



HEALTH-F4-2007-200754

www.gen2phen.org

D4.3 A Validated Code-Base for Checking Mutation Nomenclature

WP4 – Genetics G2P Databases

**V1.3
Final**

Lead beneficiary: LUMC
Date: 11/2/2010
Nature: Prototype
Dissemination level: PU (Public)



 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3–Final

TABLE OF CONTENTS

DOCUMENT INFORMATION	3
DOCUMENT HISTORY	3
DEFINITIONS	4
1. INTRODUCTION.....	5
2. CHECKING SEQUENCE VARIATION NOMENCLATURE	5
3. FUTURE WORK.....	6
REFERENCES.....	7
APPENDIXES	8
APPENDIX I - Mutalyzer Help file	
APPENDIX II - Mutalyzer Quality Assurance report	
APPENDIX III - LOVD submission and Mutalyzer check of a variant	

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3–Final

Document Information

Grant Agreement Number	HEALTH-F4-2007-200754	Acronym	GEN2PHEN
Full title	Genotype-To-Phenotype Databases: A Holistic Solution		
Project URL	http://www.gen2phen.org		
EU Project officer	Frederick Marcus (Frederick.Marcus@ec.europa.eu)		


Deliverable	Number	4.3	Title	A Validated Code-Base for Checking Mutation Nomenclature
Work package	Number	4	Title	WP4 - Genetics G2P Databases

Delivery date	Contractual	December 2009	Actual	11/02/2010
Status	Version 1.3		final <input checked="" type="checkbox"/>	
Nature	Report <input type="checkbox"/> Prototype <input checked="" type="checkbox"/> Other <input type="checkbox"/>			
Dissemination Level	Public <input checked="" type="checkbox"/> Confidential <input type="checkbox"/>			

Authors (Partner)	P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)			
Responsible Author	Johan T. den Dunnen		Email	J.T.den_Dunnen@lumc.nl
	Partner	LUMC	Phone	+31-71-5269501

Document History

Name	Date	Version	Description
P. Taschner (LUMC), I. Fokkema (LUMC), J. den Dunnen (LUMC)	13-11-09	1.0	Draft
C. Bérout, G. Collod-Bérout	04-12-09	1.1	Draft
P. Taschner (LUMC)	10-12-09	1.2	Draft
P. Taschner (LUMC)	10-02-10	1.3	Final

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases	Security: PU	
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)	Version: v1.3–Final	4/8

Definitions

- Partners of the GEN2PHEN Consortium are referred to herein according to the following codes:

ULEIC – University of Leicester (UK) – Coordinator

EMBL – European Molecular Biology Laboratory (Germany) – Beneficiary

FIMIM – Fundació IMIM (Spain) – Beneficiary

LUMC – Leiden University Medical Center (Netherlands) – Beneficiary

INSERM – Institut National de la Santé et de la Recherche Médicale (France) – Beneficiary

KI – Karolinska Institutet (Sweden) – Beneficiary

FORTH – Foundation for Research and Tecnology Hellas (Greece) – Beneficiary

CEA – Commissariat à l’Energie Atomique (France) – Beneficiary

EMC – Erasmus Universitair Medisch Centrum Rotterdam (Netherlands) – Beneficiary

UH.FGC – Helsingin Yliopisto (Finland) – Beneficiary

UAVR – Universidade de Aveiro (Portugal) – Beneficiary

UWC – University of the Western Cape (South Africa) – Beneficiary

CSIR – Council of Scientific and Industrial Research (India) – Beneficiary

SIB – Swiss Institute of Bioinformatics (Switzerland) – Beneficiary

UNIMAN – The University of Manchester (UK) – Beneficiary


BIOBASE – BioBase GmbH. (Germany) – Beneficiary

deCODE – Islensk Erfoagreining EH (Iceland) – Beneficiary

PHENO – Phenosystems S.A. (Belgium) – Beneficiary

BCP – Biocomputing Platforms Ltd. Oy (Finland) – Beneficiary

- Grant Agreement:** The agreement signed between the beneficiaries and the European Commission for the undertaking of the GEN2PHEN project (HEALTH-200754).
- Project:** The sum of all activities carried out in the framework of the Grant Agreement by the Consortium.
- Work plan:** Schedule of tasks, deliverables, efforts, dates and responsibilities corresponding to the work to be carried out for the GEN2PHEN project, as specified in Annex I to the Grant Agreement.
- Consortium:** The GEN2PHEN Consortium, conformed by the above-mentioned legal entities.
- Consortium agreement:** agreement concluded amongst GEN2PHEN participants for the implementation of the Grant Agreement. Such an agreement shall not affect the parties’ obligations to the Community and/or to one another arising from the Grant Agreement.

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3–Final

1. INTRODUCTION


DNA sequence variation information is stored in two types of databases: a) comprehensive (genomic) databases, which contain information about all genes (e.g. Online Mendelian Inheritance in Man (OMIM, 1), Single Nucleotide Polymorphism database (dbSNP, 2)) and b) Locus-specific (or specialized) databases (LSDBs) covering one or more specific genes. The latter have been developed by curators with different interests, leading to patient-centered, sequence variation-centered, disease-centered, and protein-centered databases. For efficient information retrieval and data exchange, variants need to be described using unambiguous terminology. The Human Genome Variation Society is propagating their standard sequence variation nomenclature to realize this (See <http://www.hgvs.org/mutnomen>, 3).

