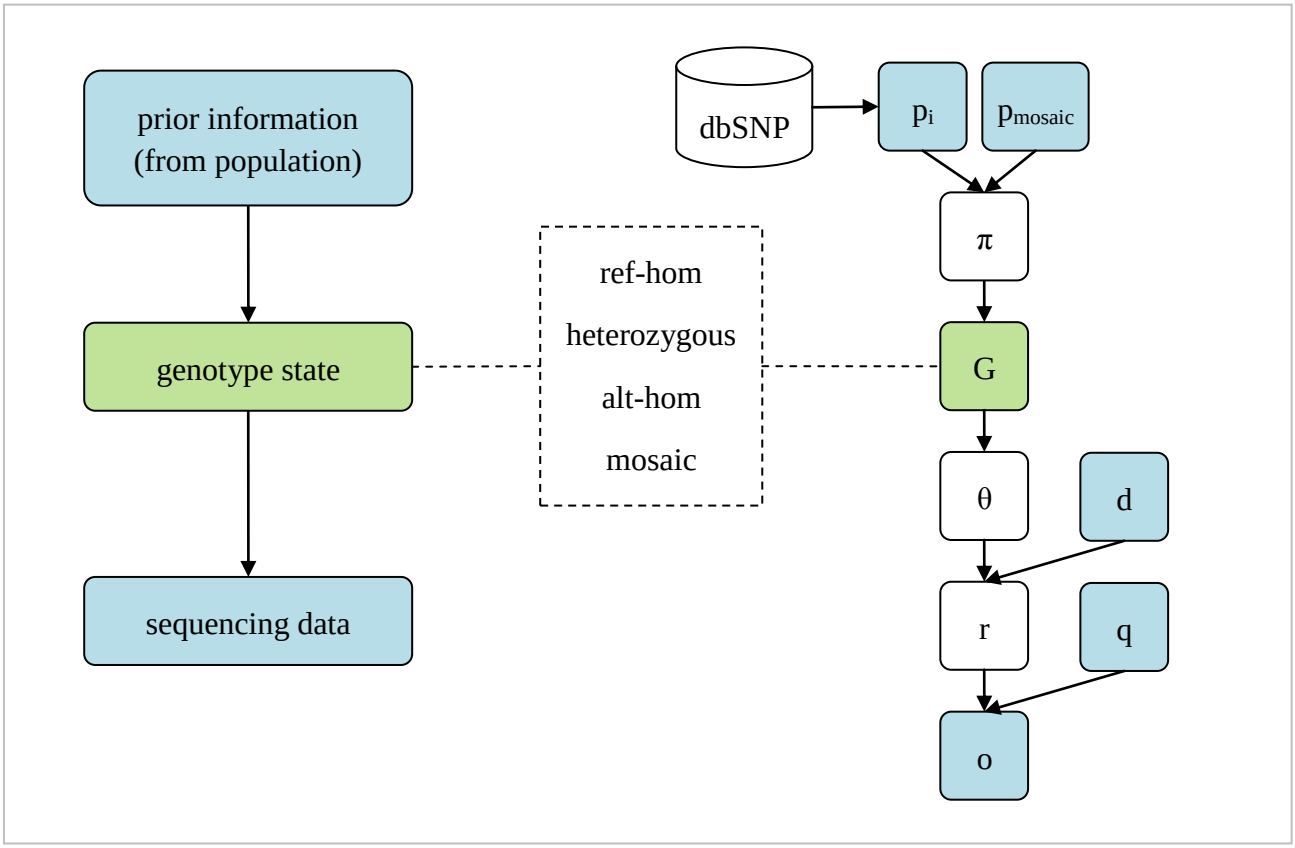
I.3.a Single mode, paired mode & trio mode

Because related samples are sometimes unavailable in practice, we firstly designed the Bayesian mosaic genotyper to call pSNMs from single sample ('single mode'). The Bayesian model could be easily generalized to trio samples ('trio mode'), in which we could better call pSNMs with allele fraction (AF) around 0.5 with the additional parents' data. For previous practice of calling mosaicism in cancer, it may be better to have matched control samples on the same individual, and then do comparison between them. We also implemented it ('paird mode').

1) Single mode

With single sample available, after users prepared its bam file, it will be quite easy to call pSNMs with MosaicHunter: just set the bam file as `input_file`, and set MosaicFilter's parameters `mode=single` (see also II.2).

The probabilistic graphical model of the Bayesian mosaic genotyper (implemented in MosaicFilter) for single sample can be depicted in this figure.

