BMC Bioinformatics

BioMed Central

Software

SNAD: sequence name annotation-based designer

Igor A Sidorov[1], Denis A Reshetov[2] and Alexander E Gorbalenya*[1,2]

Address: [1]Molecular Virology Laboratory, Department of Medical Microbiology, Center of Infectious Diseases, Leiden University Medical Center, PO Box 9600, E4-P, 2300 RC Leiden, Netherlands and [2]Faculty of Bioengineering and Bioinformatics, Moscow State University, 119899 Moscow, Russia

Email: Igor A Sidorov - i.a.sidorov@lumc.nl; Denis A Reshetov - reshetov@genebee.msu.ru; Alexander E Gorbalenya* - a.e.gorbalenya@lumc.nl

* Corresponding author

Published: 14 August 2009

BMC Bioinformatics 2009, **10**:251 doi:10.1186/1471-2105-10-251

Received: 16 February 2009

Accepted: 14 August 2009

This article is available from: http://www.biomedcentral.com/1471-2105/10/251

Abstract

Background: A growing diversity of biological data is tagged with unique identifiers (UIDs) associated with polynucleotides and proteins to ensure efficient computer-mediated data storage, maintenance, and processing. These identifiers, which are not informative for most people, are often substituted by biologically meaningful names in various presentations to facilitate utilization and dissemination of sequence-based knowledge. This substitution is commonly done manually that may be a tedious exercise prone to mistakes and omissions.

Results: Here we introduce SNAD (Sequence Name Annotation-based Designer) that mediates automatic conversion of sequence UIDs (associated with multiple alignment or phylogenetic tree, or supplied as plain text list) into biologically meaningful names and acronyms. This conversion is directed by precompiled or user-defined templates that exploit wealth of annotation available in cognate entries of external databases. Using examples, we demonstrate how this tool can be used to generate names for practical purposes, particularly in virology.

Conclusion: A tool for controllable annotation-based conversion of sequence UIDs into biologically meaningful names and acronyms has been developed and placed into service, fostering links between quality of sequence annotation, and efficiency of communication and knowledge dissemination among researchers.

Background

Identifiers (ID) composed of alphanumeric and other symbols are commonly used for tagging protein and nucleotide sequences and associated annotation in GenBank [1], UniProt [2], and other sources that either use these sequences or deal with molecular properties of associated biological entities (Table 1). For instance in UniProt, the primary accession number is the sole UID. In GenBank three fields, locus, accession and version, carry four different IDs, but only gi number (known also as primary ID) and accession number combined with version form unique IDs (UIDs). Each UID is assigned to a single entry. In contrast, a particular non-unique ID may be found in numerous entries. The UIDs ensure efficient computer-mediated data storage, maintenance and processing. To facilitate data mapping and transfer between different databases, in which ID inter-conversion plays important part, several specialized resources including GoMiner [3], ErmineJ [4], SOURCE [5], RESOURCERER [6]; IDconverter and IDClight [7], and DAVID [8] have been developed.

Table 1: Unique and non-unique identifiers in databases used by SNAD

Database	Identifier	Uniqueness	Example
GenBank[1]	gi number, primary ID	+	1234567
	version number = accession.version	+[9]	AY123456.1
	accession number	-	AY123456, NC_006558, NP_000483
	locus/sequence name	-	MTVCG, SCU49845
UniProt[2]	accession number	+	Q10AA9, P47123, A2BC19
	entry name	-	INSR_HUMAN
EMBL[3,4,5]	accession number	+	X64011, M10051, J00231, AJ000001, AY123456
	entry name	-	HSINSR, BUM
SeqHound[6,7]	gi number	+	1234567
	accession number	-	AY123456, NC_006558, NP_000483
	locus/sequence name	-	BTACHRE
EnsEMBL Gene/Transcript for Human[8]	gene ID, transcript ID	+	ENSG00000133103, ENST00000222982

1)http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=helpentrez&part=EntrezHelp#EntrezHelp.Entrez__the_Life_Sci
2) http://www.expasy.ch/sprot/userman.html#AC_line
3) [28]
4) http://www.ebi.ac.uk/embl/Documentation/User_manual/usrman.html#3_3
5) http://www.ebi.ac.uk/cgi-bin/dbfetch
6) [29]
7) http://search.cpan.org/~cjfields/BioPerl-1.6.0/Bio/DB/SeqHound.pm
8) Gene and Transcript part of EnsEMBL, http://www.ensembl.org/index.html
9) Some replaced or removed records in GenBank have no accession number but they can be retrieved from GenBank with the primary ID.

