Bolleman *et al. Journal of Biomedical Semantics* (2016) 7:39
DOI 10.1186/s13326-016-0067-z

Journal of Biomedical Semantics

RESEARCH Open Access

FALDO: a semantic standard for describing the location of nucleotide and protein feature annotation

Jerven T. Bolleman[1*], Christopher J. Mungall[2], Francesco Strozzi[3], Joachim Baran[4], Michel Dumontier[5], Raoul J. P. Bonnal[6], Robert Buels[7], Robert Hoehndorf[8], Takatomo Fujisawa[9], Toshiaki Katayama[10] and Peter J. A. Cock[11]

Abstract

Background: Nucleotide and protein sequence feature annotations are essential to understand biology on the genomic, transcriptomic, and proteomic level. Using Semantic Web technologies to query biological annotations, there was no standard that described this potentially complex location information as subject-predicate-object triples.

Description: We have developed an ontology, the Feature Annotation Location Description Ontology (FALDO), to describe the positions of annotated features on linear and circular sequences. FALDO can be used to describe nucleotide features in sequence records, protein annotations, and glycan binding sites, among other features in coordinate systems of the aforementioned "omics" areas. Using the same data format to represent sequence positions that are independent of file formats allows us to integrate sequence data from multiple sources and data types. The genome browser JBrowse is used to demonstrate accessing multiple SPARQL endpoints to display genomic feature annotations, as well as protein annotations from UniProt mapped to genomic locations.

Conclusions: Our ontology allows users to uniformly describe – and potentially merge – sequence annotations from multiple sources. Data sources using FALDO can prospectively be retrieved using federalised SPARQL queries against public SPARQL endpoints and/or local private triple stores.

Keywords: SPARQL, RDF, Semantic Web, Standardisation, Sequence ontology, Annotation, Data integration, Sequence feature

Background

Describing regions of biological sequences is a vital part of genome and protein sequence annotation, and in areas beyond this such as describing modifications related to DNA methylation or glycosylation of proteins. Such regions range from one amino acid (e.g. phosphorylation sites in singalling cascades) to multi megabase contigs mapped to a complete genome. Such annotation has been discussed in biological literature since at least 1949 [1] and recorded in biological databases since the first issue of the Atlas of Protein Sequence and Structure [2] in 1965.

There are many different conventions for storing genomic data and its annotations in plain text flat file formats such as Generic Feature Format version 3 (GFF3), Genome Variation Format (GVF) [3], Gene Transfer Format (GTF) and Variant Call Format (VCF), and more structured domain specific formats such as those from INSDC (International Nucleotide Sequence Database Collaboration) or UniProt, but none are flexible enough to discuss all aspects of genetics or proteomics. Furthermore, the fundamental designs of these formats are inconsistent, for example both zero-based and one-based counting standards exist, a regular source of off-by-one programming errors, which experienced bioinformaticians learn to look out for.

*Correspondence: jerven.bolleman@sib.swiss
[1]Swiss-Prot group, SIB Swiss Institute of Bioinformatics, Centre Medical Universitaire, 1 rue Michel, Servet, 1211 Geneva 4, Switzerland
Full list of author information is available at the end of the article

Although non-trivial, file format interconversion is a common background task in current script-centric bioinformatics pipelines, often essential for combining tools supporting different formats or format variants. As a result of this common need, file format parsing is a particular strength of community developed open source bioinformatics libraries like BioPerl [4], Biopython [5], BioRuby [6] and BioJava [7]. While using such shared libraries can reduce the programmer time spent dealing with different file formats, adopting Semantic Web technologies has even greater potential to simplify data integration tasks.

As part of the Integrated Database Project (http://lifesciencedb.mext.go.jp/en/) and the Core Technology Development Program (http://biosciencedbc.jp/en/33-en/programs/236-programs) to integrate life science databases in Japan, the National Bioscience Database Center (NBDC) and the Database Center for Life Science (DBCLS) have hosted an annual "BioHackathon" series of meetings bringing together biological database teams, open source programmers, and domain experts in Semantic Web and Linked Data [8–11]. At these meetings it was recognised that failure to standardise how to describe positions and regions on biological sequences would be an obstacle to the adoption of federalised SPARQL Protocol and RDF Query Language (SPARQL) queries which have the potential to enable cross-database queries and analyses. Discussion and prototyping with representatives from major sequence databases such as UniProt [12], DDBJ (DNA Data Bank of Japan) [13] (part of the INSDC partnership with the National Center for Biotechnology Information (NCBI)-GenBank [14] and European Molecular Biology Laboratory (EMBL)-Bank [15]), and a number of glycomics databases (BCSDB [16], GlycomeDB [17], GLYCOSCIENCES.de [18], JCGGDB, RINGS [19] and UniCarbKB [20]) and assorted open source developers during these meetings led to the development of the Feature Annotation Location Description Ontology (FALDO).

FALDO has been designed to be general enough to describe the position of annotations on nucleotide and protein sequences using the various levels of location complexity used in major databases such as INSDC (DDBJ, NCBI-GenBank and EMBL-Bank) and UniProt, their associated file formats, and other generic annotation file formats such as Browser Extensible Data (BED), GTF and GFF3. It includes compound locations, which are the combination of several regions (such as the 'join' location string in INSDC), as well as ambiguous positions. It allows us to accurately describe ambiguous positions today in such a way that future more precise knowledge does not introduce logical conflicts, which potentially could only be resolved by intervention of an expert in the field.

FALDO is suited to accurately describe the position of a feature on multiple sequences. This is expected to be most useful when lifting annotation from one draft assembly version to another. For example, a gene can start at a position for a given species' genome assembly, while the conceptually same gene can start at another position in previous/following genome assemblies for the species in question.

FALDO has a deliberately narrow scope which does not address general annotation issues about the meaning of or evidence for a location, rather FALDO is intended be used in combination with other relevant ontologies such as the Sequence Ontology (SO) [21] or database-specific ontologies. That is, it is used only to describe the loci of features, not to describe the features themselves. A FALDO position relative to a sequence record is comparable to a coordinate position on a map: it makes no claim about how that sequence record or map is related to the real world.

