MosaicHunter User Guide

Version 1.1

Nov 30th, 2017

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 for identifying postzygotic single-nucleotide mosaicism (SNM) sites from whole-genome or whole-exome sequencing data of unpaired, trio, or paired samples. It implements and wraps the Bayesian genotypers and several stringent filters in a java framework.

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

I.1 Getting Started

I.1.a   Installation & Quickstart

Prerequisite 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 UCSC blat in your PATH (or set the parameter `blat_path` for misaligned_reads_filter).

Download the MosaicHunter program and the required resource files

Clone the MosaicHunter repository

```
cd your_path
```

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

If the clonning is successful, you should see the MosaicHunter base directory as `your_path/MosaicHunter`.

Some large required resource files can be accessed by URL:

http://mosaichunter.cbi.pku.edu.cn/download/human_g1k_v37.fasta.gz

http://mosaichunter.cbi.pku.edu.cn/download/dbsnp_137.b37.tsv.gz

You also need to download these files, unzip and put them

into `your_path/MosaicHunter/resources/` directory.

```
gunzip your_path/MosaicHunter/resources/human_g1k_v37.fasta.gz
```

```
gunzip your_path/MosaicHunter/resources/dbsnp_137.b37.tsv.gz
```

Quick usage of a pre-compiled binary code release

In order to make it easy to install MosaicHunter, we have provided 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 newly compiled `mosaichunter.jar` file will overwrite the pre-compiled one in `build/` directory, and you can run MosaicHunter with:

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

I.1.b   Preparing your reference

MosaicHunter requires a .fasta file (.fasta or .fa) for your reference genome. It is better to make sure that your reference file has the same name and order of contigs as your .bam file(s) and .bed file(s).

Reads aligned to any contigs which do not appear 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 aligned reads from whole-genome or whole-exome sequencing. We recommend bwa for read mapping and GATK/Picard for read pre-processing (Picard's SortSam and MarkDuplicates, GATK's IndelRealignment and BaseRecalibration). These pre-processing steps are important in order to reduce false positives in identifying mosaic sites.

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

For the 'trio' mode, you need to specify two additional files, `father_bam_file` and `mother_bam_file`; and for the 'paired' mode, the additional `control_bam_file` must 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 this is as follows:

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

I.1.d   Preparing sample-specific files

False mosaic sites can be caused by alignment errors in the genomic regions containing INDELs and CNVs, thus we suggest that you generate a list of such error-prone regions for MosaicHunter to mask. 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 then 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/MosaicHunter/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/MosaicHunter/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`.

(For details see also I.2 and II.3)

To be concrete, here we show how to run MosaicHunter to call mosaicisms from the demo input files in `your_path/MosaicHunter/demo`. The demo input bam file was extracted from the genomic regions of 18:70500000-70600000 from a real WGS data.

The command to run the demo is:

```
cd your_path/MosaicHunter/

java -jar build/mosaichunter.jar genome \
-P input_file=demo/demo_sample.bam \
-P reference_file=demo/hg37_chr18.fa \
-P mosaic_filter.sex=M \
-P mosaic_filter.dbsnp_file=demo/dbsnp137_hg37_chr18_demo.tsv \
-P repetitive_region_filter.bed_file=resources/all_repeats.b37.bed \
-P indel_region_filter.bed_file=demo/demo_sample.indel_CNV.bed \
-P common_site_filter.bed_file=resources/WGS.error_prone.b37.bed \
-P output_dir=demo_output
```

or

```
cd your_path/MosaicHunter/

java -jar build/mosaichunter.jar -C conf/genome.properties \
-P input_file=demo/demo_sample.bam \
-P reference_file=demo/hg37_chr18.fa \
-P mosaic_filter.sex=M \
-P mosaic_filter.dbsnp_file=demo/dbsnp137_hg37_chr18_demo.tsv \
-P repetitive_region_filter.bed_file=resources/all_repeats.b37.bed \
-P indel_region_filter.bed_file=demo/demo_sample.indel_CNV.bed \
-P common_site_filter.bed_file=resources/WGS.error_prone.b37.bed \
-P output_dir=demo_output
```

When MosaicHunter is running, it will print some log messages to STDOUT, which will also be stored in the output_dir with the file name of `demo_output/stdout_*.log` in this demo.

First, it will print the version, date and time, command-line parameters of MosaicHunter, and then display the full list of parameters used in this run by combining the config file and command-line input.

Then, it will process the input file, and will show the progress, such as:

```
Tue Nov 08 16:25:08 CST 2016 Initializing...

Tue Nov 08 16:25:10 CST 2016 Reading reference from file: demo/hg37_chr18.fa

Tue Nov 08 16:25:12 CST 2016 Initializing filters...

Tue Nov 08 16:25:19 CST 2016 Scanning...
```

```
Tue Nov 08 16:25:19 CST 2016 - Time(s):0 Reads:0 Sites:0/78077248
Progress:0.00%

Tue Nov 08 16:25:28 CST 2016 - Time(s):9 Reads:100859
Sites:78077248/78077248 Progress:100.00%
```

In real cases, a typical 300GB WGS bam file usually costs about 12 hours for running MosaicHunter, whereas a typical 20GB WES bam file usually costs about 2 hours. For this demo, it should finish in several seconds to minutes.

