TrioMix

Release 0.0.1

Christopher J. Yoon

Dec 07, 2022

CONTENTS

TrioMix is a bioinformatics tool to detect **intrafamilial contamination**, **chimerism**, **uniparental disomy** by investigating inheritance patterns of SNPs. TrioMix quantifies the deviation from Mendelian inheritance patterns by using maximum likelihood esstimation (MLE) inferred from the genotypes of parent-offspring trio. TrioMix can be used on both whole-genome sequencing (WGS) of trios or whole-exome sequencing (WES) of trios.

TrioMix can be installed by cloning the github or using the pre-built docker image.

Check out the *Usage* section for further information, including how to *Install* the project.

If you use Triomix in your analysis, please cite our work below.

> **"Estimation of intrafamilial DNA contamination in family trio genome sequencing using Mendelian inconsistencies. Yoon et al., Genome Research (2022)"** [link to paper]

CHAPTER

ONE

CONTENTS

1.1 Install

1.1.1 Install

To use TrioMix, first install it using git:

```
$ git clone https://github.com/cjyoon/triomix.git
```

To verify installation, you can use the following commands. If everything is setup correctly, the following commands should result in the TrioMix help menu being printed in your terminal.

```
$ cd triomix
$ python triomix.py -h
```

1.1.2 Dependencies

TrioMix is written in Python (v3.5 or later) and R. Following Python and R packages are used in TrioMix.

- **Python**
  - pysam
  - pandas
- **R**
  - optparse
  - tidyverse
  - bbmle
  - PSCBS

1.1.3 Other dependencies

TrioMix internally uses `samtools`, `Rscript`, and `gzip`. Make sure these are in your `$PATH`. Otherwise, you can edit the absolute path of each of these in `path_config.json`.

```
$ cat path_config.json
{"SAMTOOLS": "/path/to/samtools", # default is 'samtools'
 "RSCRIPT": "/path/to/Rscript", # default is 'Rscript'
 "GZIP": "/path/to/gzip" # default is 'gzip'
}
```

1.1.4 Using the docker images

Pre-built docker image with all prerequisites installed is available.

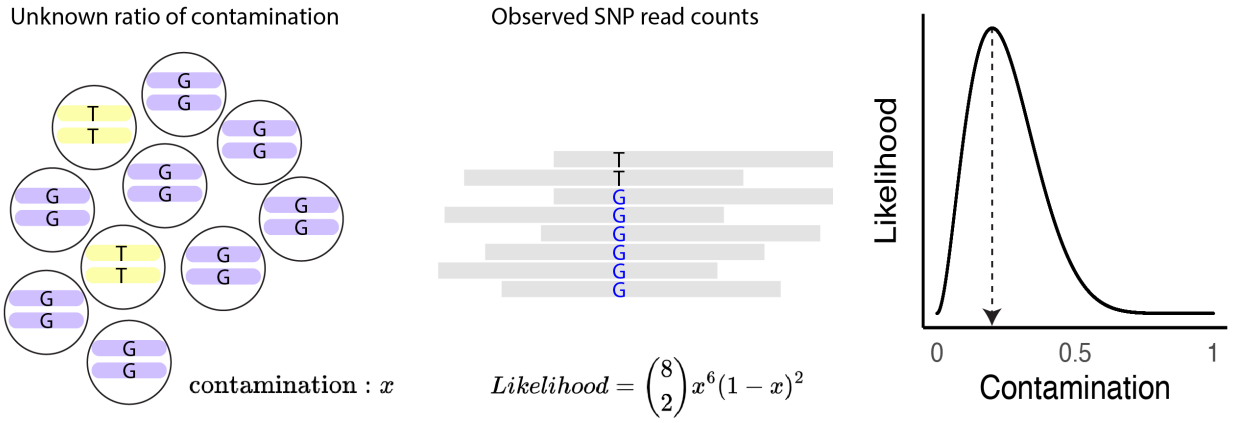
```
$ docker pull cjyoon/triomix
```

1.2 Introduction

1.2.1 Introduction

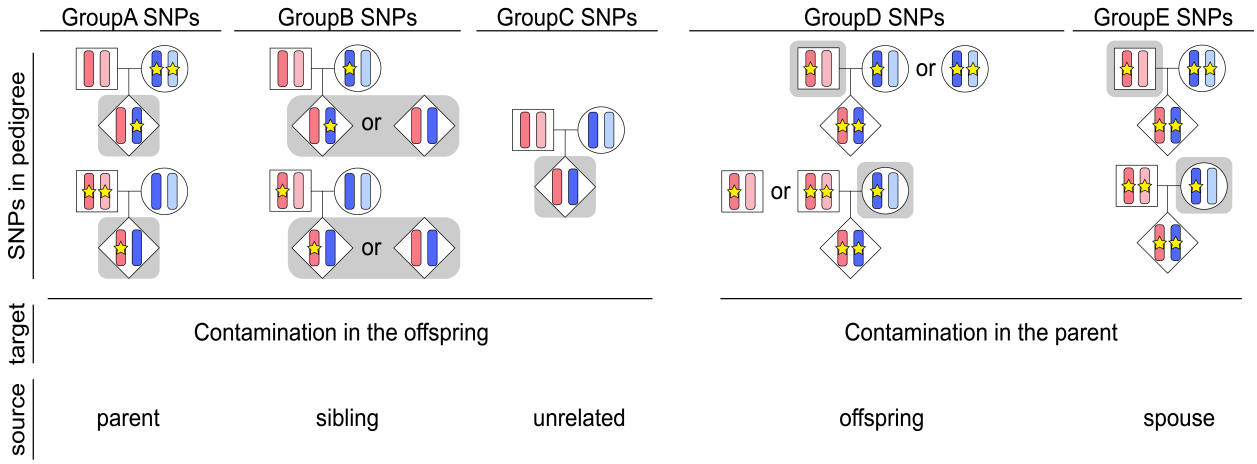
TrioMix is a bioinformatics software that quantifies within-family DNA contamination, chimerism, and uniparental disomy from a sequence data of family trio. From the genotypes of the family members, TrioMix builds a maximum likelihood estimation model to accurately quantify DNA contamination by measuring the deviation from the Mendelian inheritance.

