The Sequence Ontology Project

GENERIC FEATURE FORMAT VERSION 3

SUMMARY

```
Author:  Lincoln Stein
Date:    15 December 2010
Version: 1.20
```

Although there are many richer ways of representing genomic features via XML, the stubborn persistence of a variety of ad-hoc tab-delimited flat file formats declares the bioinformatics community's need for a simple format that can be modified with a text editor and processed with shell tools like grep. The GFF format, although widely used, has fragmented into multiple incompatible dialects. When asked why they have modified the published Sanger specification, bioinformaticists frequently answer that the format was insufficient for their needs, and they needed to extend it. The proposed GFF3 format addresses the most common extensions to GFF, while preserving backward compatibility with previous formats. The new format:

```
    1) adds a mechanism for representing more than one level
       of hierarchical grouping of features and subfeatures.
    2) separates the ideas of group membership and feature name/id
    3) constrains the feature type field to be taken from a controlled
       vocabulary.
    4) allows a single feature, such as an exon, to belong to more than
       one group at a time.
    5) provides an explicit convention for pairwise alignments
    6) provides an explicit convention for features that occupy disjunct
regions
```

Online Validator

An online GFF3 validator is available at
http://modencode.oicr.on.ca/cgi-bin/validate_gff3_online
It is limited to files of 3,000,000 lines or less. If you wish to validate larger files, please use the command-line version which can be downloaded from the same site.

DESCRIPTION OF THE FORMAT

The format consists of 9 columns, separated by tabs (NOT spaces). The following characters must be escaped using URL escaping conventions (%XX hex codes):

```
    tab - %09
    newline - %0A
    carriage return - %0D
    control characters - %00 through %1F, %7F
```

The following characters have reserved meanings and must be escaped when used in other contexts:

```
    ;  (semicolon) - %3B
    =  (equals) - %3D
    %  (percent) - %25
    &  (ampersand) - %26
    ,  (comma) - %2C
```

News

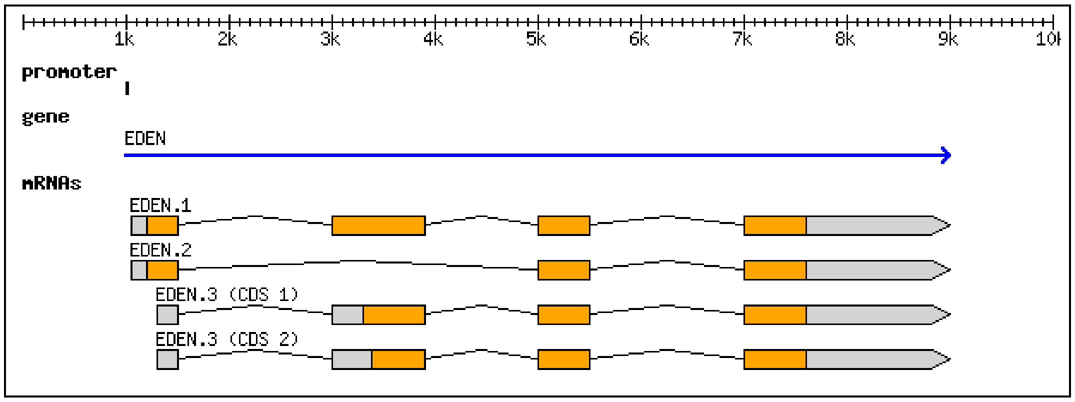
- **September 2012** Karen Eilbeck presented a paper titled Using GVF for Clinical Annotation of Personal Genomes at the AIMM workshop part of ECCB12, in Basel Switzerland.
- **July 2012** Mike Bada presented a paper titled 'Efforts toward a More Consistent and Interoperable Sequence Ontology' at the International Conference on Biomedical Ontology in Graz, Austria.
- **September 2011** Mike Bada, University of Colorado, joins to SO team, to work on interoperability with the BFO
- **September 2011** The NHGRI have funded SO's renewal R01
- **March 2011** SOBA workshop presented at GMOD Spring Training.
- **Feb 11th 2011** The CVS repository on sourceforge is back online after the hacking incident.
- **January 2011** SO presented at Synthetic Biology Workshop.
- **December 2010** 10Gen Data Set v1.04 available.
- **December 2010** GVF Specification v1.04 available.
- **November 2010** Release 2.4.4 available.

```
Unescaped quotation marks, backslashes and other ad-hoc escaping
conventions that have been added to the GFF format are explicitly
forbidden

Note that unescaped spaces are allowed within fields, meaning that
parsers must split on tabs, not spaces.

Undefined fields are replaced with the "." character, as described in
the original GFF spec.

Column 1: "seqid"

The ID of the landmark used to establish the coordinate system for the
current feature. IDs may contain any characters, but must escape any
characters not in the set [a-zA-Z0-9.:^*$@!+_?-|].  In particular, IDs
may not contain unescaped whitespace and must not begin with an
unescaped ">".

Column 2: "source"

The source is a free text qualifier intended to describe the algorithm
or operating procedure that generated this feature.  Typically this is
the name of a piece of software, such as "Genescan" or a database
name, such as "Genbank."  In effect, the source is used to extend the
feature ontology by adding a qualifier to the type creating a new
composite type that is a subclass of the type in the type column.

Column 3: "type"

The type of the feature (previously called the "method").  This is
constrained
to be either: (a)a term from the "lite" version of the Sequence Ontology -
SOFA, a term from the full Sequence Ontology - it must be an is_a child of
sequence_feature (SO:0000110) or (c) a SOFA or SO accession number.  The
latter alternative is distinguished using the syntax SO:000000.

Columns 4 & 5: "start" and "end"

The start and end of the feature, in 1-based integer coordinates,
relative to the landmark given in column 1.  Start is always less than
or equal to end.  For features that cross the origin of a circular feature
(e.g. most bacterial genomes, plasmids, and some viral genomes), the
requirement for start to be less than or equal to end is satisfied by
making end = the position of the end + the length of the landmark feature.

For zero-length features, such as insertion sites, start equals end
and the implied site is to the right of the indicated base in the
direction of the landmark.

Column 6: "score"

The score of the feature, a floating point number.  As in earlier
versions of the format, the semantics of the score are ill-defined.
It is strongly recommended that E-values be used for sequence
similarity features, and that P-values be used for ab initio gene
prediction features.

Column 7: "strand"

The strand of the feature.  + for positive strand (relative to the
landmark), - for minus strand, and . for features that are not
stranded.  In addition, ? can be used for features whose strandedness
is relevant, but unknown.

Column 8: "phase"

For features of type "CDS", the phase indicates where the feature
begins with reference to the reading frame.  The phase is one of the
integers 0, 1, or 2, indicating the number of bases that should be
removed from the beginning of this feature to reach the first base of
the next codon. In other words, a phase of "0" indicates that the next
codon begins at the first base of the region described by the current
line, a phase of "1" indicates that the next codon begins at the
second base of this region, and a phase of "2" indicates that the
codon begins at the third base of this region. This is NOT to be
confused with the frame, which is simply start modulo 3.

For forward strand features, phase is counted from the start
field. For reverse strand features, phase is counted from the end
field.

The phase is REQUIRED for all CDS features.

Column 9: "attributes"

A list of feature attributes in the format tag=value.  Multiple
tag=value pairs are separated by semicolons.  URL escaping rules are
used for tags or values containing the following characters: ",=;".
Spaces are allowed in this field, but tabs must be replaced with the
%09 URL escape.

These tags have predefined meanings:

    ID     Indicates the ID of the feature.  IDs for each feature
```