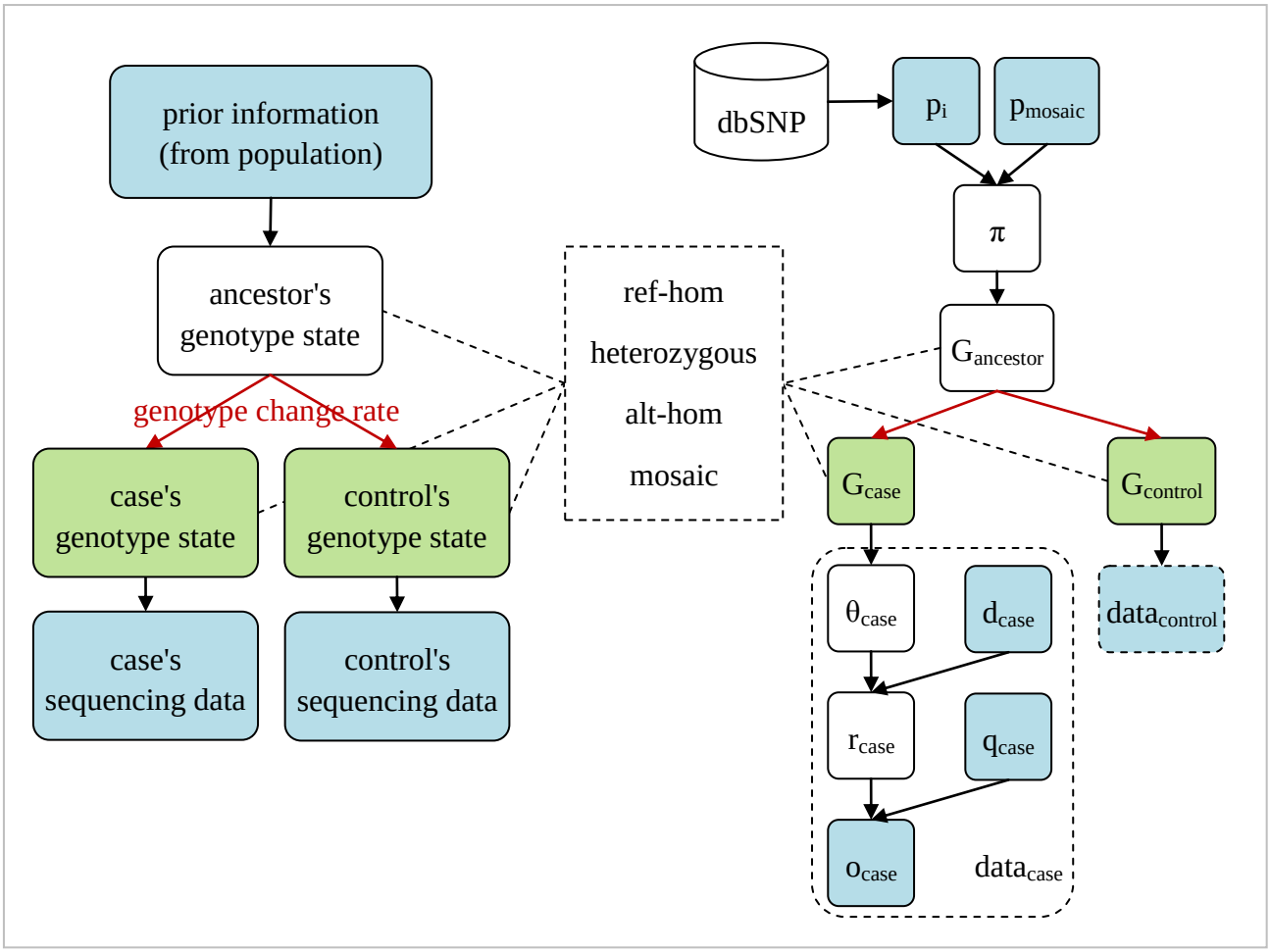


The blue boxes in the figures represent pre-specified prior information or the observed data (specified as MosaicFilter's parameter or data files), and the green box of genotype state G is what we try to infer. We use Bayesian inference to estimate the genotype state G. MosaicFilter will pass the sites if its mosaic posterior probability P(G=mosaic|data) > MosaicFilter's parameter `mosaic_threshold`.

2) Paired mode

We implemented two types of 'paired mode', namely 'naive' paired mode and 'fisher' paired mode. Both additionally need to specify the parameter `control_bam_file=<path>` in MosaicFilter.

In the 'naive' paired mode (MosaicFilter's `mode=paired_naive`), we extended the Bayesian model and incorporated a latent variable – ancestor's genotype, and a genotype change rate matrix. We inferred the joint posterior probability of the genotype states of case/control samples, and summed up the posterior probability that their genotype states were different, and will pass the sites if this probability $P(G_{case}!=G_{control}|data)$ > MosaicFilter's parameter `mosaic_threshold`. The probabilistic graphical model for naive paired mode can be shown in this figure.