If the genotypes of the parents are known, then the offspring's genotype can be inferred. Thus, the VAF of uncontaminated offspring can be estimated from the parental genotypes. If there is contamination, then there would be a deviation from the expected VAFs which can be measured with maximum likelihood estimation.



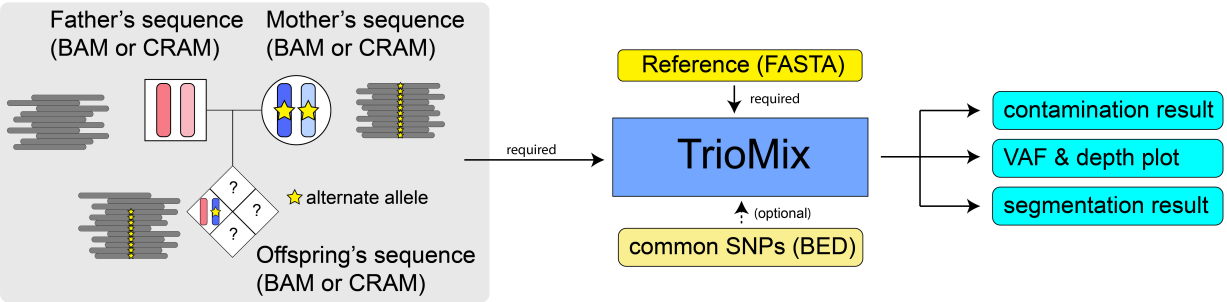
1.2.2 SNP groups used in TrioMix

TrioMix uses 5 different SNP groups in order to calculate different scenarios of contaminations. GroupA, GroupB, and GroupC are used to detect contamination in the child, and GroupD and GroupE are used to detect contamination in one of the parents.



1.3 Usage

TrioMix requires the input of sequence alignment file (BAM or CRAM files) of trios and a reference FASTA file. SNP BED file can be used as an optional argument.



1.3.1 Basic TrioMix command line: Detection of intrafamilial contamination in the offspring

By default, TrioMix uses the parental genotypes (*GroupA, B, C SNPs*) to infer the intrafamilial contamination level in the offspring. Since `-o` is commonly reserved for outputs, we use `-c, --child` to refer to the offspring. The basic command line of using TrioMix is the following:

```
$ python triomix.py -f father.bam -m mother.bam -c child.bam -r reference.fasta
```

1.3.2 TrioMix command line with common SNP only

Using a pre-selected list of common SNP would speed up the total runtime of TrioMix as the computation is limited to those regions instead of the entire genome. TrioMix provides a list of common `GRCh38` and `GRCh37` SNPs selected from the GnomAD database. These two files are included in the github repository as a common_snp folder. A `-s` argument specifies the SNP database that can be used. User can provide one's own set of SNP in BED format.

```
$ python triomix.py -f father.bam -m mother.bam -c child.bam -r reference.fasta -s␣
↪common_snps/grch38_common_snps.bed.gz
```

1.3.3 Command line arguments

```
$ python triomix.py -h
usage: triomix [-h] [--version] -f FATHER -m MOTHER -c CHILD -r REFERENCE [-s SNP] [-t␣
↪THREAD] [-o OUTPUT_DIR]
                [-p PREFIX] [--runmode {single,joint,all}] [-u {0,1}] [--parent] [-d␣
↪DOWNSAMPLE]

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  -f FATHER, --father FATHER
                        Father's BAM or CRAM file
  -m MOTHER, --mother MOTHER
                        Mother's BAM or CRAM file
  -c CHILD, --child CHILD
                        Child's BAM or CRAM file
  -r REFERENCE, --reference REFERENCE
                        Reference FASTA file
  -s SNP, --snp SNP     Optional list of SNP sites as a BED (or BED.gz) file
  -t THREAD, --thread THREAD
                        Multithread to utilize. Default=1
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        Output directory. Default=current working directory
  -p PREFIX, --prefix PREFIX
                        prefix for the output file. If not specified, will use the SM␣
↪tag from the child bam's
                        header
  --runmode {single,joint,all}
                        Runmode for mle.R script. 'single' assumes only 1 contamination␣
↪source within family.
                         'joint' calculates the fraction of all family members jointly.
↪'all' runs both modes.
                        Default=all
  -u {0,1}, --upd {0,1}
                        0: mle will filter out vaf=0 or 1 in sites where parental␣
↪genotypes are homo-ref + homo-alt
                        (GroupA SNPs) 1: mle will identify UPDs which appears as␣
↪contamination. Default=1
  --parent              Run detection of parental DNA contamination with child's DNA
  -d DOWNSAMPLE, --downsample DOWNSAMPLE
                        Downsampling for plotting.
```

1.3.4 Default output files

TrioMix produces several output files files.

