

Webinars On: Basic and Advance Next Generation Sequencing (NGS) data analysis (July 24th, 2020)



Variant calling, annotation and Visualization

By: RAHILA SARDAR ICMR- Senior Research Fellow Translational Bioinformatics Group, ICGEB



Genomic Variation

			Reference	1 2 3	
Single Nucleotide	Deletion	Insertion	Tandem	Insertion	1 2 5 3
Variant			Duplication	Deletion	1 3
$ \xrightarrow{- \cdots \rightarrow} $	→		$\xrightarrow{-} \xrightarrow{-}$	Inversion	1 3 2
				Copy Number Variation	1 1 1 2 3
Interspersed	Inversion	Translocation	Copy Number Variant	Tandem Duplication	1 1 2 3
Duplication		Tursiocution		Dispersed Duplication	1 2 1 3
\rightarrow		\rightarrow		Mobile Element Insertion	1 2 Mobile Element 3
———→	—— →	_ →			1
				Translocation	10 11 12 2 3



- Synonymous mutations: which do not alter amino acid sequences and are (sometimes) silent mutations.
- Non-synonymous mutations: nucleotide mutation that alters the amino acid sequence of a protein.
- Germline mutations: occur in gametes and can be passed onto offspring (every cell in the entire organism will be affected)
- Somatic mutations: occur in a single body cell and cannot be inherited (only tissues derived from mutated cell are affected)

Variant calling workflow

CGED

	C	F 1 (1)	
	Stage	Examples/explanation	File formats
ſ	Laboratory work	Experimental design	9
L	,	Enrichment (capture)	
_			
	Next-generation sequencing	Platforms include Illumina, SOLID Pacific Biosciences, other	Output: FASTQ-Sanger, FASTQ-Illumina
L		Sollo, i dente biosciences, other	SRA, EBI,ENA etc
	Quality assessment	Trimming, filtering Software: FastOC	FASTQ
			TrimGalore, Trimmomatic, PrinSeq
e			Reference: FASTA
oelir	Alignment to reference genome	Software: BWA, Bowtie2	Output: SAM/BAM
s piț		Single guidestide conjugate (SNVA)	
lysi		structural variants (e.g. indels)	Variant Call Format
Ana	Variant identification	Software: GATK, SAMTools	(VCF/BCF)
		Realignment, recalibration	
	Annotation	Comparison to public database	
	Annotation	functional consequence scores	
ſ	Visualization	Variant visualization; read depth;	
L		comparison to other samples Software: IGV, BEDTools, BigBED	
		-	
ſ	Prioritization	Discovery of relevant variants	VCF
L		SIFT, Panther	
-			
	Storage	Deposit data in ENA, SRA, dbGaP	BAM, VCF



Reference databases

Reference genome database

- NCBIRefSeq:<u>https://www.ncbi.nlm.nih.gov/r</u> efseq/
- UCSC : <u>https://genome.ucsc.edu</u>
- Ensembl: <u>www.ensembl.org</u>

SNP database

- **dbSNP** : Database of short Genetic Variants
- **dbVar** : Database of genomic structural variants
- **ClinVAR:** Genomic variations and their relationship to human health and disease
- **dbGap:** Database of genotypes and phyenotypes

Variant calling by samtools pipeline







Sequence alignment/map format (SAM) and BAM

- SAM is a common format having sequence reads and their alignment to a reference genome.
- BAM is the binary form of a SAM file.
- Aligned BAM files are available at repositories (Sequence Read Archive at NCBI, ENA at Ensembl)
- SAMTools is a software package commonly used to analyze SAM/BAM files.
- Visit http://samtools.sourceforge.net/

HEADER containing metadata (sequence dictionary, read group definitions etc) RECORDS containing structured read information (1 line per read record)





Align reads to reference (using BWA)



- 1. Index the reference (genome) sequence
- ≻ bwa index e.coli.fasta
- ➤ # The various index files are output in the CWD
- 2. Perform the alignment
- bwa aln [opts] e.coli.fasta input1.fastq > output.sai
- 3. Output results in SAM format
- bwa samse e.coli.fasta output.sai input1.fastq > output.sam {SIngle end}
- bwa sampe e.coli.fasta output.sai input1.fastq input2.fastq > output.sam {paired end}
- \$ bwa mem e.coli.fasta output.sai input1.fastq input2.fastq > ouput.sam {paired end}

Index file= e.coli..fasta.ann, e.coli.fasta.bwt, e.coli.fasta.fai,e.coli.fasta.pac, e.coli.fasta.sa *bwa sampe: single end , sampe for paired end. BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.