Curating sequence variant descriptions is one problem encountered by LSDB curators. Since Work package 4 “Genetics G2P Databases” is focussed on creating solutions, GEN2PHEN partners INSERM and LUMC are providing a solution for sequence variation nomenclature checking in combination with their ‘LSDB-in-a-box’ software: the Universal Mutation Database (UMD - <http://www.umd.be/>, 4) and the Leiden Open-source Variation Database (LOVD-<http://www.lovd.nl>, 5).

2. CHECKING SEQUENCE VARIATION NOMENCLATURE

In this deliverable, we describe the software components used by UMD and LOVD to check the nomenclature of sequence variants. Both systems support the existing HGVS standard.

LOVD (developed by partner LUMC) uses a modular design to enhance its basic functionality. The database administrator and database manager can select the appropriate modules for a specific function. For automated mutation nomenclature checks, the Mutalyzer module needs to be activated by the database administrator and database manager. This module connects LOVD to the web-based Mutalyzer sequence variant nomenclature checker v. 1.0.4 (6). Mutalyzer is freely accessible at <http://www.mutalyzer.nl> via a web browser interface, but also allows programmatic access. Mutalyzer has been developed to describe any sequence variant (regardless of the organism) relative to a Genbank sequence record using the HGVS standard sequence variation nomenclature rules (See Appendix I, Mutalyzer Help). The program is recommended for nomenclature checks by editors of the journal Human Mutation (7) and regularly used by the genetics community. In practice, Mutalyzer works fine with most well-annotated Genbank reference sequence records to check and correct simple changes (See Appendix II Quality Assurance report Mutalyzer). For more complex situations, further development of the HGVS nomenclature guidelines and their implementation in Mutalyzer is needed. The advantage of using Mutalyzer via an LOVD module is that LOVD developers do not have to understand and implement the increasingly complex HGVS guidelines. During submission of new variants, LOVD allows the submitter to send the reference sequence accession number and the variant information to Mutalyzer. The Mutalyzer result is subsequently displayed in a separate window

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3–Final

(Appendix III, Fig. 1). Submitters are able to correct and check their entries before submission to LOVD.

UMD (developed by partner INSERM) integrates in its data model three specific tables dedicated to the gene of interest: “genetic code”, “cDNA reference sequence” and “non-coding reference sequences”. This specific structure is the foundation used by the UMD nomenclature module, which automatically named mutations according to the HGVS nomenclature (for more details see UMD December 2009 manual, pages 16 to 33). This generic algorithm is thus adapted to any gene (only the reference sequence is imported during the LSDB creation). Users therefore do not have to understand and implement the increasingly complex HGVS guidelines. As the HGVS mutation nomenclature system is regularly updated, the generic UMD nomenclature module is also regularly updated and an automatic update process is performed on all data from each UMD-LSDB.


The correct use of the HGVS sequence variant nomenclature plays an important role in the standardization and exchange of data between LSDBs and genomic databases. Information about the use of the sequence variant nomenclature checker software can be found in the versions of the UMD and LOVD manuals (available from the UMD website (<http://www.umd.be/>) and the LOVD website (<http://www.lovd.nl>)).

3. FUTURE WORK

To check all sequence variants in a database, curators need to download the data first and submit them manually using the batch checker option of Mutalyzer. On-going development of LOVD and Mutalyzer webservices is likely to result in an automatic check of all variant descriptions in the near future.


Currently, most LSDBs describe variants using RefSeq reference sequences in GenBank format (8). This format has been implemented in Mutalyzer. To support the need for stable reference sequences in a diagnostic setting, the new Locus Reference Genomic (LRG) has been developed (See deliverable 3.3). Since LRGs will be produced in XML format only, Mutalyzer, the LOVD reference sequence parser, and the UMD reference sequence import tool will have to be adapted to use LRGs in the near future:

Additional changes may be necessary when the standard HGVS sequence variation nomenclature is updated. To support the use of newly released functionality, regular updates of the ‘LSDB-in-a-box’ user manuals will be available from the UMD and LOVD websites.

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3–Final

References

1. Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res* 33:D514–D517.
2. Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K. 2001. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res* 29:308–311.
3. Den Dunnen JT, Antonarakis SE. 2000. Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion. *Hum Mutat* 15:7-12.
4. Beroud C, et al. 2005. UMD (universal mutation database). *Hum Mutat* 26:184-191.
5. Fokkema IF, den Dunnen JT, & Taschner PE. 2005. LOVD: easy creation of a locus-specific sequence variation database using an "LSDB-in-a-Box" approach. *Hum Mutat* 26:63-68.
6. Wildeman M, van Ophuizen E, den Dunnen JT, Taschner PE. 2008. Improving sequence variation descriptions in locus-specific mutation databases and the literature using the MUTation AnaLYZER (MUTALYZER) mutation nomenclature checker. *Hum Mutat* 29:6-13.
7. Human Mutation author instructions (See <http://www3.interscience.wiley.com/journal/38515/home/ForAuthors.html>)
8. Pruitt KD, Tatusova, T, Maglott DR. 2007. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins *Nucleic Acids Res* 35(Database issue):D61-D65.

