FAQ

1. What is MosaicForecast and how to download?

   MosaicForecast (MF) is a machine learning method that leverages read-based phasing and read-level features to accurately detect mosaic SNVs (SNPs, small indels) from NGS data. It builds on existing algorithms to achieve a multifold increase in specificity.

   See more details on "https://github.com/parklab/MosaicForecast".

   You could download it through this command:

```
git clone https://github.com/parklab/MosaicForecast.git
```

2. What are the dependencies of MosaicForecast and how to install them?
   Please refer to "https://github.com/parklab/MosaicForecast" for the specific dependencies. You could either set up an environment with dependencies by using conda (see below in question #5) or use the environment we've built with docker (see below in question #3-4).

3. What is docker, how to install it and how could I use it to build an environment with all dependencies installed?

   "Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package."

   Docker could be download and install docker as described in their website: https://docs.docker.com/

   We have created a docker image with all dependencies of Mosaicforecast installed (https://hub.docker.com/r/yanmei/mosaicforecast), and you could pull the image from docker hub through the command below:

```
docker image pull yanmei/mosaicforecast:0.0.1
```

4. How to share data between the Docker container and the Host? Could I attach a local directory with all of my data in it to the docker container?

Yes, you can! For example, by using the commands below, you would be able to read and write data from your local directory from the docker container:

```
docker run -v ${your_local_directory}:/MF --rm -it yanmei/mosaicforecast:0.0.1 /bin/bash
```

Please note that "${your_local_directory}:/MF" is the absolute path of your local mosaicforecast directory. After attaching your local MF directory to the docker container, you would be able to read and write from that directory in your docker container. The attached directory in the docker container would be "/MF".

You could run MosaicForecast by using the docker image and read/write from your local directory afterwards. For example, you could run the command lines below to run phasing:

```
gunzip hs37d5.fa.gz
Phase.py /MF/demo/ /MF/demo/phasing hs37d5.fa /MF/demo/test.input 20 k24.umap.wg.bw 4
```

5. What if I cannot use docker?

   You could install conda first:

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh

bash Miniconda3-latest-Linux-x86_64.sh
```

and then set up an environment using conda through this command:

```
conda env create --name MF --file environment.yaml
```

The environment 'MF' could be activated through this command:

```
conda activate MF
```

Other dependencies and resources could be downloaded though running:

```
bash downloads.sh
```

6. How to generate the variant call set as input of MosaicForecast?
   In our paper, we used Mutect2 panel-of-normals strategy (v3.5, v3.6 or v3.7 could all work) to call raw somatic variants.

   As described on the GATK website, "A Panel of Normal or PON is a type of resource used in somatic variant analysis. Depending on the type of variant you're looking for, the PON will be generated differently. What all PONs have in common is that (1) they are made from normal samples (in this context, "normal" means derived from healthy tissue that is believed to not have any somatic alterations) and (2) their main purpose is to capture recurrent technical artifacts in order to improve the results of the variant calling analysis."

   You could learn how to generate panel-of-normals callset and use it to call mutations from single-sample from the GATK website:

   https://gatk.broadinstitute.org/hc/en-us

   GATK packages could be download from:
   https://github.com/broadinstitute/gatk/tags

   Basically, it's a four-step procedure summarized below (following procedure is based on **GATK4.1.0.0**, we also have a snakemake pipeline in under the github directory, which parallelize the running by scatter-gather different genomic regions):

Step1: Run Mutect2 in tumor-only mode for each normal sample.

```
gatk Mutect2 \
  -R ref_fasta.fa \
  -I normal1.bam \
  -tumor normal1_sample_name \
  --germline-resource af-only-gnomad.vcf.gz \
  -L intervals.list \
  --interval-padding 100 \
  -O normal1_for_pon.vcf.gz
```

   The "af-only-gnomad.vcf.gz" could be downloaded from ftp://ftp.broadinstitute.org/bundle/Mutect2/ as described in https://gatk.broadinstitute.org/hc/en-us/articles/360036212652. b37. b37 version

(ftp://gsapubftp-anonymous@ftp.broadinstitute.org/bundle/Mutect2/af-only-gnomad.raw.sites.b37.vcf.gz) and hg38 version (ftp://gsapubftp-anonymous@ftp.broadinstitute.org/bundle/Mutect2/af-only-gnomad.hg38.vcf.gz) are provided.

Step 2: Combine the normal calls from mulusing CreateSomaticPanelOfNormals.