1... convert alignments (using SAMtools)



1. Convert SAM to BAM for sorting

samtools view -S -b output.sam > output.bam

- 2. Sort BAM for SNP calling
- samtools sort output.bam output-sorted

Alignments are both:

- compressed for long term storage and
- sorted for variant discovery. Sort alignments by leftmost coordinates, or by read name when -n is used.

2) Call SNPs (using SAMtools)



1. Index the genome assembly (again!)

samtools faidx e.coli.fasta

All we did so far (roughly) is to perform a formatconversion from BAM to VCF! **faidx:** Index reference sequence in the FASTA format or extract subsequence from indexed reference sequence. If no region is specified, faidx will index the file and create <ref.fasta>.fai on the disk. If regions are specified, the subsequences will be retrieved and printed to stdout in the FASTA format.

2. Run 'mpileup' to generate VCF format

> samtools mpileup -f e.coli.fasta output-sorted1.bam output-sorted-2.bam output_sorted-n.bam > output_raw.bcf

Usage: Samtools mpileup [options] inL.bam [in2.bam []] Input options: -6,illuminal.3+ quality is in the Illumina-1.3+ encoding -A,count-orphans do not discard anomalous read pairs -b,bam-list FILE list of input BAM filenames, one per line -B,no-BAQ disable BAQ (per-Base Alignment Quality) -C,adjust-MQ INT adjust mapping quality; recommended:50, disable:0 [0] -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -G,exclude-RG FILE exclude read groups listed in FILE -l,positions FILE skip unlisted positions (Chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -q,min-BQ INT skip abases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rff,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits unset [] rf,ingnore-NG disable read-pair overlap detection -X,customized-index use customized index files Output options:	
<pre>Input options: -6,illuminal.3+ quality is in the Illumina-1.3+ encoding -A,count-orphans do not discard anomalous read pairs -b,bam-list FILE list of input BAM filenames, one per line -B,no-BAQ disable BAQ (per-Base Alignment Quality) -C,adjust-MQ INT adjust mapping quality; recommended:50, disable:0 [0] -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, gionce existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -G, -exclude-RG FILE exclude read groups listed in FILE -l, -positions FILE skip unlisted positions (Chr pos) or regions (BED) -q,min-BQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip alignments with mapQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR [INT filter flags: skip reads with mask bits unset [] ff,excl-flags STR [INT filter flags: skip reads with mask bits set [UMMAP,SECONDARY,QCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:</pre>	
 -6, -illuminal.3+ quality is in the Illumina-1.3+ encoding -A,count-orphans do not discard anomalous read pairs -b,bam-list FILE list of input BAM filenames, one per line -no-BAQ disable BAQ (per-Base Alignment Quality) -C,adjust-MQ INT adjust mapping quality; recommended:50, disable:0 [0] -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -exclude-RG FILE exclude read groups listed in FILE min-MQ INT skip alignments with mapQ smaller than INT [0] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) -rf,incl-flags STR [INT filter flags: skip reads with mask bits unset [] ff,excl-flags disable read-pair overlap detection -X,customized-index use customized index files 	
 -A,count-orphans do not discard anomalous read pairs -b,bam-list FILE list of input BAM filenames, one per line -no-BAQ disable BAQ (per-Base Alignment Quality) -C,adjust-MQ INT adjust mapping quality; recommended:50, disable:0 [0] -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -e,exclude-RG FILE exclude read groups listed in FILE -e,min-MQ INT skip alignments with mapQ smaller than INT [0] -q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) -rf,incl-flags STR [INT filter flags: skip reads with mask bits unset [] excl-flags STR [INT filter flags: overlap detection -X,customized-index use customized index files 	