Sequence IDs are used for tagging sequences also in multiple sequence alignments and sequence-based phylogenetic trees in various presentations including publications [9-12]). Notwithstanding UniProt/SwissProt [2] entry names (Table 1), these machine-readable IDs are not very informative in human communications. Consequently, sequence IDs are commonly replaced by biologically meaningful names and acronyms, e.g. species names in trees and gene/protein names in alignments. When a large-scale dataset is to be processed, manual replacement of IDs with names could be a tedious task prone to mistakes and omissions.

By some considerations, the ID-to-name conversion for viral sequences is both of particular relevance and of considerable complexity. First, a large body of results in virus research can be linked to underlying sequences, owing to relatively small number of known viruses and comparatively small sizes of their extensively characterized genomes. Second, the virus naming, at the species level and below, is not governed by either the binomial Latin name convention traditional for cellular organisms or another broadly accepted nomenclature [13] that complicates communication in virology. For species, naming viruses is under auspices of the International Committee of Virus Taxonomy (ICTV) [14]. At the isolate and strain level, it is left up to individuals or groups of researchers to decide how to proceed [15,16]. Researchers studying few economically most important viruses developed nomenclatures restricted for these viruses [17-22]. In contrast, each decision about naming that concerns a majority of viruses is an *ad hoc* and, often, lengthy process. Likewise, naming viruses with artificially designed genomes, which are not product of natural evolution [23], is formally outside of the ICTV authority and completely unregulated.

The problem of ID-to-name conversion has been partially addressed in the NEXUS alignment format [24], which has a reserved part (block) for converting names in a predefined manner, and in the stand-alone tool Phyutility [25] that allows combining a limited number of GenBank characteristics with a custom separator to form sequence names. Here we introduce a dedicated web tool dubbed Sequence Name Annotation-based Designer (SNAD) that provides a versatile facility with user-friendly interface for automatic conversion of sequence UIDs into biologically informative names and acronyms in controllable manner using annotation associated with sequences. Several examples are provided to illustrate how SNAD can be used to generate names for practical purposes, particularly in virology.

Implementation

SNAD (ver. 1.6) is accessible at http://veb.lumc.nl/SNAD. SNAD code is written in Perl (BioPerl for dynamic retrieval of entries from databases, CGI for interfacing Perl code with web server, original Perl code for analysis of sequence objects retrieved from databases, and conversion of sequence annotation into complex names, part of application programming interface – API), JavaScript (graphical user interface for uploading data, designing format of name conversion and representing results), Java (part of API) and HTML for SNAD web site pages (see http://veb.lumc.nl/SNAD where a detailed description of all SNAD options and parameters can be found). SNAD can be used as a web server with APIs in Perl and Java available on SNAD web site. SNAD recognizes input as alignment or tree with the use of BioPerl drivers (Bio::TreeIO and Bio::AlignIO). If the input is not recognized as alignment or tree, SNAD processes it as a list of identifiers separated by one or more of the following symbols: space, new line, tabulator, or comma. To map IDs on database entries, Bio::DB::(GenBank, SwissProt, EMBL or SeqHound) drivers and Bio::EnsEMBL::DBSQL::DBAdaptor are used. SNAD uses standard BioPerl functions from Bio::Seq for extracting features/annotation from the database entries (GenBank/RefSeq/GenPept, UniProt, EMBL, SeqHound) or functions from database programming interface (EnsEMBL). Detailed description of all available functions is presented in "Introduction" page of the web site. SNAD code is available for downloading by non-commercial users per request sent to SNAD@lumc.nl.

Results and discussion

The diagram of SNAD dataflow is shown in Figure 1. The user submits sequence IDs, defines a format of conversion, and selects a database to be searched. Currently available databases include GenBank/RefSeq/GenPept, UniProt, EMBL, SeqHound and EnsEMBL (Gene and Transcripts divisions for Human sequences). SNAD extracts IDs from the input and queries the database with these IDs to locate associated annotation. After retrieving annotation, SNAD designs names according to a name template that defines name structure and annotation-based content for submitted entries. The final step of the SNAD-mediated processing is substitution of submitted IDs in the input with the designed names to return results to the user.

The SNAD web site has six pages including a detailed guide to SNAD with examples ("Introduction"), program execution page ("Execution"), guided access to Java and Perl APIs ("APIs"), frequently asked questions page ("FAQs"), list of the SNAD versions ("History"), and "Reference" page. SNAD execution page has three vertically arranged sections: "Before unique ID (UID) conversion" for uploading input, "Names/format designer" for choosing and refining pre-compiled name templates, and "After UID conversion" for accessing output (Figure 2).