```
gatk CreateSomaticPanelOfNormals \
  -vcfs normals_for_pon_vcf.args \
  -O pon.vcf.gz
```

Format of normal_for_pon_vcf.args:

```
normal1_for_pon.vcf.gz
normal2_for_pon.vcf.gz
normal3_for_pon.vcf.gz
```

Step 3: call raw somatic variants with Single tumor sample mode.

```
gatk Mutect2 \
 -R ref_fasta.fa \
 -I tumor.bam \
 -tumor tumor_sample_name \
 --germline-resource af-only-gnomad.vcf.gz \
 --pon pon.vcf.gz \
 -L intervals.list \
 --interval-padding 100 \
 -O tumor_unmatched_m2_snvs_indels.vcf.gz
```

Step 4: filter somatic variants:

```
gatk FilterMutectCalls \
```

```
-R ref_fasta.fa \

-v tumor_unmatched_m2_snvs_indels.vcf.gz \

-O tumor_unmatched_m2_snvs_indels_filtered.vcf.gz
```

To generate a raw call-set from Mutect2, Variants tagged as "panel_of_normals", "str_contraction", "triallelic_site" or "t_lod_fstar" by GATK3 were excluded (python MT2_filter.py); variants within Segmental duplication regions and "clustered regions" with misalignment issues were excluded; variants with <0.02 AF calculated by Mutect2 were excluded; Variants with VAF ≥0.4 or present in the gnomAD database[1] were removed as likely germline variants. For mosaic indels specifically, variants within simple repeats[2], and RepeatMasker regions were also excluded.

For GATK4, variants tagged as "panel_of_normals", "str_contraction", "multiallelic" or "t_lod" were excluded. Refer to our Snakemake pipeline on github.

We have made a script to do preliminary filtering (allele fractions, filtrations, excluding Segmental duplication regions and hard-to-map regions) based on MuTect2-PoN calls, additional filtrations still need to be done to filter variants present in gnomAD database (as described below in question #7):

```
Python MuTect2-PoN_filter.py test demo/test.Mutect2.vcf
resources/SegDup_and_clustered.bed
```

You could also run MosaicForecast on variants generated by other software tools such as GATK Haplotype caller with ploidy number set to >2, MosaicHunter or simply by using samtools mpileup scanning, e.g., by taking all sites with ≥2% non-reference bases as potential mutations, as described in our paper, but if you would want to use our pre-trained models, Mutect2-PON calls would be the best choice since our models are trained based on Mutect2 calls.

7. How to add annotations of gnomAD population minor allele frequencies?
   As an option, you could use ANNOVAR to annotate the variants. The package of ANNOVAR could be downloaded here:

   http://annovar.openbioinformatics.org/en/latest/user-guide/download/#annovar-main-package

You could download genomAD to the directory “humandb/” through this command:

```
annotate_variation.pl -buildver hg19 -downdb -webfrom annovar gnomad_genome humandb/
```

You could then add information of the gnomAD MAFs to your variants (ex1.avinput):

```
annotate_variation.pl -filter -buildver hg19 -dbtype gnomad_genome ex1.avinput humandb/
```

Then you could choose a threshold to do MAF filtration. We suggest keeping all “filtered” sites and the “dropped” sites with MAF=0.

8. How to exclude variants within certain regions such as Segmental Duplication regions?

   You could easily exclude certain genomic regions using bedtools. We have installed bedtools in the docker image, you could also install it following the instructions from their Website (https://bedtools.readthedocs.io/en/latest/content/installation.html):