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3–Final

APPENDIXES

The Mutalyzer Help file is provided as Appendix I.


The Mutalyzer Quality Assurance report is provided as Appendix II.

The LOVD submission and Mutalyzer check of a variant are shown as Appendix III.

The UMD mutation nomenclature system is described in the UMD manual (version December

2009, available for download from the UMD website (<http://www.umd.be/>). The LOVD

Mutalyzer module (v.0.7 included in LOVD v.2.0-24) is available for download from the LOVD website (<http://www.lovd.nl>).

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3-Final

Appendix I

Mutalyzer Help

About Mutalyzer

Welcome to the sequence variant nomenclature check of the Mutalyzer project. The nomenclature of sequence variations on this site is determined according to the international standards recommended by the Human Genome Sequence Variation Society (HGVS) (For an overview, visit <http://www.genomic.unimelb.edu.au/mdi/mutnomen/index.html>).

This tool is designed to check the correct nomenclature to a user-specified sequence variation, to encourage the use of proper nomenclature in publications and reduce redundancy in sequence variation databases. In principle, Mutalyzer can also be used to check descriptions of sequence variants detected in other organisms, provided that the user applies the HGVS nomenclature guidelines. Mutalyzer is described in: Wildeman M, van Ophuizen E, den Dunnen JT, Taschner PE. Improving sequence variant descriptions in mutation databases and literature using the MUTALYZER sequence variation nomenclature checker. *Hum Mutat* 29:6-13 (2008) [PMID: [18000842](https://pubmed.ncbi.nlm.nih.gov/18000842/)].

Brief description of its functionality: Mutalyzer retrieves a Genbank record, parses its annotation, combines that with the user-defined sequence variation to check the existence of the specified positions and residues, followed by application of the HGVS sequence variation nomenclature guidelines to generate unambiguous sequence variation descriptions.

Mutalyzer Name Generator Help

Reference

This field accepts either the *GenBank accession number* with version number or the *GenInfo identifier* (gi) of the reference sequence (Example: NM_003002 or 4506864). The letters “gi” need not be included in this field. Using an accession number without version number automatically selects the most recent version of that record. The gi allows for the selection of any record in *GenBank*. In case of outdated versions, Mutalyzer will issue a warning. **We strongly recommend the use of *Genbank* records containing genomic sequences with proper annotation for optimal use of Mutalyzer's capabilities to generate descriptions for all transcripts and protein isoforms affected by the sequence variation.** All modules, except the SNP converter, accept records containing annotations for multiple genes.

Please note that Mutalyzer needs a well-annotated genomic reference sequence to apply the following HGVS Coding DNA numbering rules: Mutalyzer also accepts its own UD identifiers, which are returned by Mutalyzer during upload of a user-defined GenBank file. See GenBank uploader below for more information.

Sequence Type

Select the appropriate sequence type for your sequence variation and reference file. Specific restrictions are applied to the selected record for each sequence type. Residue numbering will also depend on the selected type. There are five possible types:

Genomic

The *Genomic* sequence type uses genomic records. The value 1 is assigned to the first base in the record and all bases are counted from there. In the output, genomic numbering is indicated by the g. notation preceding the position number(s). All *GenBank* records with 'DNA' in the first line will be accepted. Please note that well annotated genomic sequence records containing sections marked 'mRNA' and 'CDS' can be used to check descriptions on cDNA and protein level. In the output, cDNA and protein numbering will be indicated by the c. and p. notations, respectively, which precede the position number(s).

Coding DNA (cDNA)

The *Coding DNA* or *cDNA* sequence type uses either *GenBank* DNA records with a CDS entry or mRNA records. In both cases, the value 1 is assigned to the A of the ATG start codon and all the exonic bases between start and stop are counted normally.

The following records will be accepted:

Genbank records with 'DNA' in the first line, containing a section marked 'CDS' or 'mRNA'.

Records with 'mRNA' in the first line, provided that no intronic bases are involved in the mutation.

Please note that Mutalyzer needs genomic HGVS Coding DNA numbering rules:

Exonic bases upstream of (i.e. before) the ATG are numbered -1, -2, -3 and so on.

Exonic bases downstream of (i.e. beyond) the stop codon are numbered *1, *2, *3 and so on.

Intronic bases are numbered $x+1, x+2, x+3 \dots y-3, y-2, y-1$ where x is the value of the last exonic base upstream of the intron and y is the value of the first exonic base downstream of the intron.

Intronic position numbers are always counted from the closest exonic base. In case of a tie, the upstream base is used. In the output, cDNA numbering is indicated by the c. notation preceding the position number(s).

Special cases:

5' UTR split over two or more exons: Intronic bases are numbered $-x+1, -x+2, -x+3 \dots -y-3, -y-2, -y-1$ where $-x$ is the value of the last exonic base upstream of the intron and $-y$ is the value of the first exonic base downstream of the intron.

3' UTR split over two or more exons: Intronic bases are numbered $*x+1, *x+2, *x+3 \dots *y-3, *y-2, *y-1$ where $*x$ is the value of the last exonic base upstream of the intron and $*y$ is the value of the first exonic base downstream of the intron.

