# SIFT missense predictions for genomes

Robert Vaser[1,4], Swarnaseetha Adusumalli[2,4], Sim Ngak Leng[2], Mile Sikic[1,3] & Pauline C Ng[2]

[1]Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia. [2]Computational and Systems Biology Group, Genome Institute of Singapore, Agency for Science, Technology and Research (A*STAR), Singapore, Singapore. [3]Bioinformatics Institute, Agency for Science, Technology and Research, Singapore, Singapore. [4]These authors contributed equally to this work. Correspondence should be addressed to P.C.N. (ngpc4@gis.a-star.edu.sg).

Published online 3 December 2015; doi:10.1038/nprot.2015.123
This protocol is an update to *Nat. Protoc.* **4**, 1073–1081 (2009); doi:10.1038/nprot.2009.86

**The SIFT (sorting intolerant from tolerant) algorithm helps bridge the gap between mutations and phenotypic variations by predicting whether an amino acid substitution is deleterious. SIFT has been used in disease, mutation and genetic studies, and a protocol for its use has been previously published with *Nature Protocols*. This updated protocol describes SIFT 4G (SIFT for genomes), which is a faster version of SIFT that enables practical computations on reference genomes. Users can get predictions for single-nucleotide variants from their organism of interest using the SIFT 4G annotator with SIFT 4G's precomputed databases. The scope of genomic predictions is expanded, with predictions available for more than 200 organisms. Users can also run the SIFT 4G algorithm themselves. SIFT predictions can be retrieved for 6.7 million variants in 4 min once the database has been downloaded. If precomputed predictions are not available, the SIFT 4G algorithm can compute predictions at a rate of 2.6 s per protein sequence. SIFT 4G is available from http://sift-dna.org/sift4g.**



## INTRODUCTION

### Background and application of the protocol

Genome sequencing has brought about significant advances in medical and agricultural fields, as well as in basic research[1,2]. A genetic understanding of phenotypes can entail sequencing of many different breeds or strains of the same organism[3,4]. For example, over 3,000 rice genomes were sequenced in order to interpret the genetic diversity that underlies traits such as cold tolerance and grain quality[5]. These types of surveys can have a huge impact, leading some to estimate the sequencing market in agricultural and other industrial applications to be valued at more than $5 billion[6]. Basic research also benefits from the rise of genome sequencing; for example, the sequences of multiple *Drosophila* genomes have been used to better understand the fundamentals of evolutionary processes[7]. Finally, our knowledge of human disease has been improved by the availability of diverse sequence data. Studies of variations among breeds of dogs and cats have yielded the identification of genes whose orthologs are implicated in human disease[8,9]. However, after genomes are sequenced, combing through potentially tens of thousands of missense variations per breed or strain can be challenging. Thus, a tool to assist with the prioritization of functional variants is desirable.

SIFT is an algorithm that predicts whether an amino acid substitution is deleterious to protein function, and it is often used to prioritize nonsynonymous or missense variants[10–13]. A protein may be able to tolerate an amino acid change and still function normally, or it may be intolerant to the amino acid change. SIFT classifies an amino acid change as tolerated or deleterious to protein function. SIFT takes into account protein conservation with homologous sequences and the severity of the amino acid change. It has already been used in numerous disease, mutation and genetic studies[14–16]. We originally published a protocol in 2009 describing the use of SIFT, which is still functional[10]. The updated protocol, discussed here, expands on existing functionality. This protocol describes how to easily obtain SIFT predictions for nonhuman organisms, and it describes how to quickly obtain predictions on a large number of protein sequences using a single graphical processing unit (GPU). The SIFT 4G algorithm is a faster version of SIFT that enables us to provide SIFT predictions for large numbers of organisms. Comparisons of variation within wheat, melon, sunflower, finches and chicken breeds have used SIFT predictions[17–21]. SIFT 4G has recently been used to provide SIFT scores in *Sus scrofa*[22].

The original SIFT implementation took ~4 min to run on a CPU for a single protein sequence. At this speed, we had to precompute SIFT scores for every human protein and store them in a database in order to provide SIFT predictions for human genome variants[13]. This operation took many hours on a large computer cluster. The computational demands of the original SIFT algorithm thus made it challenging to scale SIFT and apply it to other organisms. The SIFT 4G algorithm is a GPU-optimized version of SIFT that allows us to obtain SIFT predictions quickly and to construct prediction databases for a large number of organisms. Although GPUs offer massive parallelism, mapping computational biology problems onto their graphics-optimized architectures often requires a trade-off in accuracy. Our contribution is migrating the original SIFT algorithm to run on a GPU with little compromise in the prediction accuracy. As a result, SIFT 4G focuses only on improving run-time performance; no new features are introduced, nor is predictive performance improved. The enhanced run-time performance enables us to provide predictions for genomes other than the human genome, thereby serving a larger research community. A side-by-side comparison of the SIFT and SIFT 4G algorithms is depicted in **Figure 1a**. Similar performances on four different data sets are observed in **Figure 1b** (see also **Supplementary Figs. 1** and **2**).

By using the SIFT 4G algorithm, we precomputed SIFT predictions for over 200 genomes. Precomputed prediction databases for organisms can be downloaded from the SIFT website. We describe two procedures for using SIFT 4G. With the first procedure, researchers with sequenced genomes of a model organism can use the SIFT 4G annotator in conjunction with the SIFT prediction databases to annotate their genomic variant files. In the second procedure, we provide instructions on how to compute predictions using the SIFT 4G algorithm on a GPU for users who want to run their protein sequences in a batch format.