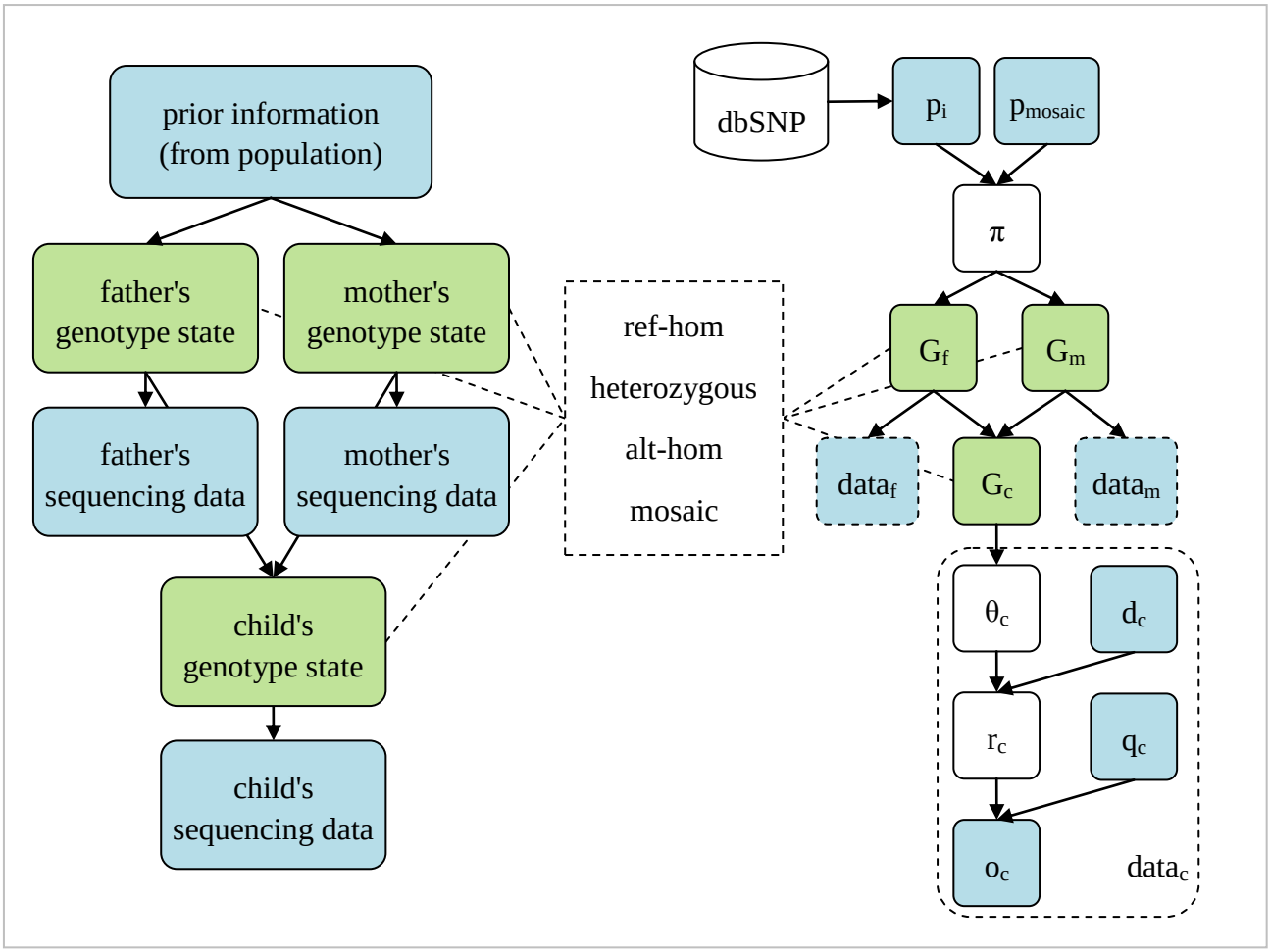


'Fisher' paired mode (MosaicFilter's `mode=paired_fisher`) does not use the Bayesian model, but just compares the refDepth and altDepth in case and control samples using Fisher's exact test for 2*2 contingency table. MosaicFilter will pass the sites with p-value < `fisher_threshold`.

3) Trio mode

If you have sequenced the individual (child) with either or both parents, you can try trio mode (MosaicFilter's `mode=trio`, and set the corresponding `father_bam_file=<path>`, `mother_bam_file=<path>`).

The probabilistic graphical model for trio mode can be shown in this figure.



The Bayesian model for mosaic calling is actually an outlier detector. Therefore, if on the site the child has pSNM, or the Mendelian inheritance is violated (such as *de novo* mutation), MosaicFilter may pass it as a candidate site (child's mosaic posterior probability > MosaicFilter's parameter `mosaic_threshold`). So under trio mode, you may need to further check the child's likelihood to judge whether the candidate site is a pSNM (child's largest likelihood is mosaic state) or a *de novo* SNV (child's largest likelihood is heterozygous state).

