

# Supplementary - trioPhaser: Using Mendelian inheritance logic to improve genomic phasing of trios.

Dustin Miller

9/12/2021

## Contents

<b>1</b>	<b><i>trioPhaser</i> Docker Image</b>	<b>1</b>
<b>2</b>	<b><i>trioPhaser</i> results on Ashkenazim and Han Chinese trios</b>	<b>2</b>
<b>3</b>	<b>Ashkenazim trio gVCF test data</b>	<b>7</b>
<b>4</b>	<b>Comparison of <i>trioPhaser</i> to linked-read phasing technology (10X)</b>	<b>8</b>
<b>5</b>	<b>Comparison of <i>trioPhaser</i> to <i>whatshap</i></b>	<b>9</b>
<b>6</b>	<b><i>trioPhaser</i> results across 50 trios</b>	<b>10</b>
<b>7</b>	<b>Test <i>trioPhaser</i>'s GRCh37 functionality</b>	<b>14</b>
	<b>References</b>	<b>16</b>

## 1 *trioPhaser* Docker Image

### 1.0.1 Install Docker and download the *trioPhaser* Docker image

*trioPhaser* is executed within a Docker container which contains all necessary software involved in the phasing process. The container can be executed on any operating system as long as the Docker engine is installed. Instructions for installing Docker are found at [here](#). Once installed, the Docker image can be downloaded using the following command:

```
docker pull dmiller903/triophaser:latest
```

### 1.0.2 Docker basics

Below we provide example code to show how Docker is used throughout this document. When running a Docker container, the command `docker run` is used. In the example below, our full docker command is `docker run -v /Data:/proj -w /proj -t dmiller903/triophaser:latest ls`. `docker run -v`

`/Data:/proj -w /proj -t dmill903/triophaser:latest` is used to set-up and run the container, and `ls` is the command being executed within the container. `-v` indicates which directory Docker has access to on your local computer. In our code below, we use `-v /Data:/proj` which means we have a directory on our local machine, `/Data`, that Docker can access, and within the container we are giving the container the name `/proj`. The name the directory is known as within the container is subjective. For example, `-v /Data:/Data`, would indicate to Docker that you want it to be able to access the `/Data` directory on your local machine and it will be known as the same name, `/Data`, within the container. We set the working directory within the container to `/proj` using `-w /proj`. The `-t` allows the container to execute commands using a “terminal”. `dmill903/triophaser:latest` is the image we want Docker to run as a container. Anything that comes after the image name is executed within and by the container. In this case, `ls` is being executed within and by the container. Therefore, the below code will show you the files and directories within the `/Data` directory. If you want to try this code, change `/Data` to a directory that is available on your local machine.

```
docker run -v /Data:/proj -w /proj -t dmill903/triophaser:latest ls
```

We often use `-d` as part of our Docker command. This executes the container in “detached” mode which allows the container to run in the background. This is useful for tasks that take a long time to execute. When the `-d` option is used, the container ID is output upon execution. The container ID is output to the terminal, or so you don’t have to keep track of the ID, `>` can be used to store the container ID to a file. In the below example, we are storing the container ID to the file `ls.out`.

```
docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest ls > ls.out
```

To view the contents output by the above command, Docker requires a container ID preceded by `docker logs`. In this case, the container ID is stored in the `ls.out` file. To view the contents of the container ID, you can use a command such as:

```
CONTAINERID="cat ls.out"
docker logs '$CONTAINERID' >> ls.out
```

`CONTAINERID="cat ls.out"` will store the container ID as a variable, then `docker logs '$CONTAINERID' >> ls.out` will use the container ID to append the container output to the `ls.out` file using the `docker logs {container ID}` command.

## 2 *trioPhaser* results on Ashkenazim and Han Chinese trios

In order to test *trioPhaser* on real trios, we used the Ashkenazim trio and Han Chinese trio available through The Genome in a Bottle (GIAB) Consortium (Zook et al. 2019).

### 2.0.1 Download GRCh38 BAM files from the Genome in a Bottle Consortium for the Ashkenazim and Han Chinese trios

```
# Create a new directory to store the GIAB files
docker run -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest mkdir ashkenazim_trio

# Download .bam and .bai files for all members of the Ashkenazim trio
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
```

```

wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/HG002_NA24385_son/\
NA24385_GRCh38.bam -O ashkenazim_trio/son_ashkenazim_GRCh38.bam \
> ashkenazim_trio/download_son_bam.out

docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/HG002_NA24385_son/\
NA24385_GRCh38.bam.bai -O ashkenazim_trio/son_ashkenazim_GRCh38.bam.bai \
> ashkenazim_trio/download_son_bai.out

docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
HG003_NA24149_father/NA24149_GRCh38.bam \
-O ashkenazim_trio/father_ashkenazim_GRCh38.bam \
> ashkenazim_trio/download_father_bam.out

docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
HG003_NA24149_father/NA24149_GRCh38.bam.bai \
-O ashkenazim_trio/father_ashkenazim_GRCh38.bam.bai \
> ashkenazim_trio/download_father_bai.out

docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
HG004_NA24143_mother/NA24143_GRCh38.bam \
-O ashkenazim_trio/mother_ashkenazim_GRCh38.bam \
> ashkenazim_trio/download_mother_bam.out

docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
HG004_NA24143_mother/NA24143_GRCh38.bam.bai \
-O ashkenazim_trio/mother_ashkenazim_GRCh38.bam.bai \
> ashkenazim_trio/download_mother_bai.out

```

*# Create a new directory to store the GIAB files*

```

docker run -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest mkdir chinese_trio

```

*# Download .bam and .bai files for all members of the Ashkenazim trio*

```
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest wget --no-check-certificate \
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/\
ChineseTrio/analysis/10XGenomics_ChromiumGenome_LongRanger\
2.1_06202017/GM24631_ChineseTrioSon_GRCh38/GM24631_\
ChineseTrioSon_GRCh38_phased_possorted_bam.bam \
-O chinese_trio/son_chinese_GRCh38.bam \
> download_chinese_son_bam.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest wget --no-check-certificate \
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/\
ChineseTrio/analysis/10XGenomics_ChromiumGenome_LongRanger\
2.1_06202017/GM24631_ChineseTrioSon_GRCh38/GM24631_\
ChineseTrioSon_GRCh38_phased_possorted_bam.bam.bai \
-O chinese_trio/son_chinese_GRCh38.bam.bai \
> download_chinese_son_bam_bai.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest wget --no-check-certificate \
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/\
ChineseTrio/analysis/10XGenomics_ChromiumGenome_LongRanger\
2.1_06202017/GM24631_ChineseTrioFather_GRCh38/GM24631_\
ChineseTrioFather_GRCh38_phased_possorted_bam.bam \
-O chinese_trio/father_chinese_GRCh38.bam \
> download_chinese_father_bam.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest wget --no-check-certificate \
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/\
data/ChineseTrio/analysis/10XGenomics_ChromiumGenome_LongRanger\
2.1_06202017/GM24631_ChineseTrioFather_GRCh38/GM24631_\
ChineseTrioFather_GRCh38_phased_possorted_bam.bam.bai \
-O chinese_trio/father_chinese_GRCh38.bam.bai \
> download_chinese_father_bam_bai.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest wget --no-check-certificate \
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/\
data/ChineseTrio/analysis/10XGenomics_ChromiumGenome_LongRanger\
2.1_06202017/GM24631_ChineseTrioMother_GRCh38/GM24631_\
ChineseTrioMother_GRCh38_phased_possorted_bam.bam \
-O chinese_trio/mother_chinese_GRCh38.bam \
> download_chinese_mother_bam.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest wget --no-check-certificate \
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/\
data/ChineseTrio/analysis/10XGenomics_ChromiumGenome_LongRanger\
2.1_06202017/GM24631_ChineseTrioMother_GRCh38/GM24631_\
ChineseTrioMother_GRCh38_phased_possorted_bam.bam.bai \
-O chinese_trio/mother_chinese_GRCh38.bam.bai \
```

```
> download_chinese_mother_bam_bai.out
```

