interactive biosoftware

SOPHiA GENETICS®

alamut BATCH 1.11

USER MANUAL

www.interactive-biosoftware.com

13/02/2019

This document and its contents are proprietary to Interactive Biosoftware, a SOPHiA GENETICS Company. They are intended solely for the contractual use of its customer in connection with the use of the product(s) described herein and for no other purpose. This document and its contents shall not be used or distributed for any other purpose and/or otherwise communicated, disclosed, or reproduced in anyway whatsoever without the prior written consent of Interactive Biosoftware. Interactive Biosoftware does not convey any license under its patent, trademark, copyright, or common-law rights nor similar rights of any third parties by this document.

The instructions in this document must be strictly and explicitly followed by qualified and properly trained personnel in order to ensure the proper and safe use of the product(s) described herein. All of the contents of this document must be fully read and understood prior to using such product(s).

© 2019 Interactive Biosoftware - a SOPHiA GENETICS Company. All rights reserved.

Table of Contents

1 Preamble – Read carefully

Alamut® Batch is a component of the Alamut® Software Suite, a set of applications dedicated to genomic variant annotation, filtration, and exploration.

Alamut® Batch is a genomic variant annotation software that does not provide recommendations for medical diagnosis. It must be used by human genetics professionals and with critical judgment. Interactive Biosoftware cannot guarantee the accuracy of information and predictions it provides.

2 About this manual

This user manual describes how to install and use Alamut® Batch version 1.11, released in Feb. 2019.

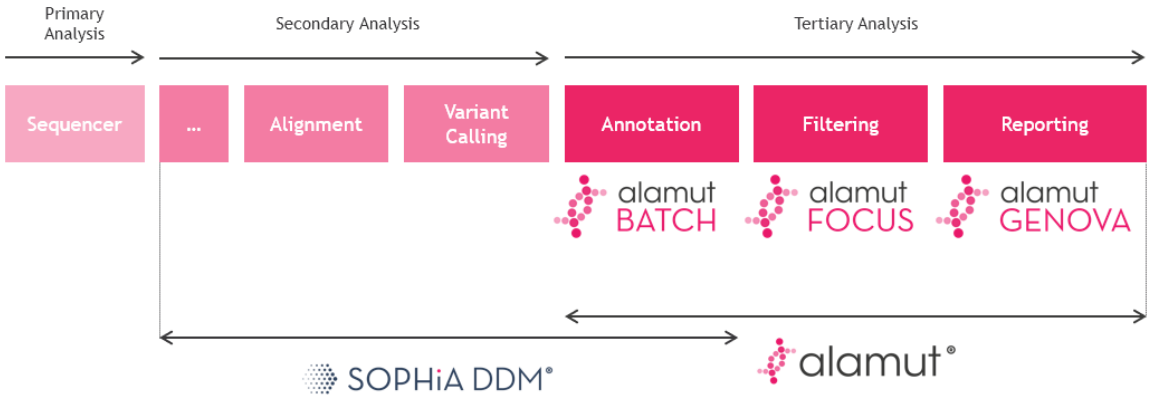
3 Product description

Alamut Batch is a high-throughput annotation engine for NGS analysis.

Designed for intensive variant analysis workflows, this software enriches raw NGS variants with multiple annotations including effects on human genes, allele frequencies, and missense and splicing predictions.

Annotations provided by Alamut Batch are similar to those available in the Alamut® Genova variant exploration software. Alamut Batch is able to annotate hundreds of thousands variants per hour.

This schematic drawing shows where Alamut Batch, Alamut Focus, and Alamut Genova take place in tertiary NGS analysis:



Alamut® Focus is an optional variant filtration companion application to Alamut Batch.

Alamut Batch can be used independently from Alamut Genova. However, results from Alamut Batch can be easily injected into Alamut Genova so as to benefit from its rich feature set, including graphical visualization.

Alamut Batch annotates variants by querying a database storing information about human genes (the Alamut database). Technically, Alamut Batch comes in two versions depending on where the gene database is located:

- The Standalone version uses a locally installed database
- The Client/Server version connects over the internet to our hosted database

Standalone version

The Standalone version of Alamut Batch provides best performance by including in a local installation all software components and the Alamut database required by the annotation process. It is most appropriate for intensive variant annotation needs such those of whole exome analyses.

Alamut Batch Standalone is a Linux command-line program.

Client/Server version

The Client/Server version of Alamut Batch connects remotely to the central Alamut database. Due to internet latency the Client/Server version is slower than the Standalone version but is

very easy to install. It is an efficient solution for moderate variant annotation needs such those of gene panels sequencing analyses.

Alamut Batch Client/Server is available as a command-line program on Windows and Linux operating systems. The software is also available with a GUI frontend on Windows.

4 System requirements

4.1 Standalone version

Alamut Batch Standalone requires the following system specifications:

- 64-bit CentOS 6.4 distribution (or other compatible Linux distribution)
- Python 2.6 or 2.7
- OpenSSL client libraries (e.g. RPM package openssl.x86_64)
- 4 GB RAM minimum
- 100 GB hard drive space
- Internet connection required for license control

4.2 Client/Server version

Alamut Batch Client/Server requires the following system specifications:

- 64-bit CentOS 6.4 distribution (or other compatible Linux distribution)
- 4 GB RAM minimum
- 100 MB hard drive space
- Internet connection required

Or:

- Windows XP, 7, 8, or 10 (32-bit or 64-bit)
- 2 GB RAM minimum
- 50 MB hard drive space
- Internet connection required

5 Release Notes – Version 1.11

Version 1.11 is a bug-fix release.

It rectifies a problem in VCF output, in case of multiallelic variants, where ',' separators in the ALAMUT_ANN INFO field could be missing.

Please note that Alamut Batch separates annotation groups with ',' for both:

1. Alternate alleles of multiallelic variants
2. Annotations on different transcripts for the same alternate allele

As an example, given the following VCF input:

```
5       37036449        .       GTATA G,GTA 7149.61     ...
```

On output the ALAMUT_ANN INFO field will have the following layout:

```
G|NIPBL|28862|Nipped-B_homolog_(Drosophila)|NM_133433.3|...
,G|NIPBL|28862|Nipped-B_homolog_(Drosophila)|NM_015384.4|...
,GTA|NIPBL|28862|Nipped-B_homolog_(Drosophila)|NM_133433.3|...
,GTA|NIPBL|28862|Nipped-B_homolog_(Drosophila)|NM_015384.4|...
```

6 Installation

Download the software from http://downloads.interactive-biosoftware.com

The downloaded file is a self-extractable archive on Windows and a tarball on Linux. Extract the contents.

6.1 Client/Server GUI frontend (Windows only)

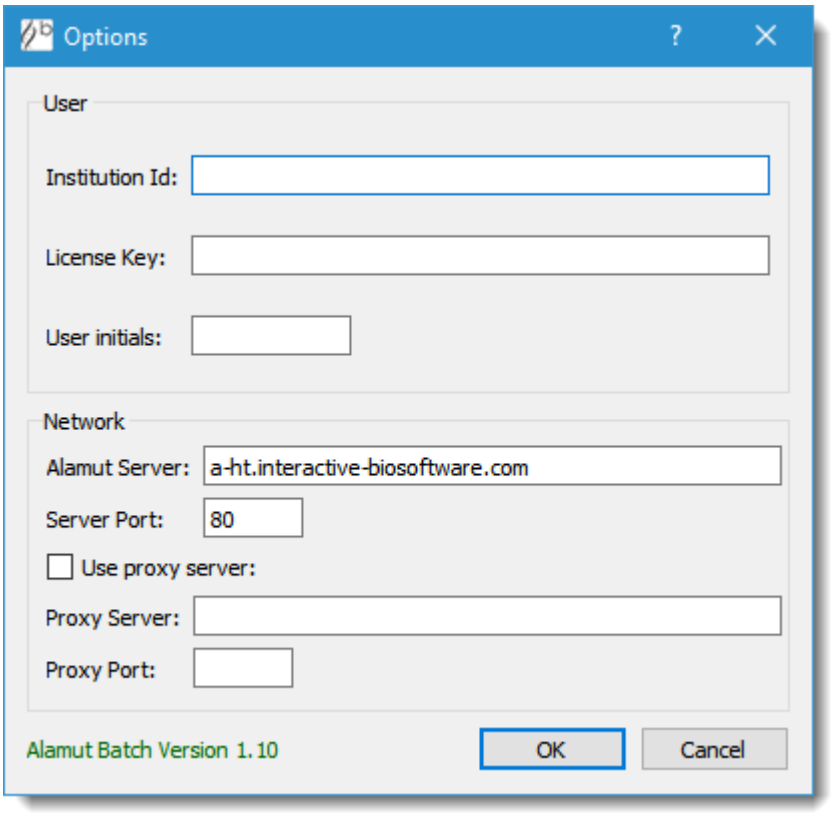
Launch the program Alamut-Batch-UI.exe

Open the Option panel and supply:

- Your Institution ID in the 'Institution' field
- Your license key in the 'Licence Key' field
- User initials as appropriate in the 'User initials' field

If your internet access is behind a proxy, you will also need to supply appropriate proxy settings.