`*.x2a.depth.tsv`: contains the depth ratio chrX vs autosome of each individual in a trio. Males are expected to have ~0.5 while female should have value ~1.0.

`*.child.counts`: contains the position of the SNP loci in either GroupA, B, or C. Contains the read depths, alternative read counts for the trios. In addition, based on the parental genotype, will determine whether the child inherited the SNP from the father (F) or the mother (M). This file is used as the input for `mle.R` which estimates the contamination level using maximum likelihood estimation.

`*.child.counts.upd.segments.tsv`: contains the VAF values for GroupA SNPs that have been segmented for UPD analysis

`*.child.counts.plot.pdf`: visualization of depth and VAF plots of GroupA and GroupB SNPs in the child.

`*.child.counts.summary.tsv`: contains the final estimated values of contamination from various sources in the child. Detailed information on each column is as follows.

```
child_contam_by_sibling_joint # contamination estimated from joint analysis of all␣
↪family members (GroupA + GroupB used)
child_contam_by_father_joint # contamination estimated from joint analysis of all family␣
↪members (GroupA + GroupB used)
child_contam_by_mother_joint # contamination estimated from joint analysis of all family␣
↪members (GroupA + GroupB used)
convergence_joint # mle function convergence status. If 0, then indicates convergence␣
↪succeeded.
child_contam_by_sibling # contamination estimated assuming only sibling contaminating␣
↪(GroupB used)
child_contam_by_father # contamination estimated assuming only father contaminating␣
↪(GroupA used)
child_contam_by_mother # contamination estimated assuming only mother contaminating␣
↪(GroupA used)
groupA_father # number of paternal GroupA variants identified
groupA_mother # number of maternal GroupA variants identified
groupB_father # number of paternal GroupB variants identified
groupB_mother # number of maternal GroupB variants identified
denovo_error_rate # fraction of alternative read count at GroupC SNPs
```

1.3.5 TrioMix with whole-exome sequencing

TrioMix can be used with whole-exome sequencing. In this case, we recommend running the command without the `-s common_snp/common_snps.bed.gz` to capture rare SNPs as well. This increases the overall number of SNPs while having minimal effect on the computational time due to smaller target in the exome sequeincing. For plotting, using `-d 1` is recommended to capture all data points in the plot without downsampling.

```
$ python triomix.py -f father.bam -m mother.bam -c child.bam -r reference.fasta -d 1
```

1.3.6 Detection of intrafamilial contamination in the parent (i.e. parent DNA contaminated by child, or by another parent)

To detect intrafamilial DNA contamination in the parent, `--parent` option can be used. This will use *GroupD SNPs* (where offspring's genotype is *homo-alt*) to detect the offspring DNA contaminating in the parents.

```
$ python triomix.py -f father.bam -m mother.bam -c child.bam -r reference.fasta -s␣
↪common_snps/grch38_common_snps.bed.gz --parent
```

1.3.7 Additional output generated with `--parent`

`*.parent.counts`: contains the position of the SNP loci in either Group D or E. Contains the read depths, alternative read counts for the trios. This file is used as the input for `mle_parent.R` which estimates the contamination level using maximum likelihood estimation.

`*.parent.counts.plot.pdf`: visualization of depth and VAF plots of GroupD and GroupE SNPs in the parents.

`*.parent.counts.summary.tsv`: contains the final estimated values of contamination from various sources in each parents. Detailed information on each column is as follows.

```
mother_contam_by_child  # contamination estimated in the mother (GroupD)
father_contam_by_child  # contamination estimated in the father (GroupD)
mother_contam_by_father # contamination estimated in the mother (GroupE)
father_contam_by_mother # contamination estimated in the fother (GroupE)
groupD_mother   # number of maternal GroupD variants identified
groupD_father   # number of paternal GroupD variants identified
groupE_mother   # number of maternal GroupE variants identified
groupE_father   # number of paternal GroupE variants identified
```

1.3.8 Running TrioMix with a docker image

Following example demonstrates how docker image can be used for runnint TrioMix.

```
# Download docker image from dockerhub
$ VERSION=v0.0.1 # download specific release version tag of TrioMix
$ docker pull cjyoon/triomix:$VERSION

# Run triomix with docker image
$ docker run \
   -v /path/to/bamfile:/path/to/bamfile \ # bind all folders where input files are␣
↪located
   -v /path/to/reference:/path/to/reference/ \
   -v /path/to/output_dir:/path/to/output_dir \ # also bind the location of output folder
   -it cjyoon/triomix:$VERSION  \
      python /tools/triomix/triomix.py \ # location of triomix.py in the docker image
         -f /path/to/bamfile/father.bam \ # location of father's bam file
         -m /path/to/bamfile/mother.bam \ # location of mother's bam file
         -c /path/to/bamfile/mother.bam \  # location of child's bam file
         -s /tools/triomix/common_snp/grch38_common_snp.bed.gz \ # location of common␣
↪SNP file in the docker image
         -r /path/to/reference/reference.fa \ # location of reference FASTA file
         -o /path/to/output_dir # location where output files are saved
```

1.4 Example

1.4.1 Example workflow

Contamination can be simulated by randomly selecting read sequences from two BAM (or CRAM) files at a defined ratio. The total number of reads in each bam file is adjusted when creating a subsampled BAM file which is later merged into a single 'contaminated' bam file.