## 2.0.2 Generate gVCF files for the Ashkenazim and Han Chinese trios

*GATK*'s (Data Sciences Platform @ Broad Institute, n.d.) HaplotypeCaller tool was used to generate gVCF files for each member of the trios using BAM and BAI files as input. Prior to executing *GATK*, the reference files (contained within the container) were decompressed so *GATK* could use them.

```
docker run -d -v /Data:/proj -w /proj -t dm11903/triophaser:latest /bin/bash -c \  
"unzip /fasta_references.zip -d /fasta_references \  
&& gzip -d /fasta_references/*.gz \  
&& gatk --java-options -Xmx8g HaplotypeCaller \  
-R /fasta_references/Homo_sapiens_assembly38.fasta \  
-I ashkenazim_trio/son_ashkenazim_GRCh38.bam \  
-O ashkenazim_trio/son_ashkenazim_GRCh38.g.vcf.gz -ERC GVCF" \  
> ashkenazim_trio/genotype_son.out
```

```
docker run -d -v /Data:/proj -w /proj -t dm11903/triophaser:latest /bin/bash -c \  
"unzip /fasta_references.zip -d /fasta_references \  
&& gzip -d /fasta_references/*.gz \  
&& gatk --java-options -Xmx8g HaplotypeCaller \  
-R /fasta_references/Homo_sapiens_assembly38.fasta \  
-I ashkenazim_trio/father_ashkenazim_GRCh38.bam \  
-O ashkenazim_trio/father_ashkenazim_GRCh38.g.vcf.gz -ERC GVCF" \  
> ashkenazim_trio/genotype_father.out
```

```
docker run -d -v /Data:/proj -w /proj -t dm11903/triophaser:latest /bin/bash -c \  
"unzip /fasta_references.zip -d /fasta_references \  
&& gzip -d /fasta_references/*.gz \  
&& gatk --java-options -Xmx8g HaplotypeCaller \  
-R /fasta_references/Homo_sapiens_assembly38.fasta \  
-I ashkenazim_trio/mother_ashkenazim_GRCh38.bam \  
-O ashkenazim_trio/mother_ashkenazim_GRCh38.g.vcf.gz -ERC GVCF" \  
> ashkenazim_trio/genotype_mother.out
```

```
docker run -d -v /Data:/proj -w /proj -t dm11903/triophaser:latest /bin/bash -c \  
"unzip /fasta_references.zip -d /fasta_references \  
&& gzip -d /fasta_references/*.gz \  
&& gatk --java-options -Xmx8g HaplotypeCaller \  
-R /fasta_references/Homo_sapiens_assembly38.fasta \  
-I chinese_trio/son_chinese_GRCh38.bam \  
-O chinese_trio/son_chinese_GRCh38.g.vcf.gz -ERC GVCF" \  
> chinese_trio/genotype_chinese_son.out
```

```
docker run -d -v /Data:/proj -w /proj -t dm11903/triophaser:latest /bin/bash -c \  
"unzip /fasta_references.zip -d /fasta_references \  
&& gzip -d /fasta_references/*.gz \  
&& gatk --java-options -Xmx8g HaplotypeCaller \  
-R /fasta_references/Homo_sapiens_assembly38.fasta \  
-I chinese_trio/father_chinese_GRCh38.bam \  
-O chinese_trio/father_chinese_GRCh38.g.vcf.gz -ERC GVCF" \  
> chinese_trio/genotype_chinese_father.out
```

```
docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest /bin/bash -c \
"unzip /fasta_references.zip -d /fasta_references \
&& gzip -d /fasta_references/*.gz \
&& gatk --java-options -Xmx8g HaplotypeCaller \
-R /fasta_references/Homo_sapiens_assembly38.fasta \
-I chinese_trio/mother_chinese_GRCh38.bam \
-O chinese_trio/mother_chinese_GRCh38.g.vcf.gz -ERC GVCF" \
> chinese_trio/genotype_chinese_mother.out
```

### 2.0.3 Run gVCF files through *trioPhaser*

Once the gVCF files were created using *GATK*, the Ashkenazim and Han Chinese trios were phased using *trioPhaser*. The `trio_phaser.py` script is located at the root directory within the container. This script has 5 required arguments: 1) (-c) The gVCF of the child, 2) (-p) the gVCF of the father, 3) (-m) the gVCF of the mother, 4) (-o) the name of the output file, and 5) (-r) where the reference files are to be stored. When `trio_phaser.py` is first executed, all necessary reference files are stored to the directory you specify with argument #5. Therefore, the first time `trio_phaser.py` is executed, the run-time will take longer as reference files are downloaded. However, as long as you point future *trioPhaser* runs to this reference directory, the previously downloaded files will be used. `trio_phaser.py` also has 3 optional arguments: 1) the number of cores available for use (default = 2, if 22 cores are specified, *SHAPEIT4* will phase all 22 autosomal chromosomes at once which speeds up the process), 2) the genome build version (default = GRCh38, GRCh37 is also supported), and 3) the minimum Phred-scaled quality score a nucleotide position is able to have (default = 30). Because *trioPhaser* is executed in the background, the log information that `trio_phaser.py` outputs will not be directly output to the terminal on your machine. Instead, all the log information is output within the container. To view the log information output by `trio_phaser.py`, the container ID can be used. In the code below, we save the container ID as a variable, then use the docker command `docker logs {container ID}` to view the log information output by `trio_phaser.py`. `trio_phaser.py` outputs helpful information about how many initial variants there were, how many of the initial variants were phased, how many variants were phased by *SHAPEIT4* (Delaneau et al. 2019) that weren't phased using Mendelian inheritance, how many variants were phased using Mendelian inheritance that *SHAPEIT4* was unable to phase, and how long *trioPhaser* took to execute. In addition, a separate VCF file (ends with "Mendel\_errors.vcf" and will be at the same path as the phased VCF file) is automatically output that only includes phased positions of the child where one of the nucleotides was not identified in either parent. These phased positions may be the result of genotype errors or de novo variants. This file is meant to be informative and allow the user to determine how these positions should be used in downstream analyses.

```
docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest \
python3 /trio_phaser.py \
-c ashkenazim_trio/son_ashkenazim_GRCh38.g.vcf.gz \
-p ashkenazim_trio/father_ashkenazim_GRCh38.g.vcf.gz \
-m ashkenazim_trio/mother_ashkenazim_GRCh38.g.vcf.gz \
-o ashkenazim_trio/giab_ashkenazim_phased.vcf.gz \
-r haplotype_references \
--number_of_tasks 22 > ashkenazim_trio/trio_phaser_giab.out

# Use the container ID to get the log of the "trio_phaser.py" output and store
# the log information to the same file where the container ID was stored.
CONTAINERID="cat ashkenazim_trio/trio_phaser_giab.out"
docker logs '$CONTAINERID' >> ashkenazim_trio/trio_phaser_giab.out
```

```

docker run -d -v /Data:/proj -w /proj -t dmiller903/triophaser:latest \
python3 /trio_phaser.py \
-c chinese_trio/son_chinese_GRCh38.g.vcf.gz \
-p chinese_trio/father_chinese_GRCh38.g.vcf.gz \
-m chinese_trio/mother_chinese_GRCh38.g.vcf.gz \
-o chinese_trio/giab_chinese_phased.vcf.gz \
-r haplotype_references \
--number_of_tasks 22 > chinese_trio/trio_phaser_chinese_trio.out

# Use the container ID to get the log of the "trio_phaser.py" output and store
# the log information to the same file where the container ID was stored.
CONTAINERID="cat chinese_trio/trio_phaser_chinese_trio.out"
docker logs '$CONTAINERID' >> chinese_trio/trio_phaser_chinese_trio.out