- **September 2010** A standard variation file format for human genome sequences in Genome Biology.
- **August 2010** 'Toward a Richer Representation of Sequence Variation in the Sequence Ontology' with Mike Bada accepted by AIMM Workshop.
- **June 2010 Release 2.4.3 available.**
- **May 2010** SOBA paper available from Nucleic Acids Research.
- **April 2010** Release 2.4.2 available.
- **March 2010** New SO paper out: 'Evolution of the Sequence Ontology terms and relationships' in the Journal of Biomedical Informatics. A pre-print of this paper is available here.
- **February 2010:** Sequence Ontology Bioinformatics Analysis (SOBA) is available. Genome annotations in GFF3 can be uploaded and analysed.
- **2009 December** Release 2.4.1 available.
- **2009 October** Release 2.4 available.
- **2009 July** SO presented at the International Conference of Biomedical Ontology.
- **2009 June** Chris Conley - Undergrad from BYU joins SO for the summer
- **2009 May** Graduate student, John Naylor joins SO.
- **2009 February 22:** A new SO related paper Quantitative Measures for the Management and Comparison of Annotated Genomes K Eilbeck *et. al.*
- **2009 January 4-6:** SO represented at the RNA Ontology consortium meeting in Cambridge UK.
- **2008 December 1:** A

```
        must be unique within the scope of the GFF file.  In the
        case of discontinuous features (i.e. a single feature that
        exists over multiple genomic locations) the same ID may
        appear on multiple lines.  All lines that share an ID
        collectively represent a single feature.

    Name    Display name for the feature.  This is the name to be
            displayed to the user.  Unlike IDs, there is no requirement
            that the Name be unique within the file.

    Alias   A secondary name for the feature.  It is suggested that
            this tag be used whenever a secondary identifier for the
            feature is needed, such as locus names and
            accession numbers.  Unlike ID, there is no requirement
            that Alias be unique within the file.

    Parent  Indicates the parent of the feature.  A parent ID can be
            used to group exons into transcripts, transcripts into
            genes, an so forth.  A feature may have multiple parents.
            Parent can *only* be used to indicate a partof
            relationship.

    Target  Indicates the target of a nucleotide-to-nucleotide or
            protein-to-nucleotide alignment.  The format of the
            value is "target_id start end [strand]", where strand
            is optional and may be "+" or "-".  If the target_id
            contains spaces, they must be escaped as hex escape %20.

    Gap    The alignment of the feature to the target if the two are
           not collinear (e.g. contain gaps).  The alignment format is
           taken from the CIGAR format described in the
           Exonerate documentation.
           (http://cvsweb.sanger.ac.uk/cgi-bin/cvsweb.cgi/exonerate
           ?cvsroot=Ensembl).  See "THE GAP ATTRIBUTE" for a description
           of this format.

    Derives_from
           Used to disambiguate the relationship between one
           feature and another when the relationship is a temporal
           one rather than a purely structural "part of" one.  This
           is needed for polycistronic genes.  See "PATHOLOGICAL CASES"
           for further discussion.

    Note    A free text note.

    Dbxref  A database cross reference.  See the section
            "Ontology Associations and Db Cross References" for
            details on the format.

    Ontology_term  A cross reference to an ontology term.  See
            the section "Ontology Associations and Db Cross References"
            for details.

    Is_circular  A flag to indicate whether a feature is circular.

Multiple attributes of the same type are indicated by separating the
values with the comma "," character, as in:

       Parent=AF2312,AB2812,abc-3

In addition to Parent, the Alias, Note, DBxref and Ontology_term
attributes can have multiple values.

Note that attribute names are case sensitive.  "Parent" is not the
same as "parent".

All attributes that begin with an uppercase letter are reserved for
later use.  Attributes that begin with a lowercase letter can be used
freely by applications.
```

THE CANONICAL GENE



paper from the BioSapiens project describing protein features in SO is published. The Protein Feature Ontology: A tool for the unification of protein feature annotations. GA Reeves *et. al.*

FIGURE 1

This section describes the representation of a protein-coding gene in GFF3. To illustrate how a canonical gene is represented, consider Figure 1 (figure1.png). This indicates a gene named EDEN extending from position 1000 to position 9000. It encodes three alternatively-spliced transcripts named EDEN.1, EDEN.2 and EDEN.3, the last of which has two alternative translational start sites leading to the generation of two protein coding sequences.

There is also an identified transcriptional factor binding site located 50 bp upstream from the transcriptional start site of EDEN.1 and EDEN2.

Here is how this gene should be described using GFF3:

```
 0  ##gff-version   3
 1  ##sequence-region   ctg123 1 1497228
 2  ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN

 3  ctg123 . TF_binding_site 1000  1012  .  +  .
ID=tfbs00001;Parent=gene00001

 4  ctg123 . mRNA            1050  9000  .  +  .
ID=mRNA00001;Parent=gene00001;Name=EDEN.1
 5  ctg123 . mRNA            1050  9000  .  +  .
ID=mRNA00002;Parent=gene00001;Name=EDEN.2
 6  ctg123 . mRNA            1300  9000  .  +  .
ID=mRNA00003;Parent=gene00001;Name=EDEN.3

 7  ctg123 . exon            1300  1500  .  +  .
ID=exon00001;Parent=mRNA00003
 8  ctg123 . exon            1050  1500  .  +  .
ID=exon00002;Parent=mRNA00001,mRNA00002
 9  ctg123 . exon            3000  3902  .  +  .
ID=exon00003;Parent=mRNA00001,mRNA00003
10  ctg123 . exon            5000  5500  .  +  .
ID=exon00004;Parent=mRNA00001,mRNA00002,mRNA00003
11  ctg123 . exon            7000  9000  .  +  .
ID=exon00005;Parent=mRNA00001,mRNA00002,mRNA00003

12  ctg123 . CDS             1201  1500  .  +  0
ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
13  ctg123 . CDS             3000  3902  .  +  0
ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
14  ctg123 . CDS             5000  5500  .  +  0
ID=cds00001;Parent=mRNA00001;Name=edenprotein.1
15  ctg123 . CDS             7000  7600  .  +  0
ID=cds00001;Parent=mRNA00001;Name=edenprotein.1

16  ctg123 . CDS             1201  1500  .  +  0
ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
17  ctg123 . CDS             5000  5500  .  +  0
ID=cds00002;Parent=mRNA00002;Name=edenprotein.2
18  ctg123 . CDS             7000  7600  .  +  0
ID=cds00002;Parent=mRNA00002;Name=edenprotein.2

19  ctg123 . CDS             3301  3902  .  +  0
ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
20  ctg123 . CDS             5000  5500  .  +  1
ID=cds00003;Parent=mRNA00003;Name=edenprotein.3
21  ctg123 . CDS             7000  7600  .  +  1
ID=cds00003;Parent=mRNA00003;Name=edenprotein.3

22  ctg123 . CDS             3391  3902  .  +  0
ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
23  ctg123 . CDS             5000  5500  .  +  1
ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
24  ctg123 . CDS             7000  7600  .  +  1
ID=cds00004;Parent=mRNA00003;Name=edenprotein.4
```

Lines beginning with ## are pragmas that provide meta-information about the document. Blank lines and lines beginning with a single # are ignored.