-b,bam-clst File fist of Input BAM filenames, one per fine -B,no-BAQ disable BAQ (per-Base Alignment Quality) -C,adjust-MQ INT adjust mapping quality; recommended:50, disable:0 [0] -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -G,exclude-RG FILE exclude read groups listed in FILE -l,positions FILE skip unlisted positions (chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rff,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set [UMMAP,SECONDARY,OCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:	
<pre>-C,adjust-MQ INT adjust BAQ (per Bask Acigament educity) -C,adjust-MQ INT adjust BAQ (per Bask Acigament educity) -C,adjust-MQ INT adjust mapping quality; recommended:50, disable:0 [0] -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -G,exclude-RG FILE exclude read groups listed in FILE -l,positions FILE skip unlisted positions (chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set</pre>	
 -d,max-depth INT max per-file depth; avoids excessive memory usage [8000] -E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -e,exclude-RG FILE exclude read groups listed in FILE -l,positions FILE skip unlisted positions (chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set _UNMAP,SECONDARY,QCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files 	
<pre>-E,redo-BAQ recalculate BAQ on the fly, ignore existing BQs -f,fasta-ref FILE faidx indexed reference sequence file -G,exclude-RG FILE exclude read groups listed in FILE -l,positions FILE skip unlisted positions (chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set [UMMAP,SECONDARY,QCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:</pre>	
 -T,Tasta-ref FILE TaidX indexed reference sequence file -G,exclude-RG FILE exclude read groups listed in FILE -l,positions FILE skip unlisted positions (chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR [INT required flags: skip reads with mask bits unset [] -ff,excl-flags STR [INT filter flags: skip reads with mask bits set [UMMAP, SECONDARY, QCFAIL, DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files 	
 -l,positions FILE exclude read groups triced in FILE -l,positions FILE skip unlisted positions (chr pos) or regions (BED) -q,min-MQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,excl-flags STR INT required flags: skip reads with mask bits unset [] -ff,excl-flags STR INT filter flags: skip reads with mask bits set	
<pre>-q,min-MQ INT skip alignments with mapQ smaller than INT [0] -Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set [UNMAP,SECONDARY,QCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:</pre>	
<pre>-Q,min-BQ INT skip bases with baseQ/BAQ smaller than INT [13] -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) rf,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set [UNMAP,SECONDARY,QCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:</pre>	
 -r,region REG region in which pileup is generated -R,ignore-RG ignore RG tags (one BAM = one sample) -rf,incl-flags STR INT required flags: skip reads with mask bits unset [] -ff,excl-flags STR INT filter flags: skip reads with mask bits set	
 rgnore-Ro Ignore Ro tags (one BAM = one Sample) rf,incl-flags STR INT required flags: skip reads with mask bits unset [] ff,excl-flags STR INT filter flags: skip reads with mask bits set	
ff,excl-flags STR INT filter flags: skip reads with mask bits set [UNMAP,SECONDARY,QCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:	
[UNMAP,SECONDARY,OCFAIL,DUP] -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options:	
 -x,ignore-overlaps disable read-pair overlap detection -X,customized-index use customized index files Output options: 	
-X,customized-index use customized index files Output options:	
Output options:	
-o,output FILE write output to FILE [standard output]	
-0,output-BP output base positions on reads	
-s, -output-may output mapping quality	
output-extra STR output extra read fields and read tag values	
output-sep CHAR set the separator character for tag lists [,]	
output-empty CHAR set the no value character for tag lists [*]	
-a output all positions (including zero depth)	

mpileup: Generate text pileup output for one or multiple BAM files.