NOTE
The Alamut Server name is 'a-ht.interactive-biosoftware.com' by default. If you are based in North America, please change the server name to 'a-ht-**na**.interactive-biosoftware.com'.



6.2 Client/Server command-line program (Windows and Linux)

Edit the `alamut-batch.ini` file and supply:

- Your Institution ID in the 'Institution' field
- Your license key in the 'Licence Key' field
- User initials as appropriate in the 'User initials' field

NOTE
The Alamut Server name (in field [Network] IBS\Server) is 'a-ht.interactive-biosoftware.com' by default.
If you are based in North America, please change the server name to 'a-ht-na.interactive-biosoftware.com'.

6.3 Standalone command-line program (Linux only)

See Section Installing Alamut Batch Standalone.

7 Variant Input file

The software takes on input a list of genomic variations, and outputs a list of annotations for each variant, when it is located on a gene available in the Alamut database.

Alamut Batch supports VCF files and tab-delimited files on input.

VCF files — This is the most common format for variant description. Alamut Batch supports VCF v4.0 and later. Alleles described as '.', '<NONREF>', '*', '<CNV>' are discarded.

Tab-delimited files — A specific tab-delimited text format can also be used for variant input. In this format each line should contain the following fields separated by tab characters:

1. Variant id (anything)
2. Chromosome (1-22, X, Y)
3. Genomic position
4. Reference nucleotide(s) (ACGT, or '-' for insertions)
5. Mutated nucleotide(s) (ACGT, or '-' for deletions)
6. Optional strand (1/+ or -1/-), used if `--strand` parameter is set to 0 (strand is related to the variant itself, not to the transcript orientation)
7. Optional transcript id, used if `--spectrans` parameter is specified
8. Optional user-defined fields (e.g. heterozygosity, number of reads, etc). These fields are not processed but merely reported as-is in the output file.

Empty lines and lines starting with a '#' character are ignored.

Example:

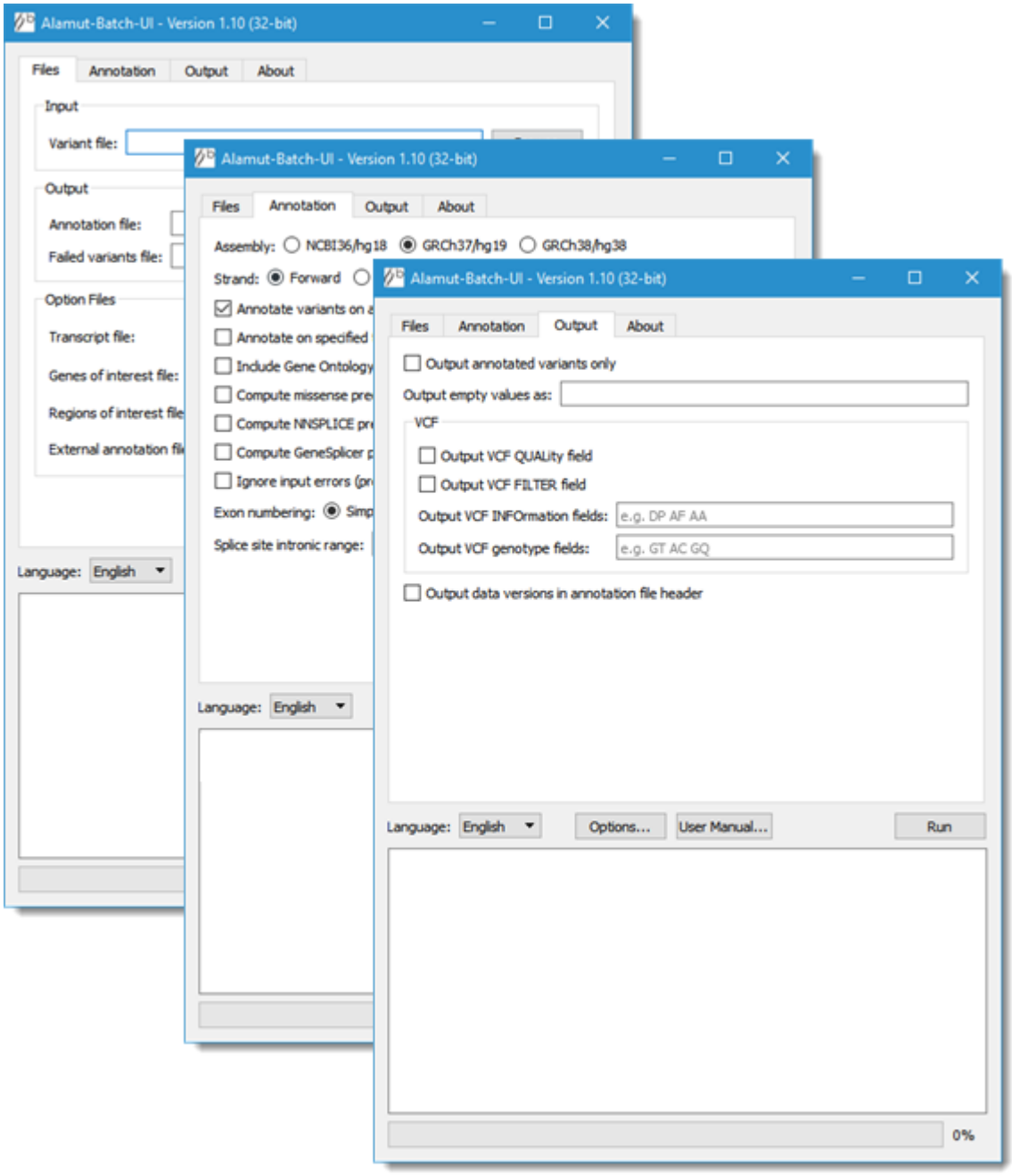
```
id00011     1     23456      T     A     42%     T>A substitution
id00022     9     876543     -     TGA   84%     TGA insertion
id00032     5     613720     AC    -     2%      AC deletion
```

8 Using Alamut Batch

8.1 GUI frontend (Windows only)

Launch the program: Alamut-Batch-UI.exe

Program options are spread over three different tabs:



Options are described in Section Software parameters.

8.2 Command-line program (Windows and Linux)

Synopsis:

```
alamut-batch
    [--help]
    [--listgenes <output file name> NCBI36|GRCh37|GRCh38]
     --in <variant file name>
     --ann <annotation file name>
     --unann <unannotated log file name>
     [--from <n>] (start annotating from the nth variant)
     [--to <n>] (annotate up to the nth variant)
     [--assbly NCBI36|GRCh37|GRCh38] (default: GRCh37)
     [--strand 1|-1|0] (default: 1; 0: per variant - not applicable to
                        VCF input)
     [--alltrans] (annotate variants on all transcripts)
     [--spectrans] (annotate variants only on specified per-variant
                    Transcripts - not applicable to VCF input)
     [--translist <transcript file name>] (annotate variants only on
                                           listed preferred transcripts)
     [--glist <gene list file name>] (list of genes of interest)
     [--roilist <ROI list BED file name>] (list of regions of interest)
     [--addGO] (add Gene Ontology annotations)
     [--nomispred] (no missense predictions; faster)
     [--donnsplice] (compute NNSPLICE predictions; slower)
     [--dogenesplicer] (compute GeneSplicer predictions; slower)
     [--ignoreInputErrors] (proceed even if input has incorrect entries)
     [--exonnums simple|custom] (default: simple)
     [--ssIntronicRange <n>] (set varLocation as 'splice site' if
                              variant is intronic and within this range)
     [--extAnnFile <external annotation file name>] (include additional
                                         annotations from external file)
     [--outputVCF] (write output in VCF format)
     [--outputannonly] (output only annotated variants
                        in annotation output)
     [--outputVCFQuality]
     [--outputVCFFilter]
     [--outputVCFInfo ID ... ID]
     [--outputVCFGenotypeData ID ... ID]
     [--outputEmptyValuesAs <value>] (e.g. NULL)
     [--outputDataVersions]
     [--proxyserver <proxy server name>]
     [--proxyport <proxy server port number>]
     [--proxyuser <proxy user login>]
     [--proxypasswd <proxy password>]
     [--tmpDir <temporary files directory>]
     [--processes <#processes>] (Standalone version only)
```

Using the `--listgenes` option puts the program in a special mode making it ouput the unsorted list of genes available in the Alamut database for the given genome assembly.