1.4.2 Test run with 1000 genomes trio

In our github a test script `test.sh` is provided which can created a simulated contamination BAM files from a 1000 genomes family.

```
$ sh test.sh
```

This will download the 1000 genomes trio CRAM files, create simulated contamination files by subsampling. Note this will requires `samtools` to be installed in your `$PATH`. Below, a few example cases in the `test.sh` are described.

1.4.3 Download a 1000 genomes trio (+ a sibling) CRAM files

The following command downloads a 1000 genomes trio (+ a sibling) into current working directory

```
# download M008 family's WGS from 1000 genomes project ftp.
# proband
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR398/ERR3989418/NA19662.final.cram  -O␣
↪proband.cram
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR398/ERR3989418/NA19662.final.cram.crai  -O␣
↪proband.cram.crai

# father
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR323/ERR3239902/NA19661.final.cram -O father.
↪cram
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR323/ERR3239902/NA19661.final.cram.crai -O␣
↪father.cram.crai

# mother
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR398/ERR3989417/NA19660.final.cram -O mother.
↪cram
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR398/ERR3989417/NA19660.final.cram.crai -O␣
↪mother.cram.crai

# sibling
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR398/ERR3989425/NA19685.final.cram -O␣
↪sibling.cram
wget -nc ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR398/ERR3989425/NA19685.final.cram.crai -O␣
↪sibling.cram.crai

# download the reference fasta file
wget -nc https://storage.googleapis.com/genomics-public-data/resources/broad/hg38/v0/
↪Homo_sapiens_assembly38.fasta
samtools faidx Homo_sapiens_assembly38.fasta
```

1.4.4 Offspring DNA contaminated by mother's DNA

The following scripts create a simulated contamination consisting of 75% offspring's DNA + 25% mother's DNA. SCRIPTPATH should be set to the folder where `triomix.py` is located.

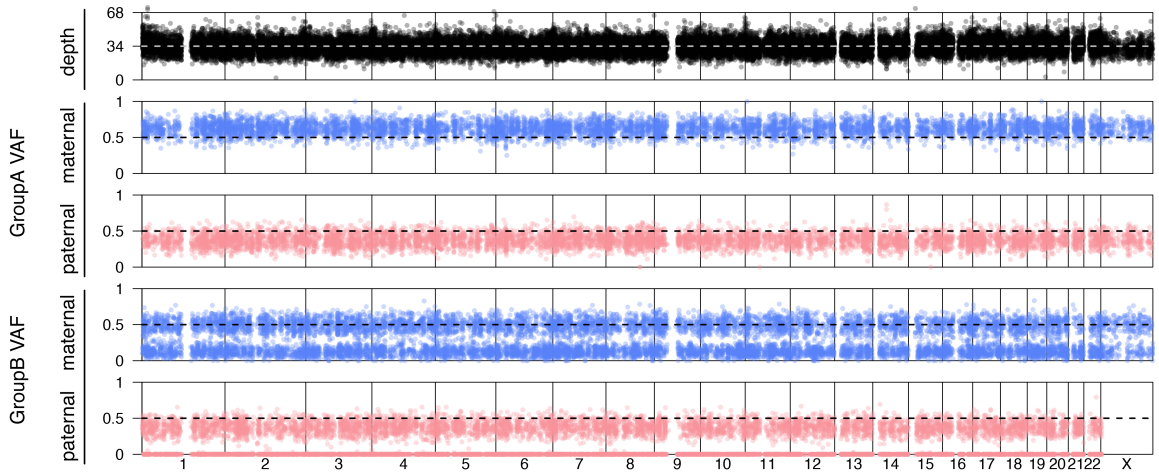
```
SCRIPTPATH=/path/to/triomix # change this to triomix's github folder path. This is␣
↪automatically detected in test.sh

#################################
# offspring contaminated by mother simulation
python $SCRIPTPATH/simulate_familial_mixture.py \
 -f father.cram \
 -m mother.cram \
 -c proband.cram \
 -s sibling.cram \
 -r 0 0.25 0.75 0 -o offspring75_mother25 # 0% father, 25% mother, 75% offspring 0%␣
↪sibling

# run TrioMix on offspring contaminated by mother
python $SCRIPTPATH/triomix.py \
 -f father.cram \
 -m mother.cram \
 -c offspring75_mother25/familymix.bam \
 -r Homo_sapiens_assembly38.fasta -t 8 \
 -s $SCRIPTPATH/common_snp/grch38_common_snp.bed.gz \
 -p offspring75_mother25 -o results
```

This will produce the contamination estimation file `offspring75_mother25.child.counts.summary.tsv`

```
$ cat offspring75_mother25.child.counts.summary.tsv
type  value
child_contam_by_sibling_joint 1.8873588344360515e-4
child_contam_by_father_joint  8.167772506667559e-5
child_contam_by_mother_joint  0.2507938532424764
convergence_joint 0
child_contam_by_sibling 0.2533023916508015
child_contam_by_father  6.474095861474213e-9
child_contam_by_mother  0.2507121759058531
groupA_father  73041
groupA_mother  73077
groupB_father  125753
groupB_mother  124975
denovo_error_rate 3.992565674974727e-4
```

Joint estimation assuming all possible contamination from all family members estimated 25% contamination only from the mother. However, if we look at the estimation assuming only one individual at a time, fitting the same data may show maximum likelihood with 25% contamination by the father and also 25% contamination of the mother. Thus, the joint method provides the most definitive contamination estimation.