```

Alternatively, *trioPhaser* can be executed using more than one trio at once. When *trioPhaser* is used this way, there are 2 required arguments: 1) (-t) a TSV file that contains one trio per line where the first, second, and third columns are paths to the child, father, and mother gVCF file, respectively, and the fourth column is the path to the output file, and 2) (-r) path where haplotype reference files will be saved. All the same optional arguments are available as described above.

```

docker run -d -v /Data:/proj -w /proj -t dmiller903/triophaser:latest \
python3 /trio_phaser.py \
-t phase_giab_trios.tsv \
-r haplotype_references \
--number_of_tasks 22 > phase_giab_trios.out

# Use the container ID to get the log of the "trio_phaser.py" output and store
# the log information to the same file where the container ID was stored.
CONTAINERID="cat phase_giab_trios.out"
docker logs '$CONTAINERID' >> phase_giab_trios.out

```

### 3 Ashkenazim trio gVCF test data

Since the gVCF files for the each trio are too large to upload to GitHub (>19 GB/per file), we have provided gVCF files for chromosome 22 of the Ashkenazim trio that can be used to test *trioPhaser*. These files can be found at <https://github.com/dmiller903/trioPhaser/tree/main/validate> and within the Docker container at `/trioPhaser/validate/`. These files do not include all chromosome 22 positions. Positions were removed so each file was less than 100 MB in size (max file size supported by GitHub). These example files can be executed using the commands below. Change `/Data` to the path you want the container to be able to access on your local machine (i.e. the path where the `haplotype_references` directory is located (or whatever you named this directory)). In this example, we set `/proj` as the working directory within the container. This directory is equivalent to `/Data` on our local machine. We use `/Data` as the working directory on our local machine. Within the `/Data` directory, the `haplotype_references` directory exists. The output file `giab_phased_chr22.vcf.gz` will be found at `/Data/giab_ashkenazim_phased_chr22.vcf.gz` on the local machine when *trioPhaser* is done executing. The log file, `trio_phaser_giab_ashkenazim_chr22.out` will be at `/Data/trio_phaser_chr22.out`.

```

docker run -d -v /Data:/proj -w /proj -t dmiller903/triophaser:latest \
python3 /trio_phaser.py \
-c /trioPhaser/validate/son_ashkenazim_GRCh38_chr22.g.vcf.gz \
-p /trioPhaser/validate/father_ashkenazim_GRCh38_chr22.g.vcf.gz \

```



```
-m /trioPhaser/validate/mother_ashkenazim_GRCh38_chr22.g.vcf.gz \
-o /proj/giab_ashkenazim_phased_chr22.vcf.gz \
-r haplotype_references > trio_phaser_giab_ashkenazim_chr22.out
```

## 4 Comparison of *trioPhaser* to linked-read phasing technology (10X)

We wanted to compare how well *trioPhaser* phased as compared to the linked-read phasing (10X) (Zheng et al. 2016) method which has been shown to be one of the most accurate phasing methods (Choi et al. 2018). The 10X-phased VCF file of the Ashkenazim and Han Chinese children are available through The Genome in a Bottle (GIAB) Consortium.

### 4.0.1 Download the Ashkenazim and Han Chinese child's 10X phased VCF files

```
docker run -v /Data:/proj -w /proj -t dm11903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/\
  giab/data/AshkenazimTrio/analysis/\
  10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
  HG002_NA24385_son/NA24385_GRCh38.vcf.gz \
  -O ashkenazim_trio/son_ashkenazim_GRCh38_longRanger.vcf.gz
```

```
docker run -v /Data:/proj -w /proj -t dm11903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
  ReferenceSamples/giab/data/ChineseTrio/analysis/\
  10XGenomics_ChromiumGenome_LongRanger2.1_06202017/\
  GM24631_ChineseTrioSon_GRCh38/\
  GM24631_ChineseTrioSon_GRCh38_phased_variants.vcf.gz \
  -O chinese_trio/son_chinese_GRCh38_longRanger.vcf.gz
```

### 4.0.2 Compare *trioPhaser* phase results to 10X

The `giab_ashkenazim_phased.vcf.gz` and `giab_chinese_phased.vcf.gz` (output by *trioPhaser*) files were compared to the `son_ashkenazim_GRCh38_longRanger.vcf.gz` and `son_chinese_GRCh38_longRanger.vcf.gz` (10X phased files) files, respectively, using the `compare_trioPhaser_to_10X.py` script.

```
docker run -v /Data:/proj -w /proj -t \
  dm11903/triophaser:latest \
  python3 trioPhaser/validate/compare_trioPhaser_to_10x.py \
  ashkenazim_trio/giab_ashkenazim_phased.vcf.gz \
  ashkenazim_trio/son_GRCh38_longRanger.vcf.gz \
  ashkenazim_trio/ > ashkenazim_trio/compare_trioPhaser_to_10x.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
  dm11903/triophaser:latest \
  python3 trioPhaser/validate/compare_trioPhaser_to_10x.py \
  chinese_trio/giab_chinese_phased.vcf.gz \
  chinese_trio/son_chinese_GRCh38_longRanger.vcf.gz \
  chinese_trio/ > chinese_trio/compare_trioPhaser_to_10x.out
```



## 5 Comparison of *trioPhaser* to *whatshap*

We wanted to compare how well *trioPhaser* phased the Ashkenazim and Han Chinese trios as compared to *whatshap* (Martin et al. 2016) which uses read-backed phasing.

### 5.0.1 Generate a combined VCF file that can be used as input into *whatshap*

First, the gVCF files that were created using HaplotypeCaller (see sub-section, “Generate gVCF files for the Ashkenazim and Han Chinese trios”) were combined using *GATK*’s CombineGVCFs tool, and then joint-genotyped using *GATK*’s GenotypeGVCFs tool. The commands to perform this process are contained within the `combine_trio.py` script. This script is found within the container at `/trioPhaser/validate/combine_trio.py`.

```
docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest \
python3 /trioPhaser/validate/combine_trio.py \
ashkenazim_trio/son_ashkenazim_GRCh38.g.vcf.gz \
ashkenazim_trio/father_ashkenazim_GRCh38.g.vcf.gz \
ashkenazim_trio/mother_ashkenazim_GRCh38.g.vcf.gz \
ashkenazim_trio/giab_ashkenazim_combined.vcf.gz > ashkenazim_trio/combined.out
```

```
docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest \
python3 /trioPhaser/validate/combine_trio.py \
chinese_trio/son_chinese_GRCh38.g.vcf.gz \
chinese_trio/father_chinese_GRCh38.g.vcf.gz \
chinese_trio/mother_chinese_GRCh38.g.vcf.gz \
chinese_trio/giab_chinese_combined.vcf.gz > chinese_trio/combined.out
```