3) Call SNPs (using bcftools)

3. Call SNPs Bcf to vcf



bcftools view output.var.bcf >output.var.vcf

Jutput options:	
<pre>-G,drop-genotypes -h/H,header-only/no-heade -l,compression-level [0-9 no-version -o,output-file <file> -0,output-type <b u z v> -r,regions <region> -R,regions-file <file> -t,targets [^]<region> -T,targets-file [^]<file> -threads <int></int></file></region></file></region></b u z v></file></pre>	<pre>drop individual genotype information (after subsetting if -s option set) print the header only/suppress the header in VCF output compression level: 0 uncompressed, 1 best speed, 9 best compression [-1] do not append version and command line to the header output file name [stdout] b: compressed BCF, u: uncompressed BCF, z: compressed VCF, v: uncompressed VCF [v] restrict to comma-separated list of regions restrict to regions listed in a file similar to -r but streams rather than index-jumps. Exclude regions with "^" prefix use multithreading with <int> worker threads [0]</int></pre>
Subset options: -a,trim-alt-alleles -I,no-update -s,samples [^] <list> -S,samples-file [^]<file> force-samples</file></list>	im ALT alleles not seen in the genotype fields (or their subset with -s/-S) not (re)calculate INFO fields for the subset (currently INFO/AC and INFO/AN) mma separated list of samples to include (or exclude with "^" prefix) le of samples to include (or exclude with "^" prefix) ly warn about unknown subset samples
<pre>Filter options: -c/C,min-ac/max-ac <int>[-f,apply-filters <list> -g,genotype [^]<hom het mp -i/e,include/exclude <exp -k/n,known/novel -m/M,min-alleles/max-alle -p/P,phased/exclude-phase -q/Q,min-af/max-af <float -u/U,uncalled/exclude-unc -v/V,types/exclude-types</float </exp </hom het mp </list></int></pre>	<pre>type>] minimum/maximum count for non-reference (nref), 1st alternate (alt1), least frequent (minor), most frequent (major) or sum of all but most frequent (nonmajor) alleles [nref] require at least one of the listed FILTER strings (e.g. "PASS") s> require one or more hom/het/missing genotype or, if prefixed with "^", exclude sites with hom/het/missing genotypes select/exclude sites for which the expression is true (see man page for details) select known/novel sites only (ID is not/is '.') s <int> minimum/maximum number of alleles listed in REF and ALT (e.gm2 -M2 for biallelic sites) select/exclude sites where all samples are phased :<type>] minimum/maximum frequency for non-reference (nref), 1st alternate (alt1), least frequent (minor), most frequent (major) or sum of all but most frequent (nonmajor) alleles [nref] led select/exclude sites without a called genotype ist> select/exclude comma-separated list of variant types: snps,indels,mnps,ref,bnd,other [null]</type></int></pre>

bcftools view: Applies the prior and does the actual calling and convert bcf to vcf

4) Filter SNPs



1. Filter SNPs

vcfutils.pl varFilter output.var.vcf > outputs.var-final.vcf

Options:	-Q INT	minimum RMS mapping quality for SNPs [10]
	-d INT	minimum read depth [2]
	-D INT	maximum read depth [10000000]
	-a INT	minimum number of alternate bases [2]
	-w INT	SNP within INT bp around a gap to be filtered [3]
	-W INT	window size for filtering adjacent gaps [10]
	-1 FLOAT	min P-value for strand bias (given PV4) [0.0001]
	-2 FLOAT	min P-value for baseQ bias [1e-100]
	-3 FLOAT	min P-value for mapQ bias [0]
	-4 FLOAT	min P-value for end distance bias [0.0001]
	-e FLOAT	min P-value for HWE (plus F<0) [0.0001]
	- p	print filtered variants
Note: Sor	me of the	filters rely on annotations generated by SAMtools/BCFtools.