Line 0 gives the GFF version using the ##gff-version pragma. Line 1 indicates the boundaries of the region being annotated (a 1,497,228 bp region named "ctg123") using the ##sequence-region pragma.

Line 2 defines the boundaries of the gene. Column 9 of this line assigns the gene an ID of gene00001, and a human-readable name of EDEN. Because the gene is not part of a larger feature, it has no Parent.

Line 3 annotates the transcriptional factor binding site. Since it is logically part of the gene, its Parent attribute is gene00001.

Lines 4-6 define this gene's three spliced transcripts, one line for the full extent of each of the mRNAs. These features are necessary to act as parents for the four CDSs which derive from them, as well as the structural parents of the five exons in the alternative splicing set.

Lines 7-11 identify the five exons. The Parent attributes indicate which mRNAs the exons belong to. Notice that several of the exons share the same parents, using the comma symbol to indicate multiple parentage.

Lines 12-24 denote this gene's four CDSs. Each CDS belongs to one of the mRNAs. cds00003 and cds00004, which correspond to alternative start codons, belong to the same mRNA.

Note that several of the features, including the gene, its mRNAs and the CDSs, all have Name attributes. This attributes assigns those features a public name, but is not mandatory. The ID attributes are only mandatory for those features that have children (the gene and mRNAs), or for those that span multiple lines. The IDs do not have meaning outside the file in which they reside. Hence, a slightly simplified version of this file would look like this:

```
  ##gff-version   3
  ##sequence-region   ctg123 1 1497228
  ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
  ctg123 . TF_binding_site 1000  1012  .  +  .  Parent=gene00001
  ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001
  ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00002;Parent=gene00001
  ctg123 . mRNA            1300  9000  .  +  .  ID=mRNA00003;Parent=gene00001
  ctg123 . exon            1300  1500  .  +  .  Parent=mRNA00003
  ctg123 . exon            1050  1500  .  +  .  Parent=mRNA00001,mRNA00002
  ctg123 . exon            3000  3902  .  +  .  Parent=mRNA00001,mRNA00003
  ctg123 . exon            5000  5500  .  +  .
Parent=mRNA00001,mRNA00002,mRNA00003
  ctg123 . exon            7000  9000  .  +  .
Parent=mRNA00001,mRNA00002,mRNA00003
  ctg123 . CDS             1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001
  ctg123 . CDS             3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001
  ctg123 . CDS             5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001
  ctg123 . CDS             7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001
  ctg123 . CDS             1201  1500  .  +  0  ID=cds00002;Parent=mRNA00002
  ctg123 . CDS             5000  5500  .  +  0  ID=cds00002;Parent=mRNA00002
  ctg123 . CDS             7000  7600  .  +  0  ID=cds00002;Parent=mRNA00002
  ctg123 . CDS             3301  3902  .  +  0  ID=cds00003;Parent=mRNA00003
  ctg123 . CDS             5000  5500  .  +  1  ID=cds00003;Parent=mRNA00003
  ctg123 . CDS             7000  7600  .  +  1  ID=cds00003;Parent=mRNA00003
  ctg123 . CDS             3391  3902  .  +  0  ID=cds00004;Parent=mRNA00003
  ctg123 . CDS             5000  5500  .  +  1  ID=cds00004;Parent=mRNA00003
  ctg123 . CDS             7000  7600  .  +  1  ID=cds00004;Parent=mRNA00003
```

NOTE 1 - SO or SOFA IDs: If using the SO (or SOFA) IDs rather than the short names1
("mRNA" etc), use the following mappings:

gene	SO:0000704
mRNA	SO:0000234
exon	SO:0000147
cds	SO:0000316

Other mRNA parts that you might wish to use are

intron	SO:0000188 (redundant with exon)
polyA_sequence	SO:0000610 (part of the three_prime_UTR)
polyA_site	SO:0000553 (part of the gene)
five_prime_UTR	SO:0000204
three_prime_UTR	SO:0000205

NOTE 2 - "Orphan" exons CDSs, and other features. Ab initio gene prediction programs call hypothetical exons and CDS's that are attached to the genomic sequence and not necessarily to a known transcript. To handle these features, you may either (1) create a placeholder mRNA and use it as the parent for the exon and CDS subfeatures; or (2) attach the exons and CDSs directly to the gene. This is allowed by SO because of the transitive nature of the part_of relationship.

NOTE 3 - UTRs, splice sites and translational start and stop sites. These are implied by the combination of exon and CDS and do not need to be explicitly annotated as part of the canonical gene. In the case of annotating predicted splice or translational start/stop sites independently of a particular gene, it is suggested that they be attached directly to the genomic sequence and not to a gene or a subpart of a gene.

NOTE 4 - CDS features MUST have have a defined phase field. Otherwise it is not possible to infer the correct polypeptides corresponding to partially annotated genes.

NOTE 5 - The START and STOP codons are included in the CDS. That is, if the locations of the start and stop codons are known, the first three base pairs of the CDS should correspond to the start codon and the last three correspond the stop codon.

CIRCULAR GENOMES

For a circular genome, the landmark feature should include Is_circular=true in column 9. In the example below, from bacteriophage f1, gene II extends across the origin from positions 6477-831. The feature end is given as length of the landmark feature, J02448, plus the distance from the origin to the end of gene II (6407 + 831 = 7238).

```
    ##gff-version 3
    # organism Enterobacteria phage f1
    # Note Bacteriophage f1, complete genome.
    J02448      GenBank region  1       6407    .       +       .
ID=J02448;Name=J02448;Is_circular=true;
    J02448      GenBank CDS         6006        7238    .       +       0
ID=geneII;Name=II;Note=protein II;
```

REPRESENTING SPLICED NON-CODING TRANSCRIPTS

For spliced non-coding transcripts, such as those produced by some processed snRNAs and viruses, use a parent feature of "noncoding_transcript" and a child of "exon."

PARENT (PART-OF) RELATIONSHIPS

The reserved Parent attribute can be used to establish a part-of relationship between two features. A feature that has the Parent attribute set is interpreted as asserting that it is a part of the specified Parent feature.

Features must respect the Sequence Ontology Part-Of relationships. A Parent relationship between two features that is not one of the Part-Of relationships listed in SO should trigger a parse exception Similarly, a set of Parent relationships that would cause a cycle should also trigger an exception.

The GFF3 format does not enforce a rule in which features must be wholly contained within the location of their parents, since some elements of the Sequence Ontology (e.g. enhancers in genes) allow for distant cis relationships.