```
wget https://github.com/arq5x/bedtools2/releases/download/v2.28.0/bedtools-2.28.0.tar.gz

tar -zxvf bedtools-2.28.0.tar.gz

cd bedtools2

make

Usage: bedtools subtract [OPTIONS] -a <bed/gff/vcf> -b <bed/gff/vcf>
```

9. Could I use case-control design instead of panel-of-normals to call mosaics in non-cancer samples?

   Unlike somatic variants in cancer samples, a large proportion of mosaic variants in non-cancer samples do not have selective advantages and would present in multiple tissues[3]. As a result, using a case-control design using a matched “normal” tissue from the same individual could cause a great loss in sensitivity.

10. Does removing recurrent variants in panel-of-normals result in removal of biologically meaningful variants?

Given the low somatic mutation rate in non-tumor samples, the likelihood of two somatic mosaic mutations occurring at the same nucleotide position across individuals is vanishingly small; much more likely are rare systematic artifacts.

In order to eliminate the possibility of removing biological meaningful signals to the lartest extent, we would suggest using healthy individuals to construct panel-of-normals. If you don't have any sequencing data from healthy individuals, we would suggest to find some public data that are technically representative of your sequencing data (i.e., select some samples from 1000 Genomes Project sequenced on the same platform using the same chemistry, and analyzed using the same toolchain). If you don't have any public data available representative of your sequencing data, even an unmatched PoN could be used to in filtering sequencing artifacts (mapping artifacts or polymerase slippage errors). Moreover, it's also ok if you don't use panel-of-normals, using gnomAD database to filter your variants might be ok, but you would expect to have slightly lower validation rate (~10% decrease).