A genome wide plot for this simulated case is shown.



1.4.5 Offspring DNA contaminated by father, mother, and a sibling

Here, we simulate a complex case where the offspring's DNA is coontaminated by the father, mother, and a sibling simultaneously.

```
###################################
# multiple contamination simulation, father=10%, mother=20%, offspring=40%, sibling=30%
python $SCRIPTPATH/simulate_familial_mixture.py \
 -f father.cram \
 -m mother.cram \
 -c proband.cram \
 -s sibling.cram  \
 -r 0.10 0.20 0.40 0.30 -o complexmix # 10% father, 20% mother, 40% offspring 30% sibling

# run TrioMix on the complex contaminated case
python $SCRIPTPATH/triomix.py \
 -f father.cram \
 -m mother.cram \
 -c complexmix/familymix.bam \
 -r Homo_sapiens_assembly38.fasta -t 8 \
 -s $SCRIPTPATH/common_snp/grch38_common_snp.bed.gz \
 -p complexmix -o results
```

This will produce the contamination estimation file `complexmix.child.counts.summary.tsv`

```
$ cat complexmix.child.counts.summary.tsv
type  value
child_contam_by_sibling_joint 0.2900164906231494
child_contam_by_father_joint  0.09759458292835317
child_contam_by_mother_joint  0.2006493758684508
convergence_joint 0
child_contam_by_sibling 0.29732892004146544
child_contam_by_father  8.909208856139373e-9
child_contam_by_mother  0.10305479295638349
groupA_father  73041
groupA_mother  73077
groupB_father  125753
```

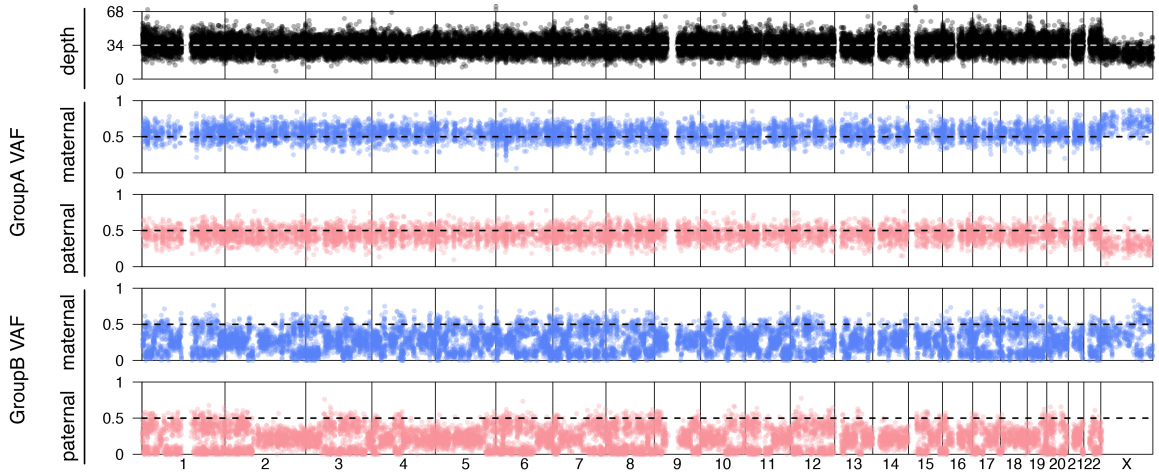
(continues on next page)

(continued from previous page)

```
groupB_mother  124975
denovo_error_rate 3.1783417104182737e-4
```

Joint estimation of all family members accurately estimated the 10% father's contamination, 20% mother's contamination, and 30% sibling's contamination in the offspring's DNA. In the single contamination estimation mode, only the difference between the father and mother is measured at 10%.

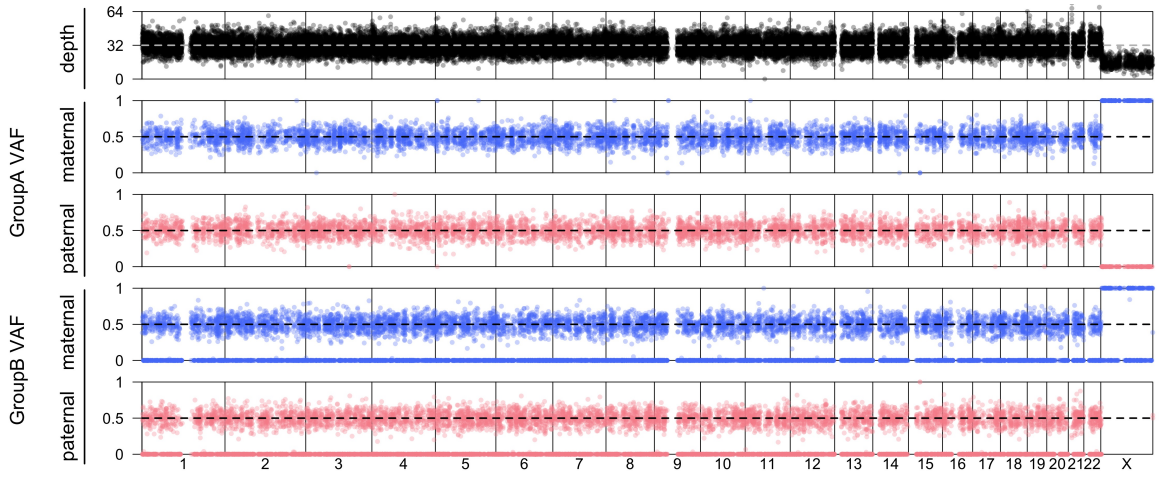
A genome wide plot for this simulated case is shown.