After the program exits in success, you should finally get a table showing the number of sites passed each filter, like this:
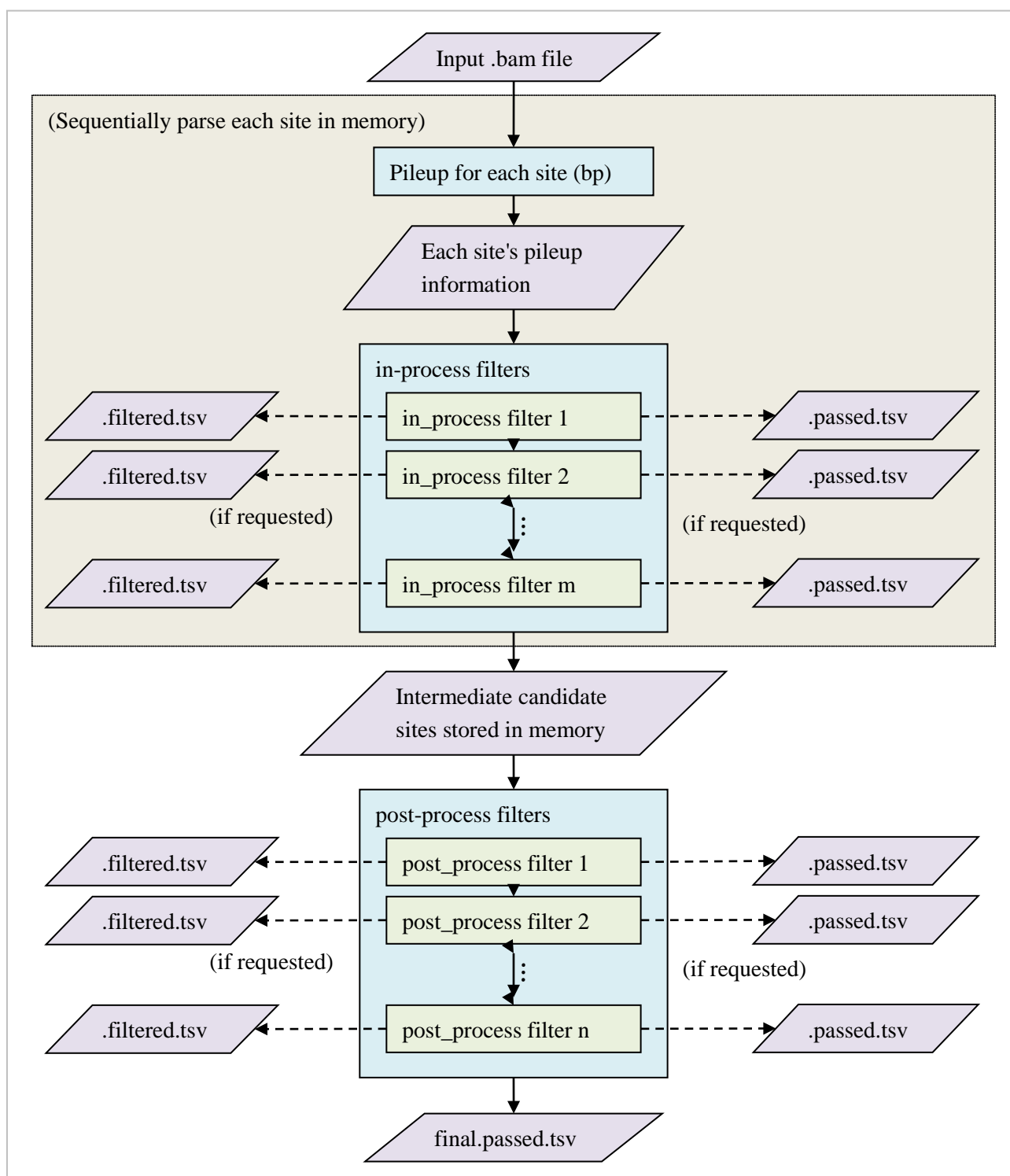
filter name	pass/all	ratio
base_number_filter	157/98748	0.16%
repetitive_region_filter	71/157	45.22%
homopolymers_filter	55/71	77.46%
indel_region_filter	52/55	94.55%
depth_filter	51/52	98.08%
common_site_filter	51/51	100.00%
strand_bias_filter	50/51	98.04%
within_read_position_filter	49/50	98.00%
mosaic_filter	1/49	2.04%
complete_linkage_filter	1/1	100.00%
mosaic_like_filter	1/51	1.96%
near_mosaic_filter	1/2	50.00%
misaligned_reads_filter	1/1	100.00%
clustered_filter	1/1	100.00%
final	1/1	100.00%

The output files are in output_dir, such as `demo_output` in this demo. You can find the final passed list in the output_dir with the file name of `demo_output/final.passed.tsv`, which should contain only one row for the candidate SNM site of 18:70512197. The output .tsv files are in tab-separated format, and please see II.3 for the definition of each column.