Implementation

FALDO is a small web ontology language version 2 (OWL2) ontology with 16 classes, 11 of these deal with the concept of a position on a sequence (Fig. 1). The instances of the `faldo:ExactPosition` represent positions that are accurately determined in respect to a reference sequence. There are two convenience subclasses of `faldo:ExactPosition` to represent positions on the N and C-terminal of a amino acid sequence. Three of those classes are used to describe accurately what we know of a position that is not precisely determined. Four classes are used to describe the concept of a position on a strand of DNA, e.g. positive, negative and on both strands. All ten of these classes are sub classes of the generic `faldo:Position` super-class. The eleventh class is the concept of a region i.e. something with a end and start position. The remaining 3 classes are used to group regions which are biologically related but for which no exact semantics are available e.g. some legacy data sources cannot be mapped cleanly without expert intervention. In contrast to other representations, FALDO has no explicit way to say that it is not "known" on which strand a position is, because this explicit statement unknown strand position can introduce contradictions when merging different data sets. For example, some positions could end up being contradictorily typed both as forward-stranded as well as being located on an unknown strand position.

There are 3 more classes (`faldo:CollectionOfRegions` and its subclasses) that are only there for backwards compatibility with INSDC join features with uncertain semantics. i.e. those join regions where a conversion program can only state that there are some regions and that the order that they are declared in the INSDC record might have biological significance. However, here the INSDC record needs intelligent

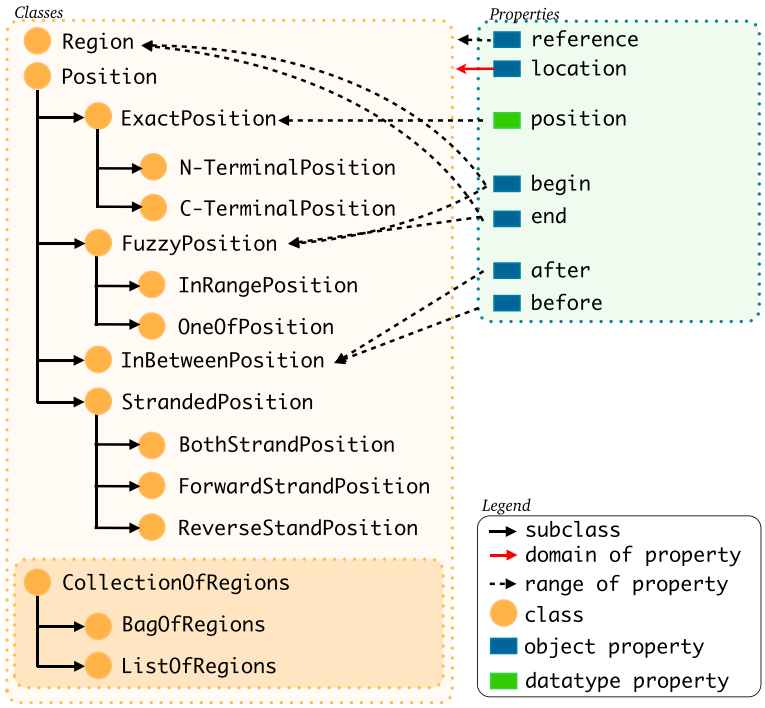

Fig. 1 The classes and object properties used in FALDO

inspection before the data can be cleanly converted to a data model with rich semantics.

FALDO defines a single datatype property, `faldo:position`, that is used to provide a one-based integer offset from the start of a reference sequence. This property, when used together with the `faldo:reference` property, links the concept of a `faldo:Position` to an instance of a biological sequence. Note that these terms are case-sensitive: `faldo:position` is a property, and `faldo:Position` is a concept.

For compatibility with a wide range of data, FALDO makes very few assumptions about the representation of the reference sequence, and can be used to describe positions on both single- and double-stranded sequences. When both strands of a double-stranded sequence are represented by a single entity (recommended over each strand being represented separately), integer `faldo:position` properties are counted from the 5' end of whichever strand is considered the "forward" strand.

A key part of the FALDO model is the separation of feature and where a feature is found in a sequence record. For this we use the `faldo:location` object property. This property is used to distinguish between a conceptual gene as an "unit of inheritance" and the corresponding representation of the DNA sequence region encoding the gene as stored in a database.

As in the INSDC data model and the associated GenBank ASN.1 notation, each location in FALDO has an identifier for the sequence it is found on [22]. This means that the position information is complete without further references to the context the position information was found in. The difference is that in FALDO, due to its RDF nature, the identifier of the sequence is a dereferencable pointer (URI) on the web, instead of just a string of characters.

Figure 2 shows how FALDO can be used to describe the position of features on a sequence, and compares it to the INSDC and GFF3/GTF text orientated formats.

Easier data integration due to OWL reasoning

Two `owl:Classes` ease data-integration with a `owl:hasKey` construct. A `faldo:ExactPosition` is the same as another position if it has the same `faldo:position` and `faldo:reference`. In practicse this means that if two sequence records are declared to be `owl:sameAs` then the features mapped to one of these sequence records is automatically mapped to the other. i.e. One extra statement allows feature annotation from a UniProt protein record to be transferred to INSDC Coding Domain Features.

Compression via OWL2 reasoning

For large databases such as INSDC or UniProt, the need to repeat the reference sequence for each position may

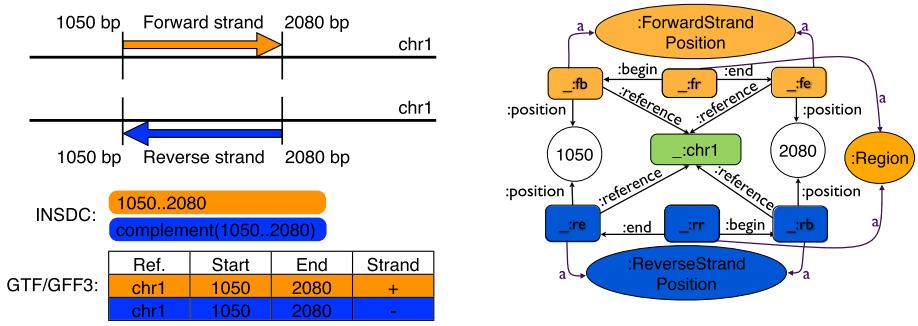

Fig. 2 Assorted conventions for regions, start, end, and strands. This figure shows two hypothetical features on a DNA sequence (labeled `chr1`), on either the forward strand (*orange*) or reverse strand (*blue*). Using the INSDC location string notation, these regions are "`1050..2080`" and "`complement(1050..2080)`" respectively if implicitly given in terms of the reference chr1. Using the GTF/GFF3 family of formats, regardless of the strand these two locations are described with *start* = 1050 and *end* = 2080, and in general, *start* ≤ *end*. Biologically speaking, in terms of transcription, the start of a genomic feature is strand dependent. For the forward strand feature (*orange*), the start is 1050 while the reverse strand feature (*blue*) starts from 2080

come with a significant cost in storage. However, this triple does not need to be materialised in the database, as it is inferrable using OWL2 property chain reasoning. With the axiom shown in Fig. 3 the `faldo:reference` triples can be inferred for any `faldo:position` described by an INSDC record. Having an OWL-capable query rewriter allows users to ignore the difference between encoding the `faldo:reference` properties explicitly and having them inferred at query time. For RDF databases that do not offer this capability, the necessary triples can be easily added using a single SPARQL insert query (Fig. 4). This flexibility allows users of the data to select the best approach for their infrastructure, rather than being constrained by the decisions of the data provider.