### 5.0.2 Run Ashkenazim and Han Chinese trios through *whatshap*

The combined VCF and previously downloaded BAM files were used as input into *whatshap*.

```
docker run -d -v /Data:/proj -w /proj -t \
quay.io/biocontainers/whatshap:1.1--py36hae55d0a_0 \
whatshap phase --ped ashkenazim_trio/ashkenazim_trio.ped \
--reference=references/Homo_sapiens_assembly38.fasta \
-o ashkenazim_trio/giab_ashkenazim_phased_whatshap.vcf \
ashkenazim_trio/giab_ashkenazim_combined.vcf.gz \
ashkenazim_trio/son_ashkenazim_GRCh38.bam \
ashkenazim_trio/father_ashkenazim_GRCh38.bam \
ashkenazim_trio/mother_ashkenazim_GRCh38.bam \
> ashkenazim_trio/giab_ashkenazim_whatshap.out
```

```
docker run -d -v /Data:/proj -w /proj -t \
quay.io/biocontainers/whatshap:1.1--py36hae55d0a_0 \
whatshap phase --ped chinese_trio/chinese_trio.ped \
--reference=references/Homo_sapiens_assembly38.fasta \
-o chinese_trio/giab_chinese_phased_whatshap.vcf \
chinese_trio/giab_chinese_combined.vcf.gz \
chinese_trio/son_chinese_GRCh38.bam \
chinese_trio/father_chinese_GRCh38.bam \
chinese_trio/mother_chinese_GRCh38.bam \
> chinese_trio/giab_chinese_whatshap.out
```

### 5.0.3 Compare *trioPhaser* results to *whatshap*

The *whatshap* phased output was compared to the *trioPhaser* phased output using the `compare_trioPhaser_to_whatshap.py` script for the Ashkenazim and Han Chinese trios.

```
docker run -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  python3 /trioPhaser/validate/compare_trioPhaser_to_whatshap.py \
  ashkenazim_trio/giab_ashkenazim_phased.vcf.gz \
  ashkenazim_trio/giab_ashkenazim_phased_whatshap.vcf \
  ashkenazim_trio/
```

```
docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  python3 trioPhaser/validate/compare_trioPhaser_to_whatshap.py \
  chinese_trio/giab_chinese_phased.vcf.gz \
  chinese_trio/giab_chinese_phased_whatshap.vcf \
  chinese_trio/ > chinese_trio/compare_trioPhaser_to_whatshap.out
```

## 6 *trioPhaser* results across 50 trios

### 6.0.1 Download gVCF files from the Gabriella Miller Kids First Data Resource Center

Controlled-access gVCF files of 50 trios were downloaded from the Gabriella Miller Kids Frist (GMKF) Data Resource Center(Heath et al. 2019). These files were generated from WGS data of normal tissue. The child in each trio had neuroblastoma. Trios were stored in a directory named with the family ID (e.g. FM\_AC8MB9RH), and gVCF file names were changed to indicate which family the individual belonged to (e.g. FM\_AC8MB9RH) and family relationship (e.g. father). Original file names and updated file names are seen in the table below:

original_file_name	download_name
057a1584-5a10-4f8a-af6c-a2e60872adcb.g.vcf.gz	FM_AC8MB9RH/FM_AC8MB9RH_father.g.vcf.gz
0781ea81-95b6-4ed5-a2f9-3466b81c8784.g.vcf.gz	FM_2JR5YVMZ/FM_2JR5YVMZ_father.g.vcf.gz
0a8ccf6b-a510-4afb-ae64-a6fe101971bb.g.vcf.gz	FM_XBG76ESE/FM_XBG76ESE_mother.g.vcf.gz
0c465840-8133-497d-b7ae-9a49510a86a5.g.vcf.gz	FM_RF8RS3YT/FM_RF8RS3YT_child.g.vcf.gz
15312212-ec1c-4c53-b0df-c66177d0da78.g.vcf.gz	FM_0FKGXBZB/FM_0FKGXBZB_mother.g.vcf.gz
16e7875f-6c31-45b1-a6f5-2c316b894987.g.vcf.gz	FM_1DG7K3TV/FM_1DG7K3TV_father.g.vcf.gz
17e821f2-8b3e-4e0c-b654-e97a116d3133.g.vcf.gz	FM_EWX4PAC4/FM_EWX4PAC4_father.g.vcf.gz
18cc83be-7ba2-475a-a695-b5fe971ac647.g.vcf.gz	FM_XBG76ESE/FM_XBG76ESE_child.g.vcf.gz
190f149e-82d9-4291-9af6-f7bbd6d6bf72.g.vcf.gz	FM_068BMXVN/FM_068BMXVN_child.g.vcf.gz
19f89a26-adb3-4214-b17a-e720c39f6783.g.vcf.gz	FM_VM04HHXA/FM_VM04HHXA_child.g.vcf.gz
1d3d816c-b8aa-46b7-884c-c996b7fc5a07.g.vcf.gz	FM_6G8GWTMG/FM_6G8GWTMG_father.g.vcf.gz
1e2e05ad-1fb0-4538-85d8-7f39bdf8467f.g.vcf.gz	FM_WR20QYM1/FM_WR20QYM1_child.g.vcf.gz
1fc33481-ebd3-419a-9ed8-640a37455e94.g.vcf.gz	FM_QNFTR3R4/FM_QNFTR3R4_mother.g.vcf.gz
21406e96-2f13-459d-b6d0-4118ee015767.g.vcf.gz	FM_9H9QAMW8/FM_9H9QAMW8_child.g.vcf.gz
2152ed58-d971-47d1-812f-071ddf43c89f.g.vcf.gz	FM_2VKNQ3DF/FM_2VKNQ3DF_father.g.vcf.gz
26c127a0-d576-48d1-98dd-af0b362f9619.g.vcf.gz	FM_T36W3RK9/FM_T36W3RK9_father.g.vcf.gz
2ba32ce1-9a7f-489e-ac41-53efac14721f.g.vcf.gz	FM_BW30548P/FM_BW30548P_child.g.vcf.gz
2c9650b7-eee0-4fc8-bf24-096592560307.g.vcf.gz	FM_8NEWFAZ7/FM_8NEWFAZ7_child.g.vcf.gz
2c9910fe-897d-430a-a00f-96e6f40a2f92.g.vcf.gz	FM_MA1QF65E/FM_MA1QF65E_mother.g.vcf.gz
2cd5c46b-bac6-4b40-baac-e59cb74e6665.g.vcf.gz	FM_JS4PRGFS/FM_JS4PRGFS_mother.g.vcf.gz