RNA

The *RNA* sequence type uses only *GenBank* mRNA records. The value 1 is assigned to the first base in the record and from there all bases are counted normally. All records with 'mRNA' in the first line will be accepted. In the output, RNA numbering is indicated by the r. notation preceding the position number(s). These records do not allow checks of intronic sequence variations.

Mitochondrial DNA (mtDNA)

The *Mitochondrial DNA (mtDNA)* sequence type uses genomic records. The value 1 is assigned to the first base in the record and from there all bases are counted normally. Only *GenBank* records with 'DNA' in the first line and ' /organelle="mitochondrion" ' in the body will be accepted.

Protein

The *Protein* sequence type uses *GenBank* protein records. The value 1 is assigned to the first amino acid in the record and from there all amino acids are counted normally. Old or New Sequence input can be given in one-letter or three-letter code, to be specified with the *radio buttons* appearing after sequence type selection. All records with 'protein' in the first line will be accepted.

EST

The *EST* sequence type uses *GenBank EST* records. The value 1 is assigned to the first base in the record and from there all bases are counted normally. Sequence variation descriptions based on EST sequences lack the c. notation to indicate that only part of the coding sequence may be present. All records with 'EST' in the first line will be accepted. These records do not allow checks of intronic

sequence variations.

Gene Symbol and Variant

The Gene Symbol is optional, when sequence variations are indicated using genomic positions. A Gene Symbol identical to that in the annotation of the Genbank record is required, when the annotation of the Reference Sequence refers to multiple (overlapping) genes AND sequence variations are indicated using cDNA position numbering. All annotated genes, transcript variants and protein isoforms annotated in the Genbank record are shown in a legend at the bottom of the output page.

The Transcript Variant and Protein Isoform field is always optional. It allows the user to indicate positions relative to the alternative transcript or protein isoform specified. When left empty, Mutalyzer will generate the description based on the longest annotated transcript of the specified gene. Mutalyzer recognizes two types of variant or isoform designations:

- A positive integer referring to the order of the transcript variants and protein isoforms in the annotation of the reference sequence record, e.g. **1, 2, 3, ...**
- The exact identifier following the underscore behind the Gene symbol in the legend, e.g. v002 for a transcript variant or i002 for a protein isoform. These exact identifiers are generated by parsing the gene product or locus tag information in the annotation of the reference sequence record.

Start and End Position

The start position is the positional value of the most upstream base or amino acid in the reference sequence affected by the mutation. The end position is the positional value of the most downstream base or amino acid in the reference sequence affected by the mutation. For single-base substitutions, the Start Position and End Position will be the same and that value must be entered into the two fields. These values should be a positive integer (whole number) for all sequence types, except *Coding DNA*. For *Coding DNA*, these values can be:

A positive integer for exonic positions in the coding region, e.g. **1, 2, 3, ...**

A negative integer for exonic positions in the 5' UTR, e.g. **-1, -2, -3, ...**

A positive integer preceded by an * for exonic positions in the 3' UTR, e.g. ***1, *2, *3, ...**

One of the previous values followed by +(integer) or -(integer) for intronic positions, e.g. **42+5, -16-3, *133-7, ...**

One of the first three cases above followed by +? or -? for unknown intronic positions. This notation is only allowed for deletions.

Start and end positions in Coding DNA may not exceed those of the transcript annotated in a genomic reference sequence record.

Mutation Type

Select the type of mutation that applies to your query. The selected mutation type determines which additional input will be required for the nomenclature check. Depending on the mutation type, input values may be restricted. Seven mutation types are supported :

Substitution

A substitution is the replacement of a single nucleotide or amino acid by another. A substitution involving multiple residues is classified as an indel. The start and end position should be identical. Old Sequence and New Sequence are both obligatory fields and must be non-identical.

Deletion

A deletion is the removal of one or more nucleotides or amino acids without replacement. New Sequence must remain empty. Old Sequence can be filled in to check the start and end positions (Optional). Please note that the start and end positions should be equal when only one nucleotide or amino acid is deleted.

Insertion

An insertion is the addition of one or more nucleotides or amino acids without removing any previously existing ones. The start and end positions (the locations between which the insertion occurs) should differ by exactly one. New Sequence must be filled in, either with the actual new sequence or with the length of the new sequence, provided it exceeds 5. If the length of the new sequence exceeds 5, the entire new sequence can optionally be included in the Comment field, to be displayed as clarification along with the official sequence variation nomenclature.

For protein sequence insertions only: The field Old Sequence must be filled in containing the amino acids at the start and end position. The preferred style for this is AA1_AA2 (in three-letter code), e.g. **Pro_Arg**, although **PR**, **P_R** or **ProArg** would also work.

Duplication

Duplication is the addition of one or more nucleotides or amino acids identical to the sequence from the specified start position to the specified end position, at the end position. Old Sequence must remain empty. New Sequence can be filled in to check the start and end positions (Optional).

Insertion/Deletion (indel)

An indel is the removal of one or more bases or amino acids, combined with the addition of one or more bases or amino acids. In the case a single residue is deleted and inserted (the length of both Old Sequence and New Sequence is 1), the mutation should be described as a substitution, not an indel. Start and end position define the boundaries of the deletion in the original sequence. Old Sequence contains the deleted sequence. New Sequence contains the inserted sequence.