I.3.b Whole genome sequencing vs. whole exome sequencing

(mean-shift &over-dispersion of allele fraction of heterozygous sites)

Another thing need to be taken into consideration is whether your data is whole genome sequencing (WGS) or whole exome sequencing (WES). We found that the heterozygous sites follow binomial distribution well in WGS, but poor in WES (heterozygous allele fraction (AF) mean < 0.5, and over-dispersion). To solve this over-dispersion problem in WES, we replace the prior of theoretical allele fraction θ P(θ|G=heterozygous) from an impulse at 0.5 to a beta prior distribution fitted from data. This adjusting method has two steps: (1) traverse the bam file once, summarize the AF information of heterozygous sites, then output the estimated shape parameters (α, β) for the beta prior. Certainly, if your data also showsmean-shift or over-dispersion of allele fraction of heterozygous sites, you can try this two-step method.

To run the 1st step, we applied several similar filters but to get heterozygous sites (MosaicFilter's `mode=heterozygous`), and finally applied ExomeParameterFilter to summarize the AF information, calculate statistics, and report the optimized shape parameters (α, β) estimates in file `output_dir`/stdout_*.log.

To run the 2nd step, just set MosaicFilter's parameters `alpha_param=<int>` and `beta_param=<int>` to be the result from 1st step, which control the beta prior used in the Bayesian model. If they are kept to their default value -1, then the original prior – impulse at 0.5 – will be used. If the estimated shape parameters (α, β) are both very large (>1000), and α/(α+β) is close to 0.5, which means that the beta prior is quite similar to the original impluse at 0.5 (no over-dispersion and mean-shift), we suggest to keep these parameters to the default 0, for running efficiency.

Except for the change in MosaicFilter, we have also modified and added some other filters for WES, such as MappingQualityFilter and SysCallFilter (see also I.3.b and II.2). For SysCallFilter, we trained the SysCall logistic model for different average depth in advance. You can check the output file `output_dir`/stdout_*.log of the 1st step to obtain the average depth.

II Manual for Parameters and Filters

II.1 Top Parameters

There are some parameters in the top level ('top parameters'), here is the list with explanation.

`input_file=<path>`

The path of input .bam file

[Required]

`reference_file=<path>`

The path of reference file

[Required]

`output_dir=<path>`

The working directory for output files

[Required]

`valid_references=<str>`

The contig name list for calling mosaicisms (delimited by ','); ie. the contigs not in this list will be ignored.

[Default: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,X,Y]

`chr_X_name=<str>`

`chr_Y_name=<str>`

The contig name for sex chromosome X and Y

[Default: X] [Default: Y]

`max_depth=<int>`

For efficiency, if the site has depth > this_value, it will be trimmed (or down-sampled) to this_value. (see also `depth_sampling=<bool>`)

[Default: 5001]

`base_quality=<int>`

The Phred shift of baseQ in the bam file

[Default: 33]

`min_mapping_quality=<int>`

The reads with mapQ < this_value will be immediately discarded.

[Default: 20]

`min_read_quality=<int>`

The bases with baseQ < this_value will be immediately discarded.

[Default: 20]

`remove_duplicates=<bool>`

Should the reads marked duplicate (with flag 0x400) be discarded?

[Default: true]

`remove_flag=<int>`

The reads with any bit on of this specified flag will be discarded. For example, to remove reads indicating secondary alignment, you can set remove_flag=0x100.

[Default: 0x0]

`seed=<int>`

The seed for random number generator; set a fixed number for replicability.

[Default: 0]

`depth_sampling=<bool>`

When the site has depth > max_depth, should I down-sampled it (true) or just trimmed it (false). (see also `max_depth=<int>`)

[Default: false]

input_sampling=false

input_sampling_regions=1

input_sampling_size=1

`chr=<str>`

If you just want to call pSNMs on one contig (chromosome) instead of all contigs, set this.

[Default: (empty)]

`start_position=<int>`

`end_position=<int>`

If you just focused on one block on one contig, you can set these two parameters.

(1-based, end-included)

[Default: 1] [Default: 300000000]

`read_buffer_size=<int>`

How many reads can be read in buffer?

[Default: 100000]

max_recent_reads=9997

stats_manager.enable_timer=false

stats_manager.enable_counter=false

enable_reference_cache=false

`in_process_filter_name=<str>`

Overall in_process filter's name (not recommend to change)

[Default: in_process_filter]

`post_process_filter_name=<str>`

Overall post_process filter's name (not recommend to change)

[Default: post_process_filter]

II.2 Filter Introduction and Parameters

We develop several filters to remove artifacts as many as possible. For convenience, we call the Bayesian genotyper as mosaic_filter in the java implementation.