The input can be either copy-pasted or uploaded from a file. SNAD accepts IDs from five databases (Table 1) including GenBank (gi or accession number) and UniProt (accession number or entry name) that are automatically recognized in and extracted from the input data prepared in several formats: as part of tree ('newick' and 'nexus') and alignment ('fasta', 'msf', 'clustalw', 'nexus', 'phylip', 'mega' and 'stockholm') or supplied as a plain list (Figure 2). Examples of IDs, unique and non-unique, in alignment, tree, and plain list are available for testing purposes. SNAD may accept input with non-unique IDs, e.g. GenBank accession number or locus name, that are most popular among researchers. Since these IDs may be associated with multiple entries, SNAD must decide which entry is to be used to retrieve annotation. The adopted approach is to use the most recent entry that might be not the one the user aimed at to retrieve annotation. The user may request automatically generated ID-to-name log (see below) for checking the quality of the conversion. To ensure unequivocal ID processing the user must use UIDs in input. UIDs from one database can be used to query another non-cognate database in which it may be not recognized at all or recognized as non-unique IDs (Table 2). For instance, UIDs from UniProt (primary accession numbers) are recognized as non-unique IDs in GenBank (accession numbers) while all UIDs from GenBank (gi or accession.version) are not recognized as valid IDs in UniProt.

The user has a choice of reading SNAD output either on the "Execution" web page, or at a separate web page in plain text mode, or receiving it via e-mail. For tree input, SNAD generates, additionally to the standard output, a respective tree drawn to facilitate visual inspection of the generated names in graphic format. The web page output can be enhanced with links to cognate entries accessible through the generated names.

There are two ways of defining format of the UID conversion: using either precompiled or user-defined name templates. The user is currently provided with several precompiled templates for name generation, which can be used "as is" or after refinement (see below). Each template is accompanied with examples of input to illustrate its usage. "Entry name" template can be useful for identifying misspelled identifiers, as well as for comparative inspection of basic information about the entries in UniProt. Two other templates ("Gene" and "Species") can be used for converting UIDs into gene and species names, respectively, as they are specified in GenBank. The "Complex species name" template combines full species name with primary ID (gi) that are neatly separated. The "Complex protein name" includes primary ID, condensed species name, protein description and CDS, and locus tag, in a size restricted manner to facilitate reading a complex name structure. These templates are suited for producing names of sequences originated from a wide range of organisms. "Acronym" template converts GenBank organism name annotation into acronym. Three templates were designed specifically for viruses to address specifics of this group for which the naming remains a matter of debate [15,17]. For several medically important viruses with large