THE GAP ATTRIBUTE

Protein and nucleotide alignment features typically consist of two sequences, the reference sequence and the "target", and are not always colinear. For example, consider the following alignment between an EST ("EST23") and a segment of the genome ("Chr3"):

```
        Chr3  (reference)  1 CAAGACCTAAACTGGAT-TCCAAT  23
        EST23 (target)     1 CAAGACCT---CTGGATATCCAAT  21
```

Previous versions of the GFF format would represent this alignment as three colinear segments, but this made it difficult to reconstruct the gapped alignment. GFF3 recommends representing gapped alignments explicitly with the "Gap" attribute. The Gap attribute's format consists of a series of (operation,length) pairs separated by space characters, for example "M8 D3 M6". Each operation is a single-letter code:

```
       Code      Operation
       ----      ---------

        M        match
        I        insert a gap into the reference sequence
        D        insert a gap into the target (delete from reference)
        F        frameshift forward in the reference sequence
        R        frameshift reverse in the reference sequence
```

In the alignment between EST23 and Chr3 shown above, Chr3 is the reference sequence referred to in the first column of the GFF3 file, and EST23 is the sequence referred to by the Target attribute. This gives a Gap string of "M8 D3 M6 I1 M6". The full GFF match line will read:

```
   Chr3 . Match 1 23 . . . ID=Match1;Target=EST23 1 21;Gap=M8 D3 M6 I1 M6
```

For protein to nucleotide matches, the M, I and D operations apply to amino acid residues in the target and nucleotide base pairs in the reference in a 1:3 residue. That is, "M2" means to match two amino residues in the target to six base pairs in the reference. Hence this alignment:

```
 100 atgaaggag---gttattgcgaatgtcggcggt
   1 M..K..E..V..V..I..-..N..V..G..G..
```

Corresponds to this GFF3 Line:

```
 ctg123 . nucleotide_to_protein 100 129 . + . ID=match008;Target=p101 1
10;Gap=M3 I1 M2 D1 M4
```

In addition, the Gap attribute provides <F>orward and <R>everse frameshift operators to allow for frameshifts in the alignment. These are in nucleotide coordinates: a forward frameshift skips forward the indicated number of base pairs, while a reverse frameshift moves backwards. Examples:

```
 100 atgaaggag---gttattgaatgtcggcggt     Gap=M3 I1 M2 F1 M4
   1 M..K..E..V..V..I...
                       N..V..G..G


 100 atgaaggag---gttataatgtcggcggt        Gap=M3 I1 M2 R1 M4
   1 M..K..E..V..V..I.
                     N..V..G..G
```

ALIGNMENTS

In the SO, an alignment between the reference sequence and another sequence is called a "match". In addition to the generic "match" type, there are the subclasses "cDNA_match," "EST_match," "translated_nucleotide_match," "nucleotide_to_protein_match," and "nucleotide_motif."

Matches typically contain gaps; matches broken up by large gaps are usually called "HSPs" (high-scoring segment pair), and previous incarnations of GFF have handled gapped alignments by breaking up the alignment into a series of ungapped HSPs.

The SO does not have an HSP type. Instead, gapped matches are represented as a single feature that occupies a discontinuous location on the reference sequence. Figure 2 shows the same gene as before, but with a new track added showing an alignment of a sequenced cDNA to the genome. For the purposes of illustration, we have shown the regions of alignment to be exact across the three exons of the second spliced transcript (EDEN.2).

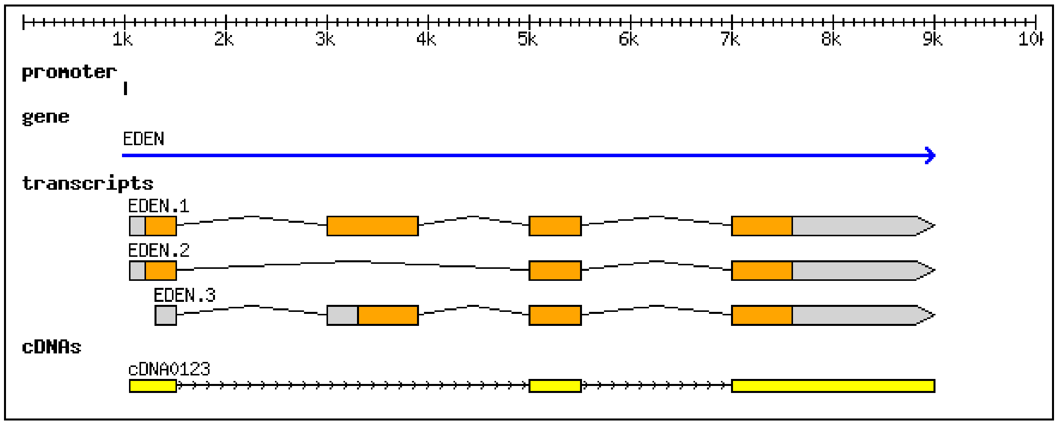


FIGURE 2

The recommended way to represent this alignment is with a single feature of type "cDNA_match" and a Gap attribute that indicates that the alignment is in three segments:

```
 ctg123 . cDNA_match 1050  9000  6.2e-45  +  .
    ID=match00001;Target=cdna0123 12 2964;Gap=M451 D3499 M501 D1499 M2001
```

Parsed out, the Target attribute indicates that the sequence named "cdna0123" between bases 12 and 2964 (in cdna coordinates) aligns to bases 1050 to 9000 of ctg123. The Gap attribute is easier to read when spaces are inserted:

```
     M451        match 451 bases
     D3499       skip 3499 bases in the reference ctg123 sequence
     M501        match the next 501 bases
     D1499       skip 1499 bases in the reference ctg123
     M2001       match the next 2001 bases
```

Note that the matched region is 2953 bases, which corresponds exactly to the matching subsequence [12,2964] of the target. Extra bases in the cDNA which would cause gaps in the reference sequence would be indicated using the CIGAR "I" notation.

Another important item to note is that the ID corresponds to the Match and not to the target sequence. This avoids the confusion that has occurred in previous incarnations of GFF which made it impossible to distinguish between a particular alignment of a target sequence to the genome and all alignments of a target sequence to the genome.

A limitation of the Gap representation is that the entire alignment shares the same score (column 6). To give each component of the match a separate score, it can be broken across multiple lines as shown here:

```
 ctg123 . cDNA_match 1050  1500  5.8e-42  +  . ID=match00001;Target=cdna0123
12 462
 ctg123 . cDNA_match 5000  5500  8.1e-43  +  . ID=match00001;Target=cdna0123
463 963
 ctg123 . cDNA_match 7000  9000  1.4e-40  +  . ID=match00001;Target=cdna0123
964 2964
```

Notice that the ID is the same across each of the three lines, indicating that these lines all refer to a single feature, the Match. Each aligning segment, however has a distinct score and Target region.

The two types of representations can be mixed, allowing large aligned segments to have their own GFF line and score, while small gaps within them are represented using a Gap attribute.