Validating data encoded with FALDO

Some databases only allow a subset of FALDO. For example INSDC requires that the start and end of a region are on the same sequence, while UniProt requires that a feature is described in relation to the reference's canonical isoform. Yet another database might annotate the location of a glycsoylation site on an UniProt isoform sequence. When added to an UniProt record in RDF, this extra RDF annotation would be ignored by applications that are not concerned with glycosylation of isoforms. The same annotation cannot be added to UniProt XML as the XSD schema does not allow for it, and the older plain text flat-file format does not allow for this kind of third party extension either. An attempt to add such information would very likely break any XML or flat-file parser and introduces the risk of importing data incorrectly. Only the UniProt RDF format allows other people to make assertions about UniProt data without breaking existing tools.

There are many ways to add constraints to the data model by applications using Semantic Web technologies [23]. In other words, data validation is an application specific concern instead of a data format concern.

Users

FALDO is already deployed and used in a number of tools and databases, in each case extended with more semantic web data using resource specific ontologies and schemas as well as other semantic standards e.g. the Sequence Ontology.

```
insdc:reference
      a       owl:ObjectProperty ;
      rdfs:subPropertyOf faldo:reference ;
      owl:propertyChainAxiom
              (faldo:endOf faldo:locationOf insdc:featureOf insdc:sequence) ,
              (faldo:locationOf insdc:featureOf insdc:sequence) ,
              (faldo:beginOf faldo:locationOf insdc:featureOf insdc:sequence) .
```

Fig. 3 OWL2 property chain axiom to infer that all positions described in an INSDC record are relative to the main sequence of the record (in RDF turtle syntax, prefixes omitted)

```
INSERT {
    ?position faldo:reference ?sequence .
}
WHERE {
    ?record a insdc:Entry ;
            insdc:feature ?feature ;
            insdc:sequence ?sequence .
    ?feature faldo:location ?location .
      { ?location faldo:begin|faldo:end ?position . }
    UNION
      { ?location a faldo:Position . }
}
```

Fig. 4 A SPARQL query to add all `faldo: reference` properties to `faldo: positions` described from a `insdc: record`

GFVO The Genomic Feature and Variation Ontology (GFVO) uses FALDO to describe loci on genomic landmarks as well as individual genomic feature positions [24].

JBrowse JBrowse can use SPARQL queries with FALDO to visualize annotations on reference sequences from semantic databases [25] (see Fig. 5).

INSDC-DDBJ DDBJ is currently working on an RDF format for the INSDC data that is stored in DDBJ/GenBank/EMBL-Bank.

BioInterchange BioInterchange makes use of FALDO in its RDF formatted output to describe genomic position information stored in the bioinformatics file formats of GFF3, GTF, GVF and VCF (http://www.codamono.com/biointerchange).

TogoGenome TogoGenome is a genome database collection provided by the DBCLS that uses FALDO in its RDF representation (http://togogenome.org/).

PhenomeBrowser The positions on the mouse genome of phenotype and disease related natural variations are described using FALDO.

BOING The "bio-ontology integrated querying of sequence annotations" framework uses FALDO to describe all feature locations [26].

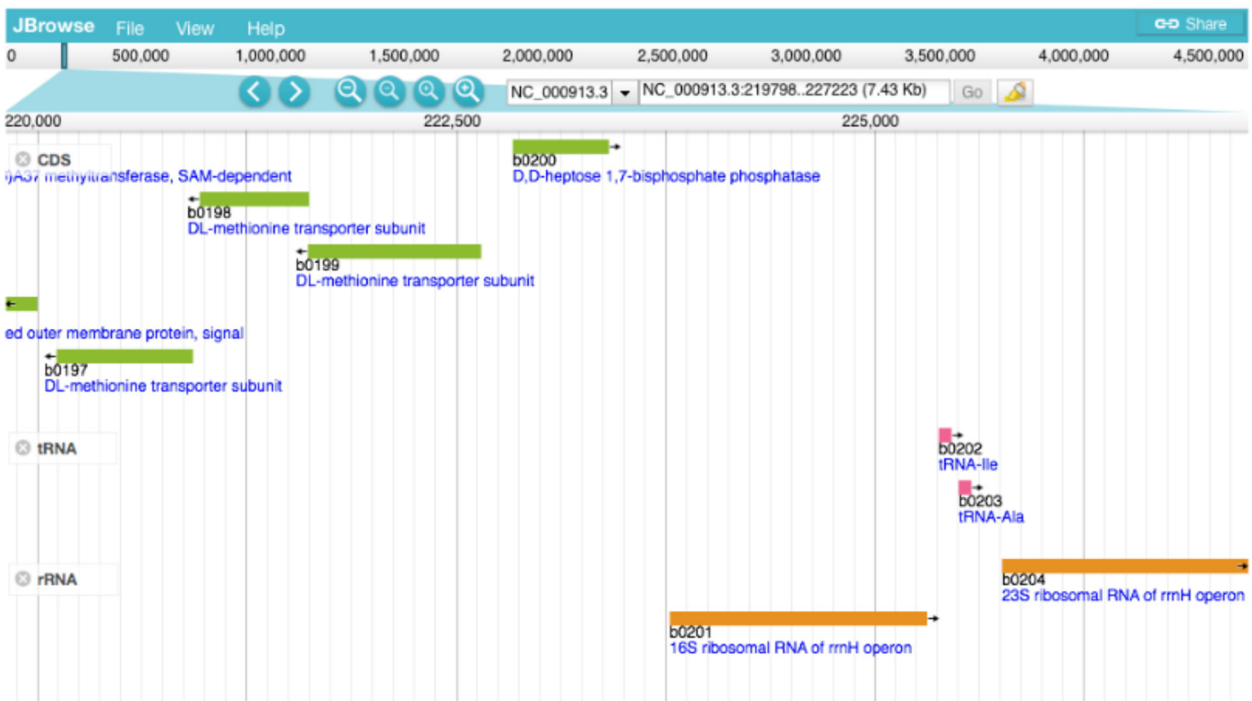

Fig. 5 JBrowse showing features, whose location is encoded using FALDO, selected via SPARQL (at e.g. http://togogenome.org/gene/1016998:SPAB_00296)

SPARQL-BED This simple tool that turns any BED file into a Web accessible SPARQL endpoint using FALDO to describe BED feature positions (https://github.com/JervenBolleman/sparql-bed).