No matter the filters are put in the in_process or the post_process, they will all be checked sequentially for each site. The only difference is that the filters put in the in_process should work on each site independently, while the filters put in the post_process can handle all remaining candidate sites.The order of some post_process filters may affect the final results. Any filters that can be put in the in_process can also be put in the post_process, although it may increase the memory burden and slow down the speed.

Each filter has a name and a class in our implementation, as some filters can actually apply the same algorithm arranged in the same class. Class names are fixed in the source code, while the filter names are written in the config.properties file. Filter names may be changed, although we recommend not changing them for better communication. Filter names will be used as prefix in the output filenames. The passed filter names for each site will be recorded in the memory, which may be further used, especially for OrFilter (eg. mosaic_and_mosaic_like_filter) and the cluster_filter, near_mosaic_filterin the whole genome sequencing analyzing pipeline (see also I.3.b).

Each filter has two general parameters and some other parameters specifically for the filter class (see also II.2).

Here is a table of our developed filter classes with filter names.

Filter Class	Filter Name	Comment
AndFilter	in_process_filter, post_process_filter, mosaic_common_filter	
OrFilter	mosaic_and_mosaic_like_filter	
DepthFilter	depth_filter	in_process
BaseNumberFilter	base_number_filter, mosaic_like_filter	in_process
RegionFilter	repetitive_region_filter, indel_region_filter, common_site_filter	in_process
HomopolymersFilter	homopolymers_filter	in_process
MosaicFilter	mosaic_filter, heterozygous_fitlter	in_process
StrandBiasFilter	strand_bias_filter	in_process
WithinReadPositionFilter	within_read_position_filter	in_process
CompleteLinkageFilter	complete_linkage_filter	in_process
NearMosaicFilter	near_mosaic_filter	post_process
MisalignedReadsFilter	misaligned_reads_filter	post_process
ClusteredFilter	clustered_filter	post_process
OutputFilter	final	post_process
ExomeParameterFilter	exome_parameter_filter	in_process
NullFilter	null_filter	in_process
MappingQualityFilter	mapping_quality_filter	in_process
SysCallFilter	syscall_filter,	in_process

Below are details about general parameters and each filter class.

General parameters

`output_filtered=<bool>`

Is the corresponding output <filter_name>.filtered.tsv requested?

[Default: false]

`output_passed=<bool>`

Is the corresponding output <filter_name>.passed.tsv requested?

[Default: false]

1) AndFilter

[in_process_filter, post_process_filter, mosaic_common_filter]

This logic filter registers some more filters. Only the sites passed every the registered filters will be passed by this filter. The registered filters will be checked sequentially.

`filters=<str>`

The registered filter names, separated by ',', and no space ' ' is allowed.

[Default: (empty)]

2) OrFilter

[mosaic_and_mosaic_like_filter]

This logic filter registers some more filters. The sites passed any of the registered filters will be passed by this filter. The registered filters will be checked sequentially.

The passed filter names for each site will be recorded in the memory, which may be further used.

`filters=<str>`

The registered filter names, separated by ',', and no space ' ' is allowed.

[Default: (empty)]

3) DepthFilter

[depth_filter]

This filter will pass sites with depth in the specified range.

`min_depth=<int>`

[Default: 25]

`max_depth=<int>`

[Default: 150]

4) BaseNumberFilter

[base_number_filter, mosaic_like_filter]

This filter checked the observed minor allele count and minor allele fraction. The sites with minor allele in specificed range will be passed.

`min_minor_allele_number=<int>`

[Default: 5]

`min_minor_allele_percentage=<int>`

[Default: 5]

`max_minor_allele_percentage=<int>`

[Default: 100]

5) RegionFilter

[repetitive_region_filter, indel_region_filter, common_site_filter]

This filter will filter out sites inside or outside the specified region blocks.

Output tsv (filtered.tsv) 11+ columns (see also II.3):

11-13: chr, start, end of the region block (1-based, end-included)

`bed_file=<path>`

The specified regions .bed file.

[Required]

`expansion=<int>`

Should I expand each block ? bp.

[Default: 5]

`include=<bool>`

Should I filter out sites inside the region blocks (false) or outside the region blocks (true).

[Default: false]

6) HomopolymersFilter

[homopolymers_filter]

This filter will filter out sites near homopolymers on the reference sequence.

`short_homopolymer_length=<int>`

Define the length of 'short' homopolymer

[Default: 4]

`short_homopolymer_expansion=<int>`

Expand each 'short' homopolymer ? bp, ie. filter out sites ? bp near a 'short' homopolymer.

[Default: 2]

`long_homopolymer_length=<int>`

Define the length of 'long' homopolymer

[Default: 6]

`long_homopolymer_expansion=<int>`

Expand each 'long' homopolymer ? bp, ie. filter out sites ? bp near a 'long' homopolymer.