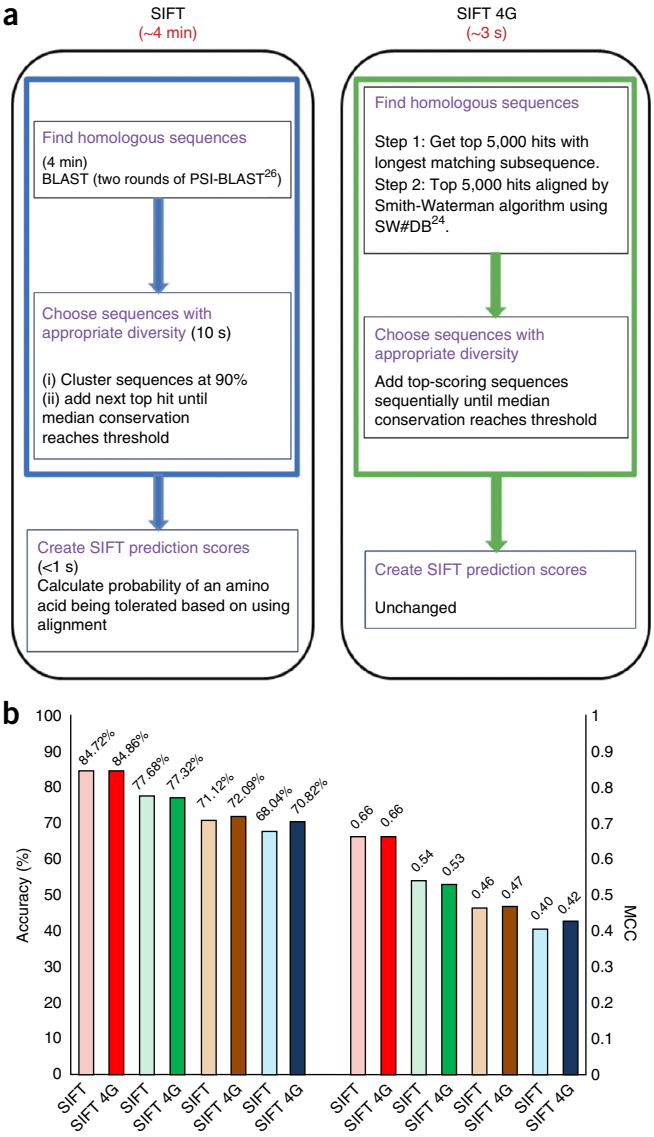


**Figure 1** | Comparison of the SIFT and SIFT 4G algorithms. (**a**) The steps of the SIFT and SIFT 4G algorithms are shown on the left and right, respectively. The principle of each step has been preserved, but the first two steps have been optimized for speed in the SIFT 4G algorithm. See (refs. 23–26). (**b**) Matthew's correlation coefficient (MCC) of SIFT (light-colored bars) and SIFT 4G (dark-colored bars) on four data sets (HumDiv (human), HumVar (human), LacI (E. coli) and lysozyme (bacteriophage) depicted in red, green, brown and blue, respectively). Accuracy is the percentage of correct predictions. MCC is a balanced measure of the true and false positives and negatives. Panel **b** is reproduced under a Creative Commons license from http://sift-dna.org/sift4g/AboutSIFT4G.html.

### SIFT and SIFT 4G methodology

SIFT is a multistep algorithm that uses sequence conservation and amino acid properties to predict whether an amino acid substitution is deleterious. Procedural details of the CPU version of SIFT have been described previously[11], and they can be seen in **Figure 1a**. SIFT 4G's implementation of SIFT's steps is described below:

(i) *Search for similar sequences against a large protein database.* SIFT 4G's heuristic search algorithm uses a seed detection algorithm coupled with the Smith-Waterman algorithm to find similar sequences. The first phase of seed detection uses the longest increasing subsequence algorithm to approximate similarities between the query and database sequences[23]. Both the query and database sequences are split into overlapping *k*-mers by a sliding window of length *k* (where *k* equals 5). Each sequence is represented by a list of *k*-mers and the *k*-mer's position in the sequence. Two pairs match if their *k*-mers are equal. To calculate a similarity score between two sequences, we find all matching pairs and retrieve their corresponding positions from the representative list. The length of the longest increasing subsequence of positions (in the newly created list) is used as the similarity score. Once all the pairs are scored, the database sequences are sorted in decreasing order of their similarity score, and the top-scoring 5,000 hits are passed on to the second phase (the Smith-Waterman algorithm). SIFT 4G uses SW#db, which is a GPU-based implementation of the Smith-Waterman algorithm[24]. SW#db reconstructs the alignment between the query and the top 5,000 hits, of which 400 sequences with the most significant *P* values are kept. At this point, the query protein sequence is aligned to its homologous sequences.

(ii) *Choose closely related sequences.* In SIFT's second step, closely related sequences are chosen to achieve a certain level of sequence diversity. Previously, SIFT clustered sequences at 90% identity into consensus sequences, and then selected the most closely related consensus sequence iteratively using Position-Specific Iterated Basic Local Alignment Search Tool (PSI-BLAST) until the chosen consensus sequences reached a median sequence conservation threshold[11]. In SIFT 4G, this step is simplified by removing the clustering step. The 400 sequences from the first step are added iteratively. The most significant hits are added first until the median sequence conservation is reached (the default is 2.75, as set by Ng and Henikoff[11]).

(iii) *Calculate probabilities for amino acid substitutions.* SIFT 4G's final step of calculating scaled probabilities is identical to that of SIFT[25]. SIFT 4G has a multithreaded implementation.