1.5 Plots

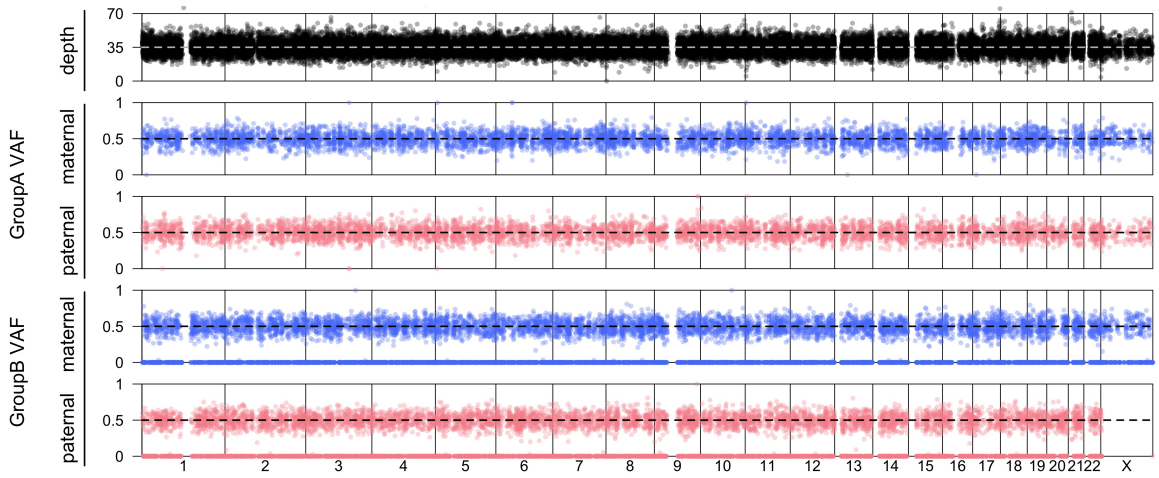
Here are some real case example plots from TrioMix.

1.5.1 Normal male offspring, uncontaminated



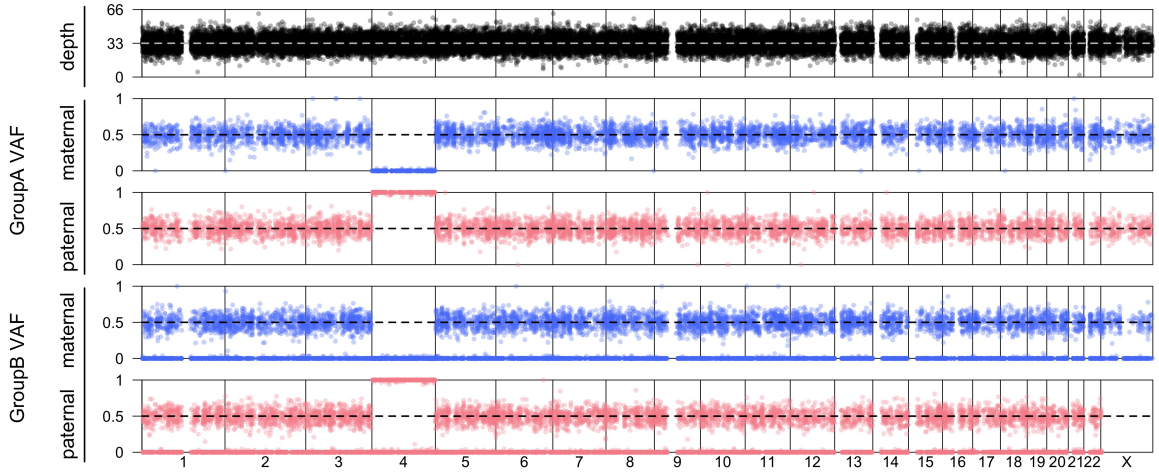
This is an example of SNP VAFs in a male offspring without contamination. Note the drop in depth and homozygosity of chrX.