Options are described in section Software parameters.

9 Software parameters

9.1 Option list

Input/Output files	Comment	Command line
Variant file (mandatory)	Variant input file full path name (refer to Section Variant Input file for details of the file format).	`--in <variant file name>`
Annotation file (mandatory)	Annotation output file full path name	`--ann <annotation file name>`
Failed variants file (mandatory)	Output log file name. This file lists the variants that could not be annotated.	`--unann <unannotated log file name>`
Transcript file	Annotate variants on preferred transcripts listed in specified file. This is used to restrict annotation to specific transcripts of listed genes. (It does not restrict annotation to specific genes – see Genes of interest below.) See Section Transcript file format	`--translist <transcript file name>`
Genes of interest file	List of genes of interest. If this is specified, only variants mapped to the listed genes are annotated. See Section Genes of interest file format	`--glist <gene list file name>`
Regions of interest file	List of regions of interest (ROIs). A tabulated file where ROIs are described as <chromosome, start, end> (BED format). Only variants located in ROIs are annotated.	`--roilist <ROI list BED file name>`
External annotation file	List of external variant annotations to be reported in output (described in Section External annotation files)	`--extAnnFile <external annotation file name>`
Annotation parameters	**Comment**	**Command line**
Variant annotation range	Not available in the Windows GUI	`--from <n>` (start annotating from the nth variant)

		`--to <n>` (annotate up to the nth variant)
Assembly	Genome assembly: NCBI36/hg18, GRCh37/hg19, or GRCh38/hg38 (NCBI36/hg18 is still supported, but you are strongly encouraged to use the latest assembly).	`--assbly NCBI36|GRCh37|GRCh38` (default: `GRCh37`)
Strand	(Not applicable to VCF input) Variants' strand must be explicitly specified, either for the entire input file or on a per variant basis (as specified in column 6 of input file).	`--strand 1|-1|0` 1: forward strand – -1: reverse strand 0: per variant (default: 1)
Annotate variants on all transcripts	Each variant will be annotated on all available transcripts if this option is specified. Otherwise only the longest transcript is used.	`--alltrans`
Annotate on specified transcript only	(Not applicable to VCF input) Each variant will be annotated on the transcript specified on a per variant basis (as specified in column 7 of input file).	`--spectrans`
Gene Ontology annotations	Add Gene Ontology annotations	`--addGO`
Compute missense predictions	Perform Align GVGD, MAPP and SIFT predictions (Align GVGD is however always computed in the Standalone version).	`--nomispred` (cancels default behavior)
Compute NNSPLICE predictions	Perform NNSPLICE predictions	`--donnsplice`
Compute GeneSplicer predictions	Perform GeneSplicer predictions	`--dogenesplicer`
Ignore input errors	Proceed even if input has invalid entries	`--ignoreInputErrors`
Exon numbering	Simple (sequential) or custom (if available) exon numbering	`--exonnums simple|custom` (default: `simple`)
Splice site intronic range	Intronic variants located within the specified range <n> from the nearest splice site are	`--ssIntronicRange <n>`

	annotated as 'splice site' in the *varLocation* annotation field.	
Output parameters	Comment	Command line
Output in VCF format	Variant input file must be in VCF format. Available in command-line version only.	`--outputVCF`
Output annotated variants only	By default variants that cannot be annotated are also reported in the annotation output file. This option cancels this behavior.	`--outputannonly`
VCF quality score	Output VCF QUAL field (applies to VCF input files only)	`--outputVCFQuality`
VCF filter	Output VCF FILTER field (applies to VCF input files only)	`--outputVCFFilter`
VCF information	Output VCF INFO fields specified by a list of IDs, e.g. 'DP AF AA' (applies to VCF input files only)	`--outputVCFInfo ID...ID`
VCF genotype data	Output VCF genotype fields specified by a list of IDs, e.g. 'GT AC GQ' (applies to VCF input files only)	`--outputVCFGenotypeData ID...ID`
Empty values	Empty output fields are populated with specified value, e.g. 'NULL'	`--outputEmptyValuesAs <value>`
Data versions	Output versions of external databases in annotation file header	`--outputDataVersions`
Proxy parameters	Comment	Command line

Internet proxy options		`--proxyserver <proxy server name>`
		`--proxyport <proxy server port number>`
		`--proxyuser <proxy user login>`
		`--proxypasswd <proxy password>`

Misc. parameters	Comment	Command line
Temporary files directory	Command-line only If `--tmpDir` is not specified:	`--tmpDir <temporary files directory>`

	• On Linux systems this is the path in the TMPDIR environment variable or /tmp if TMPDIR is not defined • On Windows this is usually the path in the TEMP or TMP environment variable	
Multi-processing	Standalone version only	`--processes <#processes>`

9.2 Transcript file format

The input file for preferred transcripts is tab-delimited and requires at least two columns: gene name (or HGNC id), and transcript name.

Gene can be specified either by HGNC official symbol (e.g. BRCA1) or HGNC id (e.g. 1100). For convenience, if HGNC id is used, it can optionally be followed by a '/' and then a comment text string (not interpreted by the software).

Multiple transcripts per gene can be specified in additional columns.

To illustrate the format, here is a hypothetical example:

```
1100/BRCA1  ->  NM_007294.3
7765   ->  NM_000267.3
MLH1   ->  NM_000249.3  ->  NM_001167618.1
```

9.3 Genes of interest file format

The input file for genes of interest requires one gene symbol or HGNC id per line.

Gene can be specified either by HGNC official symbol (e.g. BRCA1) or HGNC id (e.g. 1100). For convenience, if HGNC id is used, it can optionally be followed by a '/' and then a comment text string (not interpreted by the software).

To illustrate the format, here is a hypothetical example:

```
1100/BRCA1
7765
MLH1
```

9.4 External annotation files

External variant annotations (e.g. variant pathogenicity status as previously established in the lab) can be integrated in the annotation output.
Variants are described using the chromosome name and genomic-level nomenclature.

Variants and annotations should be supplied in tab-delimited text files using the following format:

- First line: Tab-separated list of annotation labels (preceded by 'chrom' and 'gNomen' for clarity). For example:

```
chrom  ->  gNomen  ->  Class  ->  Freq
```

(where '->' denotes tabulation characters, and 'Class' and 'Freq' are annotation labels)

- Other lines: Tab-separated variant description and annotation values, in the same order as specified in line 1. For example:

```
chr1   ->  g.45800167G>A  ->  Likely pathogenic  ->  0.001
chr13  ->  g. 32929387T>C ->  Unknown            ->  0.005
```

Annotation labels, as supplied in first line, are reported in the first line of the output file. When input variants and externally annotated variants match, the annotation output contains corresponding annotation values.

Note that multiple external variant annotation files can be supplied (using option `--extAnnFile` multiple times).