Code for SIFT and SIFT 4G can be found at http://sift-dna.org and http://sift-dna.org/sift4g, respectively. For SIFT 4G, we sought to improve run-time performance by replacing PSI-BLAST in the first step of SIFT with a faster search algorithm, and also by optimizing the second step (**Fig. 1a**). PSI-BLAST is part of the BLAST family of algorithms and it is a heuristic algorithm, so optimal answers are not guaranteed[26]. Similarly, SIFT 4G uses a heuristic method to search for homologs (by combining seed detection followed by GPU-based Smith-Waterman alignment). As different heuristic algorithms are used in the first step, the results from SIFT 4G will differ from SIFT. SIFT 4G's heuristic search algorithm achieves drastic speedup compared with PSI-BLAST at the cost of slightly less sensitivity to distant homologous sequences (**Fig. 1a**).

### Performance of SIFT and SIFT 4G

SIFT 4G and the CPU version of SIFT (v5.2.2) were assessed using UniRef90 (ref. 27; 4 August 2011) as the protein database. For both algorithms, the sequence median information was set at 2.75, and sequences were removed at 100% identity.

We used four different data sets to assess performance: HumDiv, HumVar, LacI and lysozyme[28–30]. Each data set contained (i) amino substitutions shown to affect protein function, either by *in vitro* assays (LacI and lysozyme data sets) or known to cause human disease (HumDiv and HumVar data sets), and (ii) amino acid substitutions that are functionally neutral, either by *in vitro* assays (LacI and lysozyme data sets) or by variation not known to cause disease (HumDiv and HumVar data sets).

True positives (TP) are the protein-affecting mutations correctly predicted as deleterious; false negatives (FN) are the protein-affecting mutations incorrectly predicted to be tolerated. True negatives (TN) are neutral variations correctly predicted as tolerated, and false positives (FP) are neutral variations incorrectly predicted to affect protein function. The following performance metrics were calculated:

- Sensitivity = TP/(TP + FN)
- Specificity = TN/(TN + FP)
- Accuracy = (TP + TN)/(TP + TN + FP + FN)
- Matthew's correlation coefficient (MCC) = ((TP × TN) – (FP × FN))/SQRT ((TP + FP) (TP + FN) (TN + FP) (TN + FN))

Accuracy is the percentage of correct predictions for both TP and TN. Sensitivity measures the ability to correctly identify

## Box 1 | Running the SIFT 4G algorithm on protein sequences ● TIMING ~6 h for setup, and 2.6 s per protein sequence

1. For users who want to obtain predictions for their protein sequences, download the SIFT 4G algorithm (filename SIFT4G_<version>.tar.gz) from http://sift-dna.org/sift4g/SIFT4G_codes.html.
2. Follow the instructions located at:

<SIFT_4G folder>/INSTALL_instructions/SIFT4G_Installation_Instructions.pdf to install the SIFT 4G algorithm.

3. Get predictions for amino acid substitutions in proteins. The input into SIFT 4G is a directory containing protein sequences and a directory containing corresponding amino acid substitution files, which are customized for SIFT. Each protein sequence file should have either the suffix '.fa' or '.fasta' and the protein sequence should be in FASTA format. For example, TP53_HUMAN.fa would contain the protein sequence for TP53 in FASTA format:

```
>TP53_HUMAN
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVL...
```

The substitution file shares the same name as the protein sequence so that SIFT 4G knows where to find the list of substitutions to predict on. In the above example, the protein sequence is named 'TP53_HUMAN' in the description line of the FASTA file, so the substitution file should have the filename 'TP53_HUMAN.subst'. The substitution file contains a list of amino acid substitutions that will be predicted on. Each substitution is represented by the original amino acid followed by its position in the protein, and the new amino acid to be predicted on. For example, the TP53_HUMAN.subst file could contain two amino acid substitutions:

```
D7H
P359D
```

Where D7H represents a substitution from aspartic acid (D) to histidine (H) at position 7 in TP53_HUMAN, and P359D represents a substitution from proline (P) to aspartic acid (D) at position 359 in TP53_HUMAN.
4. Run the following command line from the SIFT 4G directory to get predictions for amino acid substitutions:

```
python siftsharp/siftsharp.py --database <protein database to be searched> --queries
<query directory containing FASTA sequences> --substdir <directory containing
substitution files> --out-dir <output directory for predictions> --align-dir
<output directory for alignments>
```

Protein sequence alignments will be generated in the alignment directory specified by align-dir, and the SIFT predictions will be generated in the directory specified by out-dir. The SIFT 4G package comes with a protein database and test data sets consisting of FASTA-formatted protein sequences and amino acid substitution files. The following command line can be run as an example:

```
python siftsharp/siftsharp.py --database database/uniref90.fasta –queries
queries_and_subst/humdiv-deleterious --subst-dir queries_and_subst/humdiv-deleterious
--out-dir out --align-dir out
```

**! CAUTION** Ensure that there is adequate disk space on your system. The first step of the SIFT 4G algorithm is to search for homologous sequences against a protein database (**Fig. 1a**). To do this quickly, the SIFT 4G algorithm creates large index files of the protein database that can take eight times as much space as the protein database. For example, if the protein database is 1 GB, the index files created by the SIFT 4G algorithm can be as large as 8 GB.