Matches can align to either the + or the - strand of the reference sequence. This should be denoted in the seventh column of the GFF line and *not* by changing the order of the start and end positions in the Target attribute. To illustrate this, Figure 3 adds an EST pair to the annotation. The two ESTs, mjm1123.5 and mum1123.3 correspond to 5' and 3' EST reads from the same cDNA clone. The following GFF3 lines describe them:

```
 ctg123 . EST_match 1200  3200  2.2e-30  +  .
     ID=match00002;Target=mjm1123.5 5 506;Gap=M301 D1499 M201

 ctg123 . EST_match 7000  9000  7.4e-32  -  .
     ID=match00003;Target=mjm1123.3 1 502;Gap=M101 D1499 M401
```

Please note that the subsequence indicated by the Target always uses the coordinate system of the EST, regardless of the direction of the alignment. For the 3' EST, the seventh column contains a "-" to indicate that the match is to the reverse complement of ctg123. The Gap attribute does not change as a consequence of this reverse complementation, and is read from left to right in the usual manner.

An application may wish to group the EST pair into a single feature. This can be accomplished by creating an implied cDNA_match that extends from the left end of the first EST to the right end of the last EST, and indicating that this cDNA match is the Parent of the two ESTs. The parts of the match use the SO "match_part" term. A match_part can be used as a subpart of any type of match.

```
 ctg123 . cDNA_match 1200  9000  .        .  . ID=cDNA00001

 ctg123 . match_part  1200  3200  2.2e-30  +  .
      ID=match00002;Parent=cDNA00001;Target=mjm1123.5 5 506;Gap=M301 D1499
M201

 ctg123 . match_part  7000  9000  7.4e-32  -  .
      ID=match00003;Parent=cDNA00001;Target=mjm1123.3 1 502;Gap=M101 D1499
M401
```

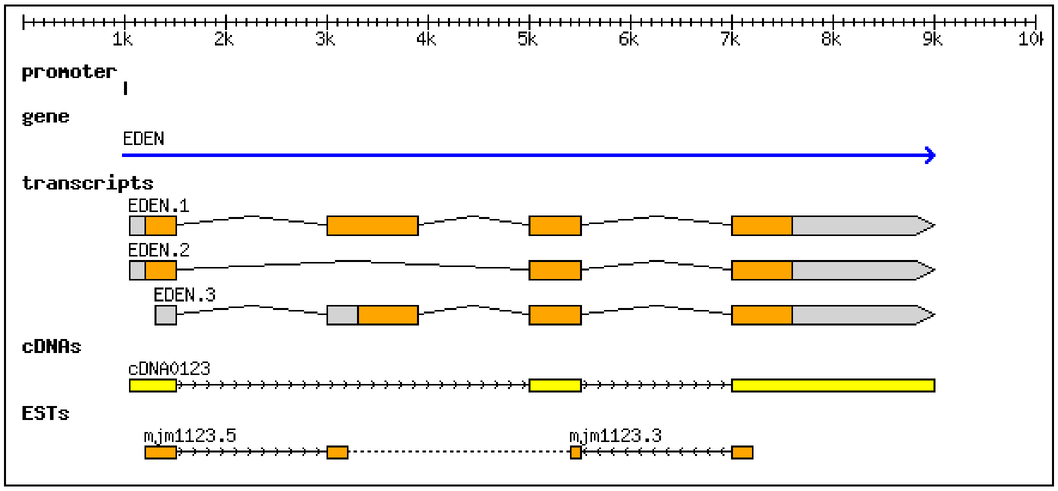


FIGURE 3

TRANSCRIPT-RELATIVE ALIGNMENTS

```
The representation of strandedness in nucleotide-to-nucleotide and
protein-to-nucleotide alignments is a common source of confusion in
GFF files.  This section will attempt to explain it.

* Case #1: alignment to a + strand transcript

Consider a pair of EST matches to the genome:

 =============================  genome
     -------------------->       transcript
     ------>        <----
      EST_A (5')    EST_B (3')

EST_A is a 5' EST and its sequence (as represented in a FASTA file,
for example) is in the same strand as the genomic sequence.  It is
represented as:

 ctg123 . EST_match  1000 1500 . + . ID=match001;Target=EST_A 1 500 +

The strand field in column #7 is "+" indicating that the match is to
the forward strand of the genome.  The optional strand field in the
Target attribute is also +, indicating that the alignment is to the
plus strand of the implied underlying transcript.

Let us now consider EST_B , which is a 3' EST. Its sequence as
represented in the FASTA file aligns to the reverse complement of the
genomic sequence.  It is represented as:

 ctg123 . EST_match  2000 2500 . + . ID=match002;Target=EST_B 1 500 -

The strand field in column #7 is "+" indicating that the match is to a
transcript feature on the forward of the genome.  The strand field in
the Target attribute is -, indicating that the EST sequence should be
reverse complemented in order to align to the underlying transcript.

* Case #2: alignment to a - strand transcript

Here is the opposite case:

 =============================  genome
     <--------------------       transcript
      ------>        <----
      EST_D (3')  EST_C (5')

In this case, the 5' EST_C aligns to the reverse complement of the
forward strand of the genome, while the 3' EST_D aligns to the forward
strand directly.  These are represented as follows:

 ctg123 . EST_match  2000 2500 . - . ID=match001;Target=EST_C 1 500 +
 ctg123 . EST_match  1000 1500 . - . ID=match001;Target=EST_D 1 500 -

The first line indicates that the transcript is on the - strand of the
genome, and that EST_C aligns to the transcripts forward strand.  The
second line uses - in the 7th column to indicate that the transcript
is on the minus strand, and - in the Target field to indicate that
EST_D aligns to the minus strand of the transcript.

  ===================================================================
  Confused?  Just remember that for purposes of display, the source
  and target strands will be multipled together.  A +/+ or -/-
  alignment indicates that the reference sequence and the target
  sequence can be aligned directly.  A +/- or -/+ alignment indicates
  that the target must be reverse complemented in order to align to
  the plus strand of the reference sequence.
  ===================================================================

A similar rule applies to TBLASTX alignments, which rely on matching
the six-frame translation of the source to the six-frame translation
of the target.  Consider the case of two genomes that align together
in the forward direction, whose alignment is supported by translations
of genes A and B, one of which is on the plus strand, and the other on
the minus strand:

  =============================>  genome X
      ------>        <----
      gene A          gene B
  =============================>  genome Y

These two alignments will be  represented as:

  X TBLASTX translated_nucleotide_match 1000 1500 . + . ID=matchA;Target=Y
500 1000 +
  X TBLASTX translated_nucleotide_match 2000 2500 . - . ID=matchB;Target=Y
1500 2000 -

Note that the first alignment is +/+ and the second is -/-.  Both
indicate that the sequences of genomes X and Y can be aligned
directly.
```

Now we look at the case of two genomes that align in the antiparallel
direction:

```
  ============================>  genome X
       ------>        <----
       gene A          gene B
  <============================  genome Y
```

These two alignments will be represented as:

```
  X TBLASTX translated_nucleotide_match 1000 1500 . + . ID=matchA;Target=Y
500 1000 -
  X TBLASTX translated_nucleotide_match 2000 2500 . - . ID=matchB;Target=Y
1500 2000 +
```

The first match indicates that a plus strand feature of genome X
aligns to a minus strand feature of genome Y. The second match
indicates that a minus strand feature of genome X aligns to a plus
strand feature of genome Y. In both cases, the result is to align the
plus strand of genome X to the minus strand of genome Y.

