nature | methods

MAQGene: software to facilitate *C. elegans* mutant genome sequence analysis

Henry Bigelow, Maria Doitsidou, Sumeet Sarin & Oliver Hobert

Supplementary figures and text:

Supplementary Table 1	Practical aspects of whole genome sequencing.
Supplementary Note 1	Installing, running, and background information on MAQGene.

Note: Supplementary Software is available on the Nature Methods website.

Nature Methods: doi:10.1038/nmeth.f.260

Supplementary Table 1: Practical Aspects of Whole Genome Sequencing

• Pre-sequence analysis:	• amount of genetic mapping depends on ease with which candidate mutations can be validated; if multiple alleles are available for validating a candidate locus through Sanger re-sequencing of these other alleles, mapping to perhaps as little as a chromosome is required. If two independently recovered alleles of the same loci are genome-sequenced then only the genes that are affected in both datasets need to be validated and in this case no mapping may be required at all.
• Outcross:	• if only a single allele of a locus is genome-sequenced, outcrosses against the starting strain (which was subjected to mutagenesis) is recommended; note that other whole-genome-sequenced mutant genomes that were similarly outcrossed, help to subtract strain background variants; also note that the closer a background mutation is linked to the locus of interest, the harder it is to outcross and therefore the ease with which a variant can be validated (RNAi, rescue, Sanger-sequencing of other alleles) determines how much outcrossing should be done. • if two independently recovered alleles of the same locus are genome-sequenced, *no* outcrossing is required as all background mutations (that affect different loci) can be subtracted as they are unique to either of the two datasets. Mutations that are unique to the two datasets but affect the same locus might identify the phenotype causing mutant locus.
• Sample information:	• 5 µg genomic *C. elegans* DNA
• Homozygosity:	• MAQgene parameters are optimized for homozygous samples. Heterozygous variants can be retrieved by adjusting the ratio of mutant versus total number of reads to values closer to 0.5. However, heterozygous variants are more difficult to call with confidence and require more coverage.
• Depth coverage:	• 20x coverage (3 lanes on a Illumina GA II flowcell) optimal, 12-14x (2 lanes, depending on yield) recommended, either with 36 bp paired end or 76 base single end runs on an Illumina GA II. 20x average coverage covers ~97% of genome at 2x or greater. 12x average coverage covers ~90% at 2x or greater. These numbers are based on the *C. elegans* genome size of 100,281,426 bases; for other genome sizes, scale the number of lanes appropriately.

See References (3) for more information.

Supplementary Note 1

Installing, running, and background information on MAQGene.

1. Summary

We provide below a detailed description of the implementation and usage of a simple web interface, MAQGene, for launching the MAQ (Mapping and Assembly with Quality) software suite to identify mutations from Illumina Genome Analyzer® fastq-formatted reads and to further classify the mutations based on associated exon annotations. Each mutation is associated to nearby exons and classified by its canonically predicted effect on splicing and translation. A results file represents the output on the web server for download and import into a single Excel spreadsheet, enabling sorting and filtering on all columns. As even a single run is computationally intensive, we do not offer MAQGene as a hosted worldwide service in itself. Rather, users must first install it on their own server and configure it for lab- or university-wide use. Once installed, however, it does not require any technical knowledge of command-line tools and can be run through the web interface entirely.

2. Installing MAQGene

We recommend that installment of MAQGene be done by a computer systems administrator (who requires no knowledge about the purpose of the program); once installed, no detailed computer knowledge is required from the end-users who routinely run MAQGene. To enable users to get a feel for it without the initial time investment, a demo version providing real reads and a reference genome of *C. elegans* is available at http://alice.cumc.columbia.edu/demo/index.php as resources allow.

The MAQGene interface consists of several shell script files, web pages and executable files besides the required MAQ software suite (which can be downloaded at http://maq.sourceforge.net/). All MAQGene specific files are available in a file maqgene-0.9.0.tar.gz, including an installation script, at http://maqweb.sourceforge.net. MAQGene requires a MySQL server and Apache2 with PHP enabled. Before running the installation script, a single configuration file must be filled in with specific values for the user environment, including URLs for downloading GFF-formatted exon annotations and matching genome sequence, directories for installation and use during runs, and a username / password of an appropriate MySQL account. Ideally, the host computer will be a server or have comparably powerful hardware, though it is not an absolute requirement.

Apache must be configured with environment variables with the following lines in the main apache configuration file httpd.conf (usually `/usr/local/apache2/conf/httpd.conf`):

```
SetEnv PIPELINE_IN_DIR /data/pipeline_in
SetEnv PIPELINE_OUT_DIR /data/pipeline_out
SetEnv PIPELINE_WORK_DIR /data/pipeline_out/work
SetEnv PIPELINE_RUNS_DIR /data/pipeline_in/Runs
SetEnv PIPELINE_GENOME_DIR /data/pipeline_in/Genomes
SetEnv PIPELINE_SCRIPT_DIR /home/sbsuser/scr
SetEnv PIPELINE_EXE_DIR /usr/local/bin
```

To install MAQGene on a particular system requires write permissions to the chosen directories. The install script will create them automatically.