I.2 Framework and Best Practice

I.2.a   Program framework and workflow

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

Input .bam file

(Sequentially parse each site in memory)

Pileup for each site (bp)

Each site's pileup information

in-process filters

.filtered.tsv ← in_process filter 1 → .passed.tsv

.filtered.tsv ← in_process filter 2 → .passed.tsv

(if requested)   (if requested)

.filtered.tsv ← in_process filter m → .passed.tsv

Intermediate candidate sites stored in memory

post-process filters

.filtered.tsv ← post_process filter 1 → .passed.tsv

.filtered.tsv ← post_process filter 2 → .passed.tsv

(if requested)   (if requested)

.filtered.tsv ← post_process filter n → .passed.tsv

final.passed.tsv

For running efficiency, by default, the in_process filters will be applied first to each site and then the post_process filters will be applied sequentially. Any filters which are independent to adjacent sites can be in_process filters, while some filters such as the clustered_filter, which requires the information of adjacent sites can only be post_process filters. It is recommended that the much slower misaligned_reads_filter (which calls UCSC blat) is run at the last step.

For 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 characters, 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 describe the suggested config properties for analyzing whole-genome sequencing (WGS) and whole-exome sequencing (WES) data obtained from either normal (non-cancer) samples or cancer samples. You can also find them in the `conf/` directory.

1)   Whole genome sequencing for normal (non-cancerous) samples

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

The flowchart and structure of filters applied to WGS data is shown in this figure. (see also I.2.a)
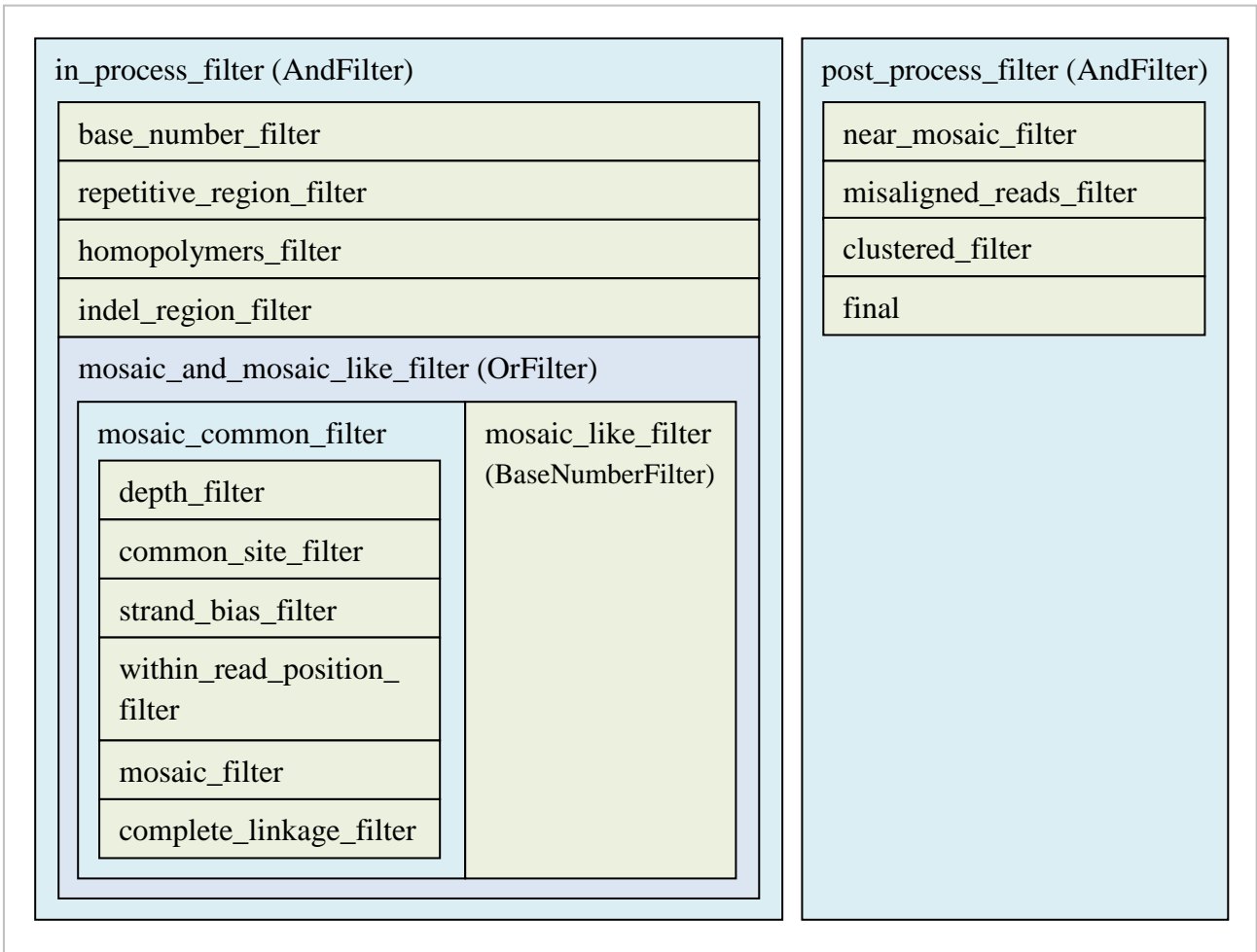
In brief, by masking the error-prone genomic regions (repetitive_region_filter, homopolymers_filter, indel_region_filter) and sites with extremely high or low sequencing depth (base_number_filter),

we identify candidate mosaic sites (mosaic_common_filter) and sites with strange allele fraction (AF) – 'mosaic_like' sites (mosaic_like_filter). We then use near_mosaic_filter to preserve the 'mosaic_like' sites which are closed to candidate mosaic sites only, then check misalignment by calling UCSC blat (misaligned_reads_filter), and remove clustered mosaic candidate sites (which are more likely, due to CNVs or unannotated repetive elements).