[Default: 3]

7) MosaicFilter

[mosaic_filter]

The main Bayesian genotyper for pSNMs calling (the mosaic caller) (see also I.2)

Output tsv (filtered.tsv, passed.tsv) will be different for different mode (single, paired, trio mode), see also II.3.

`min_read_quality=<int>`

Ignore bases with baseQ < this_value on the site

[Default: 20]

`min_mapping_quality=<int>`

Ignore bases with mapQ < this_value on the site

[Default: 20]

`dbsnp_file=<path>`

A tab-separated file providing information in dbSNP, which will provide reference/alternative allele frequency (prior information). It should have six columns:

1: chr

2: pos

3: rsId

4: reference allele (refBase)

5: alternative allele (altBase)

6: alternative allele frequency in population (1000 Genomes); -1 means no allele frequency information recorded in dbSNP.

[Required]

`unknown_af=<double>`

The specified population allele frequency for variants that have records but no allele frequency information in dbSNP.

[Default: 0.002]

`novel_af=<double>`

The specified population allele frequency for variants without records in dbSNP.

[Default: 1e-8]

`mosaic_rate=<double>`

The mosaic prior probability.

[Default: 1e-7]

`de_novo_rate=<double>`

The probability for a de novo mutation occurring

[Default: 1e-8]

`sex=<str>`

'M' for male, and 'F' for female. This affects mosaic calling for sex chromosomes.

`alpha_param=<int>`

`beta_param=<int>`

For exome sequencing, we observed over-dispersion on allele fraction of heterozygous sites, so we change the prior of theoretically allele fraction θ from impulse at 0.5 to a beta distribution, with two shape parameters (α, β) specified here. The two shape parameter (α, β) can be estimated by running MosaicHunter with exome_parameters.properties (details see also I.2.b).

The default value (0, 0) means trigger off this part, and just use the impulse at θ = 0.5. In fact, to trigger on the heterozygous beta prior, both parameters should be > 0 and their sum should be > 2.

For efficiency issue, we suggest that both parameters should be <= 1000. For larger estimated parameters, which indicates closeness of the beta prior to the θ = 0.5 impulse, we suggest to trigger off this part by setting the default (0, 0), and turn back to the θ = 0.5 impulse prior.

[Default: 0] [Default: 0]

`base_change_rate=<16*double>`

If you want to specify a prior on mosaic base change frequency, you can specify it here. It contains 16 double values(separated by ',') corresponding to A->A,C,G,T; C->A,C,G,T; G->A,C,G,T; T->A,C,G,T. You can specify the relative rate, as they will later be row-wisely normalized to sum to

1.

[Default: 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]

`mosaic_threshold=<double>`

The sites with mosaic posterior (or heterozygous posterior if `heterozygous=true`) > this_value will be passed

[Default: 0.05]

`mode=<single|trio|paired_naive|paired_fisher|heterozygous>`

Specify the mode of this filter, which can be "single", "trio", "paired_naive", "paired_fisher" or "heterozygous". (details see also I.3)

[Default: single]

`father_bam_file=<path>`

`mother_bam_file=<path>`

If you specify the trio mode (`mode=trio`), you also need to specify these parameters. Then the mosaic calling in the child will also consider parents' sequencing data.

[Default: (empty)]

`control_bam_file=<path>`

If you specify the paired mode (`mode=paired_naive` or `mode=paired_fisher`), you also need to specify these parameters.

[Default: (empty)]

`fisher_threshold=<double>`

If you specify the Fisher paired mode (`mode=paired_fisher`), then you need to specify this parameter. The filter will pass sites with the p-value of Fisher's exact test < this_value.

8) StrandBiasFilter

[strand_bias_filter]

This filter will apply Fisher's exact test on strand bias (major strand+, major strand-, minor strand+, minor strand-), and filters out site with small p-value.

Output tsv (filtered.tsv, passed.tsv) 11+ columns (see also II.3):

11-14: count of major +, minor + (forward strand), major –, minor – (reverse strand)

15: p-value of Fisher's exact test

`min_p_value=<double>`

The sites with p-value >= this_value will be passed.

[Default: 0.05]

9) WithinReadPositionFilter

[within_read_position_filter]

This filter will apply Wilcoxon rank-sum test (ie. Mann-Whitney U test) on positions within reads according to the reference orientation (major pos, minor pos), and filters out sites with small p-value.

Output tsv (filtered.tsv, passed.tsv) 11+ columns (see also II.3):

11: major read position

12: minor read position

13: p-value of Wilcoxon rank-sum test (ie. Mann-Whitney U test)

`min_p_value=<double>`

The sites with p-value >= this_value will be passed.

[Default: 0.05]

10) CompleteLinkageFilter

[complete_linkage_filter]

If there are more SNV(s) on the same read with mosaic sites, we would expect that the mosaic site should not be randomly linked or completely linked to the SNV site, which may be filtered by this filter.

