A brief guide to GO annotation using the CACAO interface

This is meant as a brief introduction to GO annotations. For a fully featured, well-written introduction that will address most of the doubts you may have after reading this, please see Balakrishnan et al. (2013) *Databases* "A guide to best practices for Gene Ontology (GO) manual annotation" [PMID: 23842463].

GO annotation basics

In this CACAO lab unit we will be making Gene Ontology (GO) annotations of gene products in a genome of interest. A GO annotation consists in establishing a link between a gene product (e.g. the Bacillus phage Troll "Tail assembly chaperone"; UniProt accession S5YQ92) and a GO term describing a specific aspect of its biology. In GO, we distinguish among three major biological components for a gene product: molecular function, biological process and cellular location. Hence, a GO annotation links a gene accession number to a GO term in any of these categories. GO is an ontology, meaning that GO terms are linked by familial relationships (e.g. "sequence specific DNA binding" GO:0043565 being a *child* of "DNA binding" GO:0003677).

Here is a brief summary from the GO Consortium site (http://geneontology.org/) on what the three biological components are meant to indicate:

Cellular Component
These terms describe a component of a cell that is part of a larger object, such as an anatomical structure (e.g. rough endoplasmic reticulum or nucleus) or a gene product group (e.g. ribosome, proteasome or a protein dimer).

Biological Process
A biological process term describes a series of events accomplished by one or more organized assemblies of molecular functions. Examples of broad biological process terms are "cellular physiological process" or "signal transduction". Examples of more specific terms are "pyrimidine metabolic process" or "alpha-glucoside transport". The general rule to assist in distinguishing between a biological process and a molecular function is that a process must have more than one distinct steps. A biological process is not equivalent to a pathway. At present, the GO does not try to represent the dynamics or dependencies that would be required to fully describe a pathway.

Molecular Function
Molecular function terms describes activities that occur at the molecular level, such as "catalytic activity" or "binding activity". GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where, when, or in what context the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products, but some activities are performed by assembled complexes of gene products. Examples of broad functional terms are "catalytic activity" and "transporter activity"; examples of narrower functional terms are "adenylate cyclase activity" or "Toll receptor binding". It is easy to confuse a gene product name with its molecular function; for that reason GO molecular functions are often appended with the word "activity".

Regular and "transfer" GO annotations

In a "regular" GO annotation, biocurators identify a peer-reviewed article where experimental evidence for the molecular function, biological process and/or cellular component of one or several genes is provided. Reading the paper, biocurators then make assertions on, say, gene `X` having molecular function `Y`, where `X` is an accession number for the gene and `Y` is a GO term.

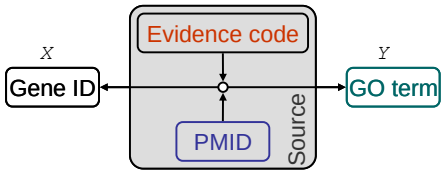


Figure 1 – Schematic diagram for "regular" GO annotations. A gene (`X`) is annotated as having GO term (`Y`), which specifies a well-defined function/process/component, using an evidence code and the PubMed ID (PMID) of a scientific article as the *source* for the annotation.

In doing so, biocurators identify the *source* of the annotation in the following way: (1) they cite the original paper with the evidence (providing its PubMed ID number) as the *reference* for the annotation, and (2) identify an appropriate *evidence code* term to summarize the type of evidence that is provided in the paper to warrant such assertion. For instance, if a study shows that protein X is part of the ribosome through immunoflueorescence techniques, a curator can use the GO evidence code *Inferred from Direct Assay (IDA)* to annotate protein X to the GO term GO:0005840. The following page provides a list of all the possible GO evidence codes you can use in a GO annotation: http://geneontology.org/page/guide-go-evidence-codes. See here for the evidence codes that you are authorized to use in CACAO annotations (http://gowiki.tamu.edu/wiki/index.php/evidence_codes)[1].