ONTOLOGY ASSOCIATIONS AND DB CROSS REFERENCES

Two reserved attributes, Ontology_term and Dbxref, can be used to
establish links between a GFF3 feature and a data record contained in
another database. Ontology_term is reserved for associations to
ontologies, such as the Gene Ontology. Dbxref is used for all other
cross references. While there is no firm boundary line between these
two concepts, curators tend to treat ontology associations differently
and hence ontology terms have been given their own reserved attribute
label.

The value of both Ontology_term and Dbxref is the ID of the cross
referenced object in the form "DBTAG:ID". The DBTAG indicates which
database the referenced object can be found in, and ID indicates the
identifier of the object within that database. IDs can contain
unescaped colons but DBTAGs cannot, so parsing code should split on
the first colon encountered in the attribute value.

The format of each type of ID varies from database to database. An
authoritative list of databases, their DBTAGs, and the URL
transformation rules that can be used to fetch the objects given their
IDs can be found at this location:

   ftp://ftp.geneontology.org/pub/go/doc/GO.xref_abbs

Further details can be found here:

   ftp://ftp.geneontology.org/pub/go/doc/GO.xref_abbs_spec

Here are some common examples:

* a dbxref to an EMBL sequence accession number:

  Dbxref="EMBL:AA816246"

* a dbxref to an NCBI gi number:

  Dbxref="NCBI_gi:10727410"

* a Ontology_term referring to a GO association

  Ontology_term="GO:0046703"

OTHER SYNTAX

Comments are preceded by the # symbol. Meta-data and directives are
preceded by ##. The following directives are recognized:

```
  ##gff-version 3
        The GFF version, always 3 in this spec.  This directive must
        be present, and must be the topmost line of the file.

  ##sequence-region seqid start end
        The sequence segment referred to by this file, in the format
        "seqid start end".  This element is optional, but strongly
        encouraged because it allows parsers to perform bounds
        checking on features. There may be multiple ##sequence-region
        directives, each corresponding to one of the reference
        sequences referred to in the body of the file.

  ##feature-ontology URI
        This directive indicates that the GFF3 file uses the ontology
        of feature types located at the indicated URI or URL.
        Multiple URIs may be added, in which case they are
        merged (or raise an exception if they cannot be merged).  The
        URIs for the released sequence ontologies are:

        Release 1: 5/12/2004
        http://song.cvs.sourceforge.net/*checkout*/song/ontology/sofa.obo?
revision=1.6

        Release 2: 5/16/2005
        http://song.cvs.sourceforge.net/*checkout*/song/ontology/sofa.obo?
revision=1.12i
    Release 2.4.3 06/01/2010
    SO:
    http://song.cvs.sourceforge.net/viewvc/*checkout*/song/ontology/so.obo?
revision=1.263
    SOFA:
    http://song.cvs.sourceforge.net/viewvc/*checkout*/song/ontology/sofa.obo?
revision=1.217

    Releases occur every two months for SO and SOFA.

    The repository for SO releases is here:
    http://sourceforge.net/projects/song/files/Sequence%20Ontology/

    The repository for SOFA releases is here:
    http://sourceforge.net/projects/song/files/SO_Feature_Annotation/

        This directive may occur several times per file.  If no
        feature ontology is specified, then the most recent release of the
        Sequence Ontology is assumed.

        If multiple directives are given and a feature type is matched
        by multiple ontologies, the matching ontology included by the
        directive highest in the file wins the reference.  The Sequence
        Ontology itself is always referenced last.

        The content referenced by URI must be in OBO or DAG-Edit
        format.


  ##attribute-ontology URI
        This directive indicates that the GFF3 uses the ontology of
        attribute names located at the indicated URI or URL.  This
        directive may appear multiple times to load multiple URIs, in
        which case they are merged (or raise an exception if merging
        is not possible).  Currently no formal attribute ontologies
        exist, so this attribute is for future extension.

  ##source-ontology URI
        This directive indicates that the GFF3 uses the ontology of
        source names located at the indicated URI or URL.  This
        directive may appear multiple times to load multiple URIs, in
        which case they are merged (or raise an exception if merging
        is not possible).  Currently no formal source ontologies
        exist, so this attribute is for future extension.

  ##species NCBI_Taxonomy_URI

        This directive indicates the species that the annotations
        apply to. The preferred format is a NCBI URL that points to
        the relevant species page in either of the following formats:

        http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=6239
        http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?
name=Caenorhabditis+elegans

  ##genome-build source buildName

        The genome assembly build name used for the coordinates given
        in the file. Please specify the source of the assembly as well
        as its name. Examples (the parentheses are comments).

        ##genome-build NCBI B36           (human)
        ##genome-build WormBase ws110     (worm)
        ##genome-build FlyBase  r4.1      (drosophila)

  ###
        This directive (three # signs in a row) indicates that all
        forward references to feature IDs that have been seen to this
        point have been resolved.  After seeing this directive, a
        program that is processing the file serially can close off any
        open objects that it has created and return them, thereby
        allowing iterative access to the file.  Otherwise, software
        cannot know that a feature has been fully populated by its
        subfeatures until the end of the file has been reached.  It
        is recommended that complex features, such as the canonical
        gene, be terminated with the ### notation.

   ##FASTA
        This notation indicates that the annotation portion of the
        file is at an end and that the remainder of the file
        contains one or more sequences (nucleotide or protein)
        in FASTA format.  This allows features and sequences to
        be bundled together.  Example:

   ##gff-version   3
   ##sequence-region   ctg123 1 1497228
   ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
   ctg123 . TF_binding_site 1000  1012  .  +  .
ID=tfbs00001;Parent=gene00001
   ctg123 . mRNA            1050  9000  .  +  .
ID=mRNA00001;Parent=gene00001;Name=EDEN.1
   ctg123 . five_prime_UTR  1050  1200  .  +  .  Parent=mRNA00001
   ctg123 . CDS             1201  1500  .  +  0  ID=cds00001;Parent=mRNA00001
   ctg123 . CDS             3000  3902  .  +  0  ID=cds00001;Parent=mRNA00001
   ctg123 . CDS             5000  5500  .  +  0  ID=cds00001;Parent=mRNA00001
   ctg123 . CDS             7000  7600  .  +  0  ID=cds00001;Parent=mRNA00001
   ctg123 . three_prime_UTR 7601  9000  .  +  .  Parent=mRNA00001
   ctg123 . cDNA_match      1050  1500  5.8e-42  +  .
ID=match00001;Target=cdna0123+12+462
   ctg123 . cDNA_match      5000  5500  8.1e-43  +  .
ID=match00001;Target=cdna0123+463+963
   ctg123 . cDNA_match      7000  9000  1.4e-40  +  .
ID=match00001;Target=cdna0123+964+2964
   ##FASTA
   >ctg123
   cttctgggcgtacccgattctcggagaacttgccgcaccattccgccttg
   tgttcattgctgcctgcatgttcattgtctacctcggctacgtgtggcta
   tctttcctcggtgccctcgtgcacggagtcgagaaaccaaagaacaaaaa
   aagaaattaaaatatttattttgctgtggtttttgatgtgtgttttttat
   aatgatttttgatgtgaccaattgtacttttcctttaaatgaaatgtaat
   cttaaatgtatttccgacgaattcgaggcctgaaaagtgtgacgccattc
   gtatttgatttgggtttactatcgaataatgagaattttcaggcttaggc
   ttaggcttaggcttaggcttaggcttaggcttaggcttaggcttaggctt
   aggcttaggcttaggcttaggcttaggcttaggcttaggcttaggcttag
   aatctagctagctatccgaaattcgaggcctgaaaagtgtgacgccattc
   ...
   >cnda0123
   ttcaagtgctcagtcaatgtgattcacagtatgtcaccaaatattttggc
   agctttctcaagggatcaaaattatggatcattatggaatacctcggtgg
   aggctcagcgctcgatttaactaaaagtggaaagctggacgaaagtcata
   tcgctgtgattcttcgcgaaattttgaaaggtctcgagtatctgcatagt
   gaaagaaaaatccacagagatattaaaggagccaacgttttgttggaccg
   tcaaacagcggctgtaaaaatttgtgattatggttaaagg

       For backward-compatibility with the GFF version output by the
      Artemis tool, a GFF line that begins with the character >
       creates an implied ##FASTA directive.
```