• Note: vcfutils.pl is a perl script that helps filtering variants according to a certain set of parameters

GATK





GATK vs Samtools



Variant callers	Samtools	GATK
Algorithm	bayesian approach to call the variants	bayesian approach to call the variants
Preprocess of alignment.	No	uses picard tools
GATK specific features	No other variant callers	haplotype caller, indel realignment,UnifiedGenotype among many others
Recalibration	No	Yes using ML algorithm
Organism specific	No	Yes
Utility	samtools is equally scalable to huge data sets in spite of its much simpler framewor	GATK is clearly the more sophisticated and the more complicated

Integrative Genomics Viewer (IGV)





Variant Call Format (VCF) file summarizes variation



Column	Mandatory	Description
CHROM	Yes	Chromosome
POS	Yes	1-based position of the start of the variant
ID	Yes	Unique identifier of the variant; the dbSNP entry rs1413368 is given in our example
REF	Yes	Reference allele
ALT	Yes	A comma-separated list of alternate nonreference alleles
QUAL	Yes	Phred-scaled quality score
FILTER	Yes	Site filtering information; in our example it is PASS
INFO	Yes	A semicolon-separated list of additional information. These fields include the gene identifier GI (here the gene is NEGR1); the transcript identifier TI (here NM_173808); and the functional consequence FC (here a synonymous change, T296T).
FORMAT	No	Defines information in subsequent genotype columns; colon separated. For example, GT:AD:DP:GQ:PL:VF:GQX in our example refers to genotype (GT), allelic depths for the ref and alt alleles in the order listed (AD), approximate read depth (reads with MQ=255 or with bad mates are filtered) (DP), genotype quality (GQ), normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification (PL), variant frequency, the ratio of the sum of the called variant depth to the total depth (VF), and minimum of {genotype quality assuming variant position, genotype quality assuming nonvariant position} (GXQ).
Sample	No	Sample identifiers define the samples included in the VCF file

How to annotate and prioritize SNP?



ASSEI (т Pre Mupr FI EVm PON_DisoSK1p FunSAV CanDrA SNPedi

Tool name	Version	Principle	Training dataset
MAPP	28.6.2005	Physicochemical properties and alignment score	No training dataset
nsSNPAnalyzer	12.2.2004	Random forest	SwissProt 3,511 neutral/502 deleterious
PANTHER	1.02	Hidden Markov model and alignment score	No training dataset
PhD-SNP	2.06	Support vector machine	SwissProt 17,983 neutral/16,330 deleterious
PolyPhen-1	1.18	Expert set of empirical rules	No training dataset
PolyPhen-2	2.2.2	Naïve Bayes	SwissProt, dbSNP 7,070 neutral/5,322 deleterio
SIFT	4.0.4	Alignment score	No training dataset
SNAP	1.1.30	Neural network	SwissProt, Protein Mutant Database 40,830 neutral/39,987 deleterious

doi:10.1371/iournal.pcbi.1003440.t001

https://genomeinterpretation.org/vipDB

SNP prioritization

PredictSNP 1.0

Consensus classifier for prediction of disease-related mutations

✓ INPUT Load example Insert protein sequence in FASTA format: >HBA HUMAN MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHG KKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTP AVHASLDKFLASVSTVLTSKYR Load ✓ MUTATIONS Manual input Select positions: 1 MVLSPADKTN VKAAWGKVGA HAGEYGAEAL ERMFLSFPTT 41 KTYFPHFDLS HGSAQVKG**HG** KK<mark>V</mark>ADAL**T**NA V**A**HVDDMP**N**A 81 ISALSD<mark>I</mark>HAH KLRVDP**V**NFK LLSHCLLVT**L** A**A**HL**P**A**E**FTP 121 AVHASLDKFL ASVSTVLTSK YR Select mutations: ALL 🗸 A Ala C Cys D Asp 📃 E Glu T Phe 🗌 G Gly 📝 🗄 His 📄 I lle K Lys M Met 📄 N Asn Pro Pro 🗌 🧕 Gin 🗖 R Arg S Ser 🗐 IT Thr 📃 🛛 Val 🕅 W Trp 📝 Ұ Тут Pos Wild-type Mutations Clear 59 H Y - Tyr 0 60 G D - Asp, V - Val 0 63 V T - Thr 0 0 68 T V - Val 72 A E - Glu, V - Val 0 Clear all mutations

✓ TOOLS FOR EVALUATION

Tool name	Time demands	Expected accuracy	
PredictSNP	38 min	73.4%	
MAPP	10 min	70.7%	
PhD-SNP	38 min	71.5%	
PolyPhen-1	15 min	68.1%	
PolyPhen-2	15 min	69.2%	
SIFT	15 min	70.3%	
SNAP	30 min	67.6%	
nsSNPAnalyzer	15 min	62.9%	
PANTHER	5 min	63.5%	
E-mail (optional) :			
	Evaluate!		