Transfer annotations

In this lab unit we may deal with an organism the sequence of which has just been recently sequenced. This means that no experimental work has been done on our organism of interest and, therefore, we cannot perform "regular" GO annotations (there are not scientific manuscripts to annotate from). Instead, what we seek to do is to *transfer* annotations from another organism in which there is experimental evidence for the annotation. The way this is most frequently done is through homology. Remember that two genes are homologous if they share similarity due to shared ancestry. Using sequence and structure search methods (such as BLAST or HHPred) we can establish that two sequences are similar. Using appropriate thresholds (listed here) and our own judgment, we can use the observed similarity to postulate homology. Once we postulate that two genes are homologous, we can make use of our knowledge of the underlying biology to decide if functional annotations made on one gene should transfer to the other or not[2].

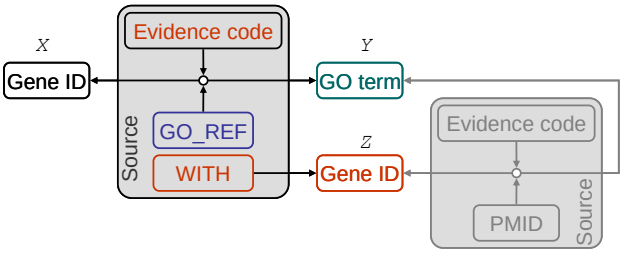


Figure 2 – Schematic diagram for "transfer" GO annotations. A gene (X) is annotated as having GO term (Y), by establishing that gene X is homologous to gene Z, where the annotated function/process/component (Y) has been established through experimental means. The source for the annotation is now the general protocol (*GO_REF*) and the specific method (evidence code) used for establishing homology between X and Z, as well as the identifier for Z in the *WITH* field.

[1] Certain evidence codes (and some types of annotations) are disabled in CACAO. CACAO is normally run as a competition where the number (and quality) of annotations determines the winning team. To avoid contestants submitting many weak annotations based on papers using high-throughput methods (e.g. protein-protein interaction yeast-to-hybrid assays) to score points, evidence codes such as Inferred from Physical Interaction (IPI) or Inferred from Expression Pattern (IEP) have been disabled. If you find that you need to use such codes, please contact your instructor.

[2] In some cases, the transfer makes complete sense, in other cases, no sense at all. For instance, a yeast protein can be annotated as being localized to the nucleus (cellullar component), but that annotation makes no sense on a bacterial homolog of the protein. Whenever you are transferring annotations between markedly different species you should attempt to explain the possible role of the protein in the recipient's biology.

Our annotation process is now, therefore, slightly different from the regular case. In the case of "transfer" annotations, we will be stating that gene X has function/process/component Y, based on its similarity with another gene (Z), for which that function/process/component has been annotated using experimental evidence.

The *source* for our annotation is therefore different than that of "regular" annotations and the evidence codes we will use are also different. To make these assertions, we use evidence codes such as Inferred from Sequence Orthology (ISO) or Inferred from Genomic Context (IGC). Because it is us, and not the authors of a paper, who are making the claim that the annotation of gene Z should be transferred to X, the source does not cite a scientific article, but rather a specialized *GO reference* (GO_REF) that describes the general procedure used by the biocurator to determine the correspondence. We will use the CACAO GO_REF (GO_REF:0000112; Gene Ontology annotation by CACAO biocurators), the description of which you can find here (https://gowiki.tamu.edu/wiki/index.php/CACAO_GO_REF). Crucially, the *source* of the "transfer" annotation includes also another element, defined by the *WITH* field of GO annotations. This is the identifier for the gene (Z) that we are transferring the annotation from.

The "transfer" annotation process

In "regular" GO annotations, biocurators typically start with a paper and then look for experimental evidence of function/process/component for one or more genes in the paper, then proceed to annotate these. The situation in transfer annotations is fundamentally reversed. Here we start with the genes of our genome of interest, for which we seek to make annotations via homology. Our workflow is therefore as follows:

1) We use search methods to identify putative homologs
2) We scan GO annotation databases and PubMed to see if *any* of the putative homologs has
   a) existent GO annotations
   b) a paper with experimental evidence of function/process/component that we can use to make GO annotations
3) We use homologs with existent GO annotations (or make *regular* GO annotations on homologs) to make our *transfer* annotations (i.e. transferring the homolog annotation to the gene in our genome of interest)

Note that, in many cases, putative homologs will not have existing GO annotations. That means that in order for you to annotate a gene in the genome of interest you will have to perform *two* annotations: a first "regular" annotation on the homolog and a second "transfer" annotation on the target genome.