1.5.2 Normal female offspring, uncontaminated



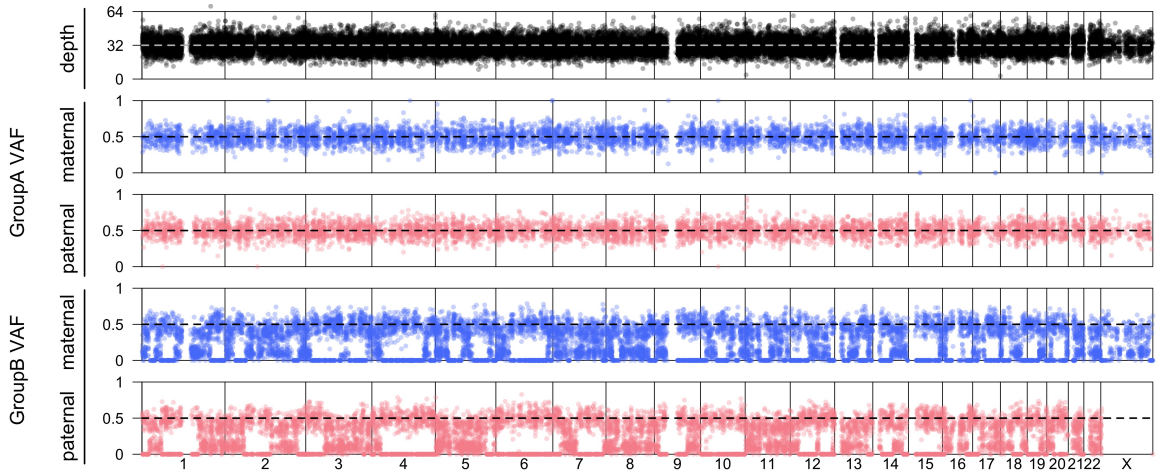
This is an example of SNP VAFs in a female offspring without contamination. Note that there is no paternal GroupB SNP on chrX since father only has one copy of chrX and cannot have heterozygous SNPs.

1.5.3 Uniparental disomy



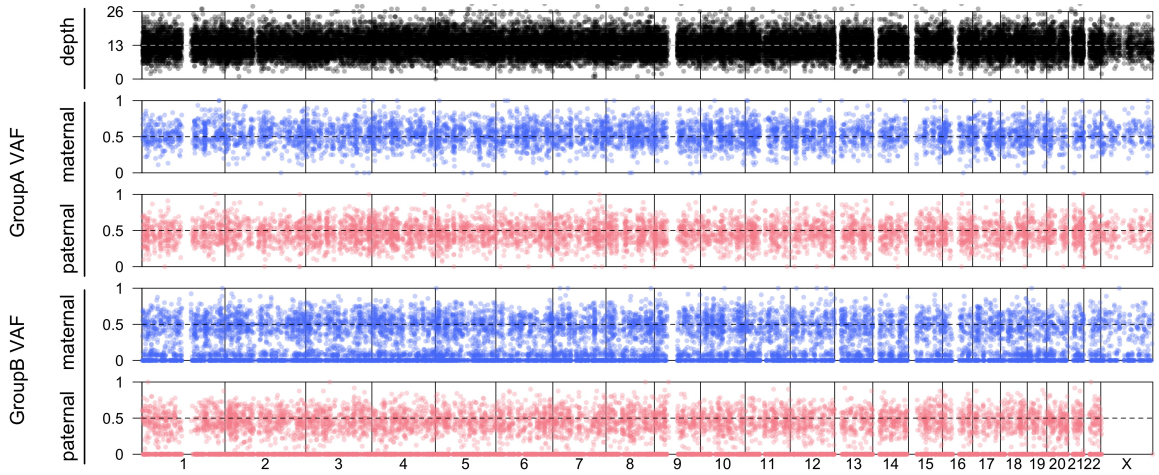
Paternal uniparental isodisomy (UPiD) of chr4 is shown. Homozygosity of GroupA SNPs suggests the presence of uniparental disomy. Homozygosity of GroupB SNPs further suggests uniparental isodisomy (UPiD).

1.5.4 Chimerism (from a sibling)



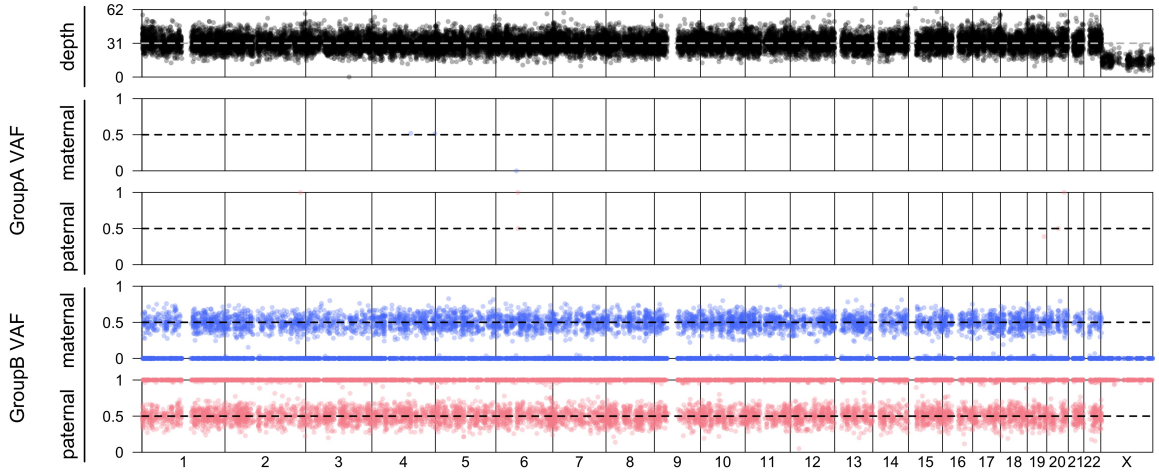
The meiotic recombination pattern between the two sibling zygotes are shown as segmental VAF patterns in GroupB SNPs. In this example TrioMix quantified the ratio between the two zygotes of the chimera 22%:78%.