10 Annotations

10.1 Basic annotations

Annotation	Label	Comment
Id	*Id*	Variant id as supplied in input file
Chromosome	*chrom*	
Input position	*inputPos*	Variant genomic position supplied in input file (annotated variant position can differ - see *gDNAstart* and *gDNAend*)
Input ref. sequence	*inputRef*	Reference sequence as given in input file
Input alt. sequence	*inputAlt*	Alternate sequence as given in input file
Failed annotation reason	*unannotatedReason*	Field not available if option `--outputannonly` is used
Gene symbol	*gene*	HUGO Gene Nomenclature Committee (HGNC) symbol
Gene id (HGNC)	*geneId*	HGNC id
Gene description	*geneDesc*	HGNC full gene name
GO biological process (requires `--addGO` option)	*goBioProcess*	Gene Ontology (GO) biological processes ('|'-separated list)

Annotation	Label	Comment
GO cellular component (requires `--addGO` option)	*goCellComp*	GO cellular components ('|'-separated list)
GO molecular function (requires `--addGO` option)	*goMolFunc*	GO molecular functions ('|'-separated list)
Transcript	*transcript*	e.g.: NM_000249.3
Transcript strand	*strand*	+/-
Transcript length	*transLen*	Full cDNA length
CDS length	*cdsLen*	Length of coding sequence
Protein	*protein*	e.g.: NP_000240.1
Uniprot	*Uniprot*	Uniprot accession, e.g.: P40692
Variant Type	*varType*	Possible values: • *substitution* • *deletion* • *insertion* • *duplication* • *delins*
Variant coding effect	*codingEffect*	Possible values: • *synonymous* • *missense* • *stop gain* • *in-frame* • *frameshift* • *start loss* • *stop loss*
Variant location	*varLocation*	Possible values: • *upstream* • *5'UTR* • *exon* • *intron* • *3'UTR* • *downstream* • *splice site* (see `--ssIntronicRange` option)
Genome assembly	*assembly*	
gDNA start	*gDNAstart*	Genomic variant position
gDNA end	*gDNAend*	Genomic variant position
HGVS genomic-level nomenclature	*gNomen*	e.g.: Chr3(GRCh37):g.37059009A>G

Annotation	Label	Comment
cDNA start	*cDNAstart*	cDNA variant position
cDNA end	*cDNAend*	cDNA variant position
HGVS cDNA-level nomenclature	*cNomen*	e.g.: NM_000249.3:c.803A>G
HGVS protein-level nomenclature	*pNomen*	e.g.: p.(Glu268Gly)
Alt. Protein-level nomenclature	*alt_pNomen*	Like *pNomen* except for synonymous variants, e.g.: p.(Leu123Leu)
Exon	*exon*	Nearest exon if intronic variant
Intron	*intron*	
OMIM® id	*omimId*	

10.2 Splicing predictions at nearest constitutive splice site

Annotation	Label	Comment
Distance to nearest splice site	*distNearestSS*	
Nearest splice site type	*nearestSSType*	5'/3'
WT seq. SpliceSiteFinder score	*wtSSFScore*	Predictions at nearest splice site
WT seq. MaxEntScan score	*wtMaxEntScore*	ditto
WT seq. NNSPLICE score	*wtNNSScore*	ditto
WT seq. GeneSplicer score	*wtGSScore*	ditto
Variant seq. SpliceSiteFinder score	*varSSFScore*	ditto
Variant seq. MaxEntScan score	*varMaxEntScore*	ditto
Variant seq. NNSPLICE score	*varNNSScore*	ditto
Variant seq. GeneSplicer score	*varGSScore*	ditto
Nearest splice site change	*nearestSSChange*	Average change predicted by MaxEntScan, NNSPLICE, and SSF

Annotation	Label	Comment
SPiCE probability	*SPiCEprob*	See Leman, R., et al. (2018). *Novel diagnostic tool for prediction of variant spliceogenicity derived from a set of 395 combined in silico/in vitro studies: an international collaborative effort*. Nucleic Acids Res.

10.3 Splicing predictions in variation vicinity

Annotation	Label	Comment
Splicing effect in variation vicinity	*localSpliceEffect*	Possible values: • *New Donor Site* • *New Acceptor Site* • *Cryptic Donor Strongly Activated* • *Cryptic Donor Weakly Activated* • *Cryptic Acceptor Strongly Activated* • *Cryptic Acceptor Weakly Activated* See Section Local splicing effect predictions
Genomic position of predicted new splice site or activated cryptic site	*localSS_pos*	
WT seq. MaxEntScan score used for computation of *localSpliceEffect*	*localSS_wtMaxEntScore*	
WT seq. NNSPLICE score used for computation of *localSpliceEffect*	*localSS_wtNNSScore*	
WT seq. SSF score used for computation of *localSpliceEffect*	*localSS_wtSSFScore*	
Variant seq. MaxEntScan score used for computation of *localSpliceEffect*	*localSS_varMaxEntScore*	
Variant seq. NNSPLICE score used for computation of	*localSS_varNNSScore*	

Annotation	Label	Comment
localSpliceEffect		
Variant seq. SSF score used for computation of *localSpliceEffect*	*localSS_varSSFScore*	
Genomic position of affected putative branch point	*branchPointPos*	
Change between WT and variant point scores	*branchPointChange*	Between -5 and -100

10.4 Protein domains

Annotation	Label	Comment
Protein domain 1	*proteinDomain1*	
Protein domain 2	*proteinDomain2*	
Protein domain 3	*proteinDomain3*	
Protein domain 4	*proteinDomain4*	

10.5 dbSNP

Annotation	Label	Comment
dbSNP variation	*rsId*	
dbSNP validated variation?	*rsValidated*	yes/no
dbSNP suspect variation?	*rsSuspect*	yes/no – Variant flagged as suspect by dbSNP
dbSNP validation labels	*rsValidations*	e.g.: Cluster/Frequency/1000G
dbSNP number of validation categories	*rsValidationNumber*	
dbSNP ancestral allele	*rsAncestralAllele*	
dbSNP variation average heterozygosity	*rsHeterozygosity*	
dbSNP variation clinical significance	*rsClinicalSignificance*	
dbSNP variation global Minor Allele Frequency	*rsMAF*	

Annotation	Label	Comment
dbSNP variation global minor allele	*rsMAFAllele*	
dbSNP variation sample size	*rsMAFCount*	

10.6 1000 Genomes

Annotation	Label	Comment
1000 genomes global allele frequency	*1000g_AF*	
1000 genomes allele frequency in African population	*1000g_AFR_AF*	
1000 genomes allele frequency in South Asian population	*1000g_SAS_AF*	
1000 genomes allele frequency in East Asian population	*1000g_EAS_AF*	
1000 genomes allele frequency in European population	*1000g_EUR_AF*	
1000 genomes allele frequency in American population	*1000g_AMR_AF*	

10.7 gnomAD

If the annotated variant belongs both to the Exomes and Genomes gnomAD datasets then frequency and count values provided are computed based on the sum of respective counts, and the read depth value is set to 0.

gnomAD population 3-letter codes used in gnomAD annotation labels:

all All populations

afr African

amr Latino

asj Ashkenazy Jewish

eas East Asian

sas South Asian

nfe Non-Finnish European

all All populations

fin Finnish European

oth Other populations

Annotation	Label	Comment
gnomAD alternate allele frequency	*gnomadAltFreq_all*	
	gnomadAltFreq_afr	
	gnomadAltFreq_amr	
	gnomadAltFreq_asj	
	gnomadAltFreq_eas	
	gnomadAltFreq_sas	
	gnomadAltFreq_nfe	
	gnomadAltFreq_fin	
	gnomadAltFreq_oth	
	gnomadAltFreq_popmax	Maximum Allele Frequency across populations (excluding OTH)
gnomAD alternate allele count	*gnomadAltCount_all*	
	gnomadAltCount_afr	
	gnomadAltCount_amr	
	gnomadAltCount_asj	
	gnomadAltCount_eas	
	gnomadAltCount_sas	
	gnomadAltCount_nfe	
	gnomadAltCount_fin	
	gnomadAltCount_oth	
gnomAD total allele count	*gnomadTotalCount_all*	
	gnomadTotalCount_afr	
	gnomadTotalCount_amr	
	gnomadTotalCount_asj	
	gnomadTotalCount_eas	

Annotation	Label	Comment
	gnomadTotalCount_sas	
	gnomadTotalCount_nfe	
	gnomadTotalCount_fin	
	gnomadTotalCount_oth	
gnomAD homozygous genotype frequency	*gnomadHomFreq_all*	
	gnomadHomFreq_afr	
	gnomadHomFreq_amr	
	gnomadHomFreq_asj	
	gnomadHomFreq_eas	
	gnomadHomFreq_sas	
	gnomadHomFreq_nfe	
	gnomadHomFreq_fin	
	gnomadHomFreq_oth	
gnomAD homozygous genotype count	*gnomadHomCount_all*	
	gnomadHomCount_afr	
	gnomadHomCount_amr	
	gnomadHomCount_asj	
	gnomadHomCount_eas	
	gnomadHomCount_sas	
	gnomadHomCount_nfe	
	gnomadHomCount_fin	
	gnomadHomCount_oth	
gnomAD heterozygous genotype count	*gnomadHetCount_all*	
	gnomadHetCount_afr	
	gnomadHetCount_amr	
	gnomadHetCount_asj	
	gnomadHetCount_eas	
	gnomadHetCount_sas	
	gnomadHetCount_nfe	