Output tsv (filtered.tsv) 11+ columns (see also II.3):

11: the close position with a possible SNV

12-16: count of four kinds of haplotypes

17: p-value of Fisher's exact test

`max_p_value=<double>`

We do Fisher's exact test on the 2*2 contingency table of count of four kinds of haplotypes. The sites with p-value > this_value will be filtered.

[Default: 0.01]

11) NearMosaicFilter

[near_mosaic_filter]

This filter drops auxiliary ('mosaic-like') sites far away from focused ('mosaic candidate') sites.

`distance=<int>`

The maximum distance allowed between passed mosaic-like sites and a mosaic candidate site.

[Default: 10000]

`auxiliary_filter_name=<str>`

The passed filter name marking the auxiliary sites.

[Default: mosaic_like_filter]

12) MisalignedReadsFilter

[misaligned_reads_filter]

Call blat to check read misalignment (very slow).

Passed sites require a enough large proportion of reads (see also `max_misalignment_percentage=<double>`): 1) blat mapping result is unique, and agrees with .bam mapping result (eg. by bwa) (see also`min_overlap_percentage=<double>`); 2) the candidate site is not near two ends (see also `min_side_distance=<int>`) or gaps (see also `min_gap_distance=<int>`) of the read.

Output tsv (filtered.tsv, passed.tsv) 11+ columns (see also II.3):

11: ALIGNMENT_OK

12: ALIGNMENT_MISSING

13: MULTIPLE_ALIGNMENTS

14: CHROM_MISMATCH

15: ALIGNMENT_MISMATCH

16: NEAR_SIDE

17: NEAR_GAP

18: NM

19: misalignment

20: alignment_ok proportion among major allele

21: alignment_ok proportion among minor allele

`blat_path=<path>`

The path of runnable blat.

[Default: blat]

`blat_param=<str>`

The additional parameters when calling blat.

[Default: -stepSize=5 -repMatch=2253 -minScore=0 -minIdentity=0.5 -noHead]

`reference_file=<path>`

The reference fasta file for blat, which can be different from the top parameter `reference_file`. We suggest putting all possible contigs in this file, such as chr6 HLA haplotypes, to reduce the misalignment due to sequence difference.

[Default: Same as top parameter `reference_file`]

`max_misalignment_percentage=<double>`

Passed sites should have proportion of 'misaligned' reads < this_value ('good' reads > (1-this_value)).

[Default: 0.5]

`min_overlap_percentage=<double>`

Each read will have one mapping block by blat and one from .bam (eg. by bwa). If the proportion of intersection of two mapping blocks on either block is < this_value, the read will be regarded as 'misaligned'.

[Default: 0.9]

`min_side_distance=<int>`

If the candidate site is near the ends of the read (the end base corresponding to distance = 1), the read will be regarded as 'misaligned'.

[Default: 15]

`min_gap_distance=<int>`

If the candidate site is near the mapping gap of the read (the marginal base corresponding to distance = 1), the read will be regarded as 'misaligned'.

[Default: 5]

`max_NM=<int>`

If the NM tag of the read in .bam is > this_value, the read will be regarded as 'misaligned'.

13) CluteredFilter

[clustered_filter]

Because some region on the genome may enrich sequencing and mapping artifacts, we see some false-positive candidate sites clustered (with close position). This filter will remove main ('mosaic') candidate sites which are clustered with other main ('mosaic') or auxiliary ('mosaic-like') sites.

Output tsv (passed.tsv, filtered.tsv) 11+ columns (see also II.3):

11: the positions of other close clustered sites

`inner_distance=<int>`

If three neighbor mosaic candidate sites are with distance < this_value bp, they are regarded as clustered sites.

[Default: 20000]

`outer_distance=<int>`

For any found clustered sites, we will expand the boundary of filtering region to upstream and downstream this_value bp, and the mosaic candidate sites falling in the region will be filtered.

`auxiliary_filter_name=<str>`

The passed filter name marking the auxiliary sites.

[Default: mosaic_like_filter]

14) OutputFilter

[final]

The final 'filter' just for outputing the result list of the passed, focused candidate sites.

15) ExomeParameterFilter

[exome_parameter_filter]

This 'filter' does statistics on passed sites (assuming heterozygous sites), and finally reports to `output_dir`/stdout_*.log the average depth and the estimated (optimized) shape parameters(α, β) for exome heterozygous allele fraction prior.

`min_group_size=<int>`

When doing statistics on standard deviation of allele fraction, we need to group together sites with close depth, if the mount of sites with the same depth is not large enough (< this_value).

[Default: 50]

`optimal_depth=<int>`

When the observed relationship between std(AF) ~ depth cannot be exactly fitted by the model formula, I will try to optimally fit the two values of depth = this_value.