original_file_name	download_name
2d1a8e6d-7615-49a8-97b1-6f145957bf91.g.vcf.gz	FM_JHCE7KM4/FM_JHCE7KM4_mother.g.vcf.gz
2eec8c1b-7eef-4bf0-9c3a-289be86f03ba.g.vcf.gz	FM_F20MF8G6/FM_F20MF8G6_child.g.vcf.gz
2fe52015-4ab9-4281-87da-f09cb52a7afc.g.vcf.gz	FM_9XR6YJ29/FM_9XR6YJ29_mother.g.vcf.gz
2ffef012-86e3-4155-9327-738d432b88e3.g.vcf.gz	FM_9XR6YJ29/FM_9XR6YJ29_child.g.vcf.gz
303b961c-9585-4547-b8cb-a8f98519561f.g.vcf.gz	FM_8NEWFAZ7/FM_8NEWFAZ7_father.g.vcf.gz
33134715-aea3-439a-973a-2067056b3166.g.vcf.gz	FM_1TQ4YE5Q/FM_1TQ4YE5Q_child.g.vcf.gz
34419302-d8c1-44e3-97d2-c9b93c64b386.g.vcf.gz	FM_2VKNQ3DF/FM_2VKNQ3DF_mother.g.vcf.gz
34e4407e-bfd9-4418-ae36-57a3f0eb0692.g.vcf.gz	FM_BHTM6B9K/FM_BHTM6B9K_child.g.vcf.gz
363e424c-266c-43a9-ac3b-bd3959648dbb.g.vcf.gz	FM_H67TXYYJ/FM_H67TXYYJ_child.g.vcf.gz
37d2510f-faa1-4003-8357-6e6fb96e0b14.g.vcf.gz	FM_02AXWKRM/FM_02AXWKRM_child.g.vcf.gz
394421cb-58d2-47fe-a311-8a674b52dd0b.g.vcf.gz	FM_FR0MMM3E/FM_FR0MMM3E_child.g.vcf.gz
39bdf5ab-bc45-4e88-afab-82c4a84825a2.g.vcf.gz	FM_MA1QF65E/FM_MA1QF65E_child.g.vcf.gz
3e5ec4f2-527b-4a21-8667-dfb5ebf93c8b.g.vcf.gz	FM_NZABV4W7/FM_NZABV4W7_father.g.vcf.gz
3f5a0fa7-180c-4c21-b1d9-e0faef7fdf84.g.vcf.gz	FM_5A7HFR0N/FM_5A7HFR0N_father.g.vcf.gz
40be2fd3-6b20-43ff-9e34-320f04ccc594.g.vcf.gz	FM_9XR6YJ29/FM_9XR6YJ29_father.g.vcf.gz
433b9cec-fe89-468f-b6e8-2207abc79651.g.vcf.gz	FM_H67TXYYJ/FM_H67TXYYJ_father.g.vcf.gz
445a06dc-5460-43dd-a763-101e9d45b6ca.g.vcf.gz	FM_068BMXVN/FM_068BMXVN_mother.g.vcf.gz
4517c80d-63bb-43d6-b404-358e7bf99250.g.vcf.gz	FM_RHPD85WC/FM_RHPD85WC_child.g.vcf.gz
45a3d2da-4dc9-4521-b8f5-235c562d5936.g.vcf.gz	FM_TEGCS8FR/FM_TEGCS8FR_mother.g.vcf.gz
46d4751c-39c8-4701-bc5e-6ea9575f1eb3.g.vcf.gz	FM_XBG76ESE/FM_XBG76ESE_father.g.vcf.gz
482a1d4d-fcb1-4621-bd36-b77be700ba5a.g.vcf.gz	FM_RF8RS3YT/FM_RF8RS3YT_father.g.vcf.gz
4927c134-114a-46d9-8f53-a0ae70eb14a8.g.vcf.gz	FM_BHTM6B9K/FM_BHTM6B9K_mother.g.vcf.gz
4b0c0f8a-861c-4149-a575-f1e896af9f69.g.vcf.gz	FM_C0QC9C8B/FM_C0QC9C8B_father.g.vcf.gz
4c0e37e1-56f5-43a3-bae6-86b8f434ac6a.g.vcf.gz	FM_02AXWKRM/FM_02AXWKRM_mother.g.vcf.gz
4e7f01f9-28f4-4051-b5c5-1dd83a8a9a67.g.vcf.gz	FM_1DG7K3TV/FM_1DG7K3TV_child.g.vcf.gz
501f7bb5-13a0-4506-b923-e4b06201020f.g.vcf.gz	FM_8F2EF55Z/FM_8F2EF55Z_father.g.vcf.gz
5177fff7-2d6d-4c88-9167-71a677666571.g.vcf.gz	FM_WR20QYM1/FM_WR20QYM1_mother.g.vcf.gz
53a7c633-07aa-4880-81b0-b914f5a15021.g.vcf.gz	FM_TEGCS8FR/FM_TEGCS8FR_child.g.vcf.gz
54f42659-9e05-4a3a-9b7e-24368fe6598c.g.vcf.gz	FM_0FKGXBZB/FM_0FKGXBZB_child.g.vcf.gz
56f33db9-2a29-4b4a-9eec-e6632c84acd0.g.vcf.gz	FM_6VKD1S0D/FM_6VKD1S0D_mother.g.vcf.gz
57eb3cb3-65c0-4c71-80a8-37a9d76fed0a.g.vcf.gz	FM_GWXRTT53/FM_GWXRTT53_father.g.vcf.gz
5874586d-c23f-419b-badc-8653c1462343.g.vcf.gz	FM_QNFTR3R4/FM_QNFTR3R4_child.g.vcf.gz
5894f13f-faaa-4ea0-a0c0-1aff9ffd866d.g.vcf.gz	FM_P5JX9P4J/FM_P5JX9P4J_mother.g.vcf.gz
59c22c44-59b0-4781-ae12-8cf01b913431.g.vcf.gz	FM_6VKD1S0D/FM_6VKD1S0D_child.g.vcf.gz
5b9337d8-3a98-495e-951a-c75d55b34507.g.vcf.gz	FM_T36W3RK9/FM_T36W3RK9_mother.g.vcf.gz
5cacc6e6-cf59-475b-b0df-278e75019180.g.vcf.gz	FM_P5JX9P4J/FM_P5JX9P4J_father.g.vcf.gz
5e1646ca-9007-4f5d-bb91-a70656df457e.g.vcf.gz	FM_6VKD1S0D/FM_6VKD1S0D_father.g.vcf.gz
5e82d3d0-c1f4-43a8-bae3-f18e6cb21246.g.vcf.gz	FM_1TQ4YE5Q/FM_1TQ4YE5Q_father.g.vcf.gz
600706ea-e048-4ebc-b028-1e7dbf739230.g.vcf.gz	FM_6G8GWTMG/FM_6G8GWTMG_child.g.vcf.gz
611d8b78-cf31-4e28-9f47-12b5b1ad46c4.g.vcf.gz	FM_VM04HHXA/FM_VM04HHXA_father.g.vcf.gz
6469fec8-43f7-4a05-a753-124f13ef1575.g.vcf.gz	FM_69MH3P0P/FM_69MH3P0P_father.g.vcf.gz
64dcb088-bb01-498c-9b98-d708727d5e60.g.vcf.gz	FM_7MF9WPNH/FM_7MF9WPNH_mother.g.vcf.gz
662d595f-5b74-4696-8b21-471ab70c48fc.g.vcf.gz	FM_0FKGXBZB/FM_0FKGXBZB_father.g.vcf.gz
673f2920-c3fb-489f-98c1-90afbc669677.g.vcf.gz	FM_58WE88C4/FM_58WE88C4_father.g.vcf.gz
68631b06-312f-4b33-8898-062653d5fe4b.g.vcf.gz	FM_QNFTR3R4/FM_QNFTR3R4_father.g.vcf.gz
6c19a5e7-b53b-47d9-93b0-86e4beddafef.g.vcf.gz	FM_RNMD1436/FM_RNMD1436_mother.g.vcf.gz
6c82ce2e-ff31-41b3-920b-3c01f5c9ac1c.g.vcf.gz	FM_C0QC9C8B/FM_C0QC9C8B_child.g.vcf.gz
6eb36ab5-60ac-4200-9bfl-35320285da66.g.vcf.gz	FM_Z53CBVAP/FM_Z53CBVAP_mother.g.vcf.gz
70ca874b-1e51-4c66-88a3-93fec1211e0b.g.vcf.gz	FM_H4GA6GWQ/FM_H4GA6GWQ_mother.g.vcf.gz
71737c84-fd85-459d-a4bf-0bec216643ee.g.vcf.gz	FM_RHPD85WC/FM_RHPD85WC_father.g.vcf.gz
721c423b-d95c-4d30-9a4d-30f82730fc1f.g.vcf.gz	FM_41RNEZ6B/FM_41RNEZ6B_mother.g.vcf.gz
73aeeb93-ffb5-4234-977d-666b2dc99731.g.vcf.gz	FM_AC8MB9RH/FM_AC8MB9RH_child.g.vcf.gz