Annotation	Label	Comment
	gnomadHetCount_fin	
	gnomadHetCount_oth	
gnomAD hemizygous genotype count	*gnomadHemCount_all*	
	gnomadHemCount_afr	
	gnomadHemCount_amr	
	gnomadHemCount_asj	
	gnomadHemCount_eas	
	gnomadHemCount_sas	
	gnomadHemCount_nfe	
	gnomadHemCount_fin	
	gnomadHemCount_oth	
gnomAD VCF filter value	*gnomadFilter*	
gnomAD read depth	*gnomadReadDepth*	
gnomAD variant origin (Exomes, genomes, or both)	*gnomadOrigin*	Possible values: • *Exomes* • *Genomes* • *Exomes+Genomes*

10.8 ESP/EVS

Annotation	Label	Comment
ESP reference allele count in European American population	*espRefEACount*	
ESP reference allele count in African American population	*espRefAACount*	
ESP reference allele count in all populations	*espRefAllCount*	
ESP alternate allele count in European American population	*espAltEACount*	
ESP alternate allele count in African American population	*espAltAACount*	
ESP alternate allele count in all populations	*espAltAllCount*	

Annotation	Label	Comment
Minor allele frequency in European American population	*espEAMAF*	
Minor allele frequency in African American population	*espAAMAF*	
Minor allele frequency in all populations	*espAllMAF*	
Alternate allele frequency in European American population	*espEAAAF*	
Alternate allele frequency in African American population	*espAAAAF*	
Alternate allele frequency in all populations	*espAllAAF*	
Average sample read depth	*espAvgReadDepth*	

10.9 ClinVar

Annotation	Label	Comment
ClinVar ids	*clinVarIds*	'|'-separated list
ClinVar origins	*clinVarOrigins*	'|'-separated list. Possible values: germline, somatic, de novo, maternal, etc
ClinVar methods	*clinVarMethods*	'|'-separated list. Possible values: clinical testing, research, literature only, etc
ClinVar clinical significances	*clinVarClinSignifs*	'|'-separated list
ClinVar review status	*clinVarReviewStatus*	'|'-separated list – Number of stars (0-4)
ClinVar phenotypes	*clinVarPhenotypes*	'|'-separated list

10.10 COSMIC

Annotation	Label	Comment
COSMIC ids	*cosmicIds*	'|'-separated list
COSMIC tissues	*cosmicTissues*	'|'-separated list
COSMIC frequencies	*cosmicFreqs*	'|'-separated list

Annotation	Label	Comment
COSMIC sample counts	*cosmicSampleCounts*	'|'-separated list

10.11 Indel-specific

Annotation	Label	Comment
Inserted nucleotides	*insNucs*	
Deleted nucleotides	*delNucs*	

10.12 SNV-specific

Annotation	Label	Comment
Type	*substType*	transition or transversion
WT nucleotide	*wtNuc*	
Variant nucleotide	*varNuc*	
Nucleotide change	*nucChange*	
PhastCons score	*phastCons*	
phyloP	*phyloP*	

10.13 Coding SNV-specific

Annotation	Label	Comment
Wild type amino acid (1 letter)	*wtAA_1*	
Wild type amino acid (3 letters)	*wtAA_3*	
Wild type amino acid codon	*wtCodon*	
Wild type amino acid codon frequency in human genome	*wtCodonFreq*	
Variant amino acid (1 letter)	*varAA_1*	
Variant amino acid (3 letters)	*varAA_3*	
Variant codon	*varCodon*	
Variant codon frequency in human genome	*varCodonFreq*	
Amino acid position in protein	*posAA*	

10.14 Missense-specific

Annotation	Label	Comment
Number of orthologues in alignment	*nOrthos*	
Number of conserved residues in alignment	*conservedOrthos*	
Most distant species in which AA is conserved	*conservedDistSpecies*	

10.15 Amino acid properties

Annotation	Label	Comment
BLOSUM45	*BLOSUM45*	
BLOSUM62	*BLOSUM62*	
BLOSUM80	*BLOSUM80*	
Wild type amino acid composition	*wtAAcomposition*	
Variant amino acid composition	*varAAcomposition*	
Wild type amino acid polarity	*wtAApolarity*	
Variant amino acid polarity	*varAApolarity*	
Wild type amino acid volume	*wtAAvolume*	
Variant amino acid volume	*varAAvolume*	
Grantham distance	*granthamDist*	

10.16 Missense effect predictions

Annotation	Label	Comment
AlignGVGD class	*AGVGDclass*	
AlignGVGD: variation (GV)	*AGVGDgv*	
AlignGVGD: deviation (GD)	*AGVGDgd*	
SIFT prediction	*SIFTprediction*	
SIFT weight	*SIFTweight*	
SIFT median	*SIFTmedian*	

Annotation	Label	Comment
MAPP prediction	*MAPPprediction*	
MAPP p-value	*MAPPpValue*	
MAPP p-value median	*MAPPpValueMedian*	

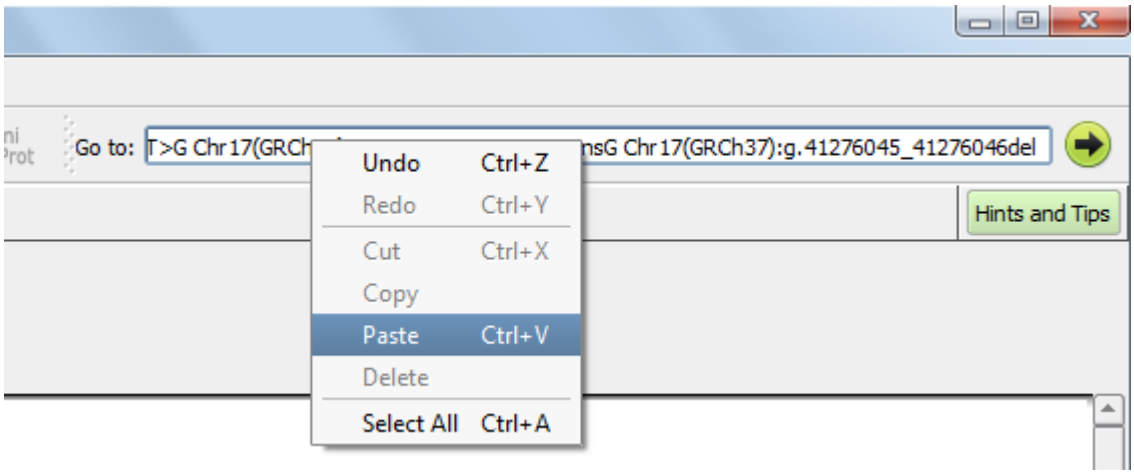
11 Viewing annotated variants in Alamut® Visual and Alamut® Genova

The genomic-level and cDNA-level HGVS descriptions generated by Alamut Batch (annotations *gNomen* and *cNomen*) can be easily copied and pasted into Alamut Visual and Alamut Genova.