Note: mosaic_and_mosaic_like_filter, mosaic_like_filter and near_mosaic_filter are designed to work coupled with clustered_filter, in which we focused on candidate 'mosaic' sites as well as some auxilary nearby 'mosaic_like' sites (see also II.2). Such filters are suitable for WGS, but not WES analysis.

You can run this pipeline with:

```
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 you modify according to your data:

```
max_depth=500
```

```
depth_filter.min_depth=25
```

```
depth_filter.max_depth=150
```

We suggest you set `depth_filter.min_depth` and `depth_filter.max_depth` to be Q10 (10% quantile) and Q90 (90% quantile) of depth in your data. Calling mosaic sites with depth less than 25 may be unreliable.

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 (non-cancerous) samples

1st step: Estimate the shape parameters (α, β) for the beta prior of heterozygous sites

(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)
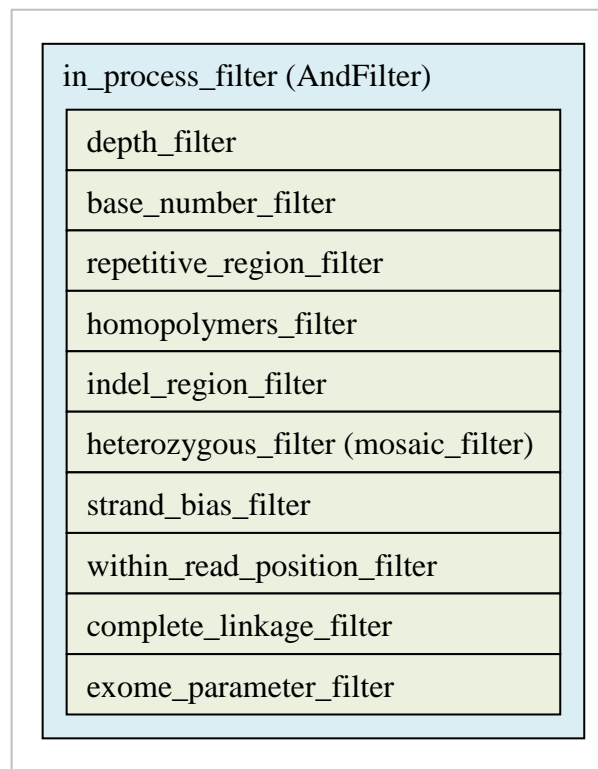
In brief, after applying filters to remove technical artifacts, we estimate the distribution of depths and AFs among heterozygous sites, which will be used in the 2nd step for distinguishing mosaic sites from heterozygous sites. The final output of the average depth and estimated shape parameters (α, β) will be reported to `output_dir`/stdout_*.log.

You can run this step with:

```
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 you modify according to your data:

```
max_depth=5001
```

```
depth_filter.min_depth=25
```
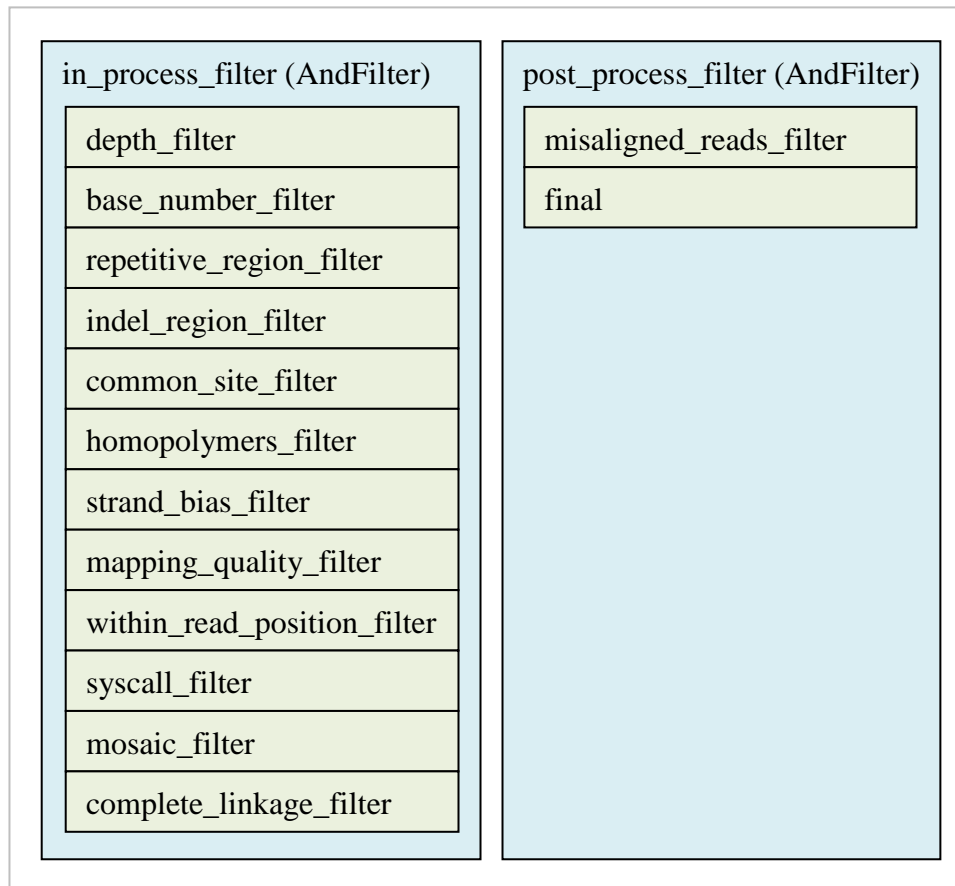
```
depth_filter.max_depth=5000
```

For efficiency issues, we suggest you limit the parameter `max_depth` <= 5000.