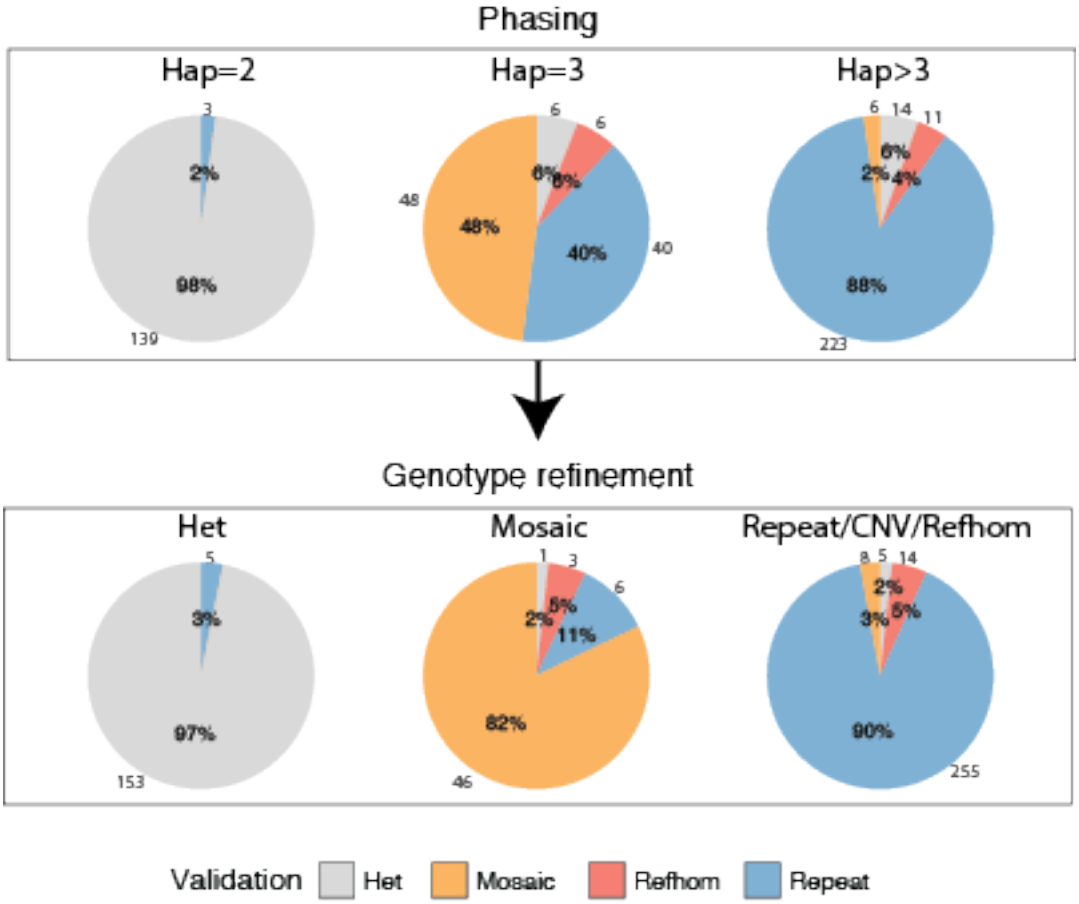
11. Could I relax or strengthen the criteria to call mosaics by using MosaicForecast, and how to?
    Yes. Given that different studies have different aims and would have different compromises between sensitivity and specificity, there are three ways to adjust the criteria:
    a. Adjust the input variant call-set:
       You could use one or multiple methods to call raw variants as input of MF, such as Mutect2-PON, GATK haplotype caller (set the ploidy number to > 2, in order to be able to call variants with low AFs), MosaicHunter and even samtools mpileup (by simply extracting variants with ≥2% alt alleles across the genome). Although our model is trained based on callstes from Mutect2-PON, we also tried the pre-trained model on other callsets and proved that our model could augment specificity of other methods in detecting mosaics. For example, the validation rate increased from 1.8% to 42.3% for samtools-mpileup by applying our model.
       To maximize the specificity and sentivity of your toolchain, you could also train a different model by using MF.
    b. Through adjusting threshold of genotyping probabilities:
       Based on our experience, higher probabilities of the "mosaic" genotype would give rise to higher validation rate. You could set a different threshold of the "mosaic" column to relax or strengthen the criteria of calling mosaics.
    c. Through adjust filters:
       You could adjust several filters of MF, such as the variant allele fractions. Now we don't include regions with mappablity issues (UMAP mappability score=0, the "mappability" column), and you could include the region back at the cost of having higher FP rate.