Once MAQGene is installed, the script `install_annotations.sh` must be run. During this run, the user is prompted for the formal and informal names of a species of interest and a preferred data source (currently ensembl, refseq, or wormbase). The script then downloads a genome sequence file and matching genome annotation file from preconfigured URLs depending on the source chosen. The script unpacks and processes the genome sequence file and saves it in three versions: the original fasta format, `maq`'s .bfa format, and `cisortho`'s .fa and .dnas format. The annotation file is parsed and used to produce three MySQL tables, named `<species>`_{`features`,`coding_dna`,`genetic_code`}. For example, if `<species>` were 'elegans': `elegans_features` contains various different genome annotations, including chromosome (or contig), start and end positions, strand, and feature type. The subset of features comprising protein-coding genomes are further processed to produce table `elegans_coding_dna`, which contains the actual dna sequence representing the open reading frame. Finally, `elegans_genetic_code` contains the genetic code for the species of interest. The genetic code is parsed from the text file `codons.txt` included in the MAQGene distribution, and contains the standard genetic code. If a non-standard genetic code is needed, the user must manually replace it with an identically formatted file reflecting the desired code.

When a run is launched (as further described below), MAQGene splits the entire set of input reads into chunks of `CHUNK_SIZE` reads (defaulted to 3 million) and runs the mapping step in parallel on `NCPU` processors. Depending on how much system memory is available, you must choose `CHUNK_SIZE` and `NCPU` appropriately during installation. MAQGene ensures that no more than `NCPU` processors are used at any one time, even when runs are submitted concurrently. In that case, the first run submitted will take up all allotted CPUs and the second run will be on hold, taking CPUs as they become available.

The following example illustrates CPU time and memory requirement: Mapping 56 million 36-base paired-end reads against the *C elegans* reference sequence using a maximum of 2 mismatches in the first 28 bases took 1.8 GB memory for each chunk of 3 million reads, roughly 8.3 bytes per mapped nucleotide. The mapping step took 41 minutes per chunk out of 19 chunks, totaling 9.5 CPU hours. The entire rest of the run, involving map merging, consensus building, and MySQL table building to classify mutations runs on a single CPU, regardless of `NCPU`. It took an additional 38 minutes and negligible memory.

During the run, MAQGene stores and merges all intermediate work files that MAQ produces. The 56 million read run described required a peak disk space of about 12 GB. The final required storage needed is negligible (in the MB range). However, if the main .map file is stored (2.4 GB) reruns will be able to avoid the majority of processing time.

