SeqVerify: A quality-assurance pipeline for whole-genome sequencing data

Merrick Pierson Smela*[1], Valerio Pepe*[1], and George M. Church[1,2]

[1]Wyss Institute at Harvard University
[2]Harvard Medical School Department of Genetics

September 27, 2023

*Equal contributions

1 Abstract

Over the last decade, advances in genome editing and pluripotent stem cell (PSC) culture have let researchers generate edited PSC lines to study a wide variety of biological questions. However, abnormalities in cell lines can arise during PSC culture or due to undesired editing outcomes. These can include aneuploidy, on-target and off-target editing errors, and microbial contamination. Any of these abnormalities can invalidate experiments, so detecting them is crucial. The ongoing decline of next-generation sequencing prices has made whole genome sequencing (WGS) an effective quality control option, since WGS can detect any abnormality involving changes to DNA sequences or presence of unwanted sequences. However, until now, this approach has suffered from a lack of easily usable data analysis software. Here, we present SeqVerify, a computational pipeline designed to take raw WGS data and a list of intended edits, and verify that the edits are present and that there are no abnormalities. We anticipate that SeqVerify will be a useful tool for researchers generating edited PSCs, and more broadly, for cell line quality control in general.

2 Introduction

2.1 The need for stem cell quality control

Pluripotent stem cells (PSCs) have found important uses in many areas of biological research, and their ability to differentiate into a variety of cell types has enabled the development of cell-based therapies derived from PSCs. Gene editing technologies such as CRISPR-Cas9 have enabled the engineering of PSC

lines containing specific alleles of interest, such as disease-relevant mutations or fluorescent reporters.

However, over the years, researchers have identified several common abnormalities that can arise during PSC culture. First, PSCs can become aneuploid due to chromosomal rearrangements or mis-segregation. The most frequent aneuploidies in PSC cultures involve chromosomal or sub-chromosomal duplications (1) (2), and some of these, such as gain of 20q11.21, have been characterized to affect the phenotypes of the cells (3) (4). Such aneuploidies are relatively common, affecting roughly twelve percent of tested PSC lines on average (2), with the frequency increasing over long-term passaging.

Second, PSCs can gain point mutations, which can be enriched during prolonged culture due to providing a growth advantage. For example, the tumor suppressor genes *TP53* and *BCOR* are recurrently mutated in PSCs (5) (6). Although mutation rates in PSCs are not abnormally high (7), harmful mutations can often be present in somatic cells used to derive induced PSCs (6), or can occasionally arise during PSC culture (5).

Third, as with other cell cultures, PSC cultures can be contaminated with microbes such as *Mycoplasma*. This is a relatively common problem; a study in 2015 was able to detect sequencing reads mapping to *Mycoplasma* in eleven percent of mammalian cell culture datasets in the NCBI Sequence Read Archive (8). Good cell culture practice, including recurrent testing, is essential for avoiding contamination.

Fourth, PSCs can have epigenetic abnormalities such as loss of imprinting (9), or erosion of X-inactivation for female PSC lines (10). Since these abnormalities do not alter the genomic DNA sequence, they are not currently detectable by short-read WGS technology, so alternative methods should be used to detect them.

Furthermore, gene editing of PSCs can introduce additional abnormalities. At the on-target site, undesired editing outcomes may be present. Traditional PCR-based genotyping can sometimes fail to detect these outcomes when they involve a large insertion of plasmid or mitochondrial DNA into the target site (11). Although off-target editing is usually rare (12), the rate may be greatly increased when less specific editing tools, such as APOBEC-based cytosine base editors, are used (13) (14).

In order to ensure valid experimental results, and the safety of PSC-derived therapeutics, it is important to detect these abnormalities and choose PSC lines without them. Whole genome sequencing (WGS) offers a cost-effective and powerful way to screen for multiple types of abnormalities simultaneously. Here, we present a computational pipeline, SeqVerify, that analyzes short-read WGS data to validate on-target genome editing and detect mutations, aneuploidies,

and microbial contamination.

3 The SeqVerify Pipeline

3.1 Pipeline Overview

SeqVerify operates on three main types of input for the majority of its results (Figure 1). These are paired-end short-read sequencing data, a reference genome to align this data to, and a list of "markers" - untargeted or targeted sequences —that the user wants to check for. The genome and the markers are first combined into an "augmented genome", a genome that includes the markers as untargeted or targeted edits. The sequencing data is then aligned to it to produce a new BAM file containing all the data necessary to parse out insertion sites for the intended edits, as well as data regarding copy number variation, which are subsequently output. Should the user also desire to check for microbial contamination through KRAKEN, a database is required, and the BAM file will additionally be used in that process. Single Nucleotide Variant analysis requires a separate reference genome and annotation database (marked as ClinVar and GRCh38 in Figure 1, but they can be substituted if the user requires it), which create a separate BAM file to be processed into an annotated VCF and filtered to a final readout.

3.2 Input Requirements

As input, SeqVerify uses FASTQ files of paired-end reads of whole genome sequencing data. We typically use a sequencing coverage of 10X. A reference genome in FASTA format is also needed. For human data, SeqVerify development was largely centered around and tested with T2T-CHM13v2.0, so its use is recommended over other popular human genomes such as GRCh38, even though any genome will work. For the most basic use of SeqVerify, the only other requirement is a list of genome edits. These can be either FASTA sequences of untargeted insertions (*e.g.* transposons or lentiviruses which integrate randomly), and/or a tab-separated list of specific genomic coordinates and edit sequences. The latter option lets SeqVerify automatically construct an edited reference genome to which reads will be aligned.