PATHOLOGICAL CASES

The following section discusses how to represent "pathological" cases that arise in prokaryotic and eukaryotic genetics. Most of these have to do with organisms' endlessly creative ways of processing transcripts.

a) Single exon genes

This is the case in which a single unspliced transcript encodes a single CDS.

```
   ----->XXXXXXX*------>
```

The preferred representation is to create a gene, a transcript, an exon and a CDS:

```
ChrX  . gene XXXX YYYY  .  +  . ID=gene01;name=resA
ChrX  . mRNA XXXX YYYY  .  +  . ID=tran01;Parent=gene01
ChrX  . exon XXXX YYYY  .  +  . Parent=tran01
ChrX  . CDS  XXXX YYYY  .  +  . Parent=tran01
```

Some groups will find this redundant. A valid alternative is to omit the exon feature:

```
ChrX  . gene XXXX YYYY  .  +  . ID=gene01;name=resA
ChrX  . mRNA XXXX YYYY  .  +  . ID=tran01;Parent=gene01
ChrX  . CDS  XXXX YYYY  .  +  . Parent=tran01
```

It is not recommended to parent the CDS directly onto the gene, because this will make it impossible to determine the UTRs (since the gene may validly include untranscribed regulatory regions).

Also note that mixing the two styles, as in the case of an organism with both spliced and unspliced transcripts, is liable to lead to the confusion of people working with the GFF3 file.

b) Polycistronic transcripts

This is the case in which a single (possibly spliced) transcript encodes multiple open reading frames that generate independent protein products.

```
 ----->XXXXXXX*-->BBBBBB*--->ZZZZ*-->AAAAAA*-----
```

Since the single transcript corresponds to multiple genes that can be identified by genetic analysis, the recommended solution here is to create four "gene" objects and make them the parent for a single

```
transcript. The transcript will contain a single exon (in the
unspliced case) and four separate CDSs:

ChrX  . gene XXXX YYYY  .  +  . ID=gene01;name=resA
ChrX  . gene XXXX YYYY  .  +  . ID=gene02;name=resB
ChrX  . gene XXXX YYYY  .  +  . ID=gene03;name=resX
ChrX  . gene XXXX YYYY  .  +  . ID=gene04;name=resZ
ChrX  . mRNA XXXX YYYY  .  +  . ID=tran01;Parent=gene01,gene02,gene03,gene04
ChrX  . exon XXXX YYYY  .  +  . ID=exon00001;Parent=tran01
ChrX  . CDS  XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene01
ChrX  . CDS  XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene02
ChrX  . CDS  XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene03
ChrX  . CDS  XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene04

To disambiguate the relationship between which genes encode which
CDSs, you may use the Derives_from relationship.

c) Gene containing an intein

An intein occurs when a portion of the protein is spliced out and the
two polypeptide fragments are rejoined to become a functional
protein. The portion that is spliced out is called the "intein," and
it may itself have intrinsic molecular activity:

----->XXXXXXyyyyyyyyyyXXXXXXX*-------

(yyyyyy is the intein)

The preferred representation is to create one gene, one transcript,
one exon, and one CDS. The CDS produces a pre-polypeptide using the
"Derives_from" tag, and this polypeptide in turn gives rise to two
mature_polypeptides, one each for the intein and the flanking protein:

ChrX  . gene                XXXX YYYY  .  +  . ID=gene01;name=resA
ChrX  . mRNA                XXXX YYYY  .  +  . ID=tran01;Parent=gene01
ChrX  . exon                XXXX YYYY  .  +  . Parent=tran01
ChrX  . CDS                 XXXX YYYY  .  +  . ID=cds01;Parent=tran01
ChrX  . polypeptide         XXXX YYYY  .  +  . ID=poly01;Derives_from=cds01
ChrX  . mature_polypeptide XXXX YYYY  .  +  . ID=poly02;Parent=poly01
ChrX  . mature_polypeptide XXXX YYYY  .  +  . ID=poly02;Parent=poly01
ChrX  . intein              XXXX YYYY  .  +  . ID=poly03;Parent=poly01

Because the flanking mature_polypeptide has discontinuous coordinates
on the genome, it appears twice with the same ID.

If the intein is immediately degraded, you may not wish to annotate it
explicitly, and its line would be deleted from the example.  However,
if it has molecular activity, it may correspond to a gene, in which
case:

ChrX  . gene                XXXX YYYY  .  +  . ID=gene01;name=resA
ChrX  . gene                XXXX YYYY  .  +  . ID=gene02;name=inteinA
ChrX  . mRNA                XXXX YYYY  .  +  . ID=tran01;Parent=gene01,gene02
ChrX  . exon                XXXX YYYY  .  +  . Parent=tran01
ChrX  . CDS                 XXXX YYYY  .  +  . ID=cds01;Parent=tran01
ChrX  . polypeptide         XXXX YYYY  .  +  . ID=poly01;Derives_from=cds01
ChrX  . mature_polypeptide XXXX YYYY  .  +  .
ID=poly02;Parent=poly01;Derives_from=gene01
ChrX  . mature_polypeptide XXXX YYYY  .  +  .
ID=poly02;Parent=poly01;Derives_from=gene01
ChrX  . intein              XXXX YYYY  .  +  .
ID=poly03;Parent=poly01;Derives_from=gene02

The term "polypeptide" is part of SO.  The terms "mature_polypeptide"
and "intein" are slated to be added in a pending release.

d) Trans-spliced transcript

This occurs when two genes contribute to a processed transcript via a
trans-splicing reaction:

   spliced
   leader
   =======>----->XXXXXXX*------>

The simplest way to represent this is to show the mRNA as being split
across two discontinuous genomic locations:

ChrX  . gene                XXXX YYYY  .  +  . ID=gene01;name=my_gene
ChrX  . mRNA                XXXX YYYY  .  +  . ID=tran01;Parent=gene01
ChrX  . mRNA                XXXX YYYY  .  +  . ID=tran01;Parent=gene01
ChrX  . exon                XXXX YYYY  .  +  . Parent=tran01
ChrX  . CDS                 XXXX YYYY  .  +  . ID=cds01;Parent=tran01

However, this does not indicate which part of the transcript comes
from the spliced leader.  A preferred representation explicitly adds
features for the spliced leader gene, the primary_transcript and the
spliced_leader_RNA:

ChrX  . gene                XXXX YYYY  .  +  . ID=gene01;name=my_gene
ChrX  . gene                XXXX YYYY  .  +  . ID=gene02;name=leader_gene
ChrX  . mRNA                XXXX YYYY  .  +  . ID=tran01;Parent=gene01,gene02
ChrX  . mRNA                XXXX YYYY  .  +  . ID=tran01;Parent=gene01,gene02
ChrX  . primary_transcript XXXX YYYY  .  +  .
```

```
ID=pt01;Parent=tran01;Derives_from=gene01
ChrX  . spliced_leader_RNA XXXX YYYY  .  +  .
ID=sl01;Parent=tran01;Derives_from=gene02
ChrX  . exon               XXXX YYYY  .  +  . Parent=tran01
ChrX  . CDS                XXXX YYYY  .  +  . ID=cds01;Parent=tran01
```