**! CAUTION** Sequences in the <query directory> that were processed by SIFT will be moved to a new folder called <query directory>_processed. This was implemented because occasionally the computer would crash after predictions had been computed for (tens of) thousands of proteins. To avoid re-processing proteins, sequences are moved to another folder called <query directory>_processed after completion. After successful completion of the job, move the protein sequences back to its original directory with a simple command: mv <query directory>_processed/*.fa* <query directory>.

the deleterious mutations, whereas specificity measures the ability to correctly identify neutral variation.

As can be seen in **Figure 1b**, the accuracies between the two algorithms are similar. SIFT 4G's accuracies differ only +0.1%, −0.4%, +1% and +2.8% from SIFT 5.2.2 on the HumDiv, HumVar, LacI and lysozyme data sets, respectively (**Fig. 1b**). When MCC is used to assess performance, the differences are negligible, ranging from −0.01 to 0.02 (**Fig. 1b**). Sensitivity and specificity are shown in **Supplementary Figure 1**; receiver operating characteristic (ROC) curves are shown in **Supplementary Figure 2**. Coverage, the number of substitutions predicted on, exceeds 99.5% for both algorithms across all data sets.

### Comparison between SIFT and SIFT 4G protocols

A protocol for SIFT was published in 2009 (ref. 10). The 2009 protocol described how to use the SIFT web server (http://www.sift-dna.org) for various inputs including protein sequences, protein alignments and dbSNP IDs. The SIFT website also provides predictions for human single-nucleotide variants and indels[13].

This current protocol describes how to obtain SIFT predictions for missense variation in organisms with sequenced genomes (in addition to humans). Predictions for all possible single-nucleotide changes in an organism's genome are computed using SIFT 4G, and they are stored in a database that is made publicly available. This protocol describes how to (i) use the SIFT 4G annotator to annotate a variant list from an organism of interest with SIFT predictions and (ii) run the SIFT 4G algorithm directly.

### Limitations

The SIFT 4G annotator loads SIFT 4G prediction databases, which are based on Ensembl gene annotation. We do not provide predictions for other gene annotations such as the University of California, Santa Cruz (UCSC) Genome Browser[31] or Phytozome[32]. If a user wants to get predictions for customized gene annotations, the SIFT 4G algorithm can be downloaded from the website (**Box 1**).

The SIFT 4G algorithm is optimized for NVIDIA GPU cards. However, it can be run without GPU support as long as the NVIDIA compiler (nvcc) is installed, albeit at a substantial performance penalty.

## MATERIALS

### EQUIPMENT

- Computer with Internet connection (see Equipment Setup)
- Data files (see Equipment Setup)

### EQUIPMENT SETUP

**System requirements**

- SIFT 4G annotator: The SIFT 4G annotator requires a computer with Java JRE (Java Runtime Environment) installed (version 1.6 or higher; http://www.java.com/en/) and enough disk space to store the database (which can range from 120 MB for *Escherichia coli* to 3.9 GB for human). The SIFT 4G annotator is platform-independent, and it can run on Windows, Linux and Mac.
- SIFT 4G algorithm: The SIFT 4G algorithm requires any Linux distribution (we have used Ubuntu 12.04) with the following compilers: gcc (version 4 or higher; https://gcc.gnu.org/) and nvcc (version 2 or higher; https://developer.nvidia.com/cuda-downloads). For fast performance, the recommended configuration should include a NVIDIA graphics card (compute capability version 1.3 or higher) and a solid-state drive (SSD).

**Data files**

- SIFT 4G annotator accepts a list of genomic variants in variant call format (VCF), which is generated by most next-generation sequencing pipelines. To annotate the input variants, the SIFT 4G annotator uses the chromosome (CHROM), position (POS), reference (REF) and alternate (ALT) alleles from the input file and appends the output to the INFO column (the eighth column).
- If the user does not have a VCF file, the user can format their list of variants in a VCF-like file, which should have at least eight columns. A sample VCF is shown in **Supplementary Table 1**. **! CAUTION** If the input file does not have at least eight columns, it will not be annotated as the prediction is appended to the eighth column. If the user's input file contains the chromosome, position, reference and alternate alleles alone, the user can append dummy columns to ensure that the input file will have at least eight columns.

**Required inputs**

- The SIFT 4G algorithm requires three inputs. The first input is a directory of '.fasta' or '.fa' files where each file contains a protein sequence in FASTA format with the protein name in the description line. The algorithm also requires a companion input file containing a list of amino acid substitutions for each protein sequence, and it will compute predictions for these substitutions. The third required input is the protein sequence database to search homologous sequences—for example, the UniRef90 (ref. 27) or NCBI nonredundant[33] protein databases.

## PROCEDURE

### Obtaining SIFT predictions via the SIFT 4G annotator

**1|** The SIFT 4G annotator takes as input a VCF file and a SIFT 4G database, and it produces as output a VCF file and a Microsoft Excel (XLS) file, both with SIFT annotations. An overview of the SIFT 4G annotator is provided in **Figure 2**. Users can annotate their genomic variants using the graphical user interface (GUI, option A) or command line (option B). For users running the SIFT 4G algorithm on protein sequences in organisms that are not present within the database, please refer to **Box 1**.

**(A) Annotate variants with the SIFT 4G annotator via the GUI ● TIMING 45 min for first-time operation; 5 min for subsequent runs**

(i) *Download SIFT 4G annotator.* Download the SIFT 4G annotator, which is an executable Java Archive (.jar) file, from http://sift-dna.org/sift4g/AnnotateVariants.html.