1.5.5 Maternal contamination in a placenta biopsy



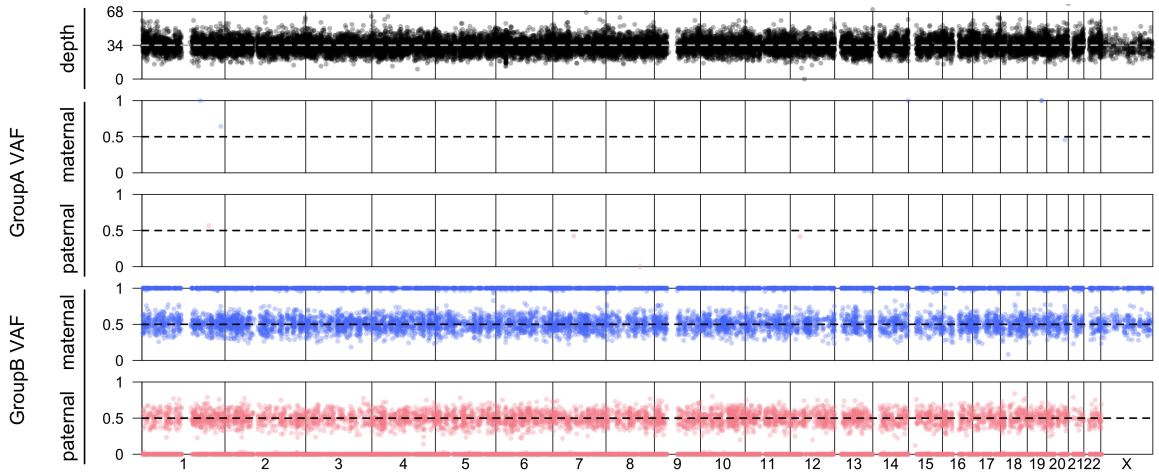
Maternal blood can contaminate the fetal portions of the placenta biopsy. Maternal GroupA VAFs are increased (mean VAF=0.53) and paternal GroupA VAFs are decreased (mean VAF=0.47). In GroupB, additional non-zero low level VAFs are seen only from the mother which suggests DNA contamination originating from the mother. In this example, TrioMix quantified that there is 6.6% maternal DNA contamination in the placenta biopsy.

1.5.6 Sample swap (father offspring)



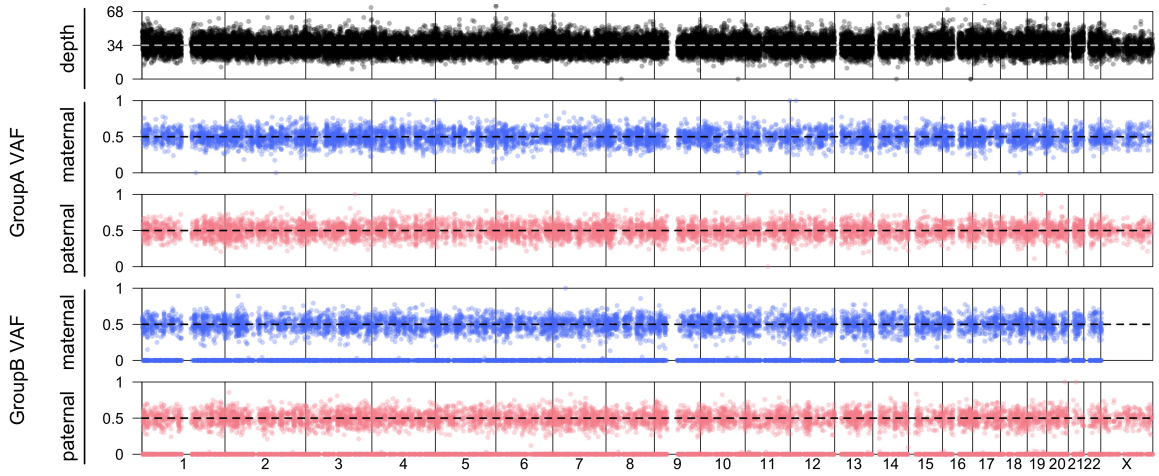
Sample swap between the father and offspring would lead to no GroupA SNPs since a offspring and another parent can both be homozygous for a different allele at the same time (i.e. offspring: *homo-alt*, father: *homo-ref*). Thus, there is no GroupA variants. For GroupB SNPs, if the offspring is a *het* genotype, then the father can be a *het* or *homo-alt* genotype. Thus a *homo-alt* (VAF=1) is seen in GroupB in the parent that is swapped with an offspring.

1.5.7 Sample swap (mother offspring)



Sample swap between the mother and offspring would lead to no GroupA SNPs since a offspring and another parent can both be homozygous for a different allele at the same time (i.e. offspring: *homo-alt*, mother: *homo-ref*). Thus, there is no GroupA variants. For GroupB SNPs, if the offspring is a *het* genotype, then the mother can be a *het* or *homo-alt* genotype. Thus a *homo-alt* (VAF=1) is seen in GroupB in the parent that is swapped with an offspring.

1.5.8 Sample swap (father mother)



In the absence of parent sample swap, GroupB is only seen with maternal SNP in chrX since the requirement for GroupB is heterozygous in that parent. For the father with XY genotype, therefore, GroupB SNP is not available. Thus, if the two parents are swapped, 'paternal' chrX GroupB SNPs will be observed instead of 'maternal' chrX GroupB.

1.6 FAQ

1.6.1 FAQ