As shown here, the mRNA derives from two genes ("my_gene" and the leader gene) and occupies disjunct coordinates on the genome. The primary_transcript, which encodes the body of the mRNA, is part of (has as its Parent) this mRNA. The same relationship applies to the spliced leader RNA. The Derives_from relationship is used to indicate which genes produced the primary transcript and spliced leader respectively.

The exon and CDS features follow in the normal fashion.

e) Programmed frameshift

This event occurs when the ribosome performs a programmed frameshift during translation in order to skip over an in-frame stop codon. The frameshift may occur forward or backward.

```
 -------------------------> mRNA
  ==========
           ============*  CDS
```

The representation of this is to make the CDS discontinuous:

```
ChrX  . gene               XXXX   YYYY  .  +  . ID=gene01;name=my_gene
ChrX  . mRNA               XXXX   YYYY  .  +  .
ID=tran01;Parent=gene01;Ontology_term=SO:1000069
ChrX  . exon               XXXX   YYYY  .  +  . Parent=tran01
ChrX  . CDS                XXXX   YYYY  0  +  . ID=cds01;Parent=tran01
ChrX  . CDS                YYYY-1 ZZZZ  1  +  . ID=cds01;Parent=tran01
```

You will also need to adjust the phase field properly so that the CDS translates.

It is suggested that the mRNA be tagged with the appropriate SO transcript attributes such as "minus_1_translational_frameshift" (SO:1000069). This will allow all such programmed frameshift mRNAs to be recovered with a query. The accession for "plus_1_translational_frameshift" is SO:1001263.

f) An operon

A classic operon occurs when the genes in a polycistronic transcript are co-regulated by cis-regulatory element(s):

```
   regulatory element
   *  ===============================================> operon
      ----->XXXXXXX*-->BBBBBB*--->ZZZZ*-->AAAAAA*-----
```

It can be indicated in GFF3 in this way:

```
ChrX  . operon   XXXX YYYY  .  +  . ID=operon01;name=my_operon
ChrX  . promoter XXXX YYYY  .  +  . Parent=operon01
ChrX  . gene     XXXX YYYY  .  +  . ID=gene01;Parent=operon01;name=resA
ChrX  . gene     XXXX YYYY  .  +  . ID=gene02;Parent=operon01;name=resB
ChrX  . gene     XXXX YYYY  .  +  . ID=gene03;Parent=operon01;name=resX
ChrX  . gene     XXXX YYYY  .  +  . ID=gene04;Parent=operon01;name=resZ
ChrX  . mRNA     XXXX YYYY  .  +  .
ID=tran01;Parent=gene01,gene02,gene03,gene04
ChrX  . exon     XXXX YYYY  .  +  . ID=exon00001;Parent=tran01
ChrX  . CDS      XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene01
ChrX  . CDS      XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene02
ChrX  . CDS      XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene03
ChrX  . CDS      XXXX YYYY  .  +  . Parent=tran01;Derives_from=gene04
```

The regulatory element ("promoter" in this example) is part of the operon via the Parent tag. The four genes are part of the operon, and the resulting mRNA is multiply-parented by the four genes, as in the earlier example.

At the time of this writing, promoters and other cis-regulatory elements cannot be part_of an operon, but this restriction is being reconsidered.

Change Log:

```
1.20 Wed Dec 15 12:35:10 MST 2010
    -Added language to the description of the ID attribute to
     clarify that discontinuous features can exist on
     multiple lines and share the same ID.

1.19 Tue Jul  6 12:51:26 MDT 2010
    -Fixed coordinate errors in the EST_match and match_part examples
     in the 'Alignments' section.
    -Constrained multiple attribute values to the Parent, Alias,
     Note, DBxref and Ontology_term attributes.

1.18 June 24 2010
     -Added the sections regarding circular genomes to the spec.

1.17 Wed June 2 2010
     -Changed the spec to include Sequence Ontology (SO) sequence_feature
      terms in column 3 as well as SOFA terms. (SOFA is a subset of SO).

1.16 Tue May 25 10:06:38 MDT 2010
     -Fixed more incorrect CDS phases throughout.
     -Changed (three|five)_prime_utr to (three|five)_prime_UTR throughout.
     -Changed (3'|5')-UTR to (three|five)_prime_UTR throughout.
     -Added ID attributes to CDS features (required for multiline features)
      in the FASTA pragma example.

1.15 Mon Aug 31 12:59:26 EDT 2009
     -Fixed incorrect CDS phases in the canonical gene example.

1.14 Mon Aug 25 10:24:02 EDT 2008
     -Add meta-directives for species and build number.

1.13 Wed May 23 10:31:01 EDT 2007
     -Insist that CDS include the start and end codon.

1.12 Thu Apr  5 17:32:32 EDT 2007
     -Use "match_part" as the subpart of cDNA_match in the paired EST
example.
     -Phase is required for all CDS features.

1.11 Fri Dec  1 16:33:39 EST 2006
     -Clarified definition of phase relative to reverse strand features

1.10 14 September 2006
     -Reformatted for new SO web site.

1.09 Wed Sep  6 17:55:32 EDT 2006
     -Information about the GFF3 validator.

1.08 Tue Jul 18 15:12:11 EDT 2006
     -Added URLs for SO releases.

1.07 Wed May 24 21:59:02 EDT 2006
     -Fixed description of phase (temporarily lost due to CVS glitches)

1.06 Wed May 24 11:44:22 EDT 2006
     -Relaxed escaping rules.
     -Fixed typos found by Gordon Gremme.

1.05 Tue May 23 10:46:25 EDT 2006
     -Fixed all IDs in the examples to make them internally consistent.
Previously,
      some examples did not validate because of inconsistent numbers of
zeroes in
      the identifiers (mRNA00001 vs mRNA0001).
```