**DETAILED METHODS**

***Pipeline Overview***

The MutantHuntWGS pipeline integrates a series of open-source bioinformatics tools and Unix commands that accept raw sequencing reads (compressed FASTQ format or .fastq.gz) and a text file containing ploidy information as input, and produces a list of sequence variants as output. The user must provide input data from at least two strains: a control strain and one or more experimental strains. The pipeline uses (1) Bowtie2 to align the reads in each input sample to the reference genome (Langmead and Salzberg 2012), (2) SAMtools to process the data and calculate genotype likelihoods (Li *et al.* 2009), (3) BCFtools to call variants (Li *et al.* 2009), (4) VCFtools (Danecek *et al.* 2011) and custom shell commands to compare variants found in experimental and control strains, and (5) SnpEff (Cingolani *et al.* 2012) and SIFT (Vaser *et al.* 2016) to assess where variants are found in relation to annotated genes and the potential impact on the expression and function of the affected gene products (Figure 1).

***Sequence Alignment***

MutantHuntWGS uses Bowtie2 version 2.2.9 (Langmead and Salzberg 2012) to first align the raw reads present in the input FASTQ files to the *S. cerevisiae* genome (S288C version = R64-2-1) (Cherry *et al.* 2012; Engel *et al.* 2014). As the default, we set Bowtie2 to search for a maximum of two distinct alignments per read (-k 2 option), which reduces alignment time and multiple mapping. The pipeline only retains sequencing reads that align (--no-unal option) in the SAM (Sequence Alignment/Map Format) output to help reduce file size. MutantHuntWGS uses SAMtools version 1.3.1 to convert the aligned-read output from Bowtie2 (SAM format) into the BAM (Binary Alignment/Map) format (view -bS options) (Li *et al.* 2009). SAMtools then sorts (sort option) and indexes (index option) the BAM file to prepare the data for variant calling. Users can view the sorted and indexed BAM files in a genome browser such as IGV (Integrative Genome Viewer) to examine the aligned reads (Thorvaldsdottir *et al.* 2013).

***Variant Calling***

Based on the aligned reads, SAMtools outputs genotype likelihoods as BCF (Binary Call Format) files (mpileup -g -f options) using the BAM file as input (Li *et al.* 2009). BCFtools version 1.3.1 then uses the genotype likelihoods recorded in the BCF file to call single nucleotide polymorphisms (SNPs), as well as insertions and deletions (INDELs) (-c -v --samples-file –ploidy-file options) (Li 2011). This variant information is saved in the Variant Call Format (VCF), the format used by the 1000 Genomes Project (Danecek *et al.* 2011). IGV can again be used to view the VCF files that are output from MutantHuntWGS (Thorvaldsdottir *et al.* 2013). At the variant calling step, BCFtools also considers a user-specified input ploidy file to account for genome copy number.

***Identifying Candidate Variants***

VCFtools version 0.1.14 compares VCF files (--diff-site option) from the control and experimental samples (Danecek *et al.* 2011). To retain the variants that are found only in the experimental dataset, MutantHuntWGS uses the Unix awk command (Aho *et al.* 1979) to remove variants from the VCFtools output that have VCF scores lower than a user-defined variant-quality-score cutoff. For each experimental dataset, it then uses the Unix grep, head, and cat commands to construct new VCF files that contain only the variants specific to the experimental strain. These VCF files can also be viewed in IGV.

***Variant Effect Prediction***

SnpEff version 4.3p (Cingolani *et al.* 2012) and SIFT4G (i.e., SIFT) (Vaser *et al.* 2016) are useful programs for (1) determining whether sequence variants are located in or near an annotated coding region and (2) predicting the effect the variant might have on gene expression or function of the protein product. SnpEff determines the locations of sequence variants relative to protein-coding genes and the severity of each variant based on how likely it is to disrupt gene expression or function (Cingolani *et al.* 2012). SnpEff also annotates variants in 5’ and 3’ UTRs as well as promoter regions. This information is vital if the causal mutation disrupts a ncRNA or DNA element rather than altering a protein-coding sequence. SIFT uses the EF4.74 library for *S. cerevisiae* toscore variants found in protein-coding genes in order to predict the impact of the resulting amino acid changes (Vaser *et al.* 2016). MutantHuntWGS saves all SnpEff and SIFT output files so the user can further filter the results to reduce the number of candidate sequence variants identified.

***Analysis of previously published data***

To demonstrate utility, we used MutantHuntWGS to analyze published datasets from paired-end sequencing experiments with DNA prepared from bulk segregants or lab-evolved strains (Baryshnikova *et al.* 2013; Goldgof *et al.* 2016; Ottilie *et al.* 2017). These data were downloaded from the sequence read archive (SRA) database (<https://www.ncbi.nlm.nih.gov/sra>; project accessions: SRP003355, SRP074482, SRP074623) and decompressed using the SRA toolkit (<https://github.com/ncbi/sra-tools/wiki>). MutantHuntWGS was run from within the Docker container, and each published mutant (experimental) file was compared to its respective published control. When processing data from bulk segregant analysis, we reduced the number of candidate variants by additionally using more stringent cutoffs: variant quality score > 130, SnpEff impact score > Moderate, and SIFT score < 0.05 (deleterious).