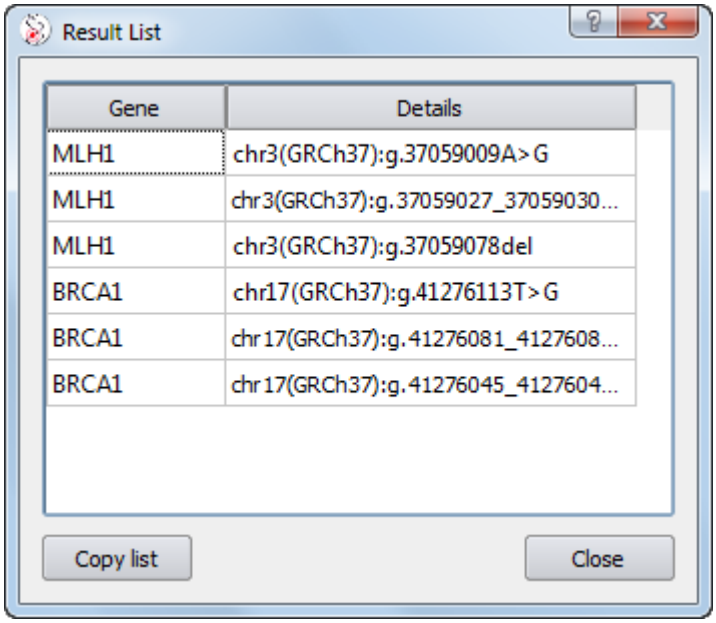
Copy a list of HGVS descriptions:

O	P	Q	R	S
gDNAend	gNomen	cDNAstart	cDNAend	cNomen
37059009	Chr3(GRCh37):g.37059009A>G	803	803	NM_000249.3:c.803A>G
37059031	Chr3(GRCh37):g.37059027_37059030dup	821	824	NM_000249.3:c.821_824dup
37059078	Chr3(GRCh37):g.37059078del	872	872	NM_000249.3:c.872del
41276113	Chr17(GRCh37):g.41276113T>G	1	1	NM_007300.3:c.1A>C
41276082	Chr17(GRCh37):g.41276081_41276082insG	32	33	NM_007300.3:c.32_33insC
41276046	Chr17(GRCh37):g.41276045_41276046del	68	69	NM_007300.3:c.68_69del

Then paste it into the Alamut Visual input field:



Variants then show up in a variant list. Double-click on an entry to jump from a variant to another:


Result List
Gene
Details
MLH1
chr3(GRCh37):g.37059009A>G
MLH1
chr3(GRCh37):g.37059027_37059030...
MLH1
chr3(GRCh37):g.37059078del
BRCA1
chr17(GRCh37):g.41276113T>G
BRCA1
chr17(GRCh37):g.41276081_4127608...
BRCA1
chr17(GRCh37):g.41276045_4127604...
Copy list
Close

12 Local splicing effect predictions

Alamut Batch interprets scores of splice site signals recognized by MaxEntScan, SpliceSiteFinder-like (SSF), and NNSPLICE in the variation vicinity in order to predict the creation of new splice sites or the activation of existing cryptic sites.

(Note that this is different from predictions at the nearest splice site, where only individual prediction scores are provided but not interpreted by Alamut Batch.)

NOTE
Prior to version 1.6.0 Alamut Batch used Human Splicing Finder (HSF) instead of SSF for the computation of these predictions.

This section describes how local splicing effect predictions are computed.

Only the MaxEntScan, SSF, and NNSPLICE scores are used in the interpretation algorithm. If option `--donnsplice` is not active, or if NNSPLICE is not installed (Standalone version), then MaxEntScan and SSF will still be used.

MaxEntScan, SSF, and NNSPLICE scores are deemed significant if they are greater than 0, 60 and 0.4 respectively.

New site creation

If, at some genomic position in the variant vicinity (excluding natural splice site positions), at least two scores are significant on the mutated sequence but not on the wild type sequence, then Alamut Batch predicts a **new splice site creation**.

Cryptic site activation

- If, at some genomic position in the variant vicinity (excluding natural splice site positions), there is at least one prediction in the wild type sequence and at least two scores are significant on the mutated sequence

and

- If scores on the mutated sequence are at least 3% greater on average than those of the wild type sequence

then Alamut Batch predicts a **cryptic splice site activation**.

If the score average increase is between 3% and 10%, activation is considered as **weak**.

If it is > 10% then it is considered as **strong**.

See detailed annotations in Section Splicing predictions in variation vicinity

13 Installing Alamut Batch Standalone

13.1 Alamut Batch Standalone components

Alamut Batch Standalone includes the following components:

1. The Alamut database. It stores all gene-related information used by the software.
2. The `alamut-batch` program. It computes variant annotations based on data provided by the database and results computed by ancillary programs.
3. Ancillary programs. These are external software tools specialized in computing missense and splicing predictions (e.g. SIFT, NNSPLICE).

The Alamut Database

The Alamut database is supplied as a single compressed file to be used as-is by the `alamut-batch` program. This file is a snapshot of the live database used by Alamut Visual and the Alamut Batch Client/Server version. Since the live Alamut database is frequently updated, monthly snapshots are provided for Alamut Batch Standalone on the Interactive Biosoftware download website.

The Alamut database is encrypted and must be queried by the `alamut-batch` program only.

The current size of the database is approximately 12 GBytes (estimated growth: 3 GBytes/year).

Software Programs

All the required programs are either Linux executables or Python 2.6 scripts. They must all be installed on the same Linux computer.

Ancillary programs include missense and splicing prediction tools that are either provided with the Alamut Batch Standalone package or can be installed separately (see Section Installation).

13.2 System Requirements

See System requirements/Standalone version.

13.3 Installation

Installing Alamut Batch Standalone requires two steps:

- Installing the `alamut_db` database
- Installing software components: `alamut-batch` and ancillary programs

Installing the alamut_db database

Go to the Alamut Batch Standalone section of http://downloads.interactive-biosoftware.com and download the latest database snapshot.

Place the donwload file anywhere in the local filesystem of the computer running Alamut Batch.

Installing Alamut Batch

Go to the Alamut Batch Standalone section of http://downloads.interactive-biosoftware.com, download the latest tarball and uncompress it.

Edit the `alamut-batch.ini` file and supply:

- Your Institution ID in the 'Institution' field
- Your license key in the 'Licence Key' field
- User initials as appropriate in the 'User' field
- The full path of the downloaded database file in the `[Database]/File` field

NOTE
The Alamut Server name (in field [Network] IBS\Server) is 'a-ht.interactive-biosoftware.com' by default. If you are based in North America, please change the server name to 'a-ht-na.interactive-biosoftware.com'.

Installing ancillary programs

All ancillary software programs must be installed in the `alamut-batch-standalone/ancillary` directory:

```
> cd ../alamut-batch-standalone/ancillary
```

SIFT

Download and uncompress:

```
> wget http://sift.jcvi.org/www/sift4.0.3b.tar.gz
> tar zxf sift4.0.3b.tar.gz
```

MAPP (optional)

Download file `MAPP.zip` from http://downloads.interactive-biosoftware.com/?Linux (Section 'Alamut Batch Standalone' > 'Other Downloads'). Unzip this file inside the `ancillary` sub-directory.

NNSPLICE (optional)

The NNSPLICE0.9 package is not officially available on the internet. If you can find a copy from colleagues unpack it in the `ancillary` directory.

Note that NNSPLICE requires glibc.i686 (GNU 32-bit libc library).

Python proxy program

A Python proxy program is needed to ease the communication between Alamut Batch and the ancillary programs: `mispred_ht.py`. It is provided in the Alamut Batch distribution and must reside in the `ancillary` directory.

Other prediction tools

Other tools are either provided within the Alamut Batch Standalone distribution (GeneSplicer and MaxEnt) or are embedded inside `alamut-batch` (Align GVGD, SSF).

13.4 Updating the alamut_db database

To update the `alamut_db` database just download the latest snapshot from http://downloads.interactive-biosoftware.com and edit the `alamut-batch.ini` file to change the `[Database]/File` field appropriately.

14 Release Notes – Previous versions

14.1 Version 1.10 (Dec. 2018)

- **New splicing prediction: SPiCE**
  SPiCE probability computed at the nearest splicing junction is output in the new *SPiCEprob* field.
  See Leman, R., et al. (2018). *Novel diagnostic tool for prediction of variant spliceogenicity derived from a set of 395 combined in silico/in vitro studies: an international collaborative effort.* Nucleic Acids Res.
- **Temporary files folder** can now be specified (command-line only, using new `--tmpDir` option)
- If the 'Include Gene Ontology annotations' (`--addGO`) option is not active then no empty GO output is produced
- Bug fix: In VCF output of multi-allelic variants, coma separators were missing in ALAMUT_ANN INFO fields. This has been fixed.
- Bug fix: gnomAD indels spanning boundaries of RefSeqGenes or LRGs could cause software crashes. This has been fixed.
- Due to the termination of our agreement with QIAGEN, HGMD options and output fields have been removed

14.2 Version 1.9.0 (Jan. 2018)

- **gnomAD**: New output field *gnomadAltFreq_popmax* reporting the maximum allele frequency across populations (excluding OTH)
- **Gene Ontology** annotations (*goBioProcess, goCellComp, goMolFunc*) are now optional. Use command-line option `--addGO` to include GO annotations in the output, or check "Include Gene Ontology annotations" in the GUI frontend.
- **HGVS nomenclature**: Recommendation to use 'm.' for mitochondrial reference sequences has been restored
- Splicing predictions from HSF (Human Splicing Finder) are no longer available in our software products.
- Bug fix: Variants spanning gene region boundaries are now discarded
- Fixed documentation related to cryptic splice site activation prediction

Alamut Batch is a genomic variant annotation software that does not provide recommendations for medical diagnosis and does not claim analytical or clinical performance related to specific diseases. This is why, starting with this version 1.9, Alamut Batch is no longer CE IVD-marked.