In addition to the basic analysis, SeqVerify can also perform contaminant detection, which requires a valid KRAKEN2 database. Finally, to detect and annotate SNVs, SeqVerify requires a VCF annotation database and a reference genome compatible with it.

3.2.1 Automatic download of reference data

The `seqverify --download_defaults` command will automatically download all the default files for a standard analysis of human cells. These are:

- T2T-CHM13v2.0 as the overall reference genome,
- GRCh38 as the reference genome for Single Nucleotide Variant calling,
- PLUSPF_8GB as the default KRAKEN2 database,
- ClinVar as the default VCF annotation database,
- snpEff.config as a fresh snpEff configuration file should the user want to manually specify advanced snpEff options.

SeqVerify uses the default `curl` command to easily download all of them from their respective FTP servers at once, such that the user does not need to find the default parameters themselves, and stores them in a `seqverify_defaults` folder in the working directory where the command is run. If, when running the pipeline, certain options are left blank (reference genome, KRAKEN database, etc), SeqVerify will automatically attempt to use these from the `seqverify_defaults` folder to correctly run the pipeline.

3.3 Alignment

Alignment of the reads to the provided reference genome is executed through BWA and its Minimal Exact Match algorithm (BWA-MEM) (15). In order for edit verification to occur, SeqVerify will copy the desired transgene sequences into a copy of the reference genome (either as additional sequences in the FASTA or at specific location within pre-existing chromosomes, specified by the user) prior to alignment, and will then align the reads to the copy of the reference genome where transgenes are included as opposed to the original. SeqVerify can also handle deletions or replacements in a similar manner to insertions.

3.4 Validation of Edits at Known Target Sites

As mentioned above, the SeqVerify pipeline can be provided an input file listing genomic coordinates, and DNA sequences to be inserted, edited, or deleted at those coordinates. SeqVerify will use this information to generate an edited reference genome, corresponding to the user's intended edits. After aligning reads to this genome, SeqVerify will automatically generate figures using IGV (16), displaying the genomic coordinates provided by the user and showing the aligned reads. On-target homozygous edits (Figure 2A) are easily distinguishable from undesired outcomes (Figure 2B). Since SeqVerify saves the BAM file output after alignment, the user can also manually open this file in IGV if more detailed inspection is desired.

3.5 Detection of Untargeted Transgene Insertions

SeqVerify will also accept untargeted transgene sequences as input. This is useful for finding the insertions of transposons and lentiviruses, or for detecting

inadvertent integration of plasmids used in editing. User-provided sequences are appended to the reference genome and treated as extra chromosomes. Similarly to targeted insertions, SeqVerify will display reads aligning to these transgenes using IGV (Figure 3A).

However, manually looking through alignments to detect insertion sites is tedious and does not scale well. Therefore, SeqVerify will also automatically find insertion sites by detecting junctions between transgenes and the host genome. After reads have been aligned to the modified reference genome, SeqVerify will parse the output SAM file and extract read pairs where a transgene is aligned to a host chromosome.

Once the data has been processed, a human-readable text file containing it is produced, formatted as follows (Figure 3B). Every sublayer of the dictionary is indented by one additional tab, so it should also be possible to easily parse automatically if needed for further post-SeqVerify processing or quality assurance.

3.6 Copy Number Variation

For copy number variation (CNV) detection and analysis, SeqVerify uses the CNVPytor package (17) and generates a Manhattan plot of the normalized read depth at a default resolution of 100kbp. These plots are useful for visualizing aneuploidies, as shown in Figure 4. CNVpytor will also call CNVs and output a list of any CNVs found in the sample.

Additionally, transgene copy numbers can be estimated using the normalized read depth. Transgene plots are produced using an automatically generated IGV batchfile, which takes screenshots of genomic coordinates corresponsing to transgene sequences. This is helpful for distinguishing full vs. partial transgene insertions. However, taking automatic screenshots requires the XVFB package which is not available on all systems (for example, MacOS). Therefore, we developed an alternative plotting system using the Python library matplotlib. The internal plotting tool uses the output from running `samtools depth` on the data to compare the amounts of reads per 30bp to an estimate of an expected value of reads per 30bp bin calculated from the average read depth. It then plots this data as a histogram of copy number against bin, and makes one of these plots per transgene.

3.7 Contaminant Detection

SeqVerify performs contaminant detection using KRAKEN2 (18). After aligning reads to the human reference genome, SeqVerify extracts all read pairs where either the read or its mate are unmapped, and runs KRAKEN2 to classify them. We recommend that a KRAKEN database containing human sequences be used (for example, PlusPF-8 which is used as default), since unmapped reads may

still contain some human sequences which may otherwise generate false positives.

Further analysis to estimate contaminant species abundance is then performed with BRACKEN, a sister tool to KRAKEN2 which does not require any additional arguments or databases (19). After having been generated, the KRAKEN2 and BRACKEN reports are placed into the same output folder as the CNV graphs and insertion site readout for ease of post-pipeline access.

We validated this contaminant detection by running it on data from twelve hiPSC lines which tested negative for *Mycoplasma* by PCR-based methods, and one which tested positive. We found no signs of contamination (less than 50 reads for any bacterial or viral species) in the negative samples, and the positive sample had 23,891 reads mapping to *Mycoplasmopsis arginini* (Supplementary File 1). Furthermore, we analyzed previously published WGS reads from HEK293 cells (SRA number: SRR18054575) to see if this method could detect adenovirus 5, which was used to transform those cells in their original derivation from fetal tissue. We successfully detected 497 reads mapping to Adenoviridae, of which 104 specifically matched human adenovirus 5. Additionally, we found that the HEK293 cells used in that dataset were contaminated with *Mesomycoplasma hyorhinis* (Supplementary File 1).