PredictSNP 1.0

Consensus classifier for prediction of disease-related mutations

RESULTS							neutral		deleterious XX	% confidence	e
Annotation	Mutation	PredictSNP	MAPP	PhD-SNP	PolyPhen-1	PolyPhen-2	SIFT	SNAP	nsSNPAnalyzer	PANTHER	
Þ	A72E	74 %	70 %	58 %	67 %	87 %	66 %	77 %	65 %	71 %	*
Þ	A72V	60 %	59 %	73 %	67 %	76 %	53 %	71 %	65 %	63 %	
Þ	N79H	74 %	72 %	55 %	67 %	87 %	53 %	67 %	65 %	65 %	
₽	V97W	52 %	46 %	45 %	74 %	81 %	79 %	58 %	65 %	76 %	
Þ	L110R	87 %	88 %	88 %	74 %	81 %	79 %	62 %	63 %	68 %	
	A112T	83 %	75 %	58 %	67 %	74 %	76 %	83 %	65 %	70 %	
Þ	P115S	63 %	72 %	59 %	67 %	75 %	79 %	77 %	65 %	68 %	
▶	E117A	83 %	46 %	55 %	67 %	87 %	77 %	67 %	65 %	67 %	=
4	L126P	79 %	81 %	82 %	74 %	81 %	79 %	50 %	63 %	72 %	
Natural Disease	variant: in Qu Hemoglobin	ong Sze;cause -H disease	es alpha- thai	assemia			maj	oped from po oped from po	osition 126 in Uniprot j osition 125 in PMD AB	20006	
Þ	L126R	79 %	88 %	86 %	74 %	68 %	79 %	50 %	63 %	69 %	
Þ	S132P	61 %	51 %	77 %	59 %	43 %	79 %	67 %	63 %	55 %	
Sumr	nary table	F	Raw results								
JOB INFORM Job ID: axiwi The results w	AATION 8i0zhtvqbj5wa rill be mailed to	adshehnrdbkyk oyou once the	x5hx4kyw70v job is comple	ww.txdfuto							
LOG RECOR	IDS										
2013-05-09 1	5:32:03	PMD search f	inished.								4
2013-05-09 1	5:32:03	PMD annotati	ions are bein	g searched.							
2013-05-09 1	5:32:03	UniProt searc	h finished.								μ
2013-05-09 1	5:31:52	UniProt anno	tations are be	ing searched.							
2013-05-09 1	5:31:52	PredictSNP co	nsensus calc	ulation success	fuly finished.						
2013-05-09 1	5:31:50	PredictSNP co	nsensus calc	ulation running.							



Ensembl "Variation Table" shows SIFT and PolyPhen scores for nonsynonymous variants

Missense variants 😑

à Show All 💌 entries Filter Show/hide columns Chr: bp Alleles SIFT ID Class. Sourc Туре AA. AA Poly Phen соог 🔺 е d rs121909815 11:5248247 A/G SNP dbSNP Missense V/A 2 0.01 0.119 variant 0.007 rs121909830 11:5248247 SNP dbSNP V/G 2 0.07 A/C Missense variant rs121909815 11:5248247 A/G SNP dbSNP V/A 0.01 0.119 Missense 2 variant dbSNP V/G 0.007 rs121909830 11:5248247 A/C SNP Missense 2 0.01 variant 11:5248248 C/T/A SNP dbSNP V/L 2 0.001 rs33958358 <u>Missense</u> 0.01 variant 11:5248248 C/T/A SNP dbSNP V/M 0.271 rs33958358 Missense 2 0 variant 11:5248248 C/T/A SNP dbSNP V/L 2 0.02 0.001 rs33958358 Missense variant rs33958358 11:5248248 C/T/A SNP dbSNP Missense V/M 2 0.271 0 variant 11:5248245 G/A SNP dbSNP. H/Y 3 0.02 0.135 rs35906307 Missense variant