14.3 Version 1.8.0 (Oct. 2017)

gnomAD

Integration of the Broad Institute's gnomAD variant data. Per-population counts and frequencies of alternate alleles and genotypes from exome and genome datasets are provided. See Annotations/gnomAD.

Note that ExAC data have been withdrawn, since they are now superseded by gnomAD Exomes.

VCF output

Alamut Batch can now output annotations in VCF format. This is available in the command-line version (using option `--outputVCF`) when the input variant file is in VCF format.

With this option input VCF variant lines are enriched with annotations in a specific '`ALAMUT_ANN`' INFO field. This field contains annotation sets, one set per allele/gene/transcript combination.

Each annotation set is a '`|`'-separated list of values starting with the *allele* value and followed by the list of values described in Annotations (starting with the *gene* value):

allele|*gene*|*geneId*|*geneDesc*|...|*MAPPpValueMedian*

Inside the '`ALAMUT_ANN`' INFO field, annotation sets are comma-separated. Due to VCF format constraints all occurrences of white-space, semi-colons, commas and equals-signs in a field value are replaced by underscore '_' characters.

When some allele/gene/transcript combination cannot be annotated by Alamut Batch a specific '`ALAMUT_UNANN`' INFO field is added to the concerned VCF line. This field contains one or multiple groups of values using this format:

allele|*gene*|*transcript*|*chrom:pos/id/ref/alt*

The *chrom:pos/id/ref/alt* sub-group is also used in the failed (unannotated) variants file, where a short description of the annotation failure reason is provided.

Note that, when using `--outputVCF`, options `--outputVCFQuality`, `--outputVCFFilter`, `--outputVCFInfo`, `--outputVCFGenotypeData`, and `--outputEmptyValuesAs` are ignored.

With `--outputVCF`, two ##INFO lines are added to the VCF header:

```
##INFO=<ID=ALAMUT_ANN,...>
##INFO=<ID=ALAMUT_UNANN,...>
```

These lines describe the format used in the corresponding INFO fields.

Another header line is also added providing the Alamut Batch version:

```
##AlamutBatchVersion=1.8
```

If option `--outputDataVersions` is used then additional VCF header lines are added, such as:

```
##AlamutBatchDataSource=Genome Aggregation Database (gnomAD) Exome variants v2.0.1 (Feb. 2017
##AlamutBatchDataSource=NCBI dbSNP build 147 (April 2016) through Ensembl e87
##AlamutBatchDataSource=NHLBI GO Exome Sequencing Project ESP (ESP6500SIV2 - July 2013)
```

. . .

Other changes

- In input files of preferred transcripts and of genes of interest, gene can now be specified either by HGNC official symbol (e.g. BRCA1) or HGNC id (e.g. 1100). For convenience, if HGNC id is used, it can optionally be followed by a comment text string (not interpreted by the software). See Transcript file format and Genes of interest file format.
- A recent change in the HGVS Sequence Variant Nomenclature states that the recommendation to use 'm.' for a mitochondrial reference sequence has been retracted, and that 'g.' should be used instead. This recommendation is implemented in this version of Alamut Batch.

Bug fixes

- A missense variation of the last amino acid of a protein can induce a software crash if the transcript has at least one 3'UTR exon. This is now fixed.
- Multiallelic VCF variant lines could induce redundant output annotations (same output line repeated). This is now fixed.
- Providing an empty VCF file as input to Alamut Batch Standalone in multiprocess mode reserved a licence token but did not release it. This is now fixed.

14.4 Version 1.7.0 (June 2017)

Version 1.7.0 mainly fixes a bug introduced in version 1.6.0 by which genomic deletions analyzed on antisense transcripts could be inappropriately 3'-shifted and in rare cases could crash the program.

Descriptions of synonymous variations at protein level now follow the latest HGVS recommendation (e.g. *p.Cys123=* instead of *p.=*).

14.5 Version 1.6.0 (May 2017)

New Features

- Alamut Batch now reads and annotates input variant files incrementally (whereas it previously loaded entire variant files in memory before starting annotation).
  This considerably reduces memory (RAM) requirements, and huge variant files can now be processed without out-of-memory failures.
- New output fields: *inputRef* and *inputAlt*
  These fields respectively report the reference and alternative sequences provided in the input variant file as-is (REF and ALT fields in VCF files, or columns 4 and 5 in tab-delimited variant files).
- New output field: *cdsLen*
  This field reports the coding DNA sequence (CDS) length of coding transcripts, or 0 for non-coding transcripts.

- The algorithm dedicated to predict *de novo* splice sites or activation of cryptic sites (reported in output field *localSpliceEffect*) now uses SpliceSiteFinder-like (SSF), along with MaxEntScan and optionally NNSPLICE, instead of HSF previously.
  SSF is now preferred over HSF since it now seems clear that it provides better sensitivity/specificity performance, based on Houdayer *et al.* (2012) and our own assessment.
  Accordingly output fields *localSS_wtHSFScore* and *localSS_varHSFScore* have been changed to *localSS_wtSSFScore* and *localSS_varSSFScore* respectively.
- Likewise, computation of the *nearestSSChange* output field now uses SSF instead of HSF.
- Command-line options `--nonnsplice` and `--nogenesplicer` have been changed to `--donnsplice` and `--dogenesplicer` respectively, thus inverting their meaning. Since NNSPLICE and GeneSplicer can considerably slow down the annotation process of large input files, and since they are less accurate than MaxEntScan, SSF, and HSF, which are much faster, using any of both now requires explicit activation through the new options.

Miscellaneous

- Alamut Batch Standalone only: The `nnsplice_ht.py` ancillary file is no longer necessary and has been removed from the Standalone distribution.
- New simpler licence agreement.

Bug fixes

- *transLen* output field could be inaccurate – This is now fixed.
- *delNucs* output field could be wrong if HGVS 3' rule applied – This is now fixed.
- Due to some genome/transcript sequence discrepancies where the genome reference holds polymorphism minor alleles and the transcript holds corresponding major alleles, some genomic variants are seen as 'non-variants' when analyzed at the transcript level. Previously Alamut Batch tried to annotate these questionable entries but failed to provide fully consistent annotations. Until a sound solution to this kind of situation is found the software now merely outputs: "`Transcript NM_...: Genome/Transcript discrepancy: Alternate genomic nucleotide (%1) same as transcript nucleotide`" in the *unannotatedReason* output field.
- Nucleotide -level deletions and insertions caused a software crash if they led to a protein missense – This is now fixed.
- ExAC and 1000G variants were not always identified due to application of the HGVS 3' rule – This is now fixed.
- If VCF ALT field contained character '*' (due to deleted alleles) the software crashed. These pseudo-alleles are now safely ignored.
- Branch point sequences detected in 3' intron sequences were supposed to be reported if they were located between positions -12 and -100, but they were actually reported even in the -12..-1 range – This is now fixed.
- Protein-level effect of deletions or delins starting at 5' exon boundaries, in genes transcribed on the reverse strand, were wrongly computed – This is now fixed.

- Protein-level HGVS descriptions of a single amino acid deletion wrongly included a position range, such as p.(Met368_Met368delinsIle), whereas a single position is needed, such as p.(Met368delinsIle) – This is now fixed.
- Each line of output annotation files had a useless trailing tab character – This is now fixed.
- Command-line `--ssIntronicRange` option is now strongly checked against inappropriate values.
- Standalone version: If options `--processes` and `--outputDataVersions` were used together then data source information was output by each sub-process, yielding to messy final output – This is now fixed.
- Standalone version: The `alamut-batch.ini` configuration file could be altered by sub-processes if Alamut Batch was run in multi-process mode, thus potentially leading to misbehavior – This is now fixed.
- Standalone version: If options `--processes` and `--outputVCFInfo` or `--outputVCFGenotypeData` were used together extra licence tokens could be used inappropriately – This is now fixed.
- Warning messages "QEventLoop: Cannot be used without QApplication" no longer appear in command-line output.