BioPerl BioPerl [4] now includes a FALDO exporter (`Bio::FeatureIO::faldo`), which allows any BioPerl-supported feature format to be translated to FALDO.

UniProt UniProt annotates many protein features and sites. Starting with UniProt RDF release 2014_01 the positions of protein feature are described using FALDO.

Results

One of the practical goals driving the development of FALDO was to be able to represent all the annotated sequences in INSDC and UniProt as RDF triples, as a step towards providing this data via SPARQL endpoints where it can be queried.

The protein examples considered here, such as the UniProt feature annotations, describe relatively simple locations within protein sequences (see the active site annotation in Figs. 6 and 7).

Complement strand

Describing biological features in relation to a genomic DNA sequence does not have to be complicated.

For example the *cheY* gene (shown in Fig. 8) *Escherichia coli* str. K-12 substr. MG1655 (accession NC_000913.2) is described in the INSDC feature table as `complement(1965072..1965461)`, which is 390 base pairs using inclusive one-based counting. This feature begins on the base complementary to $start = 1965461$ and finishes at $end = 1965072$, so the INSDC location string can be interpreted as `complement(`*end*`..`*start*`)`. FALDO respects this biological interpretation of a feature location on the reverse strand.

In contrast, other formats such as the GFF family of formats, require $start \leq end$ regardless of the strand, which is equivalent to interpreting the INSDC location

```
AC   Q6Q250;
...
DE   Flags: Precursor; Fragment;
...
FT   SIGNAL       <1     15
FT   ACT_SITE    153    153       Nucleophile (By similarity).

uniprot:Q6Q250 rdf:type up:Protein ;
   up:annotation SHA:E50-1B, SHA:E50-1C ;
   up:sequence isoform:Q6Q250-1 ;
   rdfs:seeAlso ddbj-cds:AAS67043.1 .
SHA:E50-1B a up:Signal_Peptide_Annotation ;
  faldo:location region:f1t15 .
SHA:E50-1C a up:Active_Site_Annotation ;
  faldo:location position:153 .
region:f1t15 a faldo:Region ;
   faldo:begin position:before1 ;
   faldo:end  position:15 .
position:before1 a faldo:InRangePosition ;
   faldo:reference isoform:Q6Q250-1 ;
   faldo:end position:1 .
position:1 a faldo:ExactPosition ;
   faldo:reference isoform:Q6Q250-1 ;
   faldo:position 1 .
position:15 a faldo:ExactPosition ; #End of the signal peptide region
   faldo:reference isoform:Q6Q250-1 ;
   faldo:position 15 .
position:153 a faldo:ExactPosition ; #Where the active site should be
   faldo:reference isoform:Q6Q250-1 ;
   faldo:position 153 .
isoform:P05064-1 up:fragment "single" ;
   up:precursor true .
```

Fig. 6 Excerpt from UniProt entry Q6Q250 showing the position of an active site and a signal peptide in both the UniProt flat-file format and FALDO

```
LOCUS       AY566647                948 bp    mRNA    linear   INV 22-MAR-2004
...
FEATURES             Location/Qualifiers
...
     CDS             <1..>948
                     /note="allergen Pol d 1.03"
                     /codon_start=1
                     /product="venom phospholipase A1 3 precursor"
                     /protein_id="AAS67043.1"
                     /db_xref="GI:45510891"
                     /translation="ADDLTTLRNGTLDRGITPDCTFNEKDIELHVYSRDKRNGIILKK
                     EILKNYDLFQKSQISHQIAILIHGFLSTGNNENFDAMAKALIEIDNFLVISVDWKKGA
                     CNAFASTNDVLGYSQAVGNTRHVGKYVADFTKLLVEQYKVPMSNIRLIGHSLGAHTSG
                     FAGKEVQRLKLGKYKEIIGLDPAGPSFLTSKCPDRLCETDAEYVQAIHTSAILGVYYN
                     VGSVDFYVNYGKSQPGCSEPSCSHTKAVKYLTECIKRECCLIGTPWKSYFSTPKPISQ
                     CKRDTCVCVGLNAQSYPAKGSFYVPVDKDAPYCHNEGIKL"

ddbj-feature:AAS67043.1 rdf:type :Nucleotide_Resource ;
  faldo:location [ a faldo:Region ;
                   faldo:begin [ a faldo:InRangePosition ;
                                 faldo:before [a faldo:ExactPosition ;
                                               faldo:position 1 ;
                                               faldo:reference ddbj-seq:AY566647 ]]];
                   faldo:end   [ a faldo:InRangePosition ;
                                 faldo:after [a faldo:ExactPosition ;
                                               faldo:position 948 ;
                                               faldo:reference ddbj-seq:AY566647 ]]].
```

Fig. 7 DDBJ record associated with UniProt Q6Q250 showing the related CDS sequence, with coding region outside of the known deposited mRNA sequence

```
@prefix faldo:       <http://biohackathon.org/resource/faldo#> .
@prefix ddbj-record: <http://ddbj.nig.ac.jp/entry/nucleotide/> .

# Here are example triples describing a gene, note the location line:
<_:geneCheY> a so:Gene ;
           rdfs:label "cheY" ;
           faldo:location <_:example> .

# The following triples define the location used by the feature above,
# made up of a region and its two positions:
<_:example> a faldo:Region ;
           faldo:begin [ a faldo:Position ,
                           faldo:ExactPosition ,
                           faldo:ReverseStrandPosition ;
                         faldo:position 1965461 ;
                         faldo:reference ddbj-record:U00096.3 ];
           faldo:end  [ a faldo:Position ,
                          faldo:ExactPosition ;
                          faldo:ReverseStrandPosition ;
                        faldo:position 1965072
                        faldo:reference ddbj-record:U00096.3 ].
```

Fig. 8 Using FALDO in Turtle [29] syntax to describe the location of a gene feature *cheY* at `complement(NC_000913.2:1965072..1965461)` in the INSDC record U00096.3

string as `complement(`*start*`..`*end*`)`. This convention has some practical advantages when dealing with numerical operations on features sets, such as checking for overlaps or indexing data. For example, the feature length is given by $length = end - start + 1$ under this numerically convenient scheme where the interpretation of *start* versus *end* is strand independent.