[back t

Research Article | Open Access

Volume 2016 | Article ID 9313746 | 5 pages | https://doi.org/10.1155/2016/9313746 PLOS ONE

In Silico Analysis of SNPs in *PARK2* and *PINK1* Genes That Potentially Cause Autosomal Recessive Parkinson Disease

GOPEN ACCESS DEPER-REVIEWED

In silico identification of genetic mutations conferring resistance to acetohydroxyacid synthase inhibitors: A case study of *Kochla scoparla*

Yan Li, Michael D. Netherland, Chaoyang Zhang, Huixiao Hong, Ping Gong 🖾

Abstract

Published: May 7, 2019 • https://doi.org/10.1371/journal.pone.0216116

rticle	Authors	Metrics	Comments	Media Coverage

Abstract
Introduction
Materials and methods
Results
Discussion
Disclaimer
Supporting information
Acknowledgments
References
Reader Comments (0)
Media Coverage (0)
Figures

Mutations that confer herbicide resistance are a primary concern for herbicide-based chemical control of invasive plants and are often under-characterized structurally and functionally. As the outcome of selection pressure, resistance mutations usually result from repeated long-term applications of herbicides with the same mode of action and are discovered through extensive field trials. Here we used acetohydroxyacid synthase (AHAS) of Kochia scoparia (KsAHAS) as an example to demonstrate that, given the sequence of a target protein, the impact of genetic mutations on ligand binding could be evaluated and resistance mutations could be identified using a biophysics-based computational approach. Briefly, the 3D structures of wild-type (WT) and mutated KsAHAS-herbicide complexes were constructed by homology modeling, docking and molecular dynamics simulation. The resistance profile of two AHAS-inhibiting herbicides, tribenuron methyl and thifensulfuron methyl, was obtained by estimating their binding affinity with 29 KsAHAS (1 WT and 28 mutated) using 6 molecular mechanical (MM) and 18 hybrid quantum mechanical/molecular mechanical (QM/MM) methods in combination with three structure sampling strategies. By comparing predicted resistance with experimentally determined resistance in the 29 biotypes of K. scoparia field populations, we identified the best method (i.e., MM-PBSA with single structure) out of all tested methods for the herbicide-KsAHAS system, which exhibited the highest accuracy (up to 100%) in discerning mutations

Yousuf Hacan Youcuf Bakbit Control And Antion in Cadherin Antion Ibrahim,² Mutaz Amin In silico analysis of a novel causative mutation in Cadherin 23

- Yousi gene identified in an Omani family with hearing loss
- Show Mohammed Nasser Al-Kindi, Mazin Jawad Al-Khabouri, Khalsa Ahmad Al-Lamki, Flavia Palombo, Tommaso Pippucci, Giovanni Romeo & Nadia Mohammed Al-Wardy 🖂
- Acad Journal of Genetic Engineering and Biotechnology 18, Article number: 8 (2020) | Cite this article 664 Accesses | Metrics

Abstract

Background

Hereditary hearing loss is a heterogeneous group of complex disorders with an overall incidence of one in every 500 newborns presented as syndromic and non-syndromic forms. Cadherin-related 23 (CDH23) is one of the listed deafness causative genes. It is found to be expressed in the stereocilia of hair cells and in the retina photoreceptor cells. Defective CDH23 have been associated mostly with prelingual severe-to-profound sensorineural hearing loss (SNHL) in either syndromic (USH1D) or nonsyndromic SNHL (DFNB12) deafness. The purpose of this study was to identify causative mutations in an Omani family diagnosed with severe-profound sensorineural hearing loss by whole exome sequencing technique and analyzing the detected variant in silico for pathogenicity using several in silico mutation prediction software. PUBLISH ABOUT BROWSE