original_file_name	download_name
7822f4f7-cb5d-4e9a-b5ca-8dc2422e0bd0.g.vcf.gz	FM_PTEDE0E5/FM_PTEDE0E5_father.g.vcf.gz
7db1dea5-6392-4b72-bd6c-a098c50c366a.g.vcf.gz	FM_69MH3P0P/FM_69MH3P0P_mother.g.vcf.gz
8057ee03-5199-46c8-a511-83e2f935755f.g.vcf.gz	FM_T36W3RK9/FM_T36W3RK9_child.g.vcf.gz
848f0b88-6f7f-437e-baef-3744ba915d74.g.vcf.gz	FM_W6CDBYXE/FM_W6CDBYXE_father.g.vcf.gz
84cbdcc-2c76-4aab-82c8-e19fda5b867f.g.vcf.gz	FM_58WE88C4/FM_58WE88C4_mother.g.vcf.gz
8826ca40-da75-4381-87ba-a30fd6de5c07.g.vcf.gz	FM_Z53CBVAP/FM_Z53CBVAP_child.g.vcf.gz
8968ffb4-6ac0-4c9c-b621-013e51e5d248.g.vcf.gz	FM_2VKNQ3DF/FM_2VKNQ3DF_child.g.vcf.gz
89828ca3-6074-4476-bbbb-b6a72e82ae73.g.vcf.gz	FM_2N0XG4Z1/FM_2N0XG4Z1_father.g.vcf.gz
918b3c92-a1f4-4b8c-8398-e32d45959c5c.g.vcf.gz	FM_BW30548P/FM_BW30548P_father.g.vcf.gz
91cf6103-2824-4b97-b642-e0af19dfa45f.g.vcf.gz	FM_9H9QAMW8/FM_9H9QAMW8_father.g.vcf.gz
9261fd28-3970-4403-baa8-79c913732987.g.vcf.gz	FM_RHPD85WC/FM_RHPD85WC_mother.g.vcf.gz
928d8fd4-050c-4e18-b117-0c90aade71cd.g.vcf.gz	FM_2N0XG4Z1/FM_2N0XG4Z1_mother.g.vcf.gz
92ca1a5d-e954-44de-9574-ae4405c76c5b.g.vcf.gz	FM_9H9QAMW8/FM_9H9QAMW8_mother.g.vcf.gz
9a80cac2-3416-4e93-ba88-b60bdd50a678.g.vcf.gz	FM_41RNEZ6B/FM_41RNEZ6B_father.g.vcf.gz
9cb5e7a5-716a-472f-8870-61ca7722da96.g.vcf.gz	FM_7MF9WPNH/FM_7MF9WPNH_child.g.vcf.gz
9e4f2391-c675-48fd-b4b6-dccfc9c5f38c.g.vcf.gz	FM_3055H3PW/FM_3055H3PW_child.g.vcf.gz
a518dadb-6fed-4fc7-8f11-7d65e11752b7.g.vcf.gz	FM_Y77GDZKH/FM_Y77GDZKH_mother.g.vcf.gz
a57ca7b4-4065-41c2-94b4-ae8a60f438c4.g.vcf.gz	FM_W6CDBYXE/FM_W6CDBYXE_mother.g.vcf.gz
a9d22b4c-55c3-4c11-b7ec-161c617e57a4.g.vcf.gz	FM_5A7HFR0N/FM_5A7HFR0N_mother.g.vcf.gz
aabdbcdc-4e07-4b18-ab5f-8adec6fcfd28.g.vcf.gz	FM_8F2EF55Z/FM_8F2EF55Z_child.g.vcf.gz
ab7167ff-b371-4274-ba3d-2f41be68c90e.g.vcf.gz	FM_5XT4MYNJ/FM_5XT4MYNJ_father.g.vcf.gz
ad4e29a2-9878-4aae-88ce-cc7ae4b946da.g.vcf.gz	FM_2JR5YVMZ/FM_2JR5YVMZ_child.g.vcf.gz
b04ed792-733e-472b-a2ea-d1d9af79e8f0.g.vcf.gz	FM_EWX4PAC4/FM_EWX4PAC4_mother.g.vcf.gz
b3ca6aa6-93c0-4827-87b6-38f3a622bfc2.g.vcf.gz	FM_P5JX9P4J/FM_P5JX9P4J_child.g.vcf.gz
b5404849-380b-4c84-bf9c-4bb997ee9e8d.g.vcf.gz	FM_2N0XG4Z1/FM_2N0XG4Z1_child.g.vcf.gz
b5ca8d81-e2ce-4aaf-be29-dcfd7d53a232.g.vcf.gz	FM_6G8GWTMG/FM_6G8GWTMG_mother.g.vcf.gz
b6bee03a-ca98-486d-8b50-a805951cc9aa.g.vcf.gz	FM_41RNEZ6B/FM_41RNEZ6B_child.g.vcf.gz
b7110470-3598-41a2-b513-fd8f599a71a6.g.vcf.gz	FM_C0QC9C8B/FM_C0QC9C8B_mother.g.vcf.gz
b7229611-9031-474b-a16a-13f8cbb87882.g.vcf.gz	FM_HPZPPFHN/FM_HPZPPFHN_mother.g.vcf.gz
b8469b77-59bf-46cf-b1cf-1ebb2c136ba6.g.vcf.gz	FM_JS4PRGFS/FM_JS4PRGFS_child.g.vcf.gz
b880e5ec-bafd-46c2-9f5e-da2a810269ad.g.vcf.gz	FM_Y77GDZKH/FM_Y77GDZKH_father.g.vcf.gz
ba7829ae-1545-4972-a4f0-388bcf1f53d7.g.vcf.gz	FM_VM04HHXA/FM_VM04HHXA_mother.g.vcf.gz
baef16f-2d1e-4fb1-bd57-91d92ba4a93c.g.vcf.gz	FM_3055H3PW/FM_3055H3PW_mother.g.vcf.gz
bfb7f0f-d1ad-47b1-be29-c4f3ccb6f87a.g.vcf.gz	FM_02AXWKRM/FM_02AXWKRM_father.g.vcf.gz
bfd86a7e-380d-4ef2-98f3-3a6d321bbe5f.g.vcf.gz	FM_RF8RS3YT/FM_RF8RS3YT_mother.g.vcf.gz
c29c76de-a22f-4631-91b6-2d48ca262876.g.vcf.gz	FM_BW30548P/FM_BW30548P_mother.g.vcf.gz
c523eef5-b72e-4ae5-8993-ea8956d92835.g.vcf.gz	FM_8F2EF55Z/FM_8F2EF55Z_mother.g.vcf.gz
c6a77235-a9ce-400f-91e1-90d9e797cf47.g.vcf.gz	FM_Z53CBVAP/FM_Z53CBVAP_father.g.vcf.gz
c70e5a82-41fa-4ce7-82f4-c30eecd56666.g.vcf.gz	FM_GWXRTT53/FM_GWXRTT53_mother.g.vcf.gz
ca459318-143b-436f-a62f-1c4a00656cba.g.vcf.gz	FM_FR0MMM3E/FM_FR0MMM3E_mother.g.vcf.gz
cb747fdc-c54e-42da-874f-64193f88c053.g.vcf.gz	FM_F20MF8G6/FM_F20MF8G6_mother.g.vcf.gz
cba405e9-ebea-4ef8-a669-056398de308b.g.vcf.gz	FM_PTEDE0E5/FM_PTEDE0E5_mother.g.vcf.gz
cd75f800-0e75-4f18-a682-61414778a5a2.g.vcf.gz	FM_MA1QF65E/FM_MA1QF65E_father.g.vcf.gz
cd8adb95-737b-47c2-a5ea-bb2c8c389106.g.vcf.gz	FM_GWXRTT53/FM_GWXRTT53_child.g.vcf.gz
cef847d3-3f8a-421d-b05f-46c8f95b2568.g.vcf.gz	FM_1TQ4YE5Q/FM_1TQ4YE5Q_mother.g.vcf.gz
d038d7aa-c439-4b24-bde3-1c6f72e0650b.g.vcf.gz	FM_JHCE7KM4/FM_JHCE7KM4_child.g.vcf.gz
d0726f7f-42ad-47f3-b026-9c39085b4382.g.vcf.gz	FM_H4GA6GWQ/FM_H4GA6GWQ_father.g.vcf.gz
d6ce4515-9676-4367-87f6-85c2297b5598.g.vcf.gz	FM_W6CDBYXE/FM_W6CDBYXE_child.g.vcf.gz
d967df2c-8999-4db7-a7c3-b2d3bed18083.g.vcf.gz	FM_RNMD1436/FM_RNMD1436_father.g.vcf.gz
da4ad478-6d0b-42f8-a770-ac022f6c79cd.g.vcf.gz	FM_5XT4MYNJ/FM_5XT4MYNJ_mother.g.vcf.gz
db7ab99c-2bd0-44da-8d95-15dda9320f9c.g.vcf.gz	FM_JS4PRGFS/FM_JS4PRGFS_father.g.vcf.gz
dd1c4fc1-360c-46b9-93b4-b86a5a02225d.g.vcf.gz	FM_HPZPPFHN/FM_HPZPPFHN_child.g.vcf.gz