12. Are there additional filters that you suggested to use?

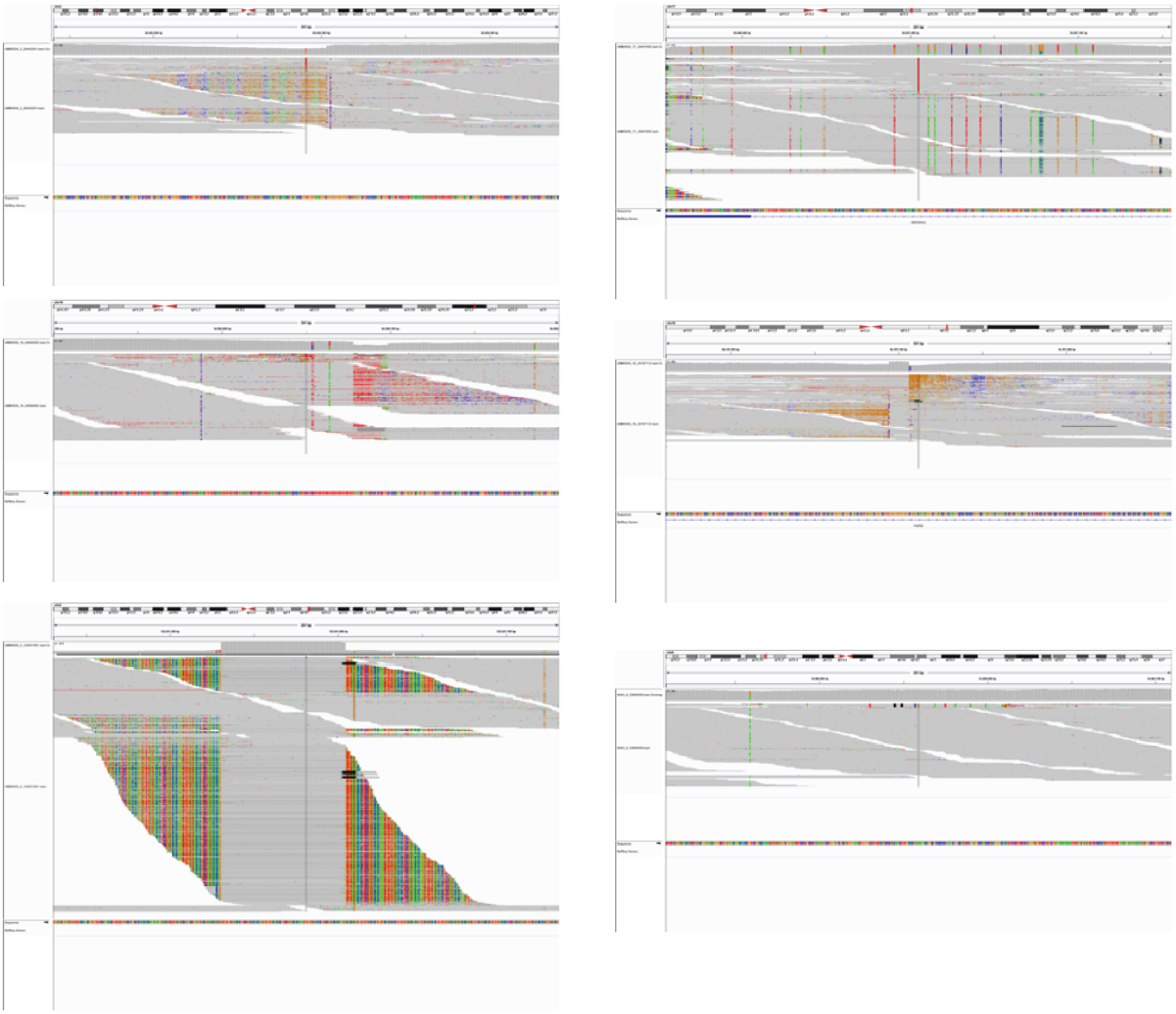
a. We would suggest checking the total number of mosaics called from each individual, and if an individual carries exceptionally high number of mosaics, it's most probably there are some contamination problems.
b. When the read depth is extremely high (i.e. twice or 1.5 time as high as the average depth), it's likely that this region has non-unique mapping problems.

13. When should I use the pre-trained model and when should I train a customized model by using my own data? Does the brain WGS-trained model generalize well to other datasets?
    Our model generalizes well to WGS data (Illumina HiSeq, paired-end sequencing, PCR-free). If you have sequencing data with a quite different chemistry (exome capture or panel sequencing), it's highly suggested to train a customized model.

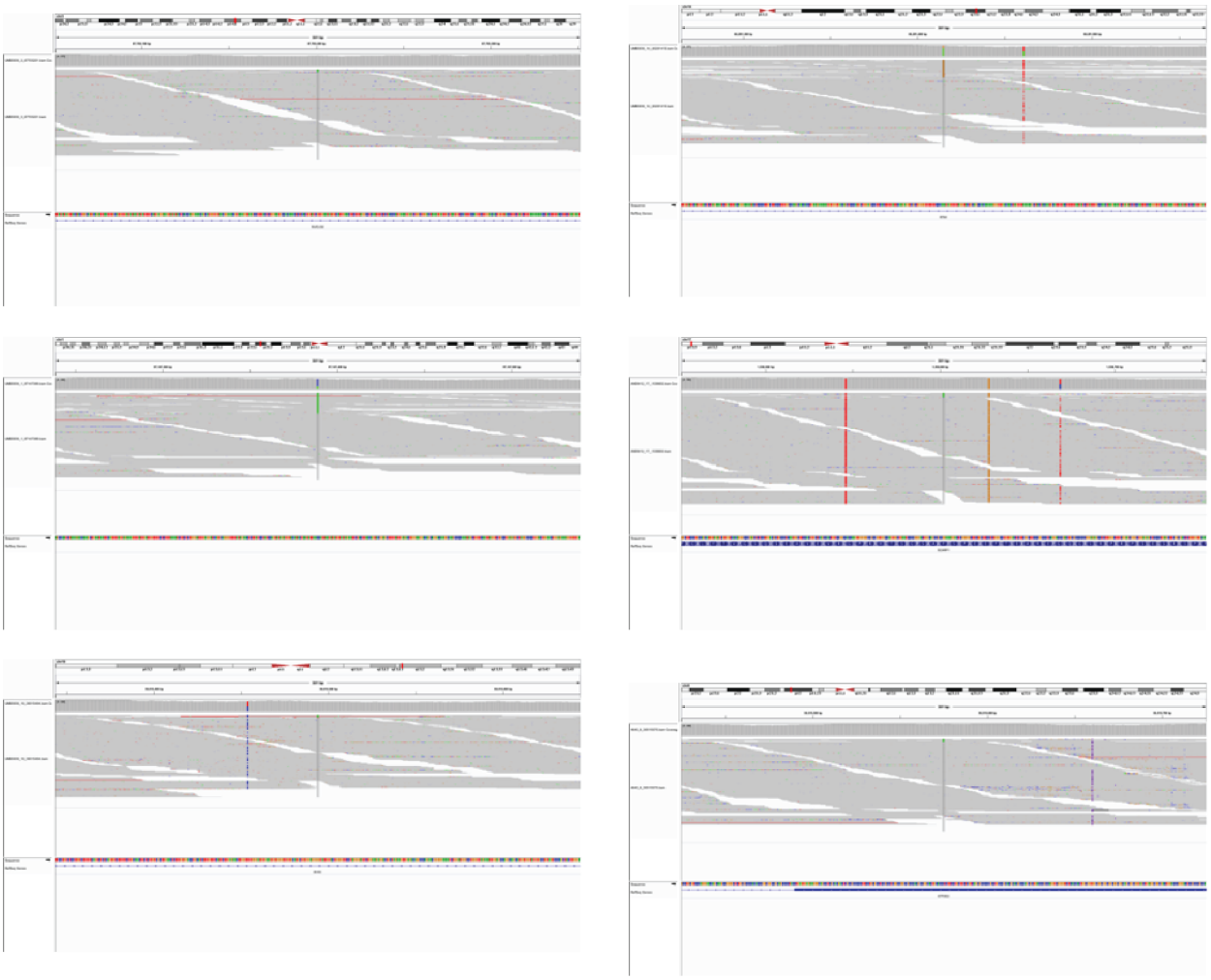
14. What if I don't have experimental evaluations to correct phasing?
    We strongly recommend using Refined genotypes instead of Phasing genotypes, since ~50% of hap=3 sites were validated as "repeat" variants in our dataset:



In case you don't have experimentally evaluated sites, it's ok to manually-check ~100 hap=3 sites with igv, and mark the sites in messy regions as "repeat". Here are some examples of "hap=3" sites experimentally evaluated as "repeat":



and here are some examples of "hap=3" sites experimentally evaluated as true positive mosaics:



15. Is model trained with data at one depth applicable to calling mosaics in other datasets with different read depths?