(ii) *Start the GUI.* When using Windows or Mac operating system, double-click on the SIFT 4G annotator icon (.jar file). When using the Linux operating system, type the following command into the terminal:

```
>java -jar <Path to SIFT4G Annotator>
```

(iii) *Select the input data.* Select the genomic variants file in VCF format (**Fig. 3**).
**? TROUBLESHOOTING**

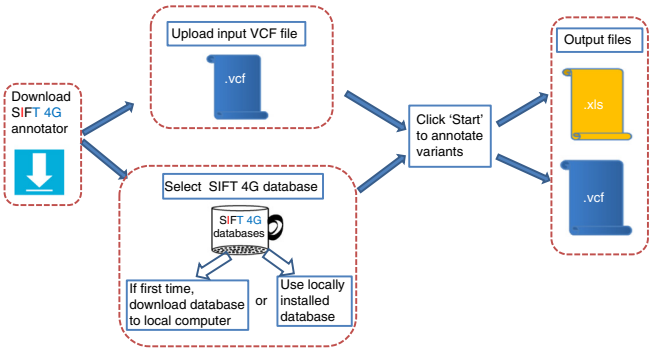


**Figure 2** | Workflow for the SIFT 4G annotator. After downloading the SIFT 4G annotator, the user selects a list of variants to be annotated and the appropriate SIFT 4G database. The SIFT 4G annotator will generate two output files with SIFT annotations (a VCF file and an XLS file).

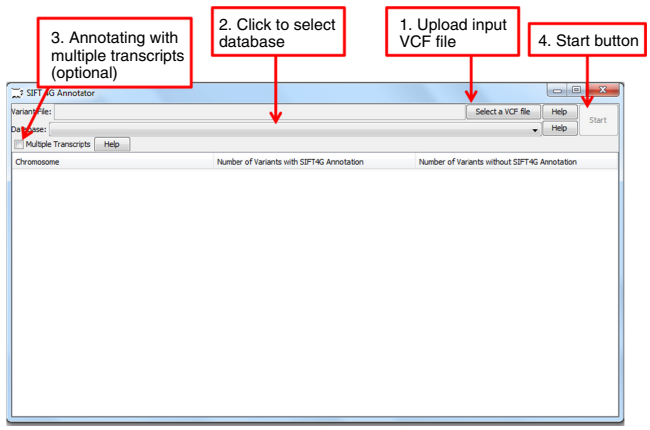


**Figure 3** | The SIFT 4G annotator graphical user interface. The user's steps are numbered. The user selects a VCF file (1), selects the database for the desired organism (2), decides on the option to annotate with multiple transcripts (3) and then clicks the start button (4).

(iv) *Load the database*. When annotating variants for a particular organism for the first time, the organism's database will need to be locally installed, and it can be subsequently selected for future annotations. If the SIFT 4G database for the desired organism has already been downloaded, it will be listed in the menu. Select the organism and proceed to Step 1A(v). Otherwise, to download the database, click on 'Select database to download' to display the list of available SIFT 4G databases. Available databases will have the scientific name of the organism, followed by the genome assembly and gene annotation versions. For example, 'Canis Familiaris (CanFam3.1.74)' refers to the dog (*Canis familiaris*) genome assembly 3.1, with Ensembl gene annotation from release 74. Double-click on the desired organism to download and extract the SIFT 4G database (**Fig. 4**). The database will be stored in the folder 'SIFT4G/Databases' located in the user's home directory. Return to the dropdown menu and select the organism after the database has been downloaded. For future use, the database will be automatically listed in the database dropdown menu.

**? TROUBLESHOOTING**

(v) *(Optional) Annotate with multiple transcripts*. Select the 'Multiple Transcripts' checkbox if the variants should be annotated with multiple transcripts. Users can ignore this step if they want to annotate the genomic variants with a single transcript (**Fig. 3**).

(vi) *Start variant annotation*. Click the 'Start' button at the top right corner of the interface to start variant annotation (**Fig. 3**). Progress for annotation will be displayed (**Fig. 5**). The variants will be annotated chromosome by chromosome. As each chromosome finishes, the 'Number of variants with SIFT 4G Annotations' and 'Number of variants without SIFT 4G Annotation' columns are updated.

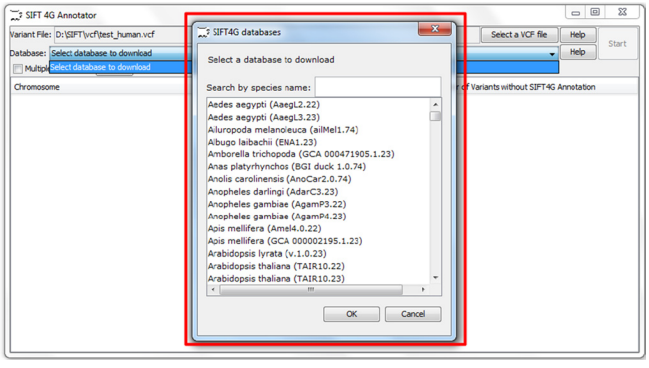


**Figure 4** | Select the database for the desired organism. The database needs to be downloaded locally on its first use. To view the available SIFT 4G databases, click 'Select database to download' in the 'Database' dropdown menu, and a list of organisms along with their assembly and gene annotation versions will appear. Users can find their organism by scrolling down the list or using the search box near the top of the list. Clicking on the organism name and then the button 'OK' will locally install the database for current and subsequent uses.