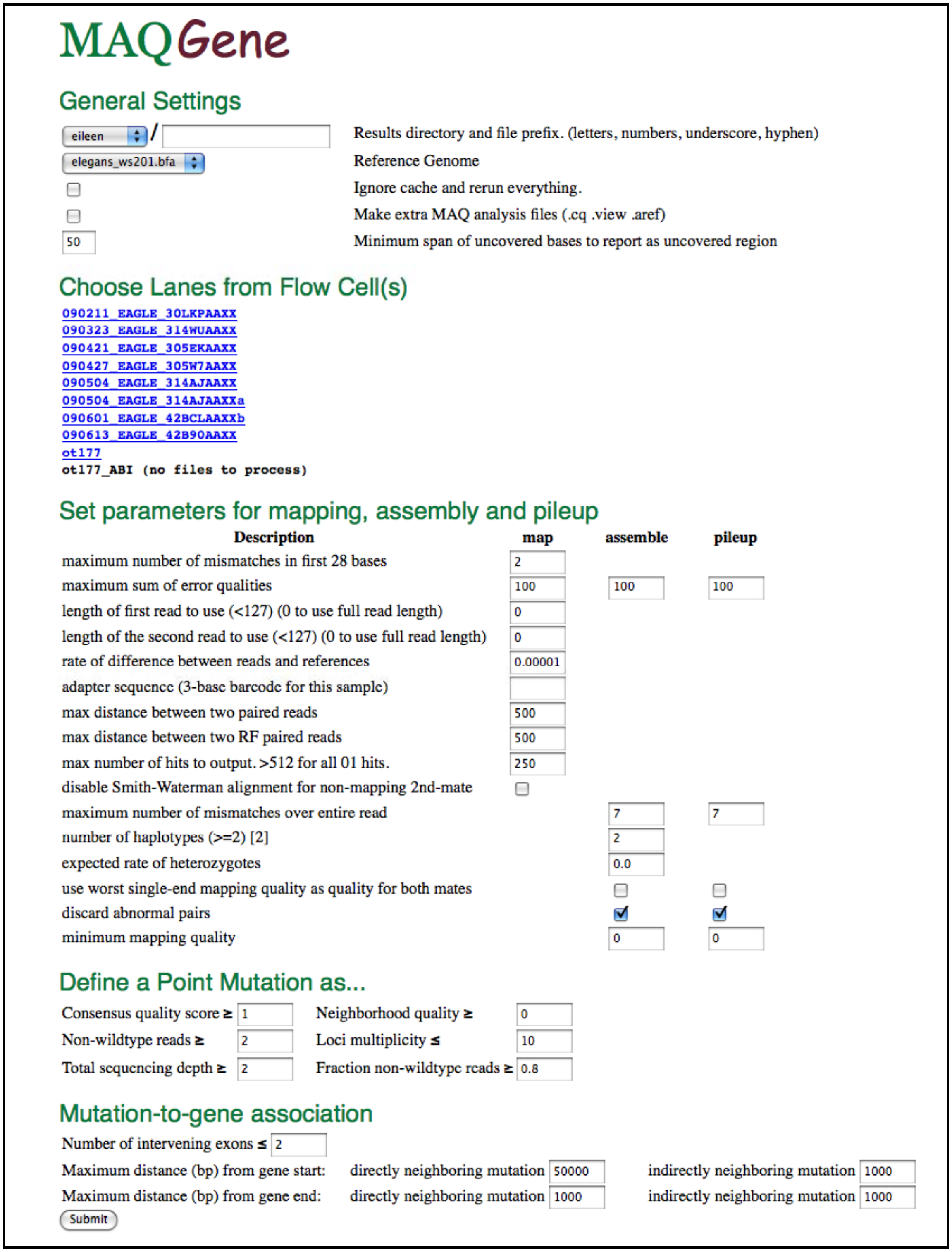
3. WGS data input

Once MAQGene is installed, the user then needs to provide the read data for each WGS dataset to be analyzed. For this, the files containing short reads (up to 127 bases long) obtained from an Illumina Genome Analyzer data must be in FASTQ format (http://maq.sourceforge.net/fastq.shtml), and placed in a subfolder of `PIPELINE_RUNS_DIR`. The organization of reads in directories is left up to the user, but in our case we use one directory per sequencing run and 8 fastq files (or pairs of files, for paired-end runs) in each directory, corresponding to each lane of a flow cell. Users who obtain fastq files independently may want to organize them differently.

Each directory will be visible in the browser (described below) and have ‘click-to-expand’ behavior. When clicked, the fastq files are shown with checkboxes and may be individually selected. In the case of paired-end reads, each pair of files may be individually selected.

4. Running MAQGene

Below is a complete screenshot of MAQGene. All parameters for running MAQ as well as further mutant classification are set here. Note that default parameters are provided that we found useful for analysis of *C. elegans* WGS data obtained from paired-end runs on an Illumina Genome Analyzer II platform (discussed further below). Each section of the website is described in the ensuing sections below.



MAQGene

General Settings

eileen /	Results directory and file prefix. (letters, numbers, underscore, hyphen)
elegans_ws201.bfa	Reference Genome
□	Ignore cache and rerun everything.
□	Make extra MAQ analysis files (.cq .view .aref)
50	Minimum span of uncovered bases to report as uncovered region

Choose Lanes from Flow Cell(s)

090211_EAGLE_30LKPAAXX
090323_EAGLE_314WUAAXX
090421_EAGLE_305EKAAXX
090427_EAGLE_305W7AAXX
090504_EAGLE_314AJAAXX
090504_EAGLE_314AJAAXXa
090601_EAGLE_42BCLAAXXb
090613_EAGLE_42B90AAXX
ot177
ot177_ABI (no files to process)

Set parameters for mapping, assembly and pileup

Description	map	assemble	pileup
maximum number of mismatches in first 28 bases	2		
maximum sum of error qualities	100	100	100
length of first read to use (<127) (0 to use full read length)	0		
length of the second read to use (<127) (0 to use full read length)	0		
rate of difference between reads and references	0.00001		
adapter sequence (3-base barcode for this sample)			
max distance between two paired reads	500		
max distance between two RF paired reads	500		
max number of hits to output. >512 for all 01 hits.	250		
disable Smith-Waterman alignment for non-mapping 2nd-mate	□		
maximum number of mismatches over entire read		7	7
number of haplotypes (>=2) [2]		2	
expected rate of heterozygotes		0.0	
use worst single-end mapping quality as quality for both mates		□	□
discard abnormal pairs		☑	☑
minimum mapping quality		0	0

Define a Point Mutation as...

Consensus quality score ≥	1	Neighborhood quality ≥	0
Non-wildtype reads ≥	2	Loci multiplicity ≤	10
Total sequencing depth ≥	2	Fraction non-wildtype reads ≥	0.8

Mutation-to-gene association

Number of intervening exons ≤ 2

Maximum distance (bp) from gene start:	directly neighboring mutation	50000	indirectly neighboring mutation	1000
Maximum distance (bp) from gene end:	directly neighboring mutation	1000	indirectly neighboring mutation	1000

Submit

• General Settings

Basic settings for the entire run must be chosen in this section.

Results directory and file prefix. If 'eileen' is chosen for **directory** and 'ot177' is chosen for **file prefix**, then MAQGene will produce all output files to PIPELINE_OUT_DIR/eileen/ot177 and additionally name those files starting with 'ot177', i.e. ot177.txt, ot177_coverage.txt, etc. The set of first level directories existing in PIPELINE_OUT_DIR will appear in the pull-down menu and must be created by the system administrator with permissions drwxrwxrwx (writable and readable by all). However, MAQGene automatically creates any subdirectories as named in the chosen file prefix; in this case, 'ot177' will be created.

Reference Genome. MAQGene will map all reads to this genome and use corresponding annotation data for variant classification based on this choice.

During MAQGene installation, the script install_annotations.sh may be run to install sequence and annotation data once for each species as will be desired for MAQGene analysis (see 'Installing MAQGene'.) In research environments with only one species of interest, there will be just a single choice here.

Ignore cache and rerun everything. Users may want to repeat a run with slightly different parameters. To avoid re-running the unaffected parts, MAQGene caches the most time-consuming steps in PIPELINE_WORK_DIR, naming each file using a 64-bit checksum of all relevant input parameters for any given stage. On reruns, MAQGene can then check if the relevant file exists, effectively determining whether a previous run with exactly those same parameters was performed before. This caching mechanism concerns large intermediate files that the user never sees. Results files, on the other hand, will be automatically over-written if the same **Results directory** and **file prefix** are used again in subsequent runs.