3.8 Single Nucleotide Variant Analysis

Single Nucleotide Variant (SNV) analysis requires re-alignment of the reads to an un-edited reference genome. Most human SNV annotation databases (for example, ClinVar) use hg38 instead of CHM13 as their reference genome. Therefore, in the SNV portion of the pipeline, the reads are re-aligned to hg38, with BCFTOOLS subsequently used to generate a VCF file of the variants found in the reads (20). SeqVerify then annotates the VCF file using SnpEff and SnpSift and (by default) the use of the ClinVar clinical database for further annotation and loss of function and effect prediction (21) (22). Finally, SeqVerify then takes the annotated VCF file and filters it, generating a human- or machine-readable readout of all mutations above a certain severity threshold, their effects, genes, any loss of function, and their homozygosity, among other data. An example of such a readout is provided in Supplementary File 2, which also contains other example output.

4 Usage

4.1 Installation

SeqVerify can be downloaded using the Bioconda package manager, where it comes pre-packaged with all the dependencies needed to run all optional features (IGV plots instead of matplotlib, KRAKEN2/BRACKEN analysis), or from

GitHub with dependencies installed separately. It was developed for and tested on Linux systems (including Windows Subsystem for Linux), although MacOS or other Unix kernel-based systems will also be able to run it, provided they can run Python and tools such as SAMTools and BWA, both of which are essential to SeqVerify's functionality. Automatic IGV plots do not currently work on MacOS. SeqVerify can also be downloaded through conda from the command line as follows:

```
conda install -c bioconda seqverify
```

In terms of technical specifications, any system powerful enough to run BWA-MEM in reasonable time will also be acceptable for SeqVerify, and there are options available for multithreading and limiting memory usage that permit users to tune SeqVerify to their needs. The overall runtime using 20 threads on a Intel® Xeon® Processor E5-2683 v4 (40M Cache, 2.10 GHz) is approximately 11-12 hours per sample at 10X sequencing depth, increasing proportionally with increasing depth.

4.2 Running

SeqVerify only includes the `seqverify` command, so all calls through the package are done through tuning its options.

```
seqverify --reads_1 sample_1.fastq
--reads_2 sample_2.fastq --inexact transgenes.fa --exact commands.txt
```

The above command is the minimal SeqVerify call with both exact and inexact (transgene) insertions: it requires the forward and reverse reads in FASTQ format, a FASTA file containing the sequences that should be reported on in the insertion site detection portion of the pipeline, and a TXT file formatted as a command file to specify what exact insertions were made such that the reference genome can be altered. With no additional options, this will use the default T2T-CHM13v2.0 reference genome for alignment, name its output folder `seqverify_output` use a single thread, a maximum of 16GB of RAM, not generate KRAKEN2 and BRACKEN reports, will report insertions within 500bp of each other as belonging to the same insertion site (except for chimeric and split reads, where an exact site is known), its CNV Manhattan plot will use 100kbp sections of the genome, and it will delete its temporary folder after it has completed running.

All of these options can be changed:

- `--output` sets the name of the sample, affecting most output filenames and the folder name. It is set to `output` by default, setting the output folder to be named `seqverify_output`.

- `--genome` takes in the file name of the reference genome to be used for everything except (usually) SNV analysis. If left blank as above, it uses T2T-CHM13v2.0, the default genome downloaded by the pipeline.

- `--kraken` can be set to enable KRAKEN2/BRACKEN analysis, as long as the `--database` flag is also set and is followed by a path to a valid KRAKEN2 database (if left blank, SeqVerify will use the default database described above).

- `--variant_calling` can be set to enable SNV analysis on the sample. It takes two additional arguments: the genome to be used to re-align the reads for SNV analysis, as well as the annotation database to use (if left blank, both of these have default options the pipeline can fall back upon). `--variant_intensity` sets the lowest intensity to be reported in the final readout, out of $MODIFIER, LOW, MODERATE, HIGH$.

- `--threads` and `--max_mem` regulate performance: the former sets how many threads should be used by the pipeline, the latter puts a cap on memory in the pipeline's most memory-intensive process, Burrows-Wheeler alignment, as well as the Java-based subprocesses that the pipeline uses.

- `--start` allows the user to start and stop at any point in the pipeline; this can be done for core efficiency (if on a job scheduler, run all the single-threaded portions on one job, and all the multi-threaded portions on another), as well as the event that parts of the pipeline have already been run but more analysis is desired, or in the event of an error such as the machine turning off, allows for the pipeline to pick back up where it left off.

- `--granularity` and `--min_matches` set the insertion site detection parameters: the former regulates how wide the window of a single insertion site is (default: 500), and the latter sets how many matches must be present at a single site for the site to appear in the readout (default: 1, but a higher number may reduce false positive alignments due to repetitive DNA or other factors).

- `--manual_plots` can be set to use the alternative matplotlib system for plotting transgene CNV, and `--bin_size` can be used to set the bins for the Manhattan plot (default: 100000).

- `--keep_temp` can be set to keep the temp folder if a user wants to keep the temporary files (e.g. the modified reference genome, a SAM file containing the unaligned reads only, etc.).

Once the desired flags have been selected and tuned, the command can be run.

4.2.1 Command Files

SeqVerify is set up to take exact gene edit sites as inputs, as well as inexact (usually transgene) edit sites, in making the insertion site readout. While inexact edits can just be specified by providing the FASTA file of the transgene that the user wants to check for (which will be appended to the genome as an extra nucleotide sequence), to place exact reads SeqVerify requires some additional information, such as the location of the edit, and whether the edit is a deletion, insertion, or replacement.