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

2nd step: Calling mosaic sites

(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)

Compared to the WGS flowchart, we add mapping_quality_filter and syscall_filter for the WES analysis, and remove clustered_filter as well as 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 `output_dir`/stdout_*.log). We recommend you set a different `output_dir` from the one in the 1st step.

You can run this step with:

```
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 you 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

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

The pipeline for cancerous samples is almost identical to the pipeline for normal (non-cancerous) samples, except that `mosaic_rate` in the mosaic_filter is changed to 1e-6 and `inner_distance` and `outer_distance` in the clustered_filter are changed to 2000, in order to increase the sensitivity to identify SNM in cancer samples.

You can run this pipeline with:

```
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 matched control samples are sometimes unavailable in practice, MosaicHunter implements a Bayesian genotyper to identify SNMs from the unpaired samples ('single' mode). In addition, MosaicHunter is able to utilize the parental sequencing data ('trio' mode) to achieve a better performance, especially for calling SNMs with AF close to 0.5. For the practice of calling somatic mutation in cancer studies, MosaicHunter can also utilize the sequencing data of matched control samples obtained from the same individual ('paired' mode).

1)   Single mode

When no related samples are available (after users prepared its .bam file) it will be easy to call SNMs with the 'single' mode of MosaicHunter: set the .bam file as `input_file`, and set mosaic_filter's parameters to `mode=single` (see also II.2).

The probabilistic graphical model of the Bayesian genotyper (implemented in mosaic_filter) for unpaired samples can be depicted in this figure.

The blue boxes in the figure represent pre-specified prior information or the observed sequencing data (specified as mosaic_filter'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. The mosaic_filter will keep the sites if its mosaic posterior probability P(G=mosaic|data) > mosaic_filter's parameter `mosaic_threshold`.

2) Paired mode

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

In the 'naive paired' mode (mosaic_filter'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 both case and control samples, and summed up the posterior probability that their genotype states were different, and will keep the sites if the probability P(G$_{case}$!=G$_{control}$|data) > mosaic_filter's parameter `mosaic_threshold`. The probabilistic graphical model for 'naive paired' mode can be shown in this figure.
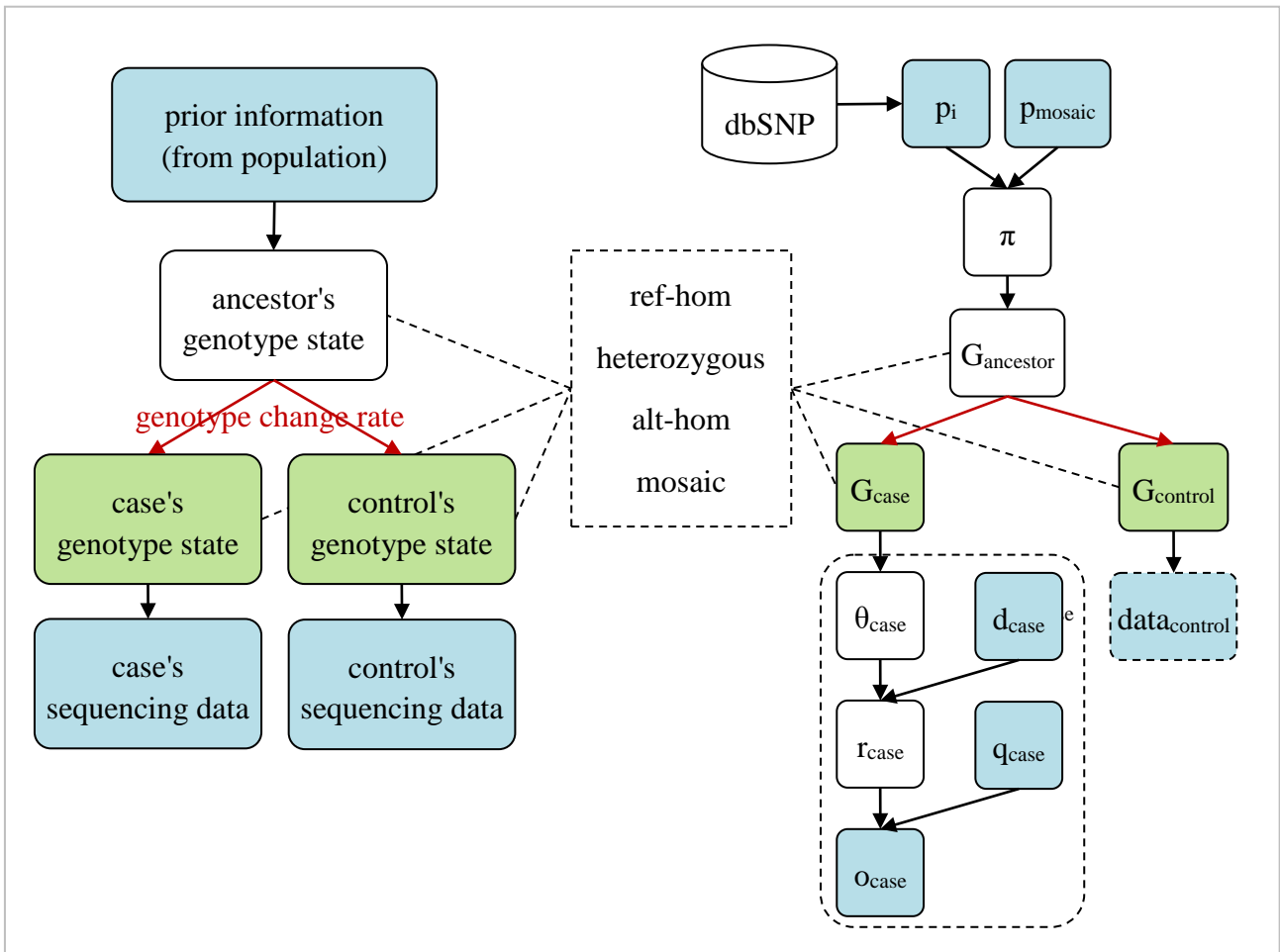