Make extra MAQ analysis files (.cq .view .aref). MAQ optionally provides extra output files. The .cq (consensus fastq) file is the deduced sample sequence with associated consensus quality scores per base. It will be the same size as the chosen reference sequence and correspond base for base, differing in positions where a mutation was inferred. The .view file shows detailed quality information and statistics for the consensus at every position in the reference. The .aref file is simply the original reference sequence in FASTA format. See original MAQ documentation at http://maq.sourceforge.net/maq-manpage.shtml for further details.

Minimum span of uncovered bases to report as uncovered region. All spans of contiguously uncovered bases longer than this minimum will be reported as intervals in the output file <prefix>_uncovered.txt, in the format CHROMOSOME START END with one interval per line. This information may be used heuristically to find potential large deletions. However, for a run of low average depth (say 6x), these regions are more likely to represent the lack of any usable sequence information due to random fluctuation in read depth rather than true deletions in the sample.

• Choose Lanes from Flow Cell(s)

All first-level directories in PIPELINE_RUNS_DIR will appear in this section as HTML links that toggle the display of all fastq files within. Each fastq has a checkbox for individually selecting it for a run. Any combination of fastq files among any number of directories may be selected simultaneously, though they must all be of the same type, whether mate-paired or single-read. Mate-paired files will automatically be detected by examining the first line of each file for matching IDs and placed in the same line. The user may create subdirectories in `PIPELINE_RUNS_DIR` and place their fastq files within them. Note that selected files remain selected even after they are hidden.

• Set parameters for mapping, assembly and pileup

This section reproduces all of the available input options for MAQ (version 0.7.1 as of this writing), simply passing them off to the relevant command. A defaults file called 'params.txt' installed in the `WEBROOT` directory (see Apache documentation) determines what defaults and options are available for selection here. It comes pre-configured with sensible defaults but may be edited during MAQGene installation to suit user preference.

We describe each option below, sometimes adding explanation or commentary of our experience using MAQ. In the spirit of flexibility we've allowed the user to set every parameter individually. However, commonly named parameters, i.e. those residing in the same rows for **map**, **assemble**, and **pileup** in the website, should be set to the same chosen value. Note also that parameters expressing a threshold logic work as absolute requirements. Thus for a read to be used, it must pass every threshold. For further details the reader is referred to the original MAQ site, http://maq.sourceforge.net/.

Maximum number of mismatches (first 28 bases or entire read). The maximum number of single-base mismatches between the read and the locus it can be mapped to. If no locus in the reference genome can be found which matches the read with this or fewer mismatches, it will not be used in the analysis at all and instead recorded in a file with extension .unmapped. For efficiency reasons, the mapping step only applies a threshold to the first 28 bases. Assembly and pileup apply the threshold to the entire read. Note that the parameters that further control the mapping step are applied to the entire length of the reads (see **Maximum sum of error qualities** and **Minimum mapping quality** below).

Number of haplotypes. In our experience, results were identical when this parameter was set to 1, 2, or 3, using MAQ version 0.7.1. Furthermore we were unable to find any useful documentation on how this parameter affects mapping or consensus generation, and provide it here in the event it becomes meaningful in a future version of MAQ.

Maximum sum of error qualities. This is the maximum value of the sum of per-base quality scores of all mismatching bases in a read. For example, a read ACGCGTAGC with quality scores 40 38 38 40 32 31 15 12 6 and mapped to the reference ACTCGGAGA mismatches at positions 3, 6, 9. These three positions have a sum of quality scores 38 + 31 + 6 = 75. The term 'error qualities' is a misnomer; it is the qualities of mismatching bases in a mapped read that are summed. Mismatching bases arise from sequencing errors, but also from true differences between sample and reference sequence, and mismapping of a read. In general this setting is a tradeoff between keeping reads that would cover true sample mutations and eliminating reads containing sequencing errors. For 36-base pair reads, we recommend a setting of 100. For a 76-single run reads we recommend the setting of 200.

Minimum mapping quality. For a read to map to a given locus, the 'mapping quality' must be at or above this value. Mapping quality, or Q_s, is an approximation of the probability of a mapping error, and expressed as a Phred-scaled quantity (MAQ original paper, equation 1 and following.)

Length of first/second read to use in analysis. Set this value to consider only a fixed length sub-portion of each read. The length can be different for first and second mate pairs, respecting the fact that the second mate-paired reads may have overall lesser quality than the first. The default value of zero means to disable truncation and use the entire length of the reads. In our experience, a zero setting is fine. However, truncating reads that show overall poor quality past a certain base position is a sensible idea and will speed up the analysis slightly. Note that first and second mate pairs are defined by the trailing '/1' and '/2' in the ID line of each read in the fastq file.

Rate of difference between reads and reference. The 'rate of difference between reads and reference', or '-m' parameter of the 'maq map' command, is also called 'Mutation rate between the reference sequences and the reads' on the website manual page (http://maq.sourceforge.net/maq-manpage.shtml). Besides this one-line description of the parameter, there is no documentation on how this value is used in the actual mapping or consensus calling algorithms.