This is done through a "command file", a specially-formatted text file that the `--exact` flag takes as its argument. One command is uniquely specified by the name of the chromosome (or other sequence) where the edit is taking place, the start and end coordinates to be deleted (unless the command is a pure insertion), and the sequence to be inserted (unless the edit is a pure deletion, in which case no sequence is required), and every line corresponds to a separate command. Commands are thus of the form "CHR:START-END SEQUENCE", where the whitespace in between CHR:START-END and SEQUENCE is a tab character.

SeqVerify will delete the bases from START to END exclusive of both (i.e. deleting bases START+1 to END-1). For example, the command `chr2:0-10` will delete the first 10 bases in chr2 and not replace them with anything. If a sequence is specified, SeqVerify will insert it after deleting the bases from base START+1 onwards: for example, `chr5:10-20 GCT` will delete the 11th to 19th bases and place GCT in the 11th, 12th, and 13th slots. A pure insertion with no deletion can be specified by using the same coordinate for both start and end: `chr1:1-1 AGCT` will not delete anything and insert AGCT after the first base (i.e. slots 2-5).

Should there be multiple commands acting on the same chromosome that may influence one another (such as a command deleting 3 bases but inserting 5, which will shift all other commands after the site of the insertion by two bases), SeqVerify will automatically handle change in the base coordinates such that the user does not need to work out the effect that a command will have on other commands themselves.

4.3 Interpreting Output

SeqVerify will output a single folder, `seqverify_output` where *output* is the value of the required `--output` argument, the name of the sample. This will contain:

- `seqverify_output_markers.bam` and its corresponding index, a BAM file containing the reads aligned to the genome with the addition of the transgene sequences.

- `seqverify_readout.txt`, the aforementioned readout for insertion site detection.
- `output.pytor`, the CNVPytor binary file, which can be used to generate further plots or further process the CNV data if necessary.
- `output.global.0000.png`, the Manhattan plot of the copy number across the genome provided.
- Transgene copy number histograms, through IGV or matplotlib, named `fig_marker.png` where *marker* is the name of the transgene as given in the FASTA file.

An example of SeqVerify output (excluding BAM and VCF files due to size limitations) is provided as Supplementary File 2.

If KRAKEN2 analysis is performed, SeqVerify will also output `classified_seqs_output.kreport`, the KRAKEN2 report on the reads, as well as `classified_output.kraken`, its binaries, and `classified_seqs_output_1.fq` and `classified_seqs_output_2.fq`, FASTQ files containing the forward and backward sequences, respectively, that were classified as contaminants.

If Single Nucleotide Variant Analysis is performed, SeqVerify will also output `seqverify_output.ann.vcf`, the full annotated VCF, as well as `seqverify_output_variants.tsv`, the readout containing easily-readable information about all mutations above a certain severity threshold.

Selecting `--keep_temp` will also keep all the temporary files, adding potentially hundreds of gigabytes and another folder `seqverify_output_temp` including the intermediate SAM files produced during alignment, the coverage map used to compute the CNV analysis, and the FASTQ files for all unaligned reads, among other files.

5 Discussion

5.1 Comparison of whole genome sequencing with previous quality control methods

Validating on-target editing PCR and Sanger sequencing is cost-effective for initial screening, but struggles to detect certain unwanted editing outcomes such as large insertions of plasmid or mitochondrial DNA (11). Additionally, some transgene delivery methods, including lentivirus and PiggyBac transposons, involve the random insertion of transgenes into the host genome. Detecting the insertion sites of these transgenes is important for quality control to ensure that essential host genes are not compromised. Furthermore, if plasmids are introduced into cells for gene editing, unwanted plasmid integration events may occur. Specialized PCR-based methods can efficiently map insertion sites

of known DNA sequences (23), although they may miss partial insertions where the primer binding site is not inserted. WGS can identify insertion sites in an unbiased manner, although if multiple different transgenes are inserted using similar transposons, short-read sequencing cannot always definitively assign the transgene present at each insertion site.

Detecting deleterious mutations When establishing a new cell line, it is important to rule out the presence of deleterious mutations. These mutations may be due to off-target editing, or may arise spontaneously. Targeted amplification and sequencing of mutation hotspots or predicted off-target edit sites can provide some information, but may miss important mutations. WGS is the only practical way to detect mutations across the entire genome.

Detecting aneuploidy The traditional method of detecting aneuploidy is by karyotyping, with G-banding or FISH. This is effective, but laborious, requiring the preparation and staining of metaphase spreads. Additionally, smaller abnormalities, such as 20q11.21 duplication which is common in PSCs, may be missed with standard karyotyping methods (3).

In recent years, DNA microarrays (for example, Thermo Fisher KaryoStat+) have also been used to detect aneuploidy. These are more sensitive, and require only a DNA sample. However, the decrease in cost of WGS over the last few years has made WGS a cost-competitive alternative for this method, with the added benefit of achieving even higher resolution. One drawback of short-read WGS (and especially microarrays) relative to karyotyping is that they are less sensitive at detecting balanced chromosomal translocations. If it is important to detect these translocations (and it may not be, given that they are rare, and typically have mild effects (24), then traditional karyotyping or long-read sequencing should be used.

Detecting microbial contamination Testing for microbial contamination, especially *Mycoplasma*, is a critical part of good cell culture practice. This can be done by multiple methods, including PCR and enzyme-based kits.
We do not recommend WGS as a routine screening method for contamination, due to higher cost, longer turnaround time, and lower sensitivity than PCR. Nonetheless, WGS data can be analyzed to check for the presence of microbial contamination. In principle, any contaminant with a DNA genome can be detected. This is an advantage over PCR-based methods, which can only detect contaminants matching the PCR primer pairs used.