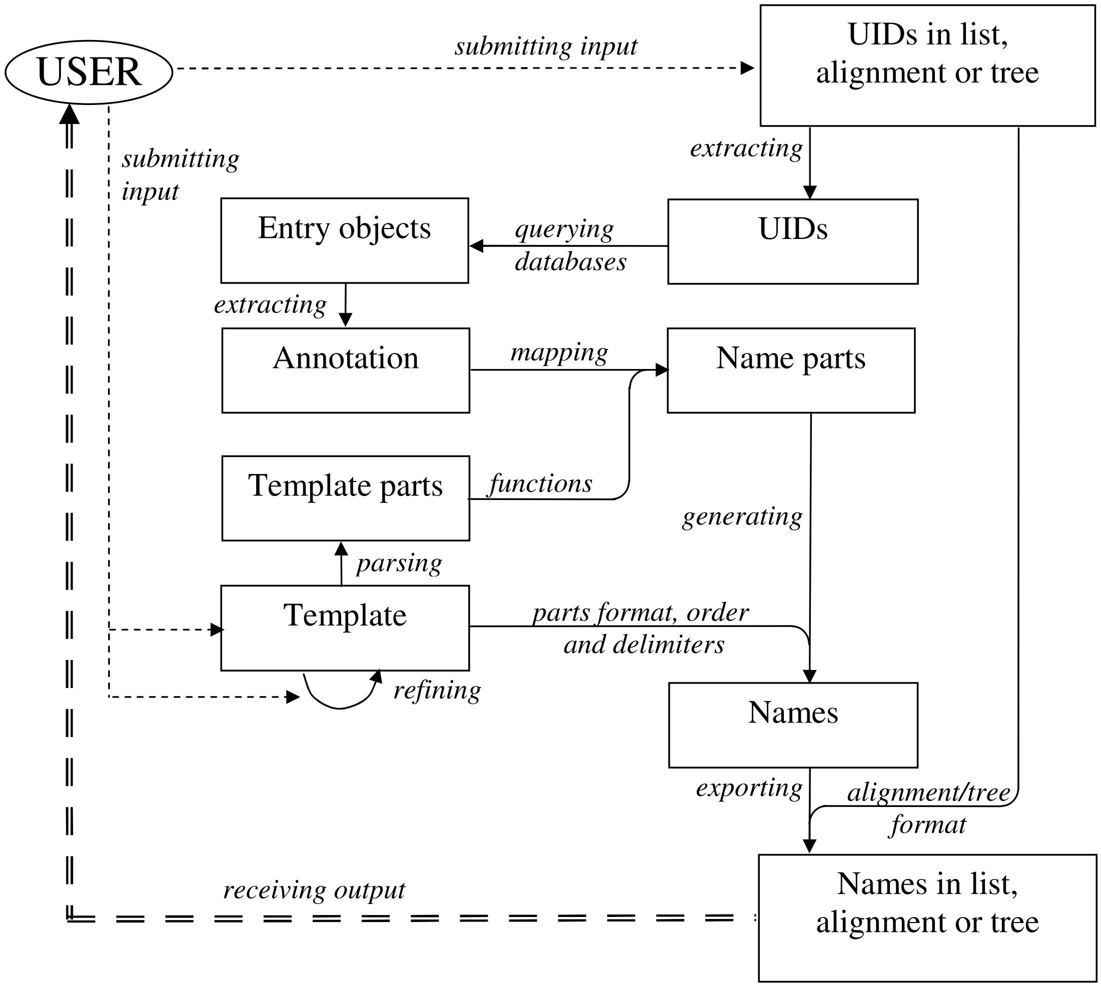
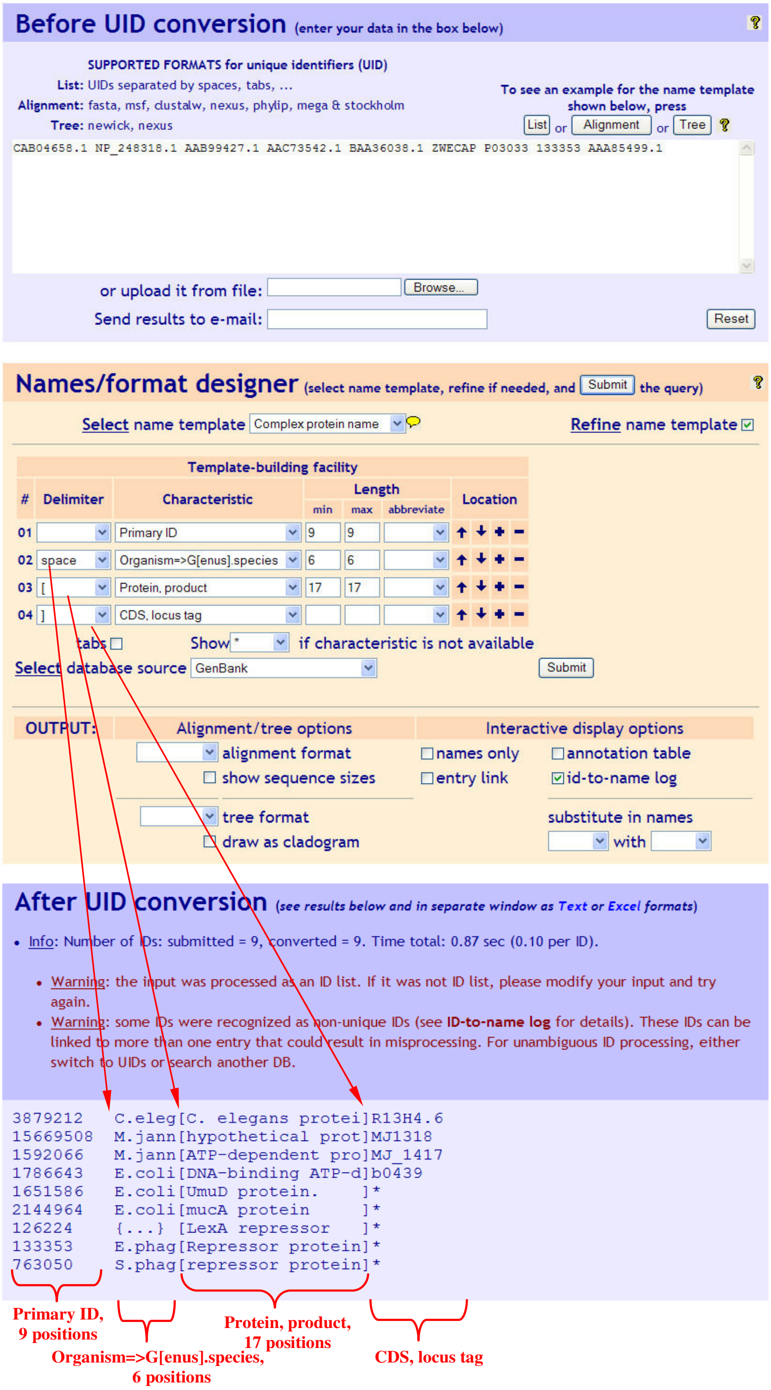