Adapter sequence. When samples are not tagged with any adapter, leave this field empty. Illumina sequencing reactions may be prepared with a tri-nucleotide 5'-end tag which serves as an identifying 'barcode' prefix for individual samples. It thus allows mixing two or more samples in a given flow cell lane. Supplying an adapter sequence causes MAQ to consider only reads starting with this particular tri-nucleotide sequence tag, and maps the rest of the read to the reference genome in otherwise normal way.

Max distance between two paired reads / RF paired reads. The first of these parameters should be set to the largest possible size of the size-fractionated DNA used in sample preparation. In paired-end mapping mode, MAQ will consider the mate-paired reads that map in the proper relative orientation and such that the total distance on the reference genome spanned by the reads is less than this threshold. If the data are generated using Illumina's long insert library, both thresholds should be set to the same value. Otherwise, it is only necessary to set the first threshold.

Max number of hits to output. Theoretically a given read may map to an arbitrarily large number of loci, even with stringent thresholds on quality of mapping. This happens frequently with reads from repetitive regions in genomes. This threshold is an upper limit on the number of such mappings to report. The set of such mappings in these (rare) cases will be a random subset of the total set of mappings that would be output otherwise. The liberal default value of 250 is effective in limiting highly promiscuous reads while remaining unobtrusive for the vast majority of reads. It is recommended to leave this parameter at its default value.

Disable Smith-Waterman alignment for non-mapping 2nd-mate. When MAQ successfully aligns one but not the other of two mate-paired reads, it will use Smith-Waterman alignment with gaps to align the second read in the proper orientation and distance threshold determined by its mate pair. Though not recommended, you may check the box to disable this feature. We provide it here to give the user complete control over running MAQ.

Expected rate of heterozygotes. First, MAQ uses this parameter to calculate the most likely genotype from a set of reads, and further the probability that such genotype is indeed correct. The calculated probability of correctness is itself only approximate.

In theory, setting this parameter to the true rate of heterozygous loci in the sample would make this calculated probability correct. In practice, the true rate is unknowable; it can however be chosen to achieve a desired effect. Setting the parameter to a smaller value causes MAQ to call loci as homozygous, and interpret the non-majority bases of mapped reads to the locus in question as sequencing or mapping errors. A higher value on the other hand, causes MAQ to interpret the bases as correct reads in a truly heterozygous locus.

To this issue the original MAQ paper states: “We usually use r = 0.001 for the discovery of new SNPs and r = 0.2 for inferring genotypes at known SNP sites”. ‘r’ here is the ‘expected rate of heterozygotes’, i.e. the fraction of loci in the genome that are heterozygous. We kept this value as recommended by MAQ.

Use single-end mapping quality. Only relevant for paired-end mapping, checking this box tells MAQ to always use the mapping quality of individual reads regardless of the quality of their mate pairs. If this box is unchecked, MAQ rewards uniquely mapping mate-pairs by setting the mapping quality of both mates as the sum of the individual calculated qualities. Since qualities are Phred-based, the sum represents a product in probability terms, and thus the joint probability that the pair as a unit is mismapped.

Discard abnormal pairs. Discards read pairs that are not mapped within the distance constraint or orientation. This option should be checked for a normal run. It exists for diagnostic purposes.

• Define a point mutation as…

The final consensus as calculated from MAQ assigns several quality descriptors to every single base position in the reference. For loci that differ from the reference, these quality descriptors allow judging between sequencing error and true mutation. This section of MAQGene allows the user to filter the results based on thresholds as described below.

Minimum consensus quality score. Only report mutants in which the supporting reads achieve at least this consensus quality score, as defined in the original MAQ paper [1], section “Consensus genotype calling”. This score is a probability of the consensus base being correct, but expressed in the Phred form (see http://maq.sourceforge.net/qual.shtml).

Neighborhood quality. Documented as an idea inspired by NQS, the Neighborhood quality standard [2]. Originally designed for assessing quality of bases in individual reads, NQS is a standard that requires the base in question to have Phred quality score >= 20, and for its five neighboring bases on either side to have Phred quality scores >= 15. MAQ applies this idea to the consensus and gives a continuously valued score to each base. The exact formula is not provided in the original documentation. From our experience the value of this variable is not by itself critical for the validity of the called variant and we therefore set it to >=0.

Non-wildtype reads. The minimum number of reads that differ from reference sequence at that position, required to consider the locus a true point mutation. Setting this threshold to 2 is advised to eliminate 1x covered loci with a mutant base (which thus have a fraction non-wildtype reads equal to 1.00.)

Loci multiplicity. The maximum number of distinct loci in the reference genome that identical reads map to. For loci in the reference genome that are the result of internal duplications, there is no way to computationally deduce which among them is the source of individual reads. Thus, although MAQ chooses among identical mapping locations at random, it reports their loci multiplicity here as an aid to filtering out such problematic locations. We recommend a loci multiplicity value of <=1 (see section below).

Total sequencing depth. Even with good (>20x) average sequencing depth across the genome, sequencing depth varies greatly across the genome. This threshold excludes regions having too little depth. To illustrate, at a depth of two and sequencing error rate of 0.01 (1 of every 100 bases), having two different reads with the same base at a given position means there is a true mutation with 99.99 % chance, or an error of 1/10,000. Due to read dependencies, unknown systematic errors, or polymerase errors in the template amplification step, however, a minimum of 2 total depth and 2 non-wildtype reads is the very least stringent sensible setting.