Inversion (nucleotide sequences only)

An inversion is a sequence of two or more bases in the same place, but in inverted order. Start and end position must be non-identical New Sequence and Old Sequence must remain empty

Old and New Sequence

The presence or absence of these fields depends on the selected Mutation Type. These fields should be used:

- to enter the original amino acid or nucleotide residue(s) present in the reference sequence (Old Sequence).
- to enter the amino acid(s) or nucleotide residue(s) introduced by the change (New Sequence). In case of insertions the length of the sequence can be entered in this field.

Comment

This field accepts any comment you might wish to add to the output, like the complete sequence of a long insert. It is never an essential field and its contents are not processed by Mutalyzer.

Output

Mutalyzer has been designed to issue warnings, when correcting entries, encountering

inconsistencies, incomplete sequences or annotation, or identifying variations with potential effects on splicing before presenting the results of the analysis. Errors will be generated when the entries can not be processed properly (see below for more information).
The sequence variation description will always be in the format:

<Accession Number>.<version number>:<sequence type>.<mutation>
(Examples: NM_003002.1:c.5delC or AL449423.14:g.61866_85191del)

or

<Accession Number>.<version number><(Gene Symbol)>:<sequence type>.<mutation>

In the latter case, the gene symbol may be followed by transcript variant or protein isoform numbers (e.g., _v001 or _i001, respectively).

Example: the fictitious sequence variation AL449423.14:g.61866_85191del corresponds with the following changes in transcript variants and protein isoforms:

AL449423.14(CDKN2A_v001):c.-271_234del

AL449423.14(CDKN2A_v002):c.5_400del

AL449423.14(CDKN2A_v003):c.1_*3352del

and

CAH70600.1(CDKN2A_i001):p.Met1?

CAH70601.1(CDKN2A_i002):p.Gly2AspfsX41

CAH70599.1(CDKN2A_i003):p.Met1?

From the example “CAD55702.1:p.Pro2Arg (missense mutation)”, you can conclude that the protein in version 1 of the record CAD55702 has a mutation denoted as Pro2Arg (which signifies an arginine substituted for a proline at position 2).

Please note the following:

- Sequence variation descriptions using genomic references in combination with Sequence Type "Coding DNA" will result in the use of reverse complement for genes transcribed in the opposite orientation.

- Genbank Identifiers are always converted to Genbank Accession Numbers, which are automatically retrieved from the annotation based on the selected Sequence Type. Example: 4506864:c.5del will be converted into NM_003002.1:c.5delC

Links to external sites

For single sequence variation checks, Mutalyzer will use the accession number provided to obtain information from external sources. When successful, the output will include links allowing the user to retrieve information about the specified gene/sequence using the UCSC Human Genome Browser or to map the sequence to the latest genome sequence.

Using the UCSC Human Genome Browser link

The accession number of the Genbank Reference sequence file is used to retrieve the position of the sequence in the UCSC Human Genome Browser. When successful, the result may be a link to Known genes and/or RefSeq Genes.

Mapping the gene/sequence to the latest genome sequence.

The accession number of the Genbank Reference sequence file is used to retrieve the chromosomal positions of the sequence in the latest build of the genome sequence. When successful, the complete genomic sequence of the corresponding region is shown with exons in bold uppercase. In addition,

links to all SNPs present in the region are highlighted in red and summarized at the bottom of the page.

Mutalyzer Name Checker Help

Users can check if a given mutation was correctly named. If the Name Checker recognizes the input, it will try to regenerate the name according to the HGVS sequence variation nomenclature guidelines. The output will be identical to that of the Name Generator.

Examples:

AB026906:c.3_4insG

AB026906:c.[1del;4G>T]

AB026906:c.[1del;6_7insAL449423.14(CDKN2A):c.[1_10;5del]]

AB026906:c.[1del;6_7insAL449423.14(CDKN2A):c.[1_10;5del]]

UD_NC_000011_22816:c.IVS2+32del

Mutalyzer SNP Converter Help

The SNP Converter will try to convert a dbSNP ID into a sequence variation description according to HGVS sequence variation guidelines using the Genbank record specified by the Accession Number by connecting to the UCSC Human Genome Browser. At the moment, the SNP converter only accepts Genbank Accession Numbers without version number. Successful SNP conversion strongly depends on the mapping of the reference sequence provided to reference sequences linked to SNPs in the UCSC Human Genome Browser. The Converter cannot be used with UD-identifiers, since these do not occur in other databases.

If the GenBank Record covers the chromosomal position of an [A/T] SNP, and contains an 'A' according to the GenBank Reference File, the sequence variation generated will be 'A>T'. The output will be identical to that of the Name Generator.

Example:

SNP Accession number: rs9919552

Nucleotide Accession number: NM_003002

Mutalyzer Batch Checker Help

The Batch checker has specifically been designed to provide locus-specific database curators the possibility to check the sequence variations in their database in one go. Although the input and output format differs from that of the Name Generator and Name Checker, the underlying analysis is the same. Users can upload a tab-delimited text file (file name **without** spaces!) with the sequence variations to be checked ([Example](#), please see the [FAQ](#) list for detailed information about the creation of a batch checker file). The first line should contain the header as shown in the example. The file may contain any combination of reference sequences and sequence types for different genes. The gene symbol field may be left empty, when it is not present in the GenBank reference sequence record annotation, but we strongly suggest to update any GenBank record following [these instructions](#).