Figure 1
SNAD dataflow. User submits sequence UIDs and defines a format of conversion by choosing a template. SNAD analyzes the input, extracts UIDs from it and queries a user-defined database to locate cognate entries. The next step is extracting annotation from the entries and parsing the user-selected template to generate separate parts. They are mapped onto annotation to produce name parts, which are combined into names according to the template. Finally, SNAD substitutes the submitted UIDs in list, alignment, or tree with the designed names and returns results to the user.

sampling from the natural diversity, e.g. influenza and noroviruses [18,22], names conventions were accepted by the field to include reference to serotype, host species, country of isolation, strain or isolate name, collection date and, for influenza viruses, subtype. These names are part of the GenBank entries where from they can be readily extracted. To demonstrate that similar names can be also generated on-the-fly from two components, using the isolate/strain and serotype fields in respective entries, a separate "Influenza virus name" template is provided. We extended this approach by designing two other virus templates, "Complex virus name I" and "Complex virus name II", that could generate names with complex structure for majority of other viruses for which no consensus on the naming has been reached. These templates combine information from up to six separate entry fields, each keeping an elementary record e.g. about date or place of virus isolation. "Complex virus name II" additionally includes reference to protein function. These templates could help researchers studying many viruses, e.g. coronaviruses and picornaviruses [15] communicate their sequence-based data in a name format similar to that accepted for influenza virus. SNAD-mediated solution to the generation of complex virus names reinforces the role of diverse fields in sequence entries as the primary source of information. If these fields are completed with records as expected, virus names can be readily generated under different and evolving conventions. Thus, SNAD provides a tool for explora-



Before UID conversion (enter your data in the box below)

SUPPORTED FORMATS for unique identifiers (UID)

List: UIDs separated by spaces, tabs, ...
Alignment: fasta, msf, clustalw, nexus, phylip, mega & stockholm
Tree: newick, nexus

To see an example for the name template shown below, press List or Alignment or Tree

```
CAB04658.1 NP_248318.1 AAB99427.1 AAC73542.1 BAA36038.1 ZWECAP P03033 133353 AAA85499.1
```

or upload it from file:
Send results to e-mail:

Names/format designer (select name template, refine if needed, and Submit the query)

Select name template: Complex protein name
Refine name template ☑

Template-building facility

#	Delimiter	Characteristic	Length min	Length max	Length abbreviate	Location
01		Primary ID	9	9		
02	space	Organism=>G[enus].species	6	6		
03	[	Protein, product	17	17		
04	]	CDS, locus tag				

tabs □ Show * if characteristic is not available
Select database source: GenBank

OUTPUT:

Alignment/tree options
alignment format
□ show sequence sizes
tree format
□ draw as cladogram

Interactive display options
□ names only
□ entry link
□ annotation table
☑ id-to-name log
substitute in names with

After UID conversion (see results below and in separate window as Text or Excel formats)

- Info: Number of IDs: submitted = 9, converted = 9. Time total: 0.87 sec (0.10 per ID).
  - Warning: the input was processed as an ID list. If it was not ID list, please modify your input and try again.
  - Warning: some IDs were recognized as non-unique IDs (see **ID-to-name log** for details). These IDs can be linked to more than one entry that could result in misprocessing. For unambiguous ID processing, either switch to UIDs or search another DB.

```
3879212   C.eleg[C. elegans protei]R13H4.6
15669508  M.jann[hypothetical prot]MJ1318
1592066   M.jann[ATP-dependent pro]MJ_1417
1786643   E.coli[DNA-binding ATP-d]b0439
1651586   E.coli[UmuD protein.    ]*
2144964   E.coli[mucA protein     ]*
126224    {...} [LexA repressor   ]*
133353    E.phag[Repressor protein]*
763050    S.phag[repressor protein]*
```

