First, you need to identify which bioinformatics file format we're dealing with. Common formats include FASTA, FASTQ, SAM/BAM, GFF, BED, and VCF.

1. FASTA/FASTQ file:

Since FASTA and FASTQ have simple structure, we can add the Gene Ontology to the header line:

>sequence\_identifier | GO:0006915 | Additional description or metadata

sequence\_data

some example code: add\_ontology\_to\_fasta.py

If we have a large number of sequences and need to map many of them to ontology terms, it might be more efficient to preload a dictionary of sequence identifiers to ontology terms rather than making a lookup for each sequence.(related to Ontology function)

1. SAM(https://samtools.github.io/hts-specs/SAMv1.pdf)

SAM (Sequence Alignment/Map) Headers contain @CO lines, which are free-form comment lines. We could use these lines to store ontology data relevant to the entire dataset or experiment.

If our ontology data pertains to individual reads/alignments, we’ll need to consider the TAGs in the alignment section. SAM format supports optional fields in the alignment section where we can define custom tags.

1. BAM

BAM need to convert to SAM and then use the function of SAM

1. GFF
   1. GTF

The GTF (Gene Transfer Format) file format is used to describe gene features, such as exons, introns, and UTR regions. The ninth column (attributes column) is particularly flexible and used for tagging a variety of metadata. This attributes column consists of tag "value" pairs and is a suitable place to add ontology terms.

1. Choose a Tag for Ontology:

Determine an appropriate tag for your ontology data. For instance, you might choose OntologyTerm as your tag.

2. Define How Ontology Relates to Features:

Decide how ontology terms correlate with each feature. For example, do they relate to genes, transcripts, or some other features?

3. Modify GTF File with Python:

You can use Python to parse and modify the GTF file: add\_ontology\_to\_gtf.py

* 1. GFF3

the GFF3 format allows for headers (lines that start with ##), which are primarily used to convey meta-information about the content or to provide a reference to the sequence data. The standard GFF3 headers include directives like ##gff-version, ##sequence-region, and ##FASTA.

1. Define the Header Structure:

You'll want to choose a descriptive header prefix. For example, you could use ##ontology followed by relevant information. This will ensure clarity for those examining the file.

2. Add to the GFF3 Header with Python: add\_ontology\_to\_gff3\_header.py

1. BED

The BED (Browser Extensible Data) format is commonly used to define genomic regions and is often used in conjunction with genome browsers like UCSC's Genome Browser. A BED file typically has a minimum of three required columns (chromosome, start, and end), but it can have up to twelve columns, depending on the depth of information you want to provide. Beyond the twelve standard columns, you can add additional custom columns.

1. Extend the BED with Custom Columns:

Add additional columns to your BED file to contain the ontology data. This is straightforward if you're working with tools or scripts that can handle BED extensions (like BED12+).

For example, a BED file with an additional ontology column might look like:

chr1 100 200 feature1 0 + 100 200 0,0,0 1 100, 0, GO:0006915

chr1 250 350 feature2 0 + 250 350 0,0,0 1 100, 0, GO:0008270

2. Incorporate Ontology into the Name (or another existing) Column:

If you don't want to extend the BED format, you can embed ontology information into one of the existing columns, such as the name column:

chr1 100 200 feature1|GO:0006915

chr1 250 350 feature2|GO:0008270

3. Use BEDtools to Modify BED Files:

BEDtools is a powerful suite of utilities for working with genomic intervals in various formats, including BED. You can use BEDtools alongside a custom script to add ontology data.

4. Python Script to Add Ontology to BED:

Here's a basic script to add ontology data to a BED file by extending it with custom columns: add\_ontology\_to\_bed.py

1. VCF

VCF (Variant Call Format) is a widely used format to represent genetic variations like SNPs, insertions, deletions, and more.

To add ontology to a VCF file, you can utilize the meta-information section, the INFO column, or even create custom tags in the INFO field. Here's how to do it:

1. Add Ontology Information in the Meta-information:

The meta-information section can store ontology-related context for the entire VCF file:

##ontology=<Source="Gene Ontology",Version="2023-05",URL="http://geneontology.org">

2. Add Ontology in the INFO Field:

The seventh column of a VCF file is the INFO field. This field provides a way to encode structured data. For adding ontology data for specific variants, you can use this field:

First, define a custom INFO field in the meta-information section:

##INFO=<ID=GO,Number=.,Type=String,Description="Gene Ontology Term">

Then, in the data lines, you can add the ontology terms:

#CHROM POS ID REF ALT QUAL FILTER INFO

chr1 12345 . A G,T 100 PASS GO=GO:0006915

chr2 67890 rs123 T G 50 PASS GO=GO:0008270,GO:0006915

3. Python Script to Add Ontology to VCF:

Here's a basic script to add ontology data to a VCF file's INFO field: add\_ontology\_to\_vcf.py

Comments:

Add a list to recording log, which file input which type of ontology

Flet GUI compliance