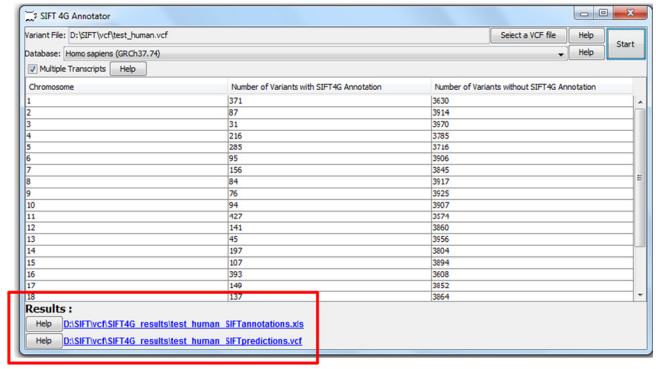


**Figure 5** | View of the SIFT 4G annotator after annotation has been completed. The number of variants with and without SIFT annotation is displayed for each chromosome. To view annotations, users can click on links to output files.

(vii) *View the result files.* Click the XLS and VCF files shown at the bottom of the interface (**Fig. 5**). The result files are stored in the same directory as the input file under the names '<*original filename*>_SIFTPredictions.vcf' and '<*original filename*>_SIFTAnnotationsOnly.xls'.
**? TROUBLESHOOTING**

**(B) Annotate variants with the SIFT 4G annotator via command line ● TIMING 45 min for first-time operation; 5 min for subsequent runs**

(i) *Download the desired database.* Go to the following link: http://sift-dna.org/sift4g/public/, where SIFT predictions for various organisms are available. Download the database (a ZIP file) for the desired organism using 'wget' or 'curl' commands. Within each organism's folder, there are different builds that correspond to the genome assembly and Ensembl version. For example, CanFam3.1.74 refers to the dog (*C. familiaris*) genome assembly 3.1, with Ensembl gene annotation from release 74.

(ii) *Extract the downloaded ZIP file.* The unzipped folder will have three files for each chromosome: a compressed chromosome file (.gz), a regions file (.regions) and a chromosome statistics file (.txt)

(iii) *Download the SIFT annotator.* Download the SIFT 4G annotator, an executable Java Archive (.jar) file, from http://sift-dna.org/sift4g/AnnotateVariants.html.

(iv) *Annotate variants.* To run the SIFT 4G annotator on Linux or Mac via command line, type the following command into the terminal:

```
>java -jar <Path to SIFT4G Annotator> -c -t -i <Path to input VCF file> -d <Path to SIFT4G database directory> -r <Path to results folder>
```

**! CAUTION** To run the SIFT 4G annotator via command line, '-c' is essential. (See **Table 1** for parameters.)
**? TROUBLESHOOTING**

**TABLE 1** | Command-line options to run SIFT 4G annotator.

| Option | Description |
|---|---|
| -c | To run on command line |
| -t | To extract annotations for multiple transcripts (optional) |
| -i | Path to the input variant file in VCF format |
| -d | Path to SIFT databases directory |
| -r | Path to the output results folder |

**? TROUBLESHOOTING**
Troubleshooting advice can be found in **Table 2**.

**TABLE 2** | Troubleshooting table.

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| 1A(iii) | The input file is not in the correct VCF format | A VCF file must have eight columns and be sorted by chromosome and position. The SIFT 4G annotator appends predictions to the eighth column | The user should append dummy columns to ensure that the input file has at least eight columns, and sort the chromosome and genomic coordinate columns in ascending order |
| 1A(iv) | The downloaded database is not seen in the database dropdown menu | The database was not downloaded completely | Download the database again |
| 1A(vii), B(iv) | Zeroes are observed in both columns 'Number of variants with SIFT 4G Annotations' and 'Number of variants without SIFT 4G Annotation'. In addition, the input file is not annotated | There is a chromosome naming discrepancy between the SIFT 4G database and the input VCF file | Rename chromosomes to match the naming convention of the SIFT 4G database. Chromosome names are listed once a SIFT 4G database is selected. Change the VCF input file to match the chromosome naming convention. For example, SIFT 4G's human database uses the chromosome naming convention '1' and not 'chr1'. Then, the input VCF file has to use '1' and not 'chr1' in its first column 'CHROM' |

**● TIMING**
Step 1(A), annotate variants with the SIFT 4G annotator via the GUI: 45 min for first-time operation; 5 min for subsequent runs. The timing for the initial setup depends on the size of the SIFT 4G database and the user's bandwidth connection. Assuming a 30 Mbps download speed, downloading and extracting a human database of size 3.2 GB takes ~40 min.

For the *E. coli* database (113 MB), the process takes 1 min. After the database has been downloaded locally, timing for variant annotation depends on the size of the variant list and the organism database. Annotation of 6.7 million human variants takes ~5 min on a CPU Intel(TM)i7-3520M CPU @ 2.90 GHz.

Step 1(B), annotate variants with the SIFT 4G annotator via command line: 45 min for first-time operation; 5 min for subsequent runs. Timing of Step 1(B) is identical to the timing of Step 1(A).

**Box 1**, running the SIFT 4G algorithm on protein sequences: ~6 h for installation and setup. For subsequent runs, an average 2.6 s per protein sequence and 40 s minimum (for loading the protein sequence database into memory). The SIFT 4G algorithm takes ~43 min (2,573 s total; 2.6 s per protein) to return predictions for 1,000 protein sequences on an NVIDIA graphics card (GeForce GTX670) accessing storage on a 1 TB solid-state drive array (2× 500 GB Samsung 840 Evo). For the same 1,000 proteins, the CPU version of SIFT takes 70 h (250,982 s total; 4.2 min per protein) on a CPU Intel Core 2 Duo (i7-2600 CPU @ 3.40 GHz).