Example:

AccNo	Genesymbol	Mutation
AB026906.1	SDHD	AB026906.1:g.7872G>T
NM_003002.1		c.3_4insG
AL449423.14	CDKN2A_v002	c.5_400del

A message containing a link to the results will be sent to the e-mail address specified (use lower case only!), when the analysis is finished, but Mutalyzer's progress can be followed in the browser window also. Performance depends on the server load and the number of reference sequence records to download from NCBI. The program will process approximately 100 variations per minute, when using a single reference sequence record.

GenBank Uploader Help

Users can upload their own reference sequence file in [GenBank Flat file format](#), retrieve the genomic sequence of a gene with its flanking regions, or specify a chromosomal range for use as a reference sequence. Mutalyzer checks whether the file is in valid GenBank Flat file format. If so, Mutalyzer stores the file on the server and returns a unique number the *UD identifier* that can be used with all different forms of the Mutalyzer Sequence Variation Nomenclature Checker. This option allows users to use reference files, which are not present in GenBank, or add information about alternative transcripts or proteins or additional genes contained within or derived from the reference sequence to an existing GenBank file. Users are encouraged to limit their use of this option by submitting annotation updates and corrections of existing GenBank files following [these instructions](#).

Uploader options:

My Genbank file is a local file

Browse to locate your Genbank Flat file with a .gb extension and press the submit button. NB. The file name should not contain spaces!

I have an URL on the Internet where my GenBank file can be found

Enter the URL, where the Genbank Flat file with a .gb extension can be found and press the submit button. NB. The file name should not contain spaces!

I want to fetch the reference genome for a (HGNC) gene symbol

This option retrieves the latest version of that part of the chromosomal reference sequence, which is annotated for this gene.

The organism name should not contain any spaces (e.g., use homo_sapiens, human or man)

Input:

Please enter a Genesymbol and specify the flanks

Genesymbol

Organism

of 5'flank nucleotides

of 3'flank nucleotides

I want to fetch a range of a chromosome

Use of NC_accession numbers without version number will result in retrieval of the latest version.

Input:

Please enter a NC accession number and the range of the chromosome you want to upload.

Chromosome Accession Number

Start Position

Stop Position

Mutalyzer output for all options:

Output: Your GenRecord was uploaded succesfully. You now can use mutalyzer with the following accession number as reference: UD_NC_000011_22816

Using Mutalyzer with sequences from other organisms

Mutalyzer can process Genbank reference files from other organisms than man and will apply the appropriate coding table to translate an open reading frame into a protein sequence. Please note that all variants will be described according to the Human Genome Sequence Variation Society (HGVS) sequence variation nomenclature guidelines. When trying to retrieve genomic reference sequences using gene symbols with the Genbank uploader or when specifying a particular gene in a genomic reference sequence, the gene symbol should be similar to that used in the (genome) sequence annotation. This means that alternative or recently updated gene symbols may not work.

Errors and feature requests

Any error message gives an indication of the problem encountered and replicates the input of the user. Most errors occurring after mistyping should be easy to understand and can be corrected immediately by altering the data in the field specified. Any LineNr in an error message has nothing to do with the reference sequence, but is included for optimal Mutalyzer problem solving and


maintenance.

Some errors might be more cryptic depending on the underlying problem. In these cases, Mutalyzer should advise you to contact us when the error persists. You can use Mutalyzer's [bugtracking system](#) to report errors and send in feature requests.

If you have any comments or suggestions be sure to let us know!

Last modified: February 10, 2010

Mutalyzer@humgen.nl

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3-Final

Appendix 2

Mutalyzer Quality Assurance report

GEN2PHEN Quality Assurance Report Mutalyzer (v. 1.0.4)

This Quality Assurance Report (QAR) is designed to accompany a Software Unit, a Database Component, or an Informatics Service used by the GEN2PHEN Consortium (www.gen2phen.org) as part of this project's mission to 'help unify human and model organism genetic variation databases towards increasingly holistic views into Genotype-To-Phenotype data, and to support linking this system into other biomedical knowledge sources via genome browser functionality'.

The QAR provides users with a concise but meaningful indication of the quality control and software development procedures that were used to generate the particular software product stated below:

Software Unit, Database Component, or Informatics Service covered by this QAR (name of product, and version details):

Mutalyzer v. 1.0.4

Purpose of the Software Unit, Database Component, or Informatics Service (brief details of the product's purpose and relationship/dependency to other products):

The web-based Mutalyzer tool (<http://www.mutalyzer.nl>) is used to check the nomenclature of sequence variants. Mutalyzer requires specification of a reference sequence accession number for download from GenBank or an upload of a custom reference sequence in GenBank format by the user. The LOVD 'LSDB-in-a-box' software is provided with the LOVD Mutalyzer module, which has to be activated by the LOVD database administrator or manager. The LOVD Mutalyzer module allows users to check sequence variant descriptions during submission via Mutalyzer's Name Checker. Subsequently, Mutalyzer returns the results of the sequence variant nomenclature check in a new window.