'Fisher paired' mode (mosaic_filter's `mode=paired_fisher`) does not use the Bayesian model, but just compares the refDepth and altDepth in case and control samples using Fisher's exact test for 2*2 contingency table. The mosaic_filter will keep the sites with p-value < `fisher_threshold`.

3) Trio mode

If you have sequenced the individual (child) with his/her parent(s), you can try trio mode (mosaic_filter'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.
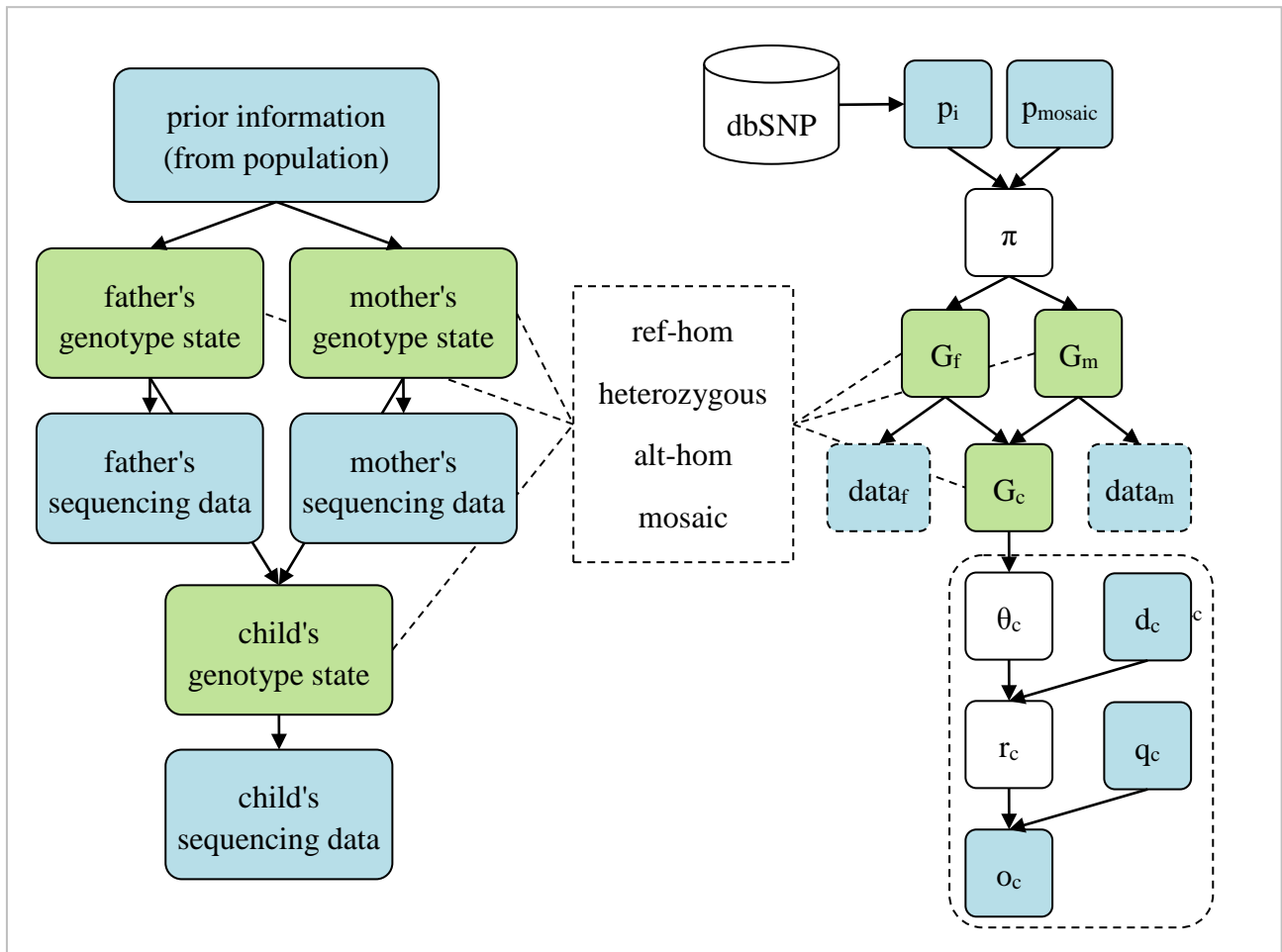
The Bayesian model will keep the sites if the child has SNM (child's mosaic posterior probability > mosaic_filter's parameter `mosaic_threshold`), or the Mendelian inheritance is violated (such as *de novo* mutation). Therefore, in 'trio' mode, you may need to further check the child's likelihood to judge whether the candidate site is an SNM (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 among heterozygous sites)