[Default: 80]

`r_data_file=<filename>`

The temporary file for counting, which will then be read in again and fitted.

[Default: "r_het_data.tsv"]

16) NullFilter

[null_filter]

This filter will discard all the sites, thus to clear up memory. Only useful after the filter doing statistics or estimating parameters (eg. ExomeParameterFilter).

17) MappingQualityFilter

[mapping_quality_filter]

This filter will apply Wilcoxon rank-sum test (ie. Mann-Whitney U test) on reads' mapQs (major mapQs, minor mapQs), and filters out sites with small p-value.

`min_p_value=<double>`

The sites with p-value >= this_value will be passed.

[Default: 0.05]

18) SysCallFilter

[syscall_filter]

This filter applies the logistic regression model suggested by SysCall (Meacham *et al.*, BMC Bioinformatics, 2011), to filter out exome false positives. The trained parameters varied across different depth of exome sequencing data, so the average depth should be specified.

`depth=<int>`

The average depth of exome sequencing data, which can be estimated by running MosaicHunter with exome_parameters.properties (details see also I.2.b and I.3.b).

[Default: 66]

II.3 Output Files and Format

All the output files are in the specified directory `output_dir`, including those requested <filter_name>.filtered.tsv, <filter_name>.passed.tsv. files, and some other temporary files, such as blat input and output.

<filter_name>.filtered.tsv and <filter_name>.passed.tsv files are in tab-separated format, whose columns' meaning is listed below:

1: chr	the contig/chromosome name
2: pos	the position/coordinate on the contig (1-based)
3: refBase	the base of reference allele
4: depth	the sequencing depth of this site
5: bases	pileuped sequencing bases at this site
6: baseQs	pileuped sequencing baseQs at this site
7: majorBase	the base of major allele
8: majorDepth	the depth of major allele
9: minorBase	the base of minor allele
10: minorDepth	the depth of minor allele

11+ columns are dependent on filters (see also Output tsv 11+ part for each filter in II.2). final.passed.tsv is the same as mosaic_filter.passed.tsv, which will be different for different mode.

1) For single mode,

11: major : minor allele, with dbSNP allele frequency in population shown in parentheses

12-15: log10 prior of major-homozygous, heterozygous, minor-homozygous, mosaic

16-19: log10 likelihood of major-homozygous, heterozygous, minor-homozygous, mosaic

20-23: log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

24: mosaic posterior probability

2) For naive paired mode,

11: case (major allele : major depth, minor allele : minor depth)

12: control (major allele : major depth, minor allele : minor depth)

13-16: case log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

17-20: control log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

21: case sum of posterior probabilities other than control's most probable genotype state

22: control posterior probability of control's most probable genotype state

3) For Fisher paired mode,

11: case (major allele : major depth, minor allele : minor depth)

12: control (major allele : major depth, minor allele : minor depth)

13-16: case log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

17-20: control log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

21: p-value of Fisher's exact test

4) For trio mode,

11: child major : minor allele, with dbSNP allele frequency in population shown in parentheses

12: father (major allele : major depth, minor allele : minor depth)

13: mother (major allele : major depth, minor allele : minor depth)

14-17: child log10 likelihood of major-homozygous, heterozygous, minor-homozygous, mosaic

18-21: father log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

22-25: mother log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

26-29: child log10 posterior of major-homozygous, heterozygous, minor-homozygous, mosaic

30: child mosaic posterior probability

31: child log10 likelihood ratio of mosaic vs. heterozygous
(log10 likelihood of mosaic – log10 likelihood of heterozygous)

5) For heterozygous mode,

11: major : minor allele, with dbSNP allele frequency in population shown in parentheses

12-14: log10 prior of major-homozygous, heterozygous, minor-homozygous

15-17: log10 likelihood of major-homozygous, heterozygous, minor-homozygous

18-20: log10 posterior of major-homozygous, heterozygous, minor-homozygous

21: heterozygous posterior probability

III Miscellaneous

III.1 Contact Infomration

Team:

August Yue Huang

Adam Yongxin Ye

Zheng Zhang

Yanmei Dou

Liping Wei (PI)

@Center for Bioinformatics, Peking University

@Liping Wei's Lab, National Institute of Biological Science

Contact us:

Room 307, Wangkezhen Building

Center for Bioinformatics

Peking University

Beijing, 100871

P. R. China

Email: mosaichunter@mail.cbi.pku.edu.cn

III.2 Acknowledgement

We thank Drs. Yu Shyr and Jinzhu Jia for their valuable suggestions and discussions about the statistical models.

Acknowledge to Viktor Persson for the "Indigo" template of the web interface.

We thank Li-Ming Xu for the source codes of the web pages.

We thank Chaozhi Hu for the design of MosaicHunter’s logo.

III.3 License

III.4 FAQ