5.2 Conclusion

In recent years, WGS has emerged as a powerful tool for genetic research and clinical diagnostics. The continued decline in sequencing prices over time has made WGS an attractive method, and we anticipate this decline will continue

in the future. Remarkably, WGS (at 10X coverage) is now cheaper than karyotyping, and cost-competitive with commercial microarray-based services such as KaryoStat. However, until now, the lack of convenient data analysis methods has presented a barrier for routine use of WGS in cell line quality control.

Therefore, we developed SeqVerify, a pipeline to analyze WGS data to validate genetic edits and check for abnormalities, including aneuploidy, mutations, and microbial contamination. SeqVerify is easily installable using bioconda, and provides a start-to-finish pipeline, taking raw sequencing reads as input and outputting quality control data. We have been using SeqVerify internally in our lab for the past year, and are now publishing it to share with others.

The main limitations of SeqVerify are based on its use of short-read DNA sequencing as input. SeqVerify is not set up to detect balanced translocations, and in the case of different transgenes with shared terminal regions, it cannot tell which transgene is present at each insertion site. Long-read sequencing would be a better option in those cases. Furthermore, epigenetic or transcriptional abnormalities cannot be detected based only on DNA sequence data.

Despite these limitations, WGS is a highly effective "all-in-one" method for detecting the most common abnormalities in cell lines, and validating on-target edits. We agree with the sentiment expressed in a 2020 paper about aneuploidy detection by Assou *et al.*, who wrote: "Ultimately, we anticipate that NGS will become the primary technique for assessing hPSC quality because sequencing costs continue to decrease." (1) We believe that SeqVerify will unlock the potential of WGS for hPSC quality control, and more broadly, for verifying the quality of any cell lines.

6 Experimental Methods

6.1 iPSC culture

Human iPSCs were cultured in mTeSR Plus medium (Stemcell Technologies) on standard polystyrene culture plates coated with Matrigel (Corning) or Geltrex (Thermo Fisher). Four lines were used: PGP1 (male) and ATCC-BSX0115, ATCC-BSX0116, and F66 (all female). Cells were passaged as small clumps using 0.5 mM EDTA and treated with 10 mM Y-27632 for 24 hr after passage. Cells were routinely tested for *Mycoplasma* using the ATCC Universal Mycoplasma Detection PCR kit. All tested negative, except one sample known to be contaminated with *Mycoplasma* that was used solely as a positive control for the KRAKEN2 analysis.

6.2 Generation of knock-in iPSC lines

Generation of knock-in lines was performed as previously described (25). Briefly, homology donor plasmids were constructed by Gibson assembly of a bacterial plasmid backbone, 5' and 3' homology arms PCR-amplified from genomic DNA, and a fluorescent protein insert. For all knock-ins except *SYCP3*, the insert also contained a PGK-PuroTK selection marker flanked by Rox sites, which was excised upon expression of Dre recombinase. sgRNA oligos targeting the knock-in site were cloned into pX330 (Addgene #42230), which expressed the sgRNA and Cas9. Plasmid sequences are given in Supplementary File 3.

To generate each line, homology donor plasmid and sgRNA/Cas9 plasmid (1 µg each) were co-electroporated into 200,000 hiPSCs using the Lonza 4D nucleofector system with 20 µL of P3 buffer and pulse setting CA-137. The cells were plated in one well of a 6-well plate and, for all knockins except *SYCP3*, selection was begun with puromycin after 48 hours. Subsequently, colonies were picked manually with a P20 pipette, transferred to a 96-well plate, and expanded. If required, a further round of electroporation was performed with pCAGGS-Dre to excise the PuroTK selection marker. During this step, selection was performed with ganciclovir (4 µM).

Preliminary genotyping was performed by PCR (primers sequences and gel images are provided in Supplementary Files 4 and 5), and editing was further confirmed by whole genome sequencing as described below.

6.3 DNA extraction and sequencing

Genomic DNA was extracted from hiPSCs using the Qiagen DNeasy Blood and Tissue kit. 1 million hiPSCs were used per sample. Library preparation and whole genome sequencing (150 bp paired-end reads to 10X coverage) were performed by Novogene Corporation.

7 Data and code availability

All code used in this study is available on Github at: https://github.com/mpiersonsmela/seqverify Raw sequencing reads for hiPSC lines derived from PGP1 are available through the NCBI Sequence Read Archive: PRJNA1019637. In order to respect donor privacy, sequencing data from other hiPSC lines are not publicly available. Sequences of plasmids used for generating knock-ins are provided in the Supplementary Information.

8 Acknowledgements

Funding for this project was provided by an NICHD F31 fellowship to Merrick Pierson Smela (F31HD108898-01A1). Portions of this research were conducted on the O2 High Performance Compute Cluster, supported by the Research Computing Group, at Harvard Medical School. We thank Dr. Chun-Ting Wu for assistance with validating *Mycoplasma* detection.