Something that must be taken into consideration is whether your data is WGS or WES. It has been noticed that the heterozygous sites follow the binomial distribution well in WGS, but poorly in

WES. To address this issue, we replace the prior of theoretical AF θ P(θ|G=heterozygous) from an impulse at 0.5 to a beta prior distribution fitted from the WES 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.

In the 1st step, we applied several error filters to get a list of high-confidence heterozygous sites (mosaic_filter's `mode=heterozygous`), and then used exome_parameter_filter to summarize the AF information, calculate statistics, and report the optimized shape parameters (α, β) estimates in the file `output_dir`/stdout_*.log.

In the 2nd step, MosaicHunter will call SNMs from the WES data with mosaic_filter's parameters `alpha_param=<int>` and `beta_param=<int>`, which are estimated in the 1st step. If they are kept to their default value of 0, 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 impulse at 0.5 (no over-dispersion and mean-shift), we suggest to keep these parameters to the default of 0, for running efficiency.

Except for the change in mosaic_filter, we have also modified and added some other filters for WES, such as mapping_quality_filter and syscall_filter (see also I.3.b and II.2). For the syscall_filter, we adapt the parameters of the SysCall logistic model with different average depth (Meacham *et al. BMC Bioinformatics* 2011). 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 explanations.

`input_file=<path>`

The path of the input .bam file

[Required]

`reference_file=<path>`

The path of the 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 ','); i.e. 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 a 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>`

The reads marked duplicate (with flag 0x400) should be discarded.

[Default: true]

`remove_flag=<int>`

The reads with any bit 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>`

Should the reads be down-sampled (true) or trimmed (false) when the site has depth > max_depth. (see also `max_depth=<int>`)

[Default: false]

`chr=<str>`

If you just want to call SNMs 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>`

The number of reads which are read in buffer.

[Default: 100000]

```
in_process_filter_name=<str>
```

Overall in_process filter's name (change not recommended)

[Default: in_process_filter]

```
post_process_filter_name=<str>
```

Overall post_process filter's name (change not recommended)

[Default: post_process_filter]

II.2 Filter Introduction and Parameters

We develop several stringent filters to remove artifacts due to technical errors as well as other types of genomic variations. For convenience, we call the Bayesian genotyper as mosaic_filter in the java implementation.

It does not matter if the filters are put into the categories of in_process or post_process, they will be checked sequentially. The only difference is that the in_process filters work on each site independently whereas the post_process filters can simultaneously handle all candidate sites altogether. 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 can be changed in the config.properties file. Filter names will be used as a prefix in the output filenames. The names of passed filters for each site will be recorded in the memory, which may be subsequently used, especially for OrFilter (e.g. mosaic_and_mosaic_like_filter), clustered_filter, and near_mosaic_filter in the WGS analysis pipeline (see also I.3.b).

Each filter has two general parameters and some specific parameters 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_filter	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 for general parameters and each filter class.

General parameters

`output_filtered=<bool>`

If the corresponding output <filter_name>.filtered.tsv is generated.

[Default: false]

`output_passed=<bool>`

If the corresponding output <filter_name>.passed.tsv is generated.

[Default: false]

1) AndFilter

[in_process_filter, post_process_filter, mosaic_common_filter]

This logic filter registers some more filters. Only the sites which have passed all the registered

filters will be kept by this filter. The registered filters will be checked sequentially.

`filters=<str>`

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

[Default: (empty)]

2) OrFilter

[mosaic_and_mosaic_like_filter]

This logic filter registers some more filters. The sites which have passed any of the registered filters will be kept 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 ' ' are allowed.

[Default: (empty)]

3) DepthFilter

[depth_filter]

This filter will keep 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 checks the observed minor allele count and minor AF. The sites with minor allele in specified range will be kept.

```
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>
```

The number of bp to expand for each block.

[Default: 5]

```
include=<bool>
```

If you want to filter out sites within 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' homopolymers.

[Default: 4]

`short_homopolymer_expansion=<int>`

The number of bp for expanding each 'short' homopolymer, i.e. filter out those sites close to a 'short' homopolymer.

[Default: 2]

`long_homopolymer_length=<int>`

Define the length of 'long' homopolymers

[Default: 6]

`long_homopolymer_expansion=<int>`

The number of bp for expanding each 'long' homopolymer, i.e. filter out those sites close to a 'long' homopolymer.