original_file_name	download_name
ddaa38f2-2243-4a73-baa7-1437dfee1381.g.vcf.gz	FM_58WE88C4/FM_58WE88C4_child.g.vcf.gz
ddc994a5-6771-4d64-99d8-2f560be7b9a7.g.vcf.gz	FM_JHCE7KM4/FM_JHCE7KM4_father.g.vcf.gz
de306be5-1cb3-4662-a18a-27273014cd04.g.vcf.gz	FM_068BMXVN/FM_068BMXVN_father.g.vcf.gz
de5711d7-c1ae-4311-bca3-ce300f9d1881.g.vcf.gz	FM_NZABV4W7/FM_NZABV4W7_mother.g.vcf.gz
deda254c-e91c-4faf-8425-491e75e7ee06.g.vcf.gz	FM_TEGCS8FR/FM_TEGCS8FR_father.g.vcf.gz
e182fe38-a33b-43bb-9e0b-aa2cc0d369ee.g.vcf.gz	FM_RNMD1436/FM_RNMD1436_child.g.vcf.gz
e18b45e4-c3b7-4b0c-aca6-a604c63c3bb1.g.vcf.gz	FM_AC8MB9RH/FM_AC8MB9RH_mother.g.vcf.gz
e2ef7450-0001-4d11-8be0-6d2a6a2a7510.g.vcf.gz	FM_BHTM6B9K/FM_BHTM6B9K_father.g.vcf.gz
e2f5ba6c-0d57-4816-ade5-e29cadd9d550.g.vcf.gz	FM_2JR5YVMZ/FM_2JR5YVMZ_mother.g.vcf.gz
e3effe1e-6ece-4d06-a4a0-681f65a2f265.g.vcf.gz	FM_EWX4PAC4/FM_EWX4PAC4_child.g.vcf.gz
e48371e4-9832-4704-b95f-e88f5e31156a.g.vcf.gz	FM_1DG7K3TV/FM_1DG7K3TV_mother.g.vcf.gz
e58ac5f7-67d9-452e-83a7-a59edd9b002d.g.vcf.gz	FM_WR20QYM1/FM_WR20QYM1_father.g.vcf.gz
ea1273ea-f02d-434b-870a-4872f5de081a.g.vcf.gz	FM_Y77GDZKH/FM_Y77GDZKH_child.g.vcf.gz
eadc3d64-3b26-4405-a68b-1f0e2602425e.g.vcf.gz	FM_FR0MMM3E/FM_FR0MMM3E_father.g.vcf.gz
ec7455b4-c632-46f6-b277-0cbd520aac7e.g.vcf.gz	FM_8NEWFAZ7/FM_8NEWFAZ7_mother.g.vcf.gz
ed0fbdc7-e442-4d08-bc1b-67a58fb9361d.g.vcf.gz	FM_7MF9WPNH/FM_7MF9WPNH_father.g.vcf.gz
ed7ecffe-83ee-4791-bbd0-49e58543593a.g.vcf.gz	FM_5XT4MYNJ/FM_5XT4MYNJ_child.g.vcf.gz
f0843574-37cc-4e42-83d0-f0287b2eb55c.g.vcf.gz	FM_3055H3PW/FM_3055H3PW_father.g.vcf.gz
f0ef2170-f382-4b86-bf03-14a84c025ffb.g.vcf.gz	FM_H4GA6GWQ/FM_H4GA6GWQ_child.g.vcf.gz
f59ee7f4-6235-4bae-b361-87fc21397a67.g.vcf.gz	FM_69MH3P0P/FM_69MH3P0P_child.g.vcf.gz
f5fc7ff8-2105-4d87-896f-2d362ebf8e80.g.vcf.gz	FM_H67TXYYJ/FM_H67TXYYJ_mother.g.vcf.gz
fbe92ff1-5788-4cd5-8f22-20d4a50d6bec.g.vcf.gz	FM_NZABV4W7/FM_NZABV4W7_child.g.vcf.gz
fc0f35af-b8db-4e8b-9d42-59a76a5799d1.g.vcf.gz	FM_5A7HFR0N/FM_5A7HFR0N_child.g.vcf.gz
fd0df59c-254f-408d-b5cd-ec9c4fcb2219.g.vcf.gz	FM_PTEDE0E5/FM_PTEDE0E5_child.g.vcf.gz
fe169f50-4710-4105-a1c2-51e610ec6911.g.vcf.gz	FM_F20MF8G6/FM_F20MF8G6_father.g.vcf.gz
ff2cb823-7195-4fc4-a9d3-66803d7c8d96.g.vcf.gz	FM_HPZPPFHN/FM_HPZPPFHN_father.g.vcf.gz

## 6.0.2 Generate phase data for each trio

Each trio was ran through *trioPhaser*. The example below shows the code used to phase one trio.

```
docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest python3 /trio_phaser.py \
  -c FM_RF8RS3YT/FM_RF8RS3YT_child.g.vcf.gz \
  -p FM_RF8RS3YT/FM_RF8RS3YT_father.g.vcf.gz \
  -m FM_RF8RS3YT/FM_RF8RS3YT_mother.g.vcf.gz \
  -o FM_RF8RS3YT/FM_RF8RS3YT_phased.vcf.gz \
  -r haplotype_references \
  --number_of_tasks 22 \
  > FM_RF8RS3YT/trio_phaser_FM_RF8RS3YT.out # Outputs container ID to a file

# Use the container ID to get the log of the "trio_phaser.py" output and store
# the log information to the same file where the container ID was stored.
CONTAINERID="cat FM_RF8RS3YT/trio_phaser_FM_RF8RS3YT.out"
docker logs '$CONTAINERID' >> FM_RF8RS3YT/trio_phaser_FM_RF8RS3YT.out
```

## 6.0.3 Average phasing results across all trios

*trio\_phaser.py* outputs helpful information about how many initial variants there were, how many of the initial variants were phased, how many variants were phased by *SHAPEIT4* that weren't phased using