Team(s) that produced the Software Unit, Database Component, or Informatics Service: (contact details for each team, and role played in the product's development):

Mutalyzer development was started before the start of the Gen2Phen project by LUMC.
Legacy code development and testing: Ernest van Ophuizen and Martin Wildeman.
Bug fixes and testing: Corine van der Horst, Gerben Stouten, Jeroen Laros.
Current development: Jeroen Laros.
HGVS nomenclature guideline interpretation and design specifications: Peter Taschner and Johan den Dunnen.
LOVD Mutalyzer module development: Ivo Fokkema

Release date for the Software Unit, Database Component, or Informatics Service:

22-7-2009

DESIGN ACTIVITIES

Configuration Identification:

State how the design concept was conceived, and to what degree the project design was formalised. Was a Rapid Application Development approach employed.

The objective was to develop software to check whether sequence variant descriptions were valid according to the standard sequence variation nomenclature of the Human Genome Variation Society (See <http://www.hgvs.org/mutnomen>). The format of variant descriptions and their individual components (e.g., positions, variant types, and residues) were determined from the HGVS guidelines. The concept was to use the individual components as (optional) input arguments, of which the format can be checked separately by the core program. Finally, the program should check whether the combination of arguments is valid in relation to the reference sequence specified. Starting with Mutalyzer's Name Generator, which accepts all description components as separate arguments, a Rapid Application Development approach was employed, because the HGVS guidelines at the start of Mutalyzer development were not entirely stable and consistent. Mutalyzer development helped to identify inconsistencies and situations, which were not covered, leading to new versions of HGVS guidelines. This co-evolution of Mutalyzer and the HGVS guidelines is still on-going. To check descriptions generated in HGVS format, we constructed a parser that identifies and separates all individual components for further analysis by the core program. This parser is used by Mutalyzer's Name Checker and Batch Checker interfaces.

Requirement Specifications:

If available, list each of the project's Requirement Specifications, and wherever possible express this in terms of a quantitative Quality Metric.

- Web-based using open technology: Linux-Apache-PHP-(MySQL)-Python
- Retrieval and parsing of Genbank reference sequence records
- Parsing of sequence variant descriptions
- Conversion between different HGVS position numbering systems
- Application of HGVS nomenclature rules to generate and check descriptions
- Basic variant description output format:

Original input:<Genbank_Accession_number>.<version_number>:<variant_description>

Mutalyzer output:

Errors/Warnings regarding incorrect or incomplete input formats and/or reference sequence files

After successful check:

Checked or corrected input in HGVS format:

<Genbank_Accession_number>.<version_number>:<variant_description>

<left_flanking_reference_sequence><original_sequence><right_flanking_reference_sequence>

<left_flanking_reference_sequence><mutated_sequence><right_flanking_reference_sequence>

<protein_reference_sequence>

<mutant protein sequence>

Configuration Control:

Except in the case of RAD projects, describe the procedures used for Configuration Control (i.e., for managing changes to the Configuration Identification).

Configuration Auditing:

Except in the case of RAD projects, describe the procedures used for Configuration Auditing (i.e., for recording changes to the Configuration Identification)

CONSTRUCTION ACTIVITIES

Revision Control System:

State the coding language(s) used for the project's implementation, and whether or not a revision control system employed to track and archive the software development work.

Language and libraries:

Development started using Python v.2.3.5 (<http://www.python.org>) with BioPython v.1.30 and its dependencies listed on <http://www.biopython.org>. Current Mutalyzer v.1.0.4. uses Python v.2.5.2 using BioPython v.1.43 and its dependencies.

Development tools:

Eclipse SDK 3.30 with PyDev 1.3.8 in combination with Subclipse 1.2.4

Revision control system:

Subversion 1.4.6

Reviews During Product Construction:

Indicate the type and frequency of reviews that were used to optimise the construction work, for example Informal Peer Reviews, Walk-Through Reviews, Inspection Reviews, or Other procedures.

Weekly Informal Peer Reviews

TESTING ACTIVITIES

Module and Integration Testing:

Describe the white box and black box testing procedures that were used to assess the final product, stating who performed these tests. Give further details if these were formal and documented tests, specifying whether or not they employed a Test Plan, a Test Procedure, one or more Test Scripts, Test Cases, Test Data, or Other means of facilitation. Were any weaknesses or defects discovered that were not resolved before product release.

White box Module and Integration testing:

- Unit tests of individual software components (Developers, testers)
- Submission of lists of correct and incorrect Accession numbers and variant descriptions with known results, errors and warnings (Approved, corrected, not recognized). (Developers and testers)

Black box Integration testing:

- Scripts automatically generating descriptions from the individual components of variant descriptions to assess server and program stability (Testers)

Weaknesses (caused by third party data):

- Reference sequence parsing problems caused by inconsistent Genbank record formats.
- Incorrect descriptions caused by the program's inability to identify exon-intron boundaries in an old version of RefSeq transcript reference sequences due to lack of exon annotation. This issue is solved by using well-annotated RefSeqGene reference sequence records.
- Inability to map CDS (protein) features to mRNA (transcripts) features due to lack of direct cross references in Genbank or RefSeqGene records. This may occur in records of genes with multiple transcripts, some of which may be non-coding.