INSDC compound locations

There are a number of implicit conventions in INSDC data that would ideally translated into a more explicit model when using FALDO. However, to enable automated bulk conversion of existing data, the FALDO class `faldo:CollectionOfRegions` and its subclasses exist to describe the compound locations used in the INSDC feature tables. Specifically, `join(...)` locations where the order is known map to FALDO's `faldo:ListOfRegions`, while `order(...)` where the order is *unknown* map to to `faldo:BagOfRegions`.

Thus while gene models with an intron/exon structure can be described this way, it is preferable when converting to RDF to explicitly describe the individual exons, each of which would have a simple location in FALDO.

One special case of INSDC compound regions is features on a circular chromosome that overlap the chromosome's origin of replication. One such feature is the "Protein II" gene from the reverse strand of f1 bacteriophage (ddbj:J02448). "Protein II" transcription starts at position 6006 on the reverse strand and ends at position 831 (see Fig. 9).

Fuzzy locations

Feature positions in, for example, INSDC or UniProt, are not always exactly known or described, but we should strive to describe our limited knowledge as accurately as possible. Take for example the position of the signal peptide annotation shown in Fig. 6, where the protein sequence is known to belong to a family of

```
{"@context" : {"comment"     : "http://www.w3.org/2000/01/rdf-schema#comment",
               "label"       : "http://www.w3.org/2000/01/rdf-schema#label",
               "taxonomy"    : "http://purl.uniprot.org/taxonomy/",
               "ddbj"        : "http://ddbj.nig.ac.jp/ontologies/sequence/",
               "ddbj-record" : "http://ddbj.nig.ac.jp/entry/nucleotide/",
               "ddbj-aa"     : "http://ddbj.nig.ac.jp/entry/protein/",
               "ddbj-seq"    : "http://ddbj.nig.ac.jp/entry/sequence/",
               "identifier"  : "http://purl.org/dc/terms/identifier",
               "faldo"       : "http://biohackathon.org/resource/faldo#",
               "so"          : "http://purl.obolibrary.org/obo/so.owl#"},
 "@id"          : "ddbj-record:J02448",
 "identifier"   : "J02448",
 "comment"      : "Bacteriophage f1, complete genome",
 "ddbj:organism": "taxonomy:10863",
 "ddbj:feature" : {"@id"           : "ddbj-protein:AAA32209.1 ",
                   "@type"         : "so:0000316",
                   "label"         : "Protein II",
                   "faldo:location": {"@type"       : "faldo:Region",
                                      "faldo:begin": {"@type": ["faldo:Position",
                                                                "faldo:ExactPosition",
                                                                "faldo:ForwardStrandPosition" ],
                                                       "faldo:position" : "6006",
                                                       "faldo:reference": "ddbj-seq:J02448" },
                                      "faldo:end" : {"@type" : ["faldo:Position",
                                                                "faldo:ExactPosition",
                                                                "faldo:ForwardStrandPosition" ],
                                                     "faldo:position"  : "831",
                                                     "faldo:reference" : "ddbj-seq:J02448" }
                                    }
                   }
}
```

Fig. 9 Partial example of using FALDO in JSON-LD [30] syntax to describe the CDS "Protein II" at `join(6006..6407,1..831)` on J02448. Notice that this is given as a single location rather than being artificially split in two as in the INSDC `join(...)` notation

proteins, but unfortunately only a part of the amino acid sequence is known. The UniProt curator deduced that the signal peptide region only partly overlaps the known sequence fragment. The same is true in the related INSDC record, were the CDS starts and ends before the known mRNA sequence (see Fig. 7). As demonstrated in the figure, this limited knowledge can be described using the FALDO classes `faldo:InRangePosition` and `faldo:OneOfPosition`.

Restriction enzymes

The task of describing the recognition sites of most restriction enzymes is quite straightforward, as is describing the cleavage site of a blunt end cutting enzyme. However, the cut site of a sticky-end cutting enzyme like HindIII that leaves an "overhang" is more challenging to specify, since it cuts in a different place on the forward and reverse strands. Figure 10 demonstrates how to describe this in FALDO by specifying start and end positions of the cut site that are on different strands.

Discussion

When designing FALDO, a broad range of use cases were considered from human genome annotations to protein domains and glycan binding sites on amino acid sequences, with the goal of developing a scheme general enough to describe regions of DNA, RNA and protein sequences.

Advantages and drawbacks of existing file formats were considered, including line based column formats like BED and GTF/GFF3, which focus on exact ranges on a given sequence, and the more complex locations supported by the INSDC feature tables used by DDBJ, NCBI-GenBank and EMBL-Bank.

The simplest non-stranded range location on a linear sequence requires a start and end coordinate, but even here there are existing competing conventions for describing open or closed end-points using zero and one-based counting (for example BED versus GTF/GFF3/INSDC).

In FALDO we always count from the start of the forward 5'–3' strand, even for features on the reverse strand. This encoding means there is no need to know the length

```
    \/
5'..AAGCTT..3'
    123456
3'..TTCGAA..5'
        /\

:enzymeRestrictionSite a example:EnzymeRestrictionSite ;
       faldo:location :enzymeRestrictionRegion .
:enzymeRestrictionRegion a faldo:Region ;
       faldo:begin :53_1 ;
       faldo:end   :35_6 .
:53cleaveageSite a example:CleavageSite ;
       faldo:location [a faldo:InBetweenPosition,
                           SO:0001690 ; # 3' restriction enzyme junction
                       faldo:after  :53_1 ;
                       faldo:before :53_2 ] .
:35cleaveageSite a example:CleavageSite ;
       faldo:location [a faldo:InBetweenPosition,
                           SO:0001689  ; # 5' restriction enzyme junction
                       faldo:before :35_5 ;
                       #Note the before after inversion due to biological start
                       faldo:after  :35_6 ].
:53_1 a faldo:ExactPosition , faldo:ForwardStrandedPosition ;
      faldo:position 1 .
:53_2 a faldo:ExactPosition , faldo:ForwardStrandedPosition ;
      faldo:position 2 .
:35_5 a faldo:ExactPosition , faldo:ReverseStrandedPosition ;
      faldo:position 5 .
:35_6 a faldo:ExactPosition , faldo:ReverseStrandedPosition ;
      faldo:position 6 .
```

Fig. 10 FALDO representation of the HindIII restriction enzyme cleavage site with sticky ends

of the sequence to compare positions on the different strands of a linear chromosome or genome. The end and start position of a region is inclusive. Unlike formats like GTF/GFF3, FALDO shares with Chado [27] the convention that the start coordinate should be the biological start (which may be a numerically higher value than the end coordinate).