Mendelian inheritance, how many variants were phased using Mendelian inheritance that *SHAPEIT4* was unable to phase, and how long *trioPhaser* took to execute. This information was averaged across all affected children using the `summarize_neuroblastoma_log_files.py` script. This script takes 2 arguments, 1) where the family/trio directories are located, and 2) where the summary stats should be saved. The summary stats file includes all the information contained in the log output by “trio\_phaser.py”, but does so in a tidy format.

```
docker run -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  python3 /trioPhaser/validate/summarize_neuroblastoma_log_files.py \
  /proj/ \
  /proj/summary_stats.tsv
```

## 7 Test *trioPhaser*’s GRCh37 functionality

### 7.0.1 Download GRCh37 BAM files from the Genome in a Bottle Consortium

We used the Ashkenazim trio from The Genome in a Bottle (GIAB) Consortium to test *trioPhaser*’s GRCh37 functionality. We used aligned (GRCh37)/index BAM files to generate genotype data that could be input into *trioPhaser*. BAM and BAI files were downloaded directly from the GIAB FTP server.

```
# Download .bam and .bai files for all members of the Ashkenazim trio
docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
  ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
  10XGenomics_ChromiumGenome_LongRanger2.0_06202016/HG002_NA24385_son/\
  NA24385_GRCh37.bam -O ashkenazim_trio/son_GRCh37.bam

docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
  ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
  10XGenomics_ChromiumGenome_LongRanger2.0_06202016/HG002_NA24385_son/\
  NA24385_GRCh37.bam.bai -O ashkenazim_trio/son_GRCh37.bam.bai

docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
  ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
  10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
  HG003_NA24149_father/NA24149_GRCh37.bam -O ashkenazim_trio/father_GRCh37.bam

docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
  ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
  10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
  HG003_NA24149_father/NA24149_GRCh37.bam.bai -O ashkenazim_trio/father_GRCh37.bam.bai

docker run -d -v /Data:/proj -w /proj -t \
  dmill903/triophaser:latest \
  wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
```



```

ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
HG004_NA24143_mother/NA24143_GRCh37.bam -O ashkenazim_trio/mother_GRCh37.bam

docker run -d -v /Data:/proj -w /proj -t \
dmill903/triophaser:latest \
wget --no-check-certificate ftp://ftp-trace.ncbi.nlm.nih.gov/\
ReferenceSamples/giab/data/AshkenazimTrio/analysis/\
10XGenomics_ChromiumGenome_LongRanger2.0_06202016/\
HG004_NA24143_mother/NA24143_GRCh37.bam.bai -O ashkenazim_trio/mother_GRCh37.bam.bai

```

### 7.0.2 Generate gVCF files for the Ashkenazim trio

*GATK*'s HaplotypeCaller tool was used to generate chr22 gVCF files for each member of the trio using GRCh37 BAM and BAI files as input. Prior to executing *GATK*, the reference files (contained within the container) were decompressed so *GATK* could use them.

```

docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest /bin/bash -c \
"unzip /fasta_references.zip -d /fasta_references \
&& gzip -d /fasta_references/*.gz \
&& gatk --java-options -Xmx8g HaplotypeCaller \
-L chr22 \
-R /fasta_references/human_g1k_v37_modified.fasta \
-I ashkenazim_trio/son_GRCh37.bam \
-O ashkenazim_trio/son_GRCh37_chr22.g.vcf.gz -ERC GVCF" > genotype_son_GRCh37.out

docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest /bin/bash -c \
"unzip /fasta_references.zip -d /fasta_references \
&& gzip -d /fasta_references/*.gz \
&& gatk --java-options -Xmx8g HaplotypeCaller \
-L chr22 \
-R /fasta_references/human_g1k_v37_modified.fasta \
-I ashkenazim_trio/father_GRCh37.bam \
-O ashkenazim_trio/father_GRCh37_chr22.g.vcf.gz -ERC GVCF" > genotype_father_GRCh37.out

docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest /bin/bash -c \
"unzip /fasta_references.zip -d /fasta_references \
&& gzip -d /fasta_references/*.gz \
&& gatk --java-options -Xmx8g HaplotypeCaller \
-L chr22 \
-R /fasta_references/human_g1k_v37_modified.fasta \
-I ashkenazim_trio/mother_GRCh37.bam \
-O ashkenazim_trio/mother_GRCh37_chr22.g.vcf.gz -ERC GVCF" > genotype_mother_GRCh37.out

```

### 7.0.3 Run chr22 of the Ashkenazim trio through *trioPhaser*

```

docker run -d -v /Data:/proj -w /proj -t dmill903/triophaser:latest \
python3 /trio_phaser.py ashkenazim_trio/son_GRCh37_chr22.g.vcf.gz \
ashkenazim_trio/father_GRCh37_chr22.g.vcf.gz \
ashkenazim_trio/mother_GRCh37_chr22.g.vcf.gz \

```



```
ashkenazim_trio/giab_phased_GRCh37_chr22.vcf.gz \
haplotype_references \
--number_of_tasks 22 \
--build_version 37 > ashkenazim_trio/trio_phaser_giab_GRCh37_chr22.out
```

## References

- Choi, Yongwook, Agnes P Chan, Ewen Kirkness, Amalio Telenti, and Nicholas J Schork. 2018. “Comparison of Phasing Strategies for Whole Human Genomes.” *PLoS Genet.* 14 (4): e1007308.
- Data Sciences Platform @ Broad Institute. n.d. “GATK | Tool Documentation Index.” [https://software.broadinstitute.org/gatk/documentation/tooldocs/3.8-0/org\\_broadinstitute\\_gatk\\_tools\\_walkers\\_phasing\\_ReadBackedPhasing.php](https://software.broadinstitute.org/gatk/documentation/tooldocs/3.8-0/org_broadinstitute_gatk_tools_walkers_phasing_ReadBackedPhasing.php).
- Delaneau, Olivier, Jean-François Zagury, Matthew R Robinson, Jonathan L Marchini, and Emmanouil T Dermitzakis. 2019. “Accurate, Scalable and Integrative Haplotype Estimation.” *Nat. Commun.* 10 (1): 5436.
- Heath, Allison P, Deanne M Taylor, Yuankun Zhu, Pichai Raman, Jena Lilly, Phillip Storm, Angela J Waanders, et al. 2019. “Abstract 2464: Gabriella Miller Kids First Data Resource Center: Harmonizing Clinical and Genomic Data to Support Childhood Cancer and Structural Birth Defect Research.” *Cancer Res.* 79 (13 Supplement): 2464–4.
- Martin, Marcel, Murray Patterson, Shilpa Garg, Sarah O Fischer, Nadia Pisanti, Gunnar W Klau, Alexander Schöenhuth, and Tobias Marschall. 2016. “WhatsHap: Fast and Accurate Read-Based Phasing.” *bioRxiv*.
- Zheng, Grace X Y, Billy T Lau, Michael Schnall-Levin, Mirna Jarosz, John M Bell, Christopher M Hindson, Sofia Kyriazopoulou-Panagiotopoulou, et al. 2016. “Haplotyping Germline and Cancer Genomes with High-Throughput Linked-Read Sequencing.” *Nat. Biotechnol.* 34 (3): 303–11.
- Zook, Justin M, Jennifer McDaniel, Nathan D Olson, Justin Wagner, Hemang Parikh, Haynes Heaton, Sean A Irvine, et al. 2019. “An Open Resource for Accurately Benchmarking Small Variant and Reference Calls.” *Nat. Biotechnol.* 37 (5): 561–66.