Fraction non-wildtype reads. The minimum fraction of non-wildtype reads that must be present overall for this locus to be considered a true mutation. This threshold is useful for filtering out higher-coverage loci that have an above-threshold *number* of non-wildtype reads but nonetheless a poor fraction of non-wildtype reads. It is also useful to eliminate heterozygous loci by setting the fraction well above 0.5. Based on our validation of variants from a number of datasets (see below), we recommend a value of fraction of non-wildtype reads >= 0.8.

• Mutation-to-Gene Association

In order to annotate each mutation relative to its associated genes, MAQGene must define distance thresholds for associating each mutation to nearby exons, and thus to the genes they belong. It does this by separately considering thresholds for directly and indirectly neighboring exons. Further, for regions in which there are many small exons within a short span, it further allows limiting the association by the number of intervening exons.

• Adjusting default parameters based on available datasets

MAQ provides a good set of suggested default parameters that are largely incorporated into MAQGene (see Screenshot above). We found a readjustment of the default parameters for the definition of point mutations to be useful as this adjustment produced better results on a test dataset. As a test dataset we used a published genome sequence dataset of an *ot177* mutant *C. elegans* strain [3]. In this study, 54 variants were called (with MAQ default settings) in a 4 MB interval. 22 of those were miscalled (i.e. Sanger-resequencing showed that they were incorrect), 18 of them were strain background variants and 4 variants were true (i.e. validated, non-background) protein-coding variants.

We defined the parameter set in MAQGene for which (a) every single confirmed variant from the *ot177* dataset was correctly called (with the exception of one variant that maps in multiple regions in the genome and its validation was ambiguous) and (b) none of the incorrect variants were called. To achieve this, we modified the default parameter provided by MAQ as described below:

1) the consensus quality score was set to >= 1.
2) mutant vs total reads ratio was set to 0.8. This value is chosen to eliminate as many false positives from the *ot177* dataset without excluding any true variants. (Further data that supports this value: we validated a total of 105 point variants from three independent genome sequence datasets. 23 of them were below a fraction of 0.6, and all were false positives. 12 were between 0.6 and 0.83 and were all false positives. 70 were above 0.87 and all were confirmed to be real variants.)
3) loci multiplicity was set <= 1. We suggest a loci multiplicity value of <=1 to obtain variants with the highest fidelity. For values higher than 1, which means that a percentage of reads map in multiple sites in the genome, the variant might be a result of mis-mapping and therefore we recommend to put a lower priority for further pursuit of those variants

These settings revealed 19 additional, novel variants in the 4MB interval of the *ot177* strains. 8 of those variants were also found in several unpublished genome sequence dataset produced in the laboratory (*ot240, ot260 and ot263* mutants). These overlaps are likely an indication of strain background variants. 2 of them were confirmed true variants that were missed in the initial *ot177* published dataset. None of the additional 9 variants were protein-coding variants (a total of 27 protein coding variants were found on the whole chromosome using the above criteria 10 of which were present also in other datasets).

5. Output

Upon clicking ‘Submit’, MAQGene launches a script in the background and meanwhile redirects the browser to the results directory where output files will be written. The files all start with the user-chosen prefix and are given extensions `.aref, .check, .cq, .input, .log, .txt, and .view`. Additionally a `<prefix>_messages.txt` displays progress. To track progress the user must visit this file and periodically refresh the page until a message ‘Finished.’ appears at the bottom. Also, the file `<prefix>_pileup.txt` displays the ‘pileup’ at every reference sequence locus, i.e. the string of [ACGTacgt.,] characters denoting the base in each read mapped to that position in the reference. For details on this format, see MAQ documentation at http://maq.sourceforge.net/maq-manpage.shtml.

The file `<prefix>.check`, produced by MAQ, shows average sequencing depth among other statistics. MAQGene also produces a histogram of sequencing depth in `<prefix>_coverage.txt` (table of sequencing depth vs. number of bases covered at that depth) and a record of uncovered intervals larger than a threshold chosen in `<prefix>_uncovered.txt`.

Mutation Classification

After mapping, assembly and pileup are finished and the list of mutations is filtered according to the above criteria, MAQGene associates each variant in the list with gene annotation data. This enables classifying each variant among several categories: `5'UTR, 3'UTR, donor`, `acceptor`, `nongenic`, `silent`, `missense`, `non_start`, `premature_stop`, `readthrough`, `intronic`, `frameshift`, and `inframe`. Technically, it achieves this by the program `associate_genes`, which is part of the CisOrtho software suite [4] and included within the MAQGene distribution. Once each variant is located relative to nearest annotated exons, a simple calculation based on the genetic code or splice site definition is done and the proper classification assigned (for details see “Methods”). Indels that occur in exonic regions are either labeled ‘`frameshift`’ or ‘`inframe`’ depending on the insert or deletion size being divisible by 3. The categories ‘`premature_stop`’ and ‘`readthrough`’ indicate the creation or destruction of a stop codon, while ‘`nonstart`’ indicates the destruction of the start codon for the gene in question. All categories except ‘`frameshift`’ and ‘`inframe`’ are potentially assignable to single point mutations.

For indel variants, MAQGene only can assign ‘`frameshift`’, ‘`inframe`’, ‘`donor`’, ‘`acceptor`’, and ‘`nongenic`’ categories. In principle it would have been possible to include the ‘`premature_stop`’, ‘`nonstart`’, and ‘`readthrough`’, and ‘`missense`’ categories, but this is technically difficult, and these categories are subsumed within the categories of ‘`frameshift`’ and ‘`inframe`’.