For a semantic description describing the strand explicitly is preferable. FALDO chooses to add the strand information to the position. This is required to accurately describe for example the sticky ends of an enzyme digestion cut site, as in the HindIII example (Fig. 10).

A major difference with other standards is that we chose to make strandedness and reference sequence a property of the position, instead of the region. This is important in a number of use cases. For example, one may need to describe the position of a gene on a draft genome assembly where the start and end are known to be on different contigs. This can be the case when RNA mapping is used in the genome assembly process. Another is when rough semantics are used in queries e.g. answering what is the start and end of a gene. In a process called transplicing, exons of one gene can be found on multiple chromosomes, or on different strands of the same chromosome. e.g. gene *mod*(*mdg4*) of *Drosophila melanogaster* (uniprot:Q86B87). In such cases the start of the gene can be on a different reference sequence or strand than the end. These biological realities cannot be described accurately if the reference sequence was a property of the region. As a side effect, it allows single nucleotide or amino acid sites to be described directly as a position without a need for an artificial region of length one.

Every `faldo:Position` refers to the sequence it is on. This allows us to say that gene *XX* starts at position 4 of assembly *Y*1, while the same conceptual gene starts at position 5 of assembly *Y*2. Even within the same assembly, FALDO offers the possibility to describe features in different contexts at the same time, allowing for instance to represent a SNP in terms of its position within a known coding region (i.e. gene coordinates) and within a chromosome region, which offers clear advantages for features annotation. Chado also allows multiple locations per feature, but unlike FALDO, the start and end of any location must be in the same region, which prohibits for example a feature that spans more than one contig, or describing the same feature on two different genome assemblies.

Efficiency of region-of-interest queries

For FALDO we also considered query efficiency in comparison to existing search technology. Region of interest (ROI) queries are common operations performed on a set of genome annotations to extract a set of features within a range. For applications such as genome browsers, it is important that these are efficient enough. Although some RDF query engines may perform poorly when performing ROI queries over large feature sets, others have special indexes (e.g. literal filter indexes) that improve query performance. There is scope for further optimisation in the context of a SPARQL query by combining efficient algorithms and indexes such as Nested Containment Lists (NCLs) [28] or spatial indexes.

As a RDF based format, FALDO can be used to represent feature position information in a wide variety of serialisations e.g. JSON-LD, RDF/XML, Turtle, RDFa (embedded in HTML). This allows developers flexibility in consideration of their usage scenario, while at the same time allowing conversion to the common RDF triple model used in RDF databases and accessed by SPARQL queries.

Conclusions

FALDO is a small ontology for describing biological features in a consistent manner that bioinformaticians can depend upon. The diverse software and high-profile databases already using FALDO show that it has enough power to describe existing biological feature locations. The uptake of this ontology means that it is now much easier for users querying biological databases on the Semantic Web to compare features on the basis of locations. This also means that visualisation tools that access positional data via SPARQL can easily reuse significant parts of queries between databases.

Abbreviations

BED: browser extensible data (file format); DDBJ: DNA data bank of Japan; EMBL: European molecular biology laboratory; FALDO: feature annotation location description ontology; GFF: generic feature format; GFF3: generic feature format version 3; GTF: gene transfer format, a variant of GFF; GVF: genome variation format, an extension to GFF3; INSDC: international nucleotide sequence database collaboration; OWL2: web ontology language (note acronym is OWL, not WOL); RDF: resource description framework; SPARQL: SPARQL protocol and RDF query language; UniProtKB: universal protein knowledgebase; VCF: variant call format.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

JTB wrote the basic ontology file and mapping to UniProt RDF and contributed to the text of this article. CM added the `Bio::FeatureIO::faldo` logic to BioPerl and contributed to the OWL modeling. FS contributed to the Cufflinks RDF converter and wrote the VCF converter that uses FALDO for sequence variations positions. JB incorporated FALDO into the BioInterchange software as well as the Genomic Feature and Variation Ontology (GFVO). MD worked on the general modelling as well as mapping FALDO to SIO as an upper ontology. RJPB adapted BioRuby to match the ontology and wrote the Cufflinks to locations.rdf converter. RB adapted JBrowse to query SPARQL endpoints that use this format to generate custom tracks. RH did the GFF3 to OWL conversion. TF and TK implemented an ontology and a tool for converting INSDC records to RDF and used FALDO to describe features on genomes in TogoGenome. PC wrote a large section of this paper, and co-ordinated the working group during the BioHackathon meetings. All authors read and approved the final manuscript.

Acknowledgments
The developers recognize the invaluable contributions from the community in helping to create this standard. We would like to especially thank the organisers and funders of the BioHackathon series of meetings for hosting the original discussions leading to FALDO (http://www.biohackathon.org/). The BioHackathon series, this work and Toshiaki Katayama were supported by National Bioscience Database Center (NBDC) of Japan Science and Technology Agency (JST) and Database Center for Life Science (DBCLS) of Research Organization of Information and Systems (ROIS) in Japan. Jerven Bolleman was supported in his role at Swiss-Prot, whose activities at the SIB Swiss Institute of Bioinformatics are supported by the Swiss Federal Government through the The State Secretariat for Education, Research and Innovation SERI. Christopher Mungall was supported by the NIH under R24OD011883 and by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Peter Cock was supported by the Scottish Government Rural and Environmental Research and Analysis Directorate. Takatomo Fujisawa was supported by the DNA Databank of Japan (DDBJ), Research Organization of Information and Systems (ROIS) in Japan.

Availability and requirements
FALDO is publicly available at the URL http://biohackathon.org/resource/faldo which is developed under source code control at https://github.com/JervenBolleman/FALDO hosted by GitHub Inc, where everyone is free to suggest extensions and improvements and if required extend FALDO to meet their unique requirements. FALDO currently uses the Creative Commons Attribution Zero 1.0 Public Domain dedication license, making FALDO available to use and reuse free of charge.
The ontology is shared in the Turtle (http://www.w3.org/TR/turtle/) RDF syntax, which can be automatically converted to another RDF syntax such as RDF/XML if required.