9 Figures

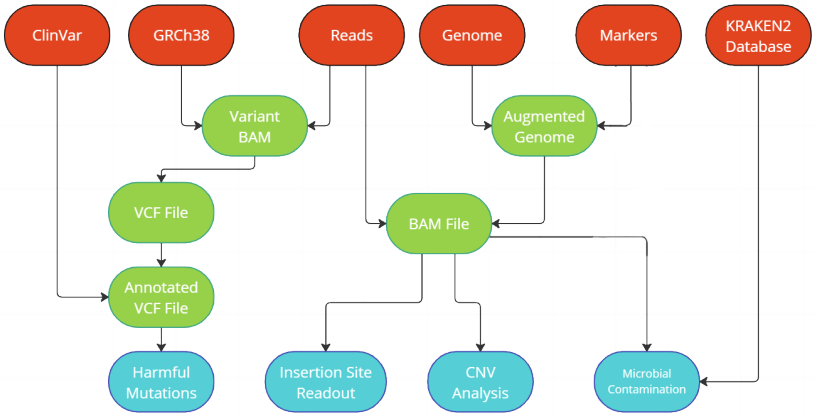


Figure 1: A flowchart showing the steps involved in the SeqVerify pipeline. In red, the possible types of inputs that the pipeline can take for the types of analysis that can be performed. In light green, the major intermediate files produced during the running of the pipeline, output in the non-temporary output folder. In light blue, the four major outputs of the pipeline: the insertion site readout and CNV analysis graphs and binaries, as well as the microbial contamination and harmful mutation readouts if the KRAKEN and SNV Analysis portions of the pipeline are enabled, respectively.

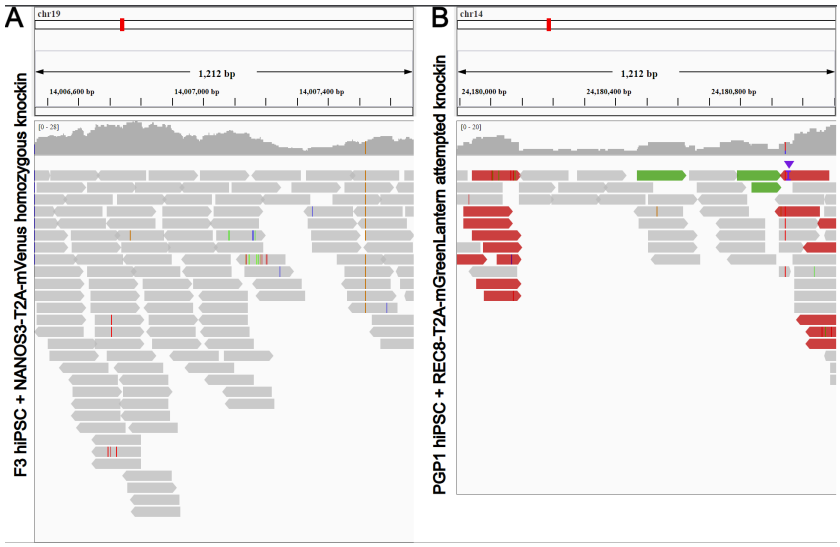
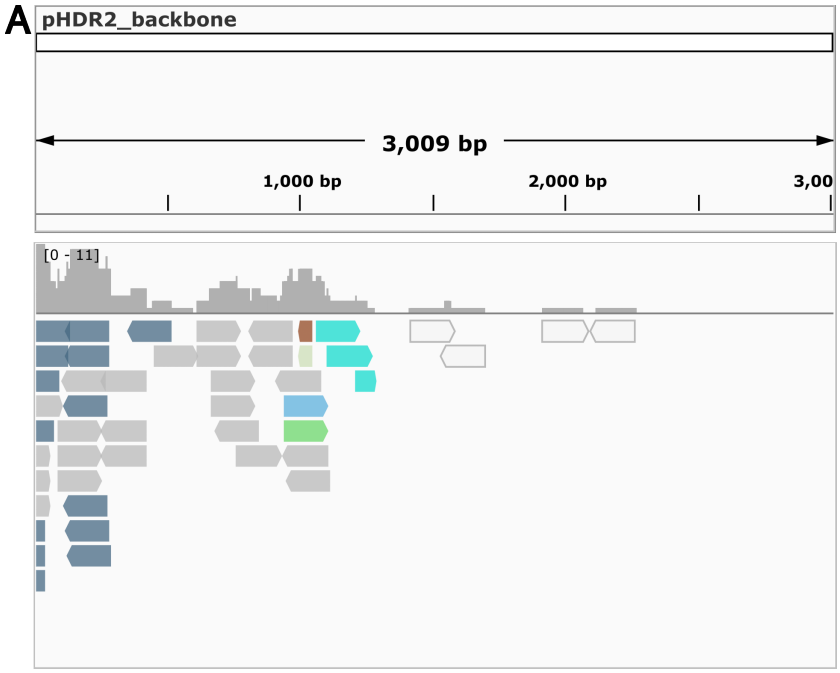


Figure 2: On-target edit validation using SeqVerify. hiPSCs with fluorescent reporter knockins (panel A: NANOS3-T2A-mVenus; panel B: REC8-T2A-mGreenLantern) were analyzed by WGS and the SeqVerify pipeline. IGV plots were automatically generated showing the insertion and 200bp of flanking wild-type sequence on either side. The NANOS3 reporter line is a homozygous knockin. A SNP present in the starting cells is visible (brown line). By contrast, the REC8 reporter line is heterozygous; read pairs highlighted in red by IGV denote a "deletion" of the T2A-mGreenLantern sequence on one of the alleles. Additionally, read pairs highlighed in green by IGV map both to the inserted sequence, and to bacterial plasmid DNA. This clearly shows an undesired editing outcome.



B

```
Insertion Sites Found:
pHDR2_backbone:
        chr5:
                56647502: 14 matched
                56647658 (Chimeric/Split Read): 5 matched
        chr7:
                88953762: 1 matched
        chr11:
                24953848: 1 matched
        chr18:
                53486674: 1 matched
        chr9:
                7676758: 1 matched
```

Figure 3: Detection of an undesired plasmid insertion in hiPSCs edited with a *DDX*4 knockin. (A) The automatically generated IGV plot shows reads aligning to the plasmid backbone. Reads highlighted in blue have their mates mapped to the human genome. (B) Automatic detection of the insertion site (at the *DDX*4 locus on chr5).