    The model is fairly robust in that regard. We examined the performance of models trained with test datasets of read depths 50-250X. With both simulated and real WGS data, the models performed robustly across different read depths, although the performance is optimal when training and testing datasets have similar coverages.

16. How to deal with haploid chromosomes?
    Our model is trained using all phasable sites from diploid chromosomes (autosomal chromosomes and chrX in female) and is applied to non-phasable sites including those from haploid chromosomes (chrX/Y in male). The performance on haploid chromosomes is only slightly worse: the IonTorrent validation rates on the 60 PCR-free samples were ~95% (154/162) for sites from diploid chromosomes and ~83% (15/18) for sites in haploid chromosomes (Supplementary Table 8).

17. Could I call somatic indels from simple repeat regions?

No. Somatic indel calling is extremely challenging given the polymerase slippage problems, and polymerase slippage problems become more serious in simple repeat regions. We would suggest excluding all of the simple repeat regions when calling mosaic indels.

18. Could I use PCR-based samples to call somatic indels?
No. In our data, none of candidate indels from PCR-based samples were validated true. It's probably because of the polymerase slippage problem is more serious in sequencing data from PCR-based samples.

1. Karczewski, K.J. *et al.* Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. *bioRxiv*, 531210 (2019).
2. Benson, G. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res* **27**, 573-580 (1999).
3. Dou, Y., Gold, H.D., Luquette, L.J. & Park, P.J. Detecting Somatic Mutations in Normal Cells. *Trends Genet* **34**, 545-557 (2018).