Submit
Browse...
Reset
Primary ID, 9 positions
Organism=>G[enus].species, 6 positions
Protein, product, 17 positions
CDS, locus tag

Figure 2
Unique IDs conversion with SNAD. Nine IDs from GenBank (section "Before UID conversion") are converted using a pre-compiled four-characteristic template "Complex protein name" that includes the following GenBank characteristics: "Primary ID" (gi number); organism name formatted as "G [enus].species"; protein product and gene locus tag. Three delimiters are used: space, " [" and "]", and none of characteristics is abbreviated. First three characteristics have their size limited to: 9, 6, and 17 symbols, respectively. Results of the conversion are shown in "After UID conversion" section. Each arrow links a delimiter to a part of converted names. {...} indicates that more than one organism name characteristic is found for a submitted ID (see [GenBank/Protein: 126224], features source/organism).

Table 2: Cross-database recognition of UIDs in databases by SNAD*

	Source DB				
Target DB	GenBank	UniProt	EMBL	SeqHound	EnsEMBL
GenBank	UID	NUID	NUID	UID	NR
UniProt	NR	UID	NR	NR	NR
EMBL	NR	NR	UID	NR	NR
SeqHound	UID**	NR	NUID	UID	NR
EnsEMBL	NR	NR	NR	NR	UID

*With the use of standard BioPerl drivers for GenBank, UniProt, EMBL and SeqHound http://search.cpan.org/~cjfields/BioPerl-1.6.0 and API drivers from EnsEMBL web site http://www.ensembl.org/info/data/api.html, UIDs from source DB can be used for searching in target DB. In some target DBs they are recognized as UID, in others as non-unique IDs (NUID) or not recognized as identifiers (NR). Although same five DBs make the source and target DB lists, the table is *not* symmetrical.

**, only gi's from GenBank are recognized as UID in SeqHound.

tion of different name arrangements that could facilitate the development of conventions for naming different viruses by the virology community.

SNAD users can initiate refining of any of the provided templates (by using "Refine name template" option) or generate *de novo* templates by choosing "User defined" template that automatically activates the refining mode. In both instances, the advanced template-building facility becomes available for the user. It empowers the user with fine control over the output name format, including the number of independently designed parts of the name, characteristic of each part, parts order, size of each part, and type of delimiter preceding each part (Figure 2). Design of each part in the name is controlled by a separate row. The name parts – from left to right – are defined by respective rows in the template-building facility from top to bottom. Rows can be moved up and down or deleted and inserted with a respective effect on the name structure and content. For retrieving sequence characteristics from database entries SNAD uses BioPerl scheme with internally controlled vocabulary of annotation [26]. Information from two or more annotation fields can also be combined in a single characteristic in a template. For instance, for virus name templates, a combination "strain OR isolate" may be used to form a characteristic. For name designing, the user can choose from a dropdown menu of sequence characteristics and features that were derived from available databases. These characteristics can be used separately or in combination to design a name format that can combine, for instance, unique reference to the original database with a concise biological description of the entry (see Figure 2 for an example). For characteristics which values are described with several words, first letters of these words can be combined using characteristic's "abbreviate" option to produce acronyms. If a selected characteristic is not available in annotation for a sequence, it is replaced in the sequence name with a symbol that the user selected from a separate drop-down menu. Due to gaps in annotation, names designed with the templates may not be complete for every entry.

SNAD has a menu that controls output display interactively. By default, output includes ID-to-name log in which selected information about generated names is detailed in table format that can be downloaded as an Excel file. This table can be useful for controlling the name generation procedure and preparing legends to accompany an alignment or a phylogenetic tree in publication. The user can also extract IDs from submitted alignment or tree by activating "names only" option, or retrieve annotation from cognate entries in table format. In the current implementation of SNAD, the most time-consuming stage is data retrieval from databases. For large queries and complex name formats, also using API, it is therefore advisable to start with less than 10 sequences to design a format to be used for the full query. Queries containing up to 500 IDs and including sequence alignments up to 50000 characters can be processed through SNAD web page. For queries containing more than 500 IDs, the user is advised to use one of the SNAD APIs.