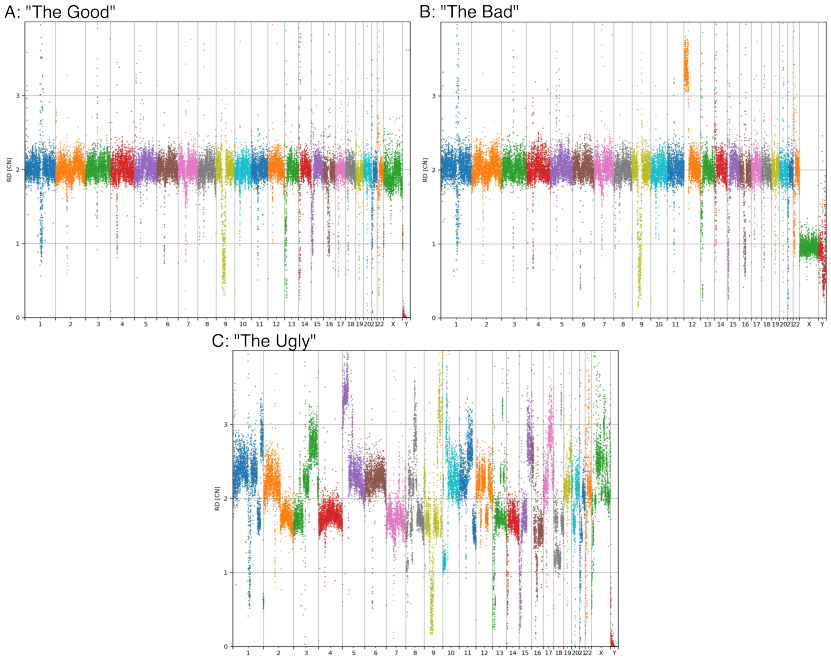


Figure 4: Aneuploidy detection using CNVpytor. WGS data from two hiPSC lines (panel A: female 46XX; panel B: male 46XY dup(12p)) and HEK293 cells (panel C) were analyzed using SeqVerify. CNVpytor plots are shown. Panel A shows a normal female karyotype, although some read depth variations are present in repetitive DNA near centromeres and telomeres. Panel B shows an aneuploid male karyotype, with a duplication of 12p that has been previously reported as a common abnormality in hPSCs (26). Panel C shows the massive aneuploidy of HEK293 cells.

References

[1] S. Assou, N. Girault, M. Plinet, J. Bouckenheimer, C. Sansac, M. Combe, J. Mianné, C. Bourguignon, M. Fieldes, E. Ahmed, T. Commes, A. Boureux, J.-M. Lemaître, and J. D. Vos, "Recurrent genetic abnormalities in human pluripotent stem cells: Definition and routine detection in culture supernatant by targeted droplet digital PCR," *Stem Cell Reports*, vol. 14, pp. 1–8, Jan. 2020.

[2] S. M. Taapken, B. S. Nisler, M. A. Newton, T. L. Sampsell-Barron, K. A. Leonhard, E. M. McIntire, and K. D. Montgomery, "Karyotypic abnormalities in human induced pluripotent stem cells and embryonic stem cells," *Nature Biotechnology*, vol. 29, pp. 313–314, Apr. 2011.

[3] C. Markouli, E. C. D. Deckersberg, M. Regin, H. Nguyen, F. Zambelli, A. Keller, D. Dziedzicka, J. D. Kock, L. Tilleman, F. V. Nieuwerburgh, L. Franceschini, K. Sermon, M. Geens, and C. Spits, "Gain of 20q11.21 in human pluripotent stem cells impairs TGF--dependent neuroectodermal commitment," *Stem Cell Reports*, vol. 13, pp. 163–176, July 2019.

[4] H. Nguyen, M. Geens, A. Mertzanidou, K. Jacobs, C. Heirman, K. Breckpot, and C. Spits, "Gain of 20q11.21 in human embryonic stem cells improves cell survival by increased expression of bcl-xL," *MHR: Basic science of reproductive medicine*, vol. 20, pp. 168–177, Nov. 2013.

[5] F. T. Merkle, S. Ghosh, N. Kamitaki, J. Mitchell, Y. Avior, C. Mello, S. Kashin, S. Mekhoubad, D. Ilic, M. Charlton, G. Saphier, R. E. Handsaker, G. Genovese, S. Bar, N. Benvenisty, S. A. McCarroll, and K. Eggan, "Human pluripotent stem cells recurrently acquire and expand dominant negative p53 mutations," *Nature*, vol. 545, pp. 229–233, Apr. 2017.

[6] F. J. Rouhani, X. Zou, P. Danecek, C. Badja, T. D. Amarante, G. Koh, Q. Wu, Y. Memari, R. Durbin, I. Martincorena, A. R. Bassett, D. Gaffney, and S. Nik-Zainal, "Substantial somatic genomic variation and selection for BCOR mutations in human induced pluripotent stem cells," *Nature Genetics*, vol. 54, pp. 1406–1416, Aug. 2022.

[7] O. Thompson, F. von Meyenn, Z. Hewitt, J. Alexander, A. Wood, R. Weightman, S. Gregory, F. Krueger, S. Andrews, I. Barbaric, P. J. Gokhale, H. D. Moore, W. Reik, M. Milo, S. Nik-Zainal, K. Yusa, and P. W. Andrews, "Low rates of mutation in clinical grade human pluripotent stem cells under different culture conditions," *Nature Communications*, vol. 11, Mar. 2020.