## ANTICIPATED RESULTS

### SIFT 4G annotator output

For a given list of genomic variants, the SIFT 4G annotator creates two output files containing SIFT 4G predictions and gene annotations (**Table 3**). SIFT 4G predictions are provided for single-nucleotide missense variants, and they take on the values 'deleterious' or 'tolerated'. For other potentially functional variants (such as synonymous, frameshift and untranslated region (UTR)), annotations are provided but not functional predictions. The two output files are as follows:

- *Output VCF file*: Results are generated in a VCF file entitled '<*original filename*>_SIFTPredictions.vcf'. For each variant from the original VCF file, SIFT 4G annotations are appended to the INFO column (eighth column), using the keyword 'SIFTINFO=' (**Supplementary Table 2**). Each data field is separated by the delimiter '|'. If the user chose the option to annotate the variants with multiple transcripts, the annotations for multiple transcripts are delimited by commas. The first seven columns of the output VCF file are identical to the input VCF file, and the output VCF file will have the same number of rows as the input VCF file.

**TABLE 3** | SIFT 4G output data fields.

| Output column name | Description |
|---|---|
| TRANSCRIPT_ID | Transcript ID |
| GENE_ID | Gene ID |
| GENE_NAME | Gene name |
| REGION | Region |
| VARIANT_TYPE | Indicates the type of variant (SYNONYMOUS, NON-SYNONYMOUS, STOP-LOSS, STOP_GAIN, START_LOSS, FRAMESHIFT-INSERTION, FRAMESHIFT-DELETION, INFRAME-INSERTION, INFRAME-DELETION, SUBSTITUTION) |
| REF_AMINO | Reference amino acid |
| ALT_AMINO | Alternate amino acid |
| AMINO_POS | Position of amino acid change |
| SIFT_SCORE | Ranges from 0 to 1. The amino acid substitution is predicted deleterious if the score is <0.05, and tolerated if the score is ≥0.05 |
| SIFT_MEDIAN | Ranges from 0 to 4.32; ideally the number would be between 2.75 and 3.25. This is used to measure the diversity of the sequences used for prediction. A warning will be given if this is greater than 3.5, because this indicates that the prediction was based on closely related sequences |
| NUM_SEQ | This is the number of sequences that have an amino acid at that particular position of prediction. If the substitution is located at the beginning or end of the protein, there may be only a few sequences represented at that position |
| dbSNP | dbSNP ID |
| SIFT_PREDICTION | Prediction can be deleterious (affects protein function) or tolerated (neutral) |

**TABLE 4** | SIFT 4G TSV output format.

| # CHROM | POS | REF_ ALLELE | ALT_ ALLELE | TRANSCRIPT_ID | GENE_ID | GENE_ NAME | REGION | VARIANT_ TYPE | REF_ AA | ALT_ AA | AA_ POS | SIFT_ SCORE | SIFT_ MEDIAN | NUM_ SEQs | dbSNP | PREDICTION |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 881918 | G | A | ENST00000327044 | ENSG00000188976 | NOC2L | CDS | NONSYNONYMOUS | S | L | 556 | 0.095 | 2.54 | 44 | rs35471880 | TOLERATED |
| 1 | 900505 | G | C | ENST00000338591 | ENSG00000187961 | KLHL17 | CDS | SYNONYMOUS | V | V | 621 | 1 | 2.63 | 79 | rs28705211 | TOLERATED |
| 1 | 900717 | CTTAT | C | ENST00000338591 | ENSG00000187961 | KLHL18 | UTR_3 | FRAMESHIFT DELETION | NA | NA | NA | NA | NA | NA | Novel | NA |

NA, not applicable.

- *Output Microsoft Excel file*: SIFT 4G annotator generates an XLS file named '<*original filename*>_SIFTAnnotationsOnly.xls'. This file contains only the variants that have been assigned annotations. As genome sequencing identifies intronic and intergenic variants that are often presumed to be neutral, removing these variants creates a smaller file that a non-bioinformaticist can easily manipulate (**Table 4**). The first four columns in the XLS file are CHROM, POS, REF ALLELE and ALT ALLELE, which are taken from the input VCF file. The remaining columns are SIFT 4G annotations. If the user opted to annotate the variants with multiple transcripts, the annotation for each transcript will be displayed in a separate row in the XLS file.

Please note that the number of rows in the output XLS file can be less than the number of rows in the input file because variants without annotation (e.g., intronic and intergenic) are not printed out in the XLS file. In addition, when there is more than one alternate allele for an input variant, the annotations for each alternate allele will be displayed on a separate row in the output XLS file.

### SIFT 4G algorithm output

We describe the output from running the SIFT 4G algorithm. For each protein sequence file (e.g., xyz123.fa) and its corresponding amino acid substitution file (e.g., xyz123.subst), the SIFT 4G algorithm creates a prediction file (e.g., xyz123.SIFTprediction). Each row of the prediction file contains the amino acid substitution from the input amino acid substitution file, followed by its SIFT_PREDICTION, SIFT_SCORE, SIFT_MEDIAN and NUM_SEQ (as described in **Table 3**). The last column in the prediction file contains the total number of sequences found in the alignment.

*Note: Any Supplementary Information and Source Data files are available in the online version of the paper.*

**ACKNOWLEDGMENTS** This work is financed in part by A*STAR and the Croatian Science Foundation (project no. 7353, Algorithms for Genome Sequence Analysis). We thank P.C.N.'s significant other for donating his gaming computer 'for science' and M. Korpar for providing the SW#db library.

**AUTHOR CONTRIBUTIONS** M.S. and P.C.N. conceived the project. R.V. implemented and tested the performance of the SIFT 4G algorithm. S.A. and S.N.L. implemented the SIFT 4G annotator. S.A. and P.C.N. wrote the manuscript.