Defects:

- Problems parsing complex variant descriptions in HGVS allele format.

Will be solved by the development of a context free grammar parser in Mutalyzer version 2.

Issues to be resolved:

- Incorrect deletion, insertion, or deletion/insertion mutant descriptions can be generated in repetitive regions at (coding) DNA and protein level. A better algorithm will solve this in the refactored Mutalyzer version 2.

Issue Tracking:

Describe what form of issue tracking was employed in the project.

Currently, Mutalyzer uses Trac as its bug tracking system (<https://www.mutalyzer.nl/projects/mutalyzer/>).

Requirements Traceability Matrix:

Was a Requirements Traceability Matrix utilised, and if so give details of the test items and their correlation to the original Quality Metrics and Requirement Specifications. Specify any tests that the product was unable to pass or areas of identified weakness in the product.

Alpha Testing:

State whether Alpha Testing was performed, and if it was then give details of the nature and extent of that testing, who performed it, and how it was documented. Specify any features of the product that raised concern in that testing if they are still to be resolved.

Submission of lists of correct and incorrect Accession numbers and variant descriptions with known results, errors and warnings (Approved, corrected, not recognized) using the web interface by testers and design specifiers.


Black box testing:

Many users of the Mutalyzer website ([http:// www.mutalyzer.nl](http://www.mutalyzer.nl)) have contributed to the improvement of the user interface by helping to identify ambiguous instructions leading to incorrect input.

Users can submit requests and error reports to Mutalyzer's bug tracking system ([https:// www.mutalyzer.nl/projects/mutalyzer/](https://www.mutalyzer.nl/projects/mutalyzer/)).


The performance of Mutalyzer v.1.0.0 has been documented in:

Wildeman M, van Ophuizen E, den Dunnen JT, Taschner PE. 2008. Improving sequence variation descriptions in locus-specific mutation databases and the literature using the MUTation AnaLYZER (MUTALYZER) mutation nomenclature checker. Hum Mutat 29:6-13.

 HEALTH-200754	D4.3 A Validated Code-Base for Checking Mutation Nomenclature		
	WP4 - Genetics G2P databases		Security: PU
	Author(s): P. Taschner (LUMC), I. Fokkema (LUMC), C. Beroud (INSERM), J. den Dunnen (LUMC)		Version: v1.3-Final

Appendix 3

LOVD submission and Mutalyzer check of a variant



LOVD - Submit new sequence variant

To add sequence variant data to your submission, please fill out the form below.

All fields, unless specified otherwise, are required to be filled in.

	Add sequence variant data to	
Variant allele	Maternal (confirmed)	Location: 119514
Exon	06	Reference: tgaaggggctcttttgaacattcagtgagg
DNA change	c.525delT	Mutation: tgaaggggctctttt-gaacattcagtgagg
DNA_pub (Optional)	Check variant with Mutalyzer	Ref Protein: MVREQYTTAT EGICIERPEN QYVKIGIYG WRKRCLYLFV LLLLILLVN LALTIWILKV MWFSPAGM GH LCVTKDGLRL EGESEFLFPL YAKEIHSRVD SLLLLQSTON VTVNARNSEG EVTGRLKVG P KMVEVQNOQF QINSNDGKPL FTVDEKEVVV GTDKLRVTGP EGALFEHSVE TPLVRADPFQ DLRLESPTRS LSM DAPRGVH IQAHAGKIEA LSQMDILFHS SDGHLVLD AE TVCLPKLVQG TWGPGSSQSQ LYEICVCPDG KLYLSVAGVS TTCQESHIC L*
RNA change (Optional)		Mut Protein: MVREQYTTAT EGICIERPEN QYVKIGIYG WRKRCLYLFV LLLLILLVN LALTIWILKV MWFSPAGM GH LCVTKDGLRL EGESEFLFPL YAKEIHSRVD SLLLLQSTON VTVNARNSEG EVTGRLKVG P KMVEVQNOQF QINSNDGKPL FTVDEKEVVV GTDKLRVTGP EGALLNIQWR HPLSEPTRFK TLD*
Protein change (Optional)		Effects on Restrictionsites: Restrictionsite Engine made an error. Get the genome Mapping for this accessionNumber Download this reference sequence
DB-ID (First ID free: SGCG_00099) (copy to field)		Legend: Only transcripts with annotated CDS are shown: SGCG: transcript = NG_008759.1 protein = NP_000222.1
Variant remarks (Optional)		Done
Variant origin (Optional)	-- select --	
Reference (Optional)		

Mutalyzer output - Mozilla Firefox

http://www.humgen.nl/mutalyzer/1.0.1/digitalBackdoor.php?method=namecheck&format=human&mu.525

NG_008759.1 (SGCG) :c.525delT

Alternative (genomic) name, relative to record:
NG_008759.1:g.119514delT

Following transcripts were affected:
NG_008759.1 (SGCG) :c.525delT

Following proteins were affected:
NP_000222.1 (SGCG) :p.Phe175Leufs*20

Figure 1. LOVD submission of an SGCG variant checked via the Mutalyzer module.