[8] A. O. Olarerin-George and J. B. Hogenesch, "Assessing the prevalence of mycoplasma contamination in cell culture via a survey of NCBI's RNA-seq archive," *Nucleic Acids Research*, vol. 43, pp. 2535–2542, Feb. 2015.

[9] S. Bar, M. Schachter, T. Eldar-Geva, and N. Benvenisty, "Large-scale analysis of loss of imprinting in human pluripotent stem cells," *Cell Reports*, vol. 19, pp. 957–968, May 2017.

[10] M. Cloutier, S. Kumar, E. Buttigieg, L. Keller, B. Lee, A. Williams, S. Mojica-Perez, I. Erliandri, A. M. D. Rocha, K. Cadigan, G. D. Smith, and S. Kalantry, "Preventing erosion of x-chromosome inactivation in human embryonic stem cells," *Nature Communications*, vol. 13, May 2022.

[11] D. Simkin, V. Papakis, B. I. Bustos, C. M. Ambrosi, S. J. Ryan, V. Baru, L. A. Williams, G. T. Dempsey, O. B. McManus, J. E. Landers, S. J. Lubbe, A. L. George, and E. Kiskinis, "Homozygous might be hemizygous: CRISPR/cas9 editing in iPSCs results in detrimental on-target defects that escape standard quality controls," *Stem Cell Reports*, vol. 17, pp. 993–1008, Apr. 2022.

[12] A. Veres, B. S. Gosis, Q. Ding, R. Collins, A. Ragavendran, H. Brand, S. Erdin, C. A. Cowan, M. E. Talkowski, and K. Musunuru, "Low incidence of off-target mutations in individual CRISPR-cas9 and TALEN targeted human stem cell clones detected by whole-genome sequencing," *Cell Stem Cell*, vol. 15, pp. 27–30, July 2014.

[13] E. Zuo, Y. Sun, W. Wei, T. Yuan, W. Ying, H. Sun, L. Yuan, L. M. Steinmetz, Y. Li, and H. Yang, "Cytosine base editor generates substantial off-target single-nucleotide variants in mouse embryos," *Science*, vol. 364, pp. 289–292, Apr. 2019.

[14] E. McGrath, H. Shin, L. Zhang, J.-N. Phue, W. W. Wu, R.-F. Shen, Y.-Y. Jang, J. Revollo, and Z. Ye, "Targeting specificity of APOBEC-based cytosine base editor in human iPSCs determined by whole genome sequencing," *Nature Communications*, vol. 10, Nov. 2019.

[15] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *Bioinformatics*, vol. 25, pp. 1754–1760, May 2009.

[16] H. Thorvaldsdottir, J. T. Robinson, and J. P. Mesirov, "Integrative genomics viewer (IGV): high-performance genomics data visualization and exploration," *Briefings in Bioinformatics*, vol. 14, pp. 178–192, Apr. 2012.

[17] M. Suvakov, A. Panda, C. Diesh, I. Holmes, and A. Abyzov, "CNVpytor: a tool for copy number variation detection and analysis from read depth and allele imbalance in whole-genome sequencing," *GigaScience*, vol. 10, Nov. 2021.

[18] D. E. Wood, J. Lu, and B. Langmead, "Improved metagenomic analysis with kraken 2," *Genome Biology*, vol. 20, Nov. 2019.

[19] J. Lu, F. P. Breitwieser, P. Thielen, and S. L. Salzberg, "Bracken: estimating species abundance in metagenomics data," *PeerJ Computer Science*, vol. 3, p. e104, Jan. 2017.

[20] H. Li, "A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data," *Bioinformatics*, vol. 27, pp. 2987–2993, Sept. 2011.

[21] P. Cingolani, A. Platts, L. L. Wang, M. Coon, T. Nguyen, L. Wang, S. J. Land, X. Lu, and D. M. Ruden, "A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff," *Fly*, vol. 6, pp. 80–92, Apr. 2012.

[22] P. Cingolani, V. Patel, M. Coon, T. Nguyen, S. Land, D. Ruden, and X. Lu, "Using drosophila melanogaster as a model for genotoxic chemical mutational studies with a new program, snpsift," *Frontiers in Genetics*, vol. 3, 2012.

[23] E. Sherman, C. Nobles, C. C. Berry, E. Six, Y. Wu, A. Dryga, N. Malani, F. Male, S. Reddy, A. Bailey, K. Bittinger, J. K. Everett, L. Caccavelli, M. J. Drake, P. Bates, S. Hacein-Bey-Abina, M. Cavazzana, and F. D. Bushman, "INSPIIRED: A pipeline for quantitative analysis of sites of new DNA integration in cellular genomes," *Molecular Therapy - Methods &amp Clinical Development*, vol. 4, pp. 39–49, Mar. 2017.

[24] D. Warburton, "De novo balanced chromosome rearrangements and extra marker chromosomes identified at prenatal diagnosis: clinical significance and distribution of breakpoints.," *American journal of human genetics*, vol. 49, no. 5, p. 995, 1991.

[25] M. D. P. Smela, C. C. Kramme, P. R. Fortuna, J. L. Adams, R. Su, E. Dong, M. Kobayashi, G. Brixi, V. S. Kavirayuni, E. Tysinger, R. E. Kohman, T. Shioda, P. Chatterjee, and G. M. Church, "Directed differentiation of human iPSCs to functional ovarian granulosa-like cells via transcription factor overexpression," *eLife*, vol. 12, Feb. 2023.

[26] S. E. Peterson and J. F. Loring, "Genomic instability in pluripotent stem cells: Implications for clinical applications," *Journal of Biological Chemistry*, vol. 289, pp. 4578–4584, Feb. 2014.