The availability of description for characteristics in a particular set of entries from databases is a major factor determining how well names designed with SNAD match expectation. The amount and accuracy of annotation in sequence databases vary markedly between entries that has been a major concern for the bioinformatics community continuously working toward improving this critical aspect of sequence databases [7,8,27]. Involving other researchers in this effort, especially those who submit original sequence annotation in the first place, could accelerate the process. We hope that through using SNAD service, researchers could see immediate benefits from better annotation for their research that may create impetus for contributing quality annotation to public databases. To improve the user experience with SNAD, we plan to expand the diversity of databases and templates upon future upgrades of SNAD. Any template that the users designed and considered useful could be included in the future SNAD releases.

Conclusion

A tool for controllable annotation-based conversion of sequence UIDs into biologically meaningful names and acronyms has been developed and placed into service, fostering links between quality of sequence annotation, and efficiency of communication and knowledge dissemination among researchers.

Availability and requirements

Project name: SNAD, Sequence Name Annotation-based Designer. Project home page: http://veb.lumc.nl/SNAD. Operating system(s): web service, platform independent. Programming languages: Java, JavaScript, Perl, HTML. Restrictions to use SNAD web site by non-academics: none.

Authors' contributions

AEG conceived SNAD; IAS, DAR, and AEG developed SNAD design and performed testing; DAR and IAS carried out programming; IAS and AEG wrote the manuscript. All authors read and approved the final version of the manuscript.

Acknowledgements

We thank Chris Lauber, Michael Rozanov, Johan Faase and Kalina Zlateva (Leiden University Medical Center, Leiden, Netherlands) for discussion and useful suggestions. This work was partially supported by the Collaborative Agreement in Bioinformatics between LUMC and MSU (CRDF GAP1473), and EU FP6 IP Vizier LSHG-CT-2004-511960.