14.6 Version 1.5.2 (July 2016)

Version 1.5.2 fixes a bug in 1.5.0 whereby stop loss variations could cause software crashes.

14.7 Version 1.5.1 (June 2016)

Version 1.5.1 fixes a bug in 1.5.0 whereby missense predictions were not reported.

It also fixes a bug whereby deletions in the 3'UTR were reported as coding variants.

14.8 Version 1.5.0 (June 2016)

Local splicing effects

Predictions of cryptic splice site activation or new splice site creation in variation vicinity are now provided with the site position and raw prediction scores computed at the putative site. Here are the related output fields:

localSpliceEffect (already in previous versions)	One of: *New Donor Site, New Acceptor Site, Cryptic Donor Strongly Activated, Cryptic Donor Weakly Activated, Cryptic Acceptor Strongly Activated, Cryptic Acceptor Weakly Activated*
localSS_pos	Genomic position of predicted new splice site or activated cryptic site
localSS_wtMaxEntScore	WT sequence MaxEntScan score used for cryptic splice site activation predictions

localSS_wtNNSScore	WT sequence NNSPLICE score used for cryptic splice site activation predictions
localSS_wtHSFScore	WT sequence HSF score used for cryptic splice site activation predictions
localSS_varMaxEntScore	Variant sequence MaxEntScan score used for new site creation or cryptic splice site activation predictions
localSS_varNNSScore	Variant sequence NNSPLICE score used for new site creation or cryptic splice site activation predictions
localSS_varHSFScore	Variant sequence HSF score used for new site creation or cryptic splice site activation predictions

For details please see Section Local splicing effect predictions.

Branch point predictions

Furthermore Alamut Batch now provides branch point predictions.

If a variant disrupts a putative branch point the following new output fields are populated:

branchPointPos	Genomic position of putative branch point
branchPointChange	Amount of change between wild type branch point sequence score and mutated sequence score (between -5 and -100)

Branch point sequences are detected in 3' intron sequences between positions -12 and -100, using the consensus sequence weight matrix described by Zhang (1998) and a score threshold of 60.

Coding effect nomenclature

To better stick with the Sequence Ontology, variants causing a premature stop codon in the coding sequence are now reported as 'stop gain' in the *codingEffect* output field instead of 'nonsense' which is now deprecated.

Data versions

Versions of external variant databases used for annotation can be output in the header of annotation files (option `--outputDataVersions`) and are displayed in the output window (GUI version) or standard output (command-line version).

Additional ExAC output fields

ExAC frequency data fields have been renamed and new fields have been added: per-population alternate allele counts, total allele counts, and homozygote allele counts (see Section ExAC).

14.9 Version 1.4.4 (Feb. 2016)

Release 1.4.4 adds the following new annotations:

- Full HGNC gene name (e.g. 'Cystic fibrosis transmembrane conductance regulator' for CFTR)
- Gene Ontology (GO) biological processes, cellular components and molecular functions

Also, starting with this release, coding genes with non-AUG translation initiation codons are now handled (eg. WT1).

14.10 Version 1.4.3 (Dec. 2015)

Release 1.4.3 is a bug-fix release.

1000 Genomes population frequencies were in a muddle:

- *1000G_AFR_AF* field contained EAS population frequency
- *1000G_AMR_AF* field contained SAS population frequency
- *1000G_EAS_AF* field contained AFR population frequency
- *1000G_EUR_AF* field contained AMR population frequency
- *1000G_SAS_AF* field contained EUR population frequency

Frequency values are now placed where they belong.

14.11 Version 1.4.2 (Sep. 2015)

Release 1.4.2 is a bug-fix release.

In lists of genes of interest, entries with official symbols containing lower-case letters (eg. *C2orf16*) were discarded. They are now handled properly.

14.12 Version 1.4.1 (July 2015)

Release 1.4.1 is a minor release fixing the following bugs:

- Output field *pos* was intended to report the original genomic variant position as supplied in the input file. In previous versions the value of this field could be different from the original input value (due to adjustments, mainly in case of indels).
  This has now been fixed, and the field has been renamed to *inputPos* for clarity. Whatever the computed variant position, reflected by fields *gDNAstart* and *gDNAend*, *inputPos* always holds the value supplied in the input file.
- The program crashed if the multi-process (`--processes`) option was used on a single-variant input file. This has now been fixed and multi-processing is automatically disabled if the input file contains only a few variants.

14.13 Version 1.4 (Feb. 2015)

Version 1.4 adds annotations from **ClinVar**, **COSMIC** and **ExAC**.

Note that, since a given genomic variant can match multiple ClinVar or COSMIC records, annotations from these datasets are output as lists where each item is separated by a '|' character. Lists in each field are ordered by dataset entries.

For example variant MLH1 NM_000249.2:c.793C>T has 3 entries in ClinVar, yielding the following ClinVar annotation fields:

clinVarIds	RCV000022502.22|RCV000075872.1|RCV000034802.1
clinVarOrigins	germline|germline|germline
clinVarMethods	literature only|research|research
clinVarClinSignifs	Pathogenic|Pathogenic|VUS
clinVarReviewStatus	1|3|1
clinVarPhenotypes	Lynch syndrome ii|Lynch syndrome|Not provided

14.14 Version 1.3 (Nov. 2014)

Version 1.3 adds support for the **GRCh38** (hg38) human genome assembly, and includes **1000 genomes** Phase 3 version 5 variant frequencies for five sub-populations (African, East Asian, South Asian, European, American).

14.15 Version 1.2 (Apr. 2014)

Version 1.2 introduces the following new features:

- Support for non-protein coding genes now available in the Alamut gene database
- Output annotation lines now include the original variant position provided in the input variant file. (This helps in reconciling variants between the output annotation file and other variant files, which could previously show problematic in case of variant position changes due to application of HGVS rules.)
- VCF quality, filter, information, and genotype fields are now reported in the output even for not-annotated variants
- Annotation can now be restricted to a list of preferred transcripts specified in a gene/transcripts file (`--translist` option)
- Annotation can also be restricted to a range of variants of the input file [`--from` and `--to` options] (not available in the Windows GUI)

Two other new features are specific to the Standalone version:

- Multi-process support: Annotation jobs can now be split among multiple processes on the same computer (`--processes` option)

- Access to local HGMD® Professional database installations has been changed since BIOBASE no longer provides a query API (see Using a local HGMD® Professional database installation).

14.16 Version 1.1 (Nov. 2012)

Here are the new features introduced in version 1.1:

- Integration of HGMD (the Human Gene Mutation Database) data, available to HGMD® Professional subscribers
- Integration of NHLBI GO Exome Sequencing Project (ESP) data
- Unannotated variants are now reported in the annotation output file (and in the failed variants output file as well) unless the `--outputannonly` option is specified
- If the new option `--ssIntronicRange <n>` is used, intronic variants located within the specified range <n> from the nearest splice site are annotated as 'splice site' in the varLocation annotation field
- Variants can now be filtered by regions of interest defined in a BED format file (`--roilist <ROI list BED file name>`)
- External annotations supplied in variant annotation files can now be integrated in the output (`--extAnnFile <external annotation file name>`)
- Version 1.1.3 adds three output fields reporting validation details of dbSNP entries
- Version 1.1.3 also adds three output fields reporting frequencies of ESP alternate alleles (alternate alleles not always being minor alleles)
- Version 1.1.4 adds an option to allow processing even if the input file has invalid entries
- Version 1.1.5 fixes a bug where variants affecting multiple genes where not processed on all genes
- Version 1.1.6 adds the HGMD variant sub-category output field and fixes a network proxy bug for HTTPS
- Version 1.1.7 brings improvements to the GUI version: all command-line options are now also available in the graphical interface
- With version 1.1.7 it is now possible to input variant alleles that are the same as the transcript allele (e.g. when the genome reference sequence has the minor allele of a SNP and the transcript has the major allele)
- Version 1.1.7 can query a local HGMD database installation
- Version 1.1.8 fixes a bug occurring when a gene cannot be loaded
- Version 1.1.9 features performance improvements and support for mitochondrial variants
- Version 1.1.10 fixes a bug causing software crashes on transcripts where the STOP codon is isolated in a 3'UTR exon
- Version 1.1.11 supports a wider range of VCF variant descriptions (i.e. descriptions that don't strictly comply with the format specification) and can now output VCF genotype fields of all input samples

Index

- 1 -

- A -

- C -

- D -

- E -

- G -

- H -

- I -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- V -

- W -