Author details
[1]Swiss-Prot group, SIB Swiss Institute of Bioinformatics, Centre Medical Universitaire, 1 rue Michel, Servet, 1211 Geneva 4, Switzerland. [2]Genomics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, US. [3]CeRSA, Parco Tecnologico Padano, Lodi 26900, Italy. [4]CODAMONO, 5-121 Marion Street, Toronto, Ontario M6R 1E6, Canada. [5]Stanford Center for Biomedical Informatics Research, 1265 Welch Road, Room X223, Stanford, CA 94305-5479, US. [6]Integrative Biology Program, Istituto Nazionale Genetica Molecolare, Milan, Italy. [7]University of California, Berkeley, Berkeley, CA, USA. [8]Department of Computer Science, Aberystwyth SY23 3DB, UK. [9]Center for Information Biology, National Institute of Genetics, Research Organization of Information and Systems, 1111 Yata, Mishima, Shizuoka 411-08540, Japan. [10]Database Center for Life Science, Research Organization of Information and Systems, 2-11-16, Yayoi, Bunkyo-ku, Tokyo, 113-0032, Japan. [11]The James Hutton Institute, Dundee DD2 5DA, UK.

Received: 5 February 2014 Accepted: 17 March 2016
Published online: 13 June 2016

References

1. Sanger F. The terminal peptides of insulin. Biochem J. 1949;45(5):563–74.
2. Dayhoff MO, Eck RV, Foundation NBR. Atlas of Protein Sequence and Structure. Silver Spring (Maryland): National Biomedical Research Foundation; 1965.
3. Reese MG, Moore B, Batchelor C, Salas F, Cunningham F, Marth GT, Stein L, Flicek P, Yandell M, Eilbeck K. A standard variation file format for human genome sequences. Genome Biol. 2010;11(R88):. doi:10.1186/gb-2010-11-8-r88.
4. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JGR, Korf I, Lapp H, Lehväslaiho H, Matsalla C, Mungall CJ, Osborne BI, Pocock MR, Schattner P, Senger M, Stein LD, Stupka E, Wilkinson MD, Birney E. The Bioperl toolkit: Perl modules for the life sciences. Genome Res. 2002;12(10):1611–8. doi:10.1101/gr.361602.
5. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, de Hoon MJL. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422–3. doi:10.1093/bioinformatics/btp163.
6. Goto N, Prins P, Nakao M, Bonnal R, Aerts J, Katayama T. BioRuby: bioinformatics software for the Ruby programming language. Bioinformatics. 2010;26(20):2617–9. doi:10.1093/bioinformatics/btq475.
7. Prlić A, Yates A, Bliven SE, Rose PW, Jacobsen J, Troshin PV, Chapman M, Gao J, Koh CH, Foisy S, Holland R, Rimsa G, Heuer ML, Brandstätter-Müller H, Bourne PE, Willis S. BioJava: an open-source framework for bioinformatics in 2012. Bioinformatics. 2012;28(20):2693–5. doi:10.1093/bioinformatics/bts494.
8. Katayama T, Arakawa K, Nakao M, Ono K, Aoki-Kinoshita KF, Yamamoto Y, Yamaguchi A, Kawashima S, Chun HW, Aerts J, Aranda B, Barboza LH, Bonnal RJ, Bruskiewich R, Bryne JC, Fernández JM, Funahashi A, Gordon PM, Goto N, Groscurth A, Gutteridge A, Holland R, Kano Y, Kawas EA, Kerhornou A, Kibukawa E, Kinjo AR, Kuhn M, Lapp H, Lehvaslaiho H, Nakamura H, Nakamura Y, Nishizawa T, Nobata C, Noguchi T, Oinn TM, Okamoto S, Owen S, Pafilis E, Pocock M, Prins P, Ranzinger R, Reisinger F, Salwinski L, Schreiber M, Senger M, Shigemoto Y, Standley DM, Sugawara H, Tashiro T, Trelles O, Vos RA, Wilkinson MD, York W, Zmasek CM, Asai K, Takagi T. The DBCLS BioHackathon: standardization and interoperability for bioinformatics web services and workflows. J Biomed Semantics. 2010;1:8. doi:10.1186/2041-1480-1-8.
9. Katayama T, Wilkinson MD, Vos R, Kawashima T, Kawashima S, Nakao M, Yamamoto Y, Chun HW, Yamaguchi A, Kawano S, Aerts J, Aoki-Kinoshita KF, Arakawa K, Aranda B, Bonnal RJ, Fernández JM, Fujisawa T, Gordon PM, Goto N, Haider S, Harris T, Hatakeyama T, Ho I, Itoh M, Kasprzyk A, Kido N, Kim YJ, Kinjo AR, Konishi F, Kovarskaya Y, von Kuster G, Labarga A, Limviphuvadh V, McCarthy L, Nakamura Y, Nam Y, Nishida K, Nishimura K, Nishizawa T, Ogishima S, Oinn T, Okamoto S, Okuda S, Ono K, Oshita K, Park KJ, Putnam N, Senger M, Severin J, Shigemoto Y, Sugawara H, Taylor J, Trelles O, Yamasaki C, Yamashita R, Satoh N, Takagi T. The 2nd DBCLS BioHackathon: interoperable bioinformatics Web services for integrated applications. J Biomed Semantics. 2011;2:6. doi:10.1186/2041-1480-2-4.
10. Katayama T, Wilkinson MD, Micklem G, Kawashima S, Yamaguchi A, Nakao M, Yamamoto Y, Okamoto S, Oouchida K, Chun HW, Aerts J, Afzal H, Antezana E, Arakawa K, Aranda B, Belleau F, Bolleman J, Bonnal RJP, Chapman B, Cock PJA, Eriksson T, Gordon PMK, Goto N, Hayashi K, Horn H, Ishiwata R, Kaminuma E, Kasprzyk A, Kawaji H, Kido N, Kim YJ, Kinjo AR, Konishi F, Kwon KH, Labarga A, Lamprecht AL, Lin Y, Lindenbaum P, McCarthy L, Morita H, Murakami K, Nagao K, Nishida K, Nishimura K, Nishizawa T, Ogishima S, Ono K, Oshita K, Park KJ, Prins P, Saito TL, Samwald M, Satagopam VP, Shigemoto Y, Smith R, Splendiani A, Sugawara H, Taylor J, Vos RA, Withers D, Yamasaki C, Zmasek CM, Kawamoto S, Okubo K, Asai K, Takagi T. The 3rd DBCLS BioHackathon: improving life science data integration with Semantic Web technologies. J Biomed Semantics. 2013;4:6. doi:10.1186/2041-1480-4-6.
11. Katayama T, Wilkinson MD, Aoki-Kinoshita KF, Kawashima S, Yamamoto Y, Yamaguchi A, Okamoto S, Kawano S, Kim JD, Wang Y, Wu H, Kano Y, Ono H, Bono H, Kocbek S, Aerts J, Akune Y, Antezana E, Arakawa K, Aranda B, Baran J, Bolleman J, Bonnal RJP, Buttigieg PL, Campbell MP, Chen Y-A, Chiba H, Cock PJA, Cohen KB, Constantin A, Duck G, Dumontier M, Fujisawa T, Fujiwara T, Goto N, Hoehndorf R, Igarashi Y, Itaya H, Ito M, Iwasaki W, Kalaš M, Katoda T, Kim T, Kokubu A, Komiyama Y, Kotera M, Laibe C, Lapp H, Lütteke T, Marshall MS, Mori T, Mori H, Morita M, Murakami K, Nakao M, Narimatsu H, Nishide H, Nishimura Y, Nystrom-Persson J, Ogishima S, Okamura Y, Okuda S, Oshita K, Packer NH, Prins P, Ranzinger R, Rocca-Serra P, Sansone S, Sawaki H, Shin SH, Splendiani A, Strozzi F, Tadaka S, Toukach P, Uchiyama I, Umezaki M, Vos R, Whetzel PL, Yamada I, Yamasaki C, Yamashita R, York WS, Zmasek CM, Kawamoto S, Takagi T. Biohackathon series in 2011 and 2012: penetration of ontology and Linked Data in life science domains. J Biomed Semantics. 2014;5:5. doi:10.1186/2041-1480-5-5.
12. The UniProt Consortium. Update on activities at the Universal Protein Resource (UniProt) in 2013. Nucleic Acids Res. 2013;41(D1):43–7. doi:10.1093/nar/gks1068. http://nar.oxfordjournals.org/content/41/D1/D43.full.pdf+html.
13. Ogasawara O, Mashima J, Kodama Y, Kaminuma E, Nakamura Y, Okubo K, Takagi T. Ddbj new system and service refactoring. Nucleic Acids Res. 2013;41(D1):25–9. doi:10.1093/nar/gks1152. http://nar.oxfordjournals.org/content/41/D1/D25.full.pdf+html.
14. Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. Genbank. Nucleic Acids Res. 2013;41(D1):36–42.