**COMPETING FINANCIAL INTERESTS** The authors declare no competing financial interests.



1. Xia, Q. *et al.* Complete resequencing of 40 genomes reveals domestication events and genes in silkworm (*Bombyx*). *Science* **326**, 433–436 (2009).
2. The Bovine HapMap Consortium. Genome-wide survey of SNP variation uncovers the genetic structure of cattle breeds. *Science* **324**, 528–532 (2009).
3. Huang, X. *et al.* A map of rice genome variation reveals the origin of cultivated rice. *Nature* **490**, 497–501 (2012).
4. McNally, K.L. *et al.* Sequencing multiple and diverse rice varieties. Connecting whole-genome variation with phenotypes. *Plant Physiol.* **141**, 26–31 (2006).
5. The 3,000 rice genomes project. The 3,000 rice genomes project. *Gigascience* **3**, 7 (2014).
6. Herper, M. *Gene Machine* (Forbes, 2010).
7. Drosophila 12 Genomes Consortium. Evolution of genes and genomes on the *Drosophila* phylogeny. *Nature* **450**, 203–218 (2007).
8. Atanur, S.S. *et al.* The genome sequence of the spontaneously hypertensive rat: analysis and functional significance. *Genome Res.* **20**, 791–803 (2010).
9. Seppälä, E.H. *et al.* LGI2 truncation causes a remitting focal epilepsy in dogs. *PLoS Genet.* **7**, e1002194 (2011).
10. Kumar, P., Henikoff, S. & Ng, P.C. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat. Protoc.* **4**, 1073–1081 (2009).
11. Ng, P.C. & Henikoff, S. Accounting for human polymorphisms predicted to affect protein function. *Genome Res.* **12**, 436–446 (2002).
12. Ng, P.C. & Henikoff, S. SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Res.* **31**, 3812–3814 (2003).
13. Sim, N.L. *et al.* SIFT web server: predicting effects of amino acid substitutions on proteins. *Nucleic Acids Res.* **40**, W452–W457 (2012).
14. Henikoff, S., Till, B.J. & Comai, L. TILLING. Traditional mutagenesis meets functional genomics. *Plant Physiol.* **135**, 630–636 (2004).
15. Mitsui, J. *et al. CSF1R* mutations identified in three families with autosomal dominantly inherited leukoencephalopathy. *Am. J. Med. Genet. B Neuropsychiatr. Genet.* **159B**, 951–957 (2012).
16. Tennessen, J.A. *et al.* Evolution and functional impact of rare coding variation from deep sequencing of human exomes. *Science* **337**, 64–69 (2012).
17. Lamichhaney, S. *et al.* Evolution of Darwin's finches and their beaks revealed by genome sequencing. *Nature* **518**, 371–375 (2015).
18. Leida, C. *et al.* Variability of candidate genes, genetic structure and association with sugar accumulation and climacteric behavior in a broad germplasm collection of melon (*Cucumis melo* L.). *BMC Genet.* **16**, 28 (2015).
19. Moreira, G.C. *et al.* Variant discovery in a QTL region on chromosome 3 associated with fatness in chickens. *Anim. Genet.* **46**, 141–147 (2015).

20. Ortega, R., Guzmán, C. & Alvarez, J. *Wx* gene in diploid wheat: molecular characterization of five novel alleles from einkorn (*Triticum monococcum* L. ssp. *monococcum*) and *T. urartu*. *Mol. Breeding* **34**, 1137–1146 (2014).
21. Renaut, S. & Rieseberg, L.H. The accumulation of deleterious mutations as a consequence of domestication and improvement in sunflowers and other Compositae crops. *Mol. Biol. Evol.* **32**, 2273–2283 (2015).
22. Choi, J.W. *et al.* Whole-genome resequencing analyses of five pig breeds, including Korean wild and native, and three European origin breeds. *DNA Res.* **22**, 259–267 (2015).
23. Schensted, C. Longest increasing and decreasing subsequences. *Can. J. Math.* **13**, 179–191 (1961).
24. Korpar, M., Sosic, M., Blazeka, D. & Sikic,, M. SW#db: GPU-accelerated exact sequence similarity database search. doi:10.1101/013805 (14 January 2015).
25. Ng, P.C. & Henikoff, S. Predicting deleterious amino acid substitutions. *Genome Res.* **11**, 863–874 (2001).
26. Altschul, S.F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
27. Suzek, B.E., Huang, H., McGarvey, P., Mazumder, R. & Wu, C.H. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics* **23**, 1282–1288 (2007).
28. Adzhubei, I.A. *et al.* A method and server for predicting damaging missense mutations. *Nat. Methods* **7**, 248–249 (2010).
29. Pace, H.C. *et al.* Lac repressor genetic map in real space. *Trends Biochem. Sci.* **22**, 334–339 (1997).
30. Rennell, D., Bouvier, S.E., Hardy, L.W. & Poteete, A.R. Systematic mutation of bacteriophage T4 lysozyme. *J. Mol. Biol.* **222**, 67–88 (1991).
31. Kent, W.J. *et al.* The human genome browser at UCSC. *Genome Res.* **12**, 996–1006 (2002).
32. Goodstein, D.M. *et al.* Phytozome: a comparative platform for green plant genomics. *Nucleic Acids Res.* **40**, D1178–D1186 (2012).
33. Pruitt, K.D., Tatusova, T. & Maglott, D.R. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.* **33**, D501–D504 (2005).