References

1. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL: **GenBank.** *Nucl Acids Res* 2008, **36:**D25-D30.
2. Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, *et al.*: **UniProt: the Universal Protein knowledgebase.** *Nucl Acids Res* 2004, **32:**D115-D119.
3. Zeeberg BR, Feng W, Wang G, Wang MD, Fojo AT, Sunshine M, Narasimhan S, Kane DW, Reinhold WC, Lababidi S, *et al.*: **GoMiner: a resource for biological interpretation of genomic and proteomic data.** *Genome Biology* 2003, **4:**R28.
4. Lee HK, Braynen W, Keshav K, Pavlidis P: **ErmineJ: Tool for functional analysis of gene expression data sets.** *BMC Bioinformatics* 2005, **6:**269.
5. Diehn M, Sherlock G, Binkley G, Jin H, Matese JC, Hernandez-Boussard T, Rees CA, Cherry JM, Botstein D, Brown PO, *et al.*: **SOURCE: a unified genomic resource of functional annotations, ontologies, and gene expression data.** *Nucl Acids Res* 2003, **31:**219-223.
6. Tsai J, Sultana R, Lee Y, Pertea G, Karamycheva S, Antonescu V, Cho J, Parvizi B, Cheung F, Quackenbush J: **RESOURCERER: a database for annotating and linking microarray resources within and across species.** *Genome Biology* 2001, **2:**1-4.
7. Alibes A, Yankilevich P, Canada A, az-Uriarte R: **IDconverter and IDClight: Conversion and annotation of gene and protein IDs.** *BMC Bioinformatics* 2007, **8:**9-17.
8. Sherman BT, Huang DW, Tan Q, Guo Y, Bour S, Liu D, Stephens R, Baseler MW, Lane CH, Lempicki RA: **DAVID Knowledgebase: a gene-centered database integrating heterogeneous gene annotation resources to facilitate high-throughput gene functional analysis.** *BMC Bioinformatics* 2007, **8:**426-436.
9. Birghan C, Mundt E, Gorbalenya AE: **A non-canonical lon proteinase lacking the ATPase domain employs the ser-Lys catalytic dyad to exercise broad control over the life cycle of a double-stranded RNA virus.** *EMBO Journal* 2000, **19:**114-123.
10. Charrel RN, de Lamballerie X, Emonet S: **Phylogeny of the genus Arenavirus.** *Current Opinion in Microbiology* 2008, **11:**362-368.
11. Kapoor A, Victoria J, Simmonds P, Slikas E, Chieochansin T, Naeem A, Shaukat S, Sharif S, Alam MM, Angez M, *et al.*: **A highly prevalent and genetically diversified *Picornaviridae* genus in South Asian children.** *Proceedings of National Academy of Sciences* 2008, **105:**20482-20487.
12. Gonzalez JM, Gomez-Puertas P, Cavanagh DA, Gorbalenya AE, Enjuanes L: **A comparative sequence analysis to revise the current taxonomy of the family Coronaviridae.** *Archives of Virology* 2003, **148:**2207-2235.
13. van Regenmortel MH, Mayo MA, Fauquet CM, Maniloff J: **Virus nomenclature: consensus versus chaos.** *Archives of Virology* 2000, **145:**2227-2232.
14. Fauquet CM, Mayo MA, Maniloff J, Desselberger U, Ball LA: *Virus Taxonomy, Eighth report of the International Committee on Taxonomy of Viruses* Amsterdam: Elsevier, Academic Press; 2005.
15. Cavanagh DA: **A nomenclature for avian coronavirus isolates and the question of species status.** *Avian Pathology* 2001, **30:**109-115.
16. Leitch EC, Bendig J, Cabrerizo M, Cardosa J, Hyypia T, Ivanova OE, Kelly A, Kroes AC, Lukashev A, MacAdam A, *et al.*: **Transmission Networks and Population Turnover of Echovirus 30.** *J Virol* 2009, **83:**2109-2118.
17. Green KY, Ando T, Balayan MS, Berke T, Clarke IN, Estes MK, Matson DO, Nakata S, Neill JD, Studdert MJ, *et al.*: **Taxonomy of the Caliciviruses.** *The Journal of Infectious Diseases* 2000, **181:**S322-S330.
18. Kawaoka Y, Cox NJ, Haller O, Hongo O, Kaverin N, Klenk H-D, Lamb A, McCauley J, Palese P, Rimstad E, *et al.*: **Family Orthomyxoviridae.** In *Virus Taxonomy, Eighth report of the International Committee on Taxonomy of Viruses* Edited by: Fauquet CM, Mayo MA, Maniloff J, Desselberger U, Ball LA. Amsterdam: Elsevier, Academic Press; 2005:681-693.
19. Fauquet CM, Briddo RW, Brown JK, Moriones E, Stanley J, Zerbini M, Zhou X: **Geminivirus strain demarcation and nomenclature.** *Archives of Virology* 2008, **153:**783-821.
20. Schreiber MJ, Ong SH, Holland RCG, Hibberd ML, Vasudevan SG, Mitchell WP, Holmes EC: **DengueInfo: A web portal to dengue information resources.** *Infection, Genetics and Evolution* 2007, **7:**540-541.
21. Simmonds P, Bukh J, Combet C, Deleage G, Enomoto N, Feinstone S, Halfon P, Inchauspe G, Kuiken C, Maertens G, *et al.*: **Consensus proposals for a unified system of nomenclature of hepatitis C virus genotypes.** *Hepatology* 2005, **42:**962-973.
22. Zheng DP, Ando T, Fankhauser RL, Beard RS, Glass RI, Monroe SS: **Norovirus classification and proposed strain nomenclature.** *Virology* 2006, **346:**312-323.
23. Coleman JR, Papamichail D, Skiena S, Futcher B, Wimmer E, Mueller S: **Virus Attenuation by Genome-Scale Changes in Codon Pair Bias.** *Science* 2008, **320:**1784-1787.
24. Maddison DR, Swofford DL, Maddison WP: **NEXUS: an extensible file format for systematic information.** *Systematic Biology* 1997, **46:**590-621.
25. Smith SA, Dunn CW: **Phyutility: a phyloinformatics tool for trees, alignments and molecular data.** *Bioinformatics* 2008, **24:**715-716.
26. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JGR, Korf I, Lapp H, *et al.*: **The Bioperl Toolkit: Perl Modules for the Life Sciences.** *Genome Res* 2002, **12:**1611-1618.
27. Camon E, Barrel D, Brooksbank C, Magrane M, Apweiler R: **The Gene Ontology Annotation (GOA) Project–Application of GO in SWISS-PROT, TrEMBL and InterPro.** *Comarative and Functional Genomics* 2003, **4:**71-74.
28. Stoesser G, Baker W, Broek A van den, Garcia-Pastor M, Kanz C, Kulikova T, Leinonen R, Lin Q, Lombard V, Lopez R, *et al.*: **The EMBL Nucleotide Sequence Database: major new developments.** *Nucl Acids Res* 2003, **31:**17-22.
29. Michalikova K, Bader GD, Dumontier M, Lieu H, Betel D, Isserlin R, Hogue CWV: **SeqHound: biological sequence and structure database as a platform for bioinformatics research.** *BMC Bioinformatics* 2002, **3:**32.