For variants occurring in introns or intergenic regions, the logic of gene association is more complex. To define the limits of such an association, the user must enter a few thresholds in the section ‘Mutation-to-gene association’ of the webpage. In this section there is one set of thresholds for direct associations (in which there is no other exon between the mutation and gene in question) and a separate, usually more stringent set of thresholds for indirect associations. Finally, to limit associations in relatively complex areas of the genome with interleaved genes, the total number of intervening exons between a given gene boundary and a mutation may be limited as well.

Finally, there are rich sources of other genomic annotation data. From a functional standpoint, the most important ones are probably noncoding RNAs, such as miRNAs and other types of non-coding RNAs with possible regulatory functions (21U RNAs, predicted ncRNAs etc.). MAQGene reports all variants that directly overlap such features, setting their ‘class’ and ‘description’ fields according to the source and feature descriptions from the genomic annotation itself. During database installation with install_annotations.sh, all of these annotations are installed by default. For users who prefer to ignore some or all such nongenic features, instructions in the README file detail how to do so.

Some variants will overlap with more than one genomic feature, and thus potentially be reported multiple times in the final output. To avoid clutter and circumvent this, MAQGene reports only the most important association in order of decreasing importance: protein product effect, noncoding feature overlap, and finally, intronic or intergenic association. Thus, for example, a variant occurring in an intronic or intergenic region but also in a miRNA will be reported as miRNA, not the former. Similarly, a variant occurring simultaneously in a noncoding feature and affecting a protein product of a coding gene will be solely reported as the latter.

Sample Output

The main output of MAQGene consists of a single text-based file with one line per mutated locus in the entire genome. Shown below is one single row from a recent run (total 5,538 rows), transposed to column format. The sample is derived from a locus, *lsy-12(ot177)* that was later confirmed by re-sequencing to be the causal mutation for a phenotype of interest [3].

variant_id	72595
mutant_strain	ot177
dna	CHROMOSOME_V
start_position	9846724
6-species_conservation	0.203
reference_base	G
sample_base	A
consensus_score	26
loci_multiplicity	1
mapping_quality	42
neighbor_quality	26
number_wildtype_reads	0
number_variant_reads	23
sequencing_depth	23
sample_reads	@AaaAAAaAAAAAaAaaAaAaAAa
variant_type	Point
indel_size	0
class	premature_stop
description	CAG->TAG [Gln->stop]
exon_associations	R07B5.9()[+2364]
intron_associations	None
head_associations	None
tail_associations	None

The results file is initially sorted by dna, then start_position. Each locus is assigned several quality metrics that can be used for further sorting and filtering, usually using Excel. For point mutations, the relevant measures are consensus_score, loci_multiplicity, mapping_quality, neighbor_quality and sequencing_depth. The fields exon_associations, intron_associations, head_associations and tail_associations show genes associated with the mutation in question according to where the mutation resides relative to the gene. The format in these fields is GENE_ID(common name)[start offset], where start offset is negative for upstream (head) and positive for gene-internal and downstream associations (exon, intron, and tail).

`sample_reads` show a pileup at each reference locus of identified bases in the form of a list of symbols among [AaCcGgTg.,] reflecting the nucleotide mapped reads. [ACGT] and [acgt] indicate places where individual reads differ from the reference for the reads on the positive (upper case) or negative (lower case) strand, respectively. Dot and comma [.,] indicate nucleotides in reads on positive (dot) and negative (comma) strands that are identical to the reference. The total number of symbols at any given reference locus is called the 'sequencing depth' and is where the term 'deep sequencing' comes from.

For indel mutations, quality metrics except for sequencing depth do not apply. Since indels are not directly covered by any reads, sequencing depth in these cases is reported as the lesser of the sequencing depth of left and right-flanking regions of the indel. Also, the `reference_base` and `sample_base` are both set to 'X'. `indel_size` is the difference in length of the sample dna to reference dna at that position, and so is 0 for point mutations, positive for insertions and negative for deletions. Some indels are reported by MAQ that have zero mutation size, for unknown reasons.

`class` takes on any of the values listed above in the section 'Mutation Classification'. `variant_type` is either 'point' or 'indel'. Finally, any user-defined data that can be associated on a one-per-base basis may be configured during MAQGene installation and will appear here. In our case, we adapted the `6-species_conservation` field from a [0,1] conservation measure derived from 6-species alignment conservation measure derived from a phylogenetic hidden Markov model [5], downloaded from http://hgdownload.cse.ucsc.edu/goldenPath/ce6/phastCons6way/.

The column `mutant_strain` exists for ease in combining and comparing results of separate runs. Comparing runs can be very important, as it allows to subtract common background variants from two different mutant genomes. In our proof-of-concept study [3], we have in fact found that there is a substantial background variation between the published wild-type genome sequence and the sequence of the wild-type strain that we use in the lab. Rather than genome-sequencing one's own wild-type strain, we recommend to rather compare genome sequences of various mutant genomes. In the most extreme case, one may genome-sequence two independent alleles of the same mutant locus; comparison of the output data then allows to easily determine which locus is mutated in both samples, with all other variants being background variants.

To do such comparative analysis, the user may cut-and-paste lines from different runs' Excel tables. We suggest then sorting by `dna`, then `start_position`, then `mutant_strain` columns (and possibly applying a range filter for the range of interest). In this way, variants that are common between both strains will appear as consecutive lines in the merged, sorted table. Alternatively, one may use existing excel functions that are designed to compare two datasets (see excel user's manual).