doi:10.1093/nar/gks1195. http://nar.oxfordjournals.org/content/41/D1/D36.full.pdf+html.
15. Cochrane G, Alako B, Amid C, Bower L, Cerdeño-Tárraga A, Cleland I, Gibson R, Goodgame N, Jang M, Kay S, Leinonen R, Lin X, Lopez R, McWilliam H, Oisel A, Pakseresht N, Pallreddy S, Park Y, Plaister S, Radhakrishnan R, Rivière S, Rossello M, Senf A, Silvester N, Smirnov D, ten Hoopen P, Toribio A, Vaughan D, Zalunin V. Facing growth in the European Nucleotide Archive. Nucleic Acids Res. 2013;41(D1):30–5. doi:10.1093/nar/gks1175. http://nar.oxfordjournals.org/content/41/D1/D30.full.pdf+html.
16. Toukach PV. Bacterial carbohydrate structure database 3: Principles and realization. J Chem Inform Modeling. 2011;51(1):159–70. doi:10.1021/ci100150d. http://pubs.acs.org/doi/pdf/10.1021/ci100150d.
17. Ranzinger R, Herget S, von der Lieth C-W, Frank M. GlycomeDB – a unified database for carbohydrate structures. Nucleic Acids Res. 2011;39(suppl 1):373–6. doi:10.1093/nar/gkq1014. http://nar.oxfordjournals.org/content/39/suppl_1/D373.full.pdf+html.
18. Lütteke T, Bohne-Lang A, Loss A, Goetz T, Frank M, von der Lieth C-W. Glycosciences.de: an internet portal to support glycomics and glycobiology research. Glycobiology. 2006;16(5):71–81. doi:10.1093/glycob/cwj049. http://glycob.oxfordjournals.org/content/16/5/71R.full.pdf+html.
19. Akune Y, Hosoda M, Kaiya S, Shinmachi D, Aoki-Kinoshita KF. The RINGS resource for glycome informatics analysis and data mining on the web. OMICS: J Integrative Biol. 2010;14(4):475–86. doi:10.1089/omi.2009.0129.
20. Campbell MP, Peterson R, Mariethoz J, Gasteiger E, Akune Y, Aoki-Kinoshita KF, Lisacek F, Packer NH. Unicarbkb: Building a knowledge platform for glycoproteomics. Underconsideration NAR Databases. 2014. doi:10.1093/nar/gkt1128.
21. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M. The Sequence Ontology: A tool for the unification of genome annotations. Genome Biol. 2005;6(5):44. doi:10.1186/gb-2005-6-5-r44.
22. Vakatov D E. Biological sequence data model. Technical report, NCBI. 2013. http://www.ncbi.nlm.nih.gov/toolkit/doc/book/ch_datamod/#ch_datamod.Locations_on_Biologi.
23. Le Hors A, Solbrig H, Eric Prud'hommeaux. Rdf validation workshop report, practical assurances for quality rdf data. Technical report, W3C. 2013. http://www.w3.org/2012/12/rdf-val/report.
24. Baran J, Durgahee BSB, Eilbeck K, Antezana E, Hoehndorf R, Dumontier M. GFVO: the Genomic Feature and Variation Ontology. 2015. PeerJ3:e933. https://doi.org/10.7717/peerj.933.
25. Skinner ME, Uzilov AV, Stein LD, Mungall CJ, Holmes IH. JBrowse: A next-generation genome browser. Genome Res. 2009;19:1630–8. doi:10.1101/gr.094607.109.
26. Devisscher M, De Meyer T, Van Criekinge W, Dawyndt P. An ontology based query engine for querying biological sequences. EMBnet J. 2013;19(B):51–5.
27. Mungall CJ, Emmert DB, The FlyBase Consortium. A Chado case study: an ontology-based modular schema for representing genome-associated biological information. Bioinformatics. 2007;23(13):337–46. doi:10.1093/bioinformatics/btm189. http://bioinformatics.oxfordjournals.org/content/23/13/i337.full.pdf+html.
28. Alekseyenko AV, Lee CJ. Nested containment list (NCList): a new algorithm for accelerating interval query of genome alignment and interval databases. Bioinformatics. 647. 2007. doi:10.1093/bioinformatics/btl647.
29. Beckett D, Berners-Lee T, Prud'hommeaux E, Carothers G. Rdf 1.1 turtle - terse rdf triple language. Technical report, W3C. 2014. http://www.w3.org/TR/turtle/.
30. Sporny M, Kellogg G, Lanthaler M. Json-ld 1.0 - a json-based serialization for linked data. Technical report, W3C. 2014. http://www.w3.org/TR/json-ld/.