[Default: 3]

7) MosaicFilter

[mosaic_filter]

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

Output .tsv files (filtered.tsv, passed.tsv) will be different for different modes (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 and 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; -1 means that the variant is recorded in dbSNP but has no allele frequency information.

[Required]

`unknown_af=<double>`

The specified population allele frequency for those variants that are recorded in dbSNP but has no allele frequency information.

[Default: 0.002]

`novel_af=<double>`

The specified population allele frequency for those 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 of AF on heterozygous sites, so we changed the prior of theoretically AF θ from impulse at 0.5 to a beta distribution, with two shape parameters (α, β) specified here. The two shape parameters (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 $\theta = 0.5$. In fact, to trigger on the heterozygous beta prior, both parameters should be > 0 and their sum should be > 2.

For efficiency issues, we suggest that both parameters should be <= 1000. For larger estimated parameters, which indicate closeness of the beta prior to the $\theta = 0.5$ impulse, we suggest to trigger off this part by setting the default (0, 0), and turn back to the $\theta = 0.5$ 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 a sum of 1.

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

`mosaic_threshold=<double>`

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

[Default: 0.05]

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

Specify the Bayesian 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 the 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 keep 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 by strand bias (major strand+, major strand-, minor strand+, minor strand-), and filters out sites with small p-values.

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 kept.

[Default: 0.05]

9) WithinReadPositionFilter

[within_read_position_filter]

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

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

11: major read position

12: minor read position

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

`min_p_value=<double>`

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

[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: counts of four kinds of haplotypes

17: p-value of Fisher's exact test

`max_p_value=<double>`

We perform Fisher's exact test on the 2*2 contingency table of counts 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 kept 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).

Kept sites require an adequately large proportion of reads with high-confidence alignment (see also `max_misalignment_percentage=<double>`): 1) the blat alignment result of a read is unique, and agrees with the alignment result of the corresponding read in the .bam file (e.g. by bwa) (see also `min_overlap_percentage=<double>`); 2) the candidate site isn't 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 differences.

[Default: Same as top parameter `reference_file`]

`max_misalignment_percentage=<double>`

Kept sites should have a proportion of 'misaligned' reads < this_value.

[Default: 0.5]

`min_overlap_percentage=<double>`

Each read will have one mapping block by blat and one from .bam (e.g. 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 end 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) ClusteredFilter

[clustered_filter]

Because some regions on the genome may enrich sequencing and mapping artifacts, we see some false-positive candidate sites clustered (with close positioning). This filter will remove the 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 have a distance of < 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 regions to upstream and downstream this_value bp, and the mosaic candidate sites falling in the region will be filtered.

[Default: 20000]

`auxiliary_filter_name=<str>`

The passed filter name marking the auxiliary sites.

[Default: mosaic_like_filter]

14) OutputFilter

[final]

The final 'filter' for outputting the result list of the kept candidate sites.

15) ExomeParameterFilter

[exome_parameter_filter]

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

`min_group_size=<int>`

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

[Default: 50]

`optimal_depth=<int>`

When the observed relationship between the 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 again and fitted.

[Default: "r_het_data.tsv"]

16) NullFilter

[null_filter]

This filter will discard all the sites to clear up memory. Only useful after the filter has done statistics or estimating parameters (e.g. ExomeParameterFilter).

17) MappingQualityFilter

[mapping_quality_filter]

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

```
min_p_value=<double>
```

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

[Default: 0.05]

18) SysCallFilter

[syscall_filter]

This filter applies the logistic regression model developed and introduced by SysCall (Meacham *et al. BMC Bioinformatics* 2011), to filter out exome false positives. The trained parameters vary across different depths 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 of the output files are in the specified directory `output_dir`, including the final candidate list final.passed.tsv, the filtered and remaining variant lists of each filters (<filter_name>.filtered.tsv, <filter_name>.passed.tsv), and other temporary files, such as blat input and output.

The output .tsv files are in tab-separated format, whose columns' meanings are 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 modes.

1) 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) 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

21-24: case log10 likelihood of major-homozygous, heterozygous, minor-homozygous, mosaic

25-28: control log10 likelihood of major-homozygous, heterozygous, minor-homozygous, mosaic

29-44: log10 joint posterior matrix

   29-32 case = major-homozygous,   33-36 case = heterozygous,
   37-40 case = minor-homozygous,   41-44 case = mosaic;

   29 33 37 41 control = major-homozygous,   30 34 38 42 control = heterozygous,
   31 35 39 43 control = minor-homozygous,   32 36 40 44 control = mosaic

45: log10 posterior probability that the genotype state of case != control

3) 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)   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)   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 Information

Team:

August Yue Huang

Adam Yongxin Ye

Zheng Zhang

Yanmei Dou

Liping Wei (PI)

@Center for Bioinformatics, Peking University

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

III.3 Citation

Please Cite: Huang, A.Y., Zhang, Z., Ye, A.Y., Dou, Y., Yan, L., Yang, X., Zhang, Y., and Wei, L. (2017) MosaicHunter: accurate detection of postzygotic single-nucleotide mosaicism through next-generation sequencing of unpaired, trio, and paired samples. *Nucleic Acids Res* **45**, e76.

III.4 License

MosaicHunter is licensed under the MIT License.

III.5 FAQ