6. Methods

Mutation Classification Detail

Variants may be either indel or point mutations. MAQGene has a robust classification system for point mutations, but only a very simplistic one for indels.

In order to classify point mutations according to their effects on transcription and translation, MAQGene implements the canonical rules of these processes *in silico* as follows. First, the point mutation in question is associated with all exons within a user-defined distance, chosen as the larger of the two distance thresholds for direct exon association in section ‘Mutation-to-gene association’. Second, each gene is spliced into its coding sequence. Third, exonic point mutations are located in their relevant codons and translated according to a user-provided genetic code (provided during MAQGene installation).

Nearby and overlapping exons are found by first building a *nested containment list* of exons [6]. Briefly, an NClist is a data structure of intervals that allows efficient search of all intervals overlapping the query interval. Without loss of generality, we use this structure to retrieve all exons within a certain distance of an interval defining the point mutation. The interval distance is defined for two intervals A = [a1, a2] and B = [b1, b2], as min(abs(a2-b1), abs(a1-b2)). Intuitively this is the minimum displacement necessary to make the two intervals adjoin such that there is no gap in between them and without overlapping. In all calculations, intervals are defined in boundary coordinates, so that for example in the string **`abcdefghi`**, the substring **`def`** is defined by interval [3,6], *not* [4,6]. For distance calculations, this coordinate choice is more consistent.

The *in silico* splicing and translation processes are defined as follows. For each gene (and each isoform of each gene if there are multiple isoforms), the actual DNA sequence defined by its coding exons is extracted and catenated together, with the logic adjusted appropriately for negative-strand genes (MySQL table `<species>_coding_dna`). This results in a complete coding sequence to be used later. Then, the point mutation’s position in the coding sequence is calculated as the sum of the lengths of all preceding exons plus the offset from the start of the point mutation’s containing exon. For negative strand genes, the logic is reversed appropriately. From this information, the codon in which the point mutation occurs may be determined as the nearest multiple-of-3 position before and after the point mutation interval. Then, using the genetic code as a table lookup (table `<species>_genetic_code`), the amino acid for the wild-type and mutant are determined. Using this information, exonic point mutants are sub-classified as missense, silent, premature_stop, readthrough or nonstart.

Non-exonic point mutations are either intergenic or intronic. For intergenic point mutations, no further details are produced. For intronic point mutations, another NClist query is performed to find all point mutations at an interval distance of 0 or 1 base from any exon (i.e. the first and second intronic bases next to the exon). These are classified as splice donor or acceptor, appropriately taking into account the relative genomic position of the point mutation and exon, and the strand of the gene.

Indels are treated more simplistically. Those affecting splice sites are still classified as ‘`donor`’ or ‘`acceptor`’, but exonic indels are not classified as completely as they could be. Merely, if the length (negative for deletions, positive for insertions) of the indel is divisible by 3, the indel is classified as ‘`inframe`’, otherwise ‘`frameshift`’. In particular though, an inframe indel, divisible by 3, does *not* have to occur on a codon/codon boundary. If not, it will create a new codon in the region of merging, that could possibly contain a stop, or destroy a start. For insertions, there is currently no easy way to determine the identity of the inserted sequence from MAQ output and we do not provide one either. We urge the user to manually check indels in their regions of interest to make up for this lack in MAQGene. We note that most of the indels that we have identified in our runs are single base indels.

System Design

The main index page, `index.php` uses PHP scripting to query certain directories on the host to fill in various pull-down menus and populate tables of run-specific information for ease in selecting appropriate input. It is made aware of these specific directories through Apache's httpd.conf file, in the section that exports MAQGene-specific environment variables (see 'Installing MAQGene').

During the installation of MAQGene, the script `install_annotations.sh` automates the process of transferring and filtering a GFF formatted annotations file into several SQL tables of related exon annotation information, as chosen in `config.sh`. It may be run several times, each time after editing `config.sh` to point to a different set of genomic information (GFF file plus fasta DNA sequence).

During a run, MAQGene creates and deletes a number of SQL tables representing the relationships between mutations and exons, and makes heavy use of UNIX pipes to streamline the process together.

MAQGene is modular in its treatment of core functions. The steps of read mapping, consensus generation, and variant-to-gene association are all governed by separate and independent shell scripts whose inputs and outputs may be combined with new programs as they become available. Most notably, MAQGene produces the standardized .bam and .sam file formats in addition to the .map files it uses internally. This provides the future opportunity to customize MAQGene to use a different software than MAQ for initial reads mapping.

7. References

1. H. Li, J. Ruan, and R. Durbin, *Genome Res* **18** (11), 1851 (2008).
2. D. Altshuler, V. J. Pollara, C. R. Cowles et al., *Nature* **407** (6803), 513 (2000).
3. S. Sarin, S. Prabhu, M. M. O'Meara et al., *Nat Methods* **5** (10), 865 (2008).
4. H. R. Bigelow, A. S. Wenick, A. Wong et al., *BMC Bioinformatics* **5** (1), 27 (2004).
5. A. Siepel, G. Bejerano, J. S. Pedersen et al., *Genome Res* **15** (8), 1034 (2005).
6. A. V. Alekseyenko and C. J. Lee, *Bioinformatics* **23** (11), 1386 (2007).