Alternative workflow

Notice that it is possible to use an alternative workflow. This alternative workflow depends on first identifying one or more model organisms that are evolutionary "close" to our organism of interest. These model organisms will have abundant experimental literature that we can annotate on (or maybe even already-made annotations). We just need to check before we start annotating that the gene in the model organism will transfer to our organism of interest according to the criteria outlined in the CACAO GO_REF and made explicit here: https://gowiki.tamu.edu/wiki/index.php/Category:CACAO_GO_REF.

Annotations using the CACAO interface

To facilitate and standardize the annotation process in genomes of interest, we use the CACAO/GONUTS interface, developed by Jim Hu at Texas A&M University. CACAO is an intercampus

annotation competition. For reference on CACAO and GONUTS, see all the great instructional material already available here: http://gowiki.tamu.edu/wiki/index.php/Category:CACAO (see *Help for Students* section)

Getting gene products for our genomes of interest

In order for our annotations to be incorporated into the Gene Ontology (GO), we must annotate gene products with an assigned accession. Most gene products are proteins, and hence we will use UniProt identifiers. Proteins get UniProt accessions after the genome sequence has been successfully submitted to the NCBI GenBank or EBI ENA databases. Once you have identified a gene to annotate in the genome of interest, get its protein accession from the NCBI RefSeq database (e.g. YP_008051028.1). Go to the UniProt website and search with this accession. You should get as a result an item with a UniProt accession (R4TBI6).

Checking for annotations

The first thing to do once you have a candidate gene for annotation is to check that it has no previous annotations. Go to QuickGO and search with your UniProt accession.

The fact that your gene is not on QuickGO does not mean that it has not been annotated; it might have been picked up by another student and may be already annotated in CACAO. Check this by searching with the code `9CAUD:` followed by the UniProt accession (R4TBI6; that is: 9CAUD:R4TBI6) in the CACAO interface.

Creating a gene page

Chances are that your gene will not even be in CACAO to start with. To add your gene to CACAO, click on the *Create New Gene Page* link on the left panel and paste your UniProt accession into it, then hit *Create Page*.

Figure 3 – Gene page creation in CACAO.

Making an annotation

Once you have created a gene page (or using an existing one), you can make annotations using the *edit table* link and then clicking on the *Add row* button. This will bring up a form to create the annotation (Figure 5). The annotation fields are as follows:

- **Qualifier**: this allows you to modify the annotation to indicate, for instance, that the GO term used is *NOT* applicable to your gene.
- **GO ID**: this is the identifier of the GO term. You can use QuickGO and AmiGO to browse GO terms and find the one that is most adequate for describing the function/process/component you want to annotate.

- **Reference**: this will either be the CACAO GO_REF or the PubMed ID of the article you are annotating from.
- **Evidence code**: this is the evidence code term that best captures the method used to make the annotation.
- **with/from**: if you are making a "transfer" annotation using the GO_REF, you will use *WITH* and enter here the UniProt ID for the identified homolog you are annotating from. Note that you can use multiple homologs to make your annotation.
- **Aspect**: this just refers to the type (function/process/component) of annotation our term belongs to
- **Notes**: here you should summarize the process used (e.g. HHpred with probability *X* and coverage *Y* identifies gene *Z* as a putative homolog for this gene). You can see examples in any annotation already on CACAO, such as this one. Take also a look at the instructions on the GO_REF (Figure 4). You should also note here the rationale for transferring the function/process/component from the homolog to your gene, especially when the homolog is not a gene from a similar organism (that is, why you think the same function/process/component applies to your gene).

```
go_ref_id: GO_REF:0000112
title: Gene Ontology annotation by CACAO biocurators
authors: Ivan Erill, James Hu, Community Assessment of Community Annotation with Ontologies
year: 2017
abstract: This GO reference describes the criteria used by biocurators participating in the Community
Assessment of Community Annotation with Ontologies (CACAO) to annotate gene products from genomes of
interest through the use of computational methods to establish and manually validate function or homology
to gene products. In particular, this GO reference describes the criteria used to make annotations based
on evidence codes ISS, ISA, ISO, ISM and IGC. To perform ISS-, ISA-, and ISO-based annotations on a gene
product, CACAO biocurators use sequence- and structure-based search algorithms (e.g. BLASTP, HHPred) to
establish homology, conservation of sequence and structure functional determinants between the target gene
product and gene products from other organisms with published GO annotations supported by experimental
codes and lacking NOT qualifiers. These gene products are referenced in the WITH field of the annotation
using their xref database accession. ISM-based annotations make use of published computational methods
(e.g. TMHMM, SignalP) to predict gene product structure, localization or function. IGC-based annotations
are made on the basis of suggestive evidence for function based on synteny. Parameters and criteria for
use of all computational methods (e.g. e-value) are listed and versioned in the publicly available CACAO
documentation (http://gowiki.tamu.edu/). Annotations made by CACAO biocurators are reviewed by CACAO team
instructors before their release.
```

Figure 4 – The Gene Ontology annotation by CACAO biocurators GO_REF:0000112.

Figure 5 – Making an annotation in CACAO. Adding a row to the annotation table (left) and making the annotation (right).

Once you have filled up the required fields, click *Save Row* and then, on the *TableEdit* page don't forget to click on the *Save Table to wiki page* button (Figure 5). Otherwise your annotation will NOT be saved.

Making a GO "transfer" annotation for a gene in our genome of interest

Making "transfer" annotations on a genome of interest is not easy. First, and foremost, you must not rely on the assigned function (if any) in the genome annotation. These annotations are likely automatic and do not intend to be a permanent and validated functional association for the gene. The following example describes the process of making an annotation for the YP_008430876.1 - TROLL_93 "tail assembly chaperone" gene product in *Bacillus phage Troll*, a recently sequenced bacteriophage genome. It is intended to be an illustration for the process, not a direct template you should follow in your annotations, and the main steps and concepts introduced apply to any other genome of interest.

BLASTP and HHpred

The first thing to do is to run a search with the main programs we use in this unit: BLAST and HHpred. In this case, we modify the BLASTP parameters to ask for 5,000 targets. This is a good trick, because it allows you to detect similarity with more distant things that the lot of closely related genomes (usually poorly annotated in bacteriophages) that populate the first ~100 rows. Another convenient trick is to exclude Eukarya to speed up and focus the search (since we will rarely be able to make use of hits with eukaryotic organisms to faithfully annotate a bacteriophage gene).

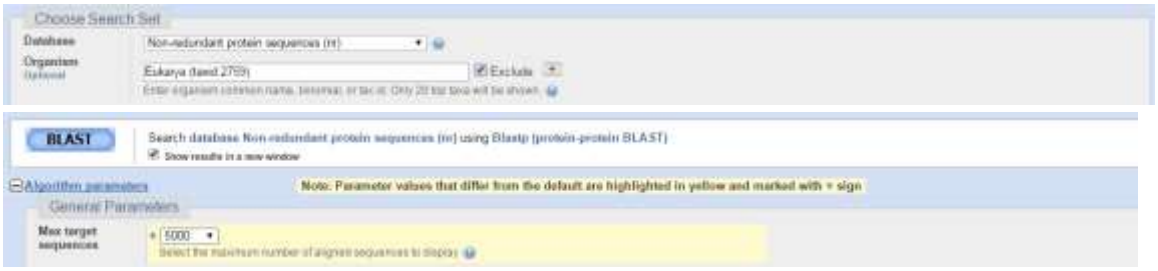


Figure 6 – Setting up the BLAST search.

When trying to annotate with these search tools, the first thing to do is to look for hits on relevant model organisms. Most experimental work and serious annotation on bacteriophages has been done on a handful of them. These include Enterobacteria phage lambda, *Enterobacteria phage T4* and *Enterobacteria phage T7*. Closer to *Bacillus phage Troll*, several *Mycobacteriophages* (L5, L1, TM4 and D29) have been carefully annotated, and the same is true for *Bacillus phage SPO1*, *Listeria* phages A511, PSA and A118, and *Staphylococcus aureus* phages G1, phiMR11 or SA4, *Bacillus cereus* bacteriophages BCP78 and B4, and *Bacillus phage vB_BceM-Bc431v3* (this is by no means an exclusive list).

The BLASTP and HHpred results in this case are rather disappointing. HHpred returns only high-quality hits to generic domains, which we cannot use. BLASTP does not return any slam-dunk hits (such as a hit to an *Enterobacteria phage lambda* tail chaperone, which would come in handy). In fact, only a few entries in the BLASTP result list hit genes annotated as "tail chaperones"[3]. The first one comes from

[3] BLASTP will not provide you with an extensive list of results. Identical proteins, for instance, will be masked and only one representative will be reported. If you find experimental evidence for what looks (from the given name/function) like a homolog of your gene, it is good practice to perform a BLASTP limiting your search to that specific *Organism*.

Bacillus phage Moonbeam. Following the protein accession [AIW03469.1], we can see on the right *Related information* tab that we are lucky, since there seems to be at least one article in PubMed citing this gene. This is a recent *Genome Announcement* paper on the "Complete Genome Sequence of Bacillus megaterium Myophage Moonbeam" by Cadungog *et al.* [PMID:25593264]. The protein is also accessible at UniProt, with accession number A0A0A0RPE2. Reading the paper, we find the following statement:

> Several functional proteins were identified using BLASTp and InterProScan analyses (**6**, **7**). Genes encoding structural proteins include a capsid protein, portal, prohead protease, tail proteins, tail chaperones, tape measure protein, tail proteins, and multiple components of the baseplate. The tail chaperone had an unusual +1 frameshift to its secondary product, where most Caudovirales use a −1 frameshift to encode their secondary tail chaperone (**8**).

This leads us to reference 8: Xu *et al.* "Conserved translational frameshift in dsDNA bacteriophage tail assembly genes." [PMID:15469818]. Here we find this:

> The gene encoding the tape measure protein is easily recognizable in the genome because it is very long (usually more than 2 kb) and the encoded protein is predicted to be largely α-helical. Furthermore, the order of the tail genes is highly conserved. Notably, the major tail protein gene is always upstream of the tape measure protein gene, and, as we show here, between these two genes there are typically two overlapping open reading frames (ORFs) that are related by a programmed translational frameshift.
>
> In bacteriophage λ, between the major tail protein gene *V* and tape measure protein gene *H*, two proteins−gpG and gpGT−are encoded, the second by a −1 translational frameshift (Figure 1A) (Levin et al., 1993). Both proteins are required for tail assembly even though neither is part of the mature tail structure. Near the end of gene *G*, a "slippery sequence" in the mRNA, 5′-GGGAAAG-3′, causes about 3.5% of the ribosomes to slip back one nucleotide, with the shifted ribosomes then continuing to read in the −1 reading frame to make a larger fusion protein, gpGT.

and this:

> comparative genomic studies show that dsDNA-tailed phages with very similar virion morphology often have structural genes with very different primary sequences (Brussow and Hendrix, 2002). Yet the head and tail genes of these phages normally have the same or similar functional gene order despite the frequent lack of demonstrable homology (Casjens et al., 1992).

Enterobacteria phage lambda gene *G* (lambdap14; NP_040593.1) has a UniProt accession (P03734), and QuickGO shows three associated annotations using evidence code IEA (Inferred from Electronic Annotation).

Database	Gene Product ID	Symbol	Qualifier	GO Identifier	GO Term Name	Aspect	Evidence	Reference	With	Taxon	Date	Assigned By	Product Form ID
					Process								
UniProtKB	P03734	G		GO:0019079	viral release from host cell	P	IEA	UniProt Keywords2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-KW:KW-1188	10710	20150314	UniProt	
					Component								
UniProtKB	P03734	G		GO:0033643	host cell cytoplasm	C	IEA	UniProt Keywords2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-KW:KW-1035	10710	20150314	UniProt	
UniProtKB	P03734	G		GO:0033643	host cell cytoplasm	C	IEA	UniProt Subcellular Location2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-SubCell:SL-0331	10710	20150314	UniProt	

Figure 7 – Annotations for Enterobacteriophage lambda gene *G* (P03734).

Making a "regular" annotation

IEA codes are not really what we are aiming for. IEA annotations are tags automatically assigned to genes using rudimentary mappings to function (e.g. the presence of an InterProt domain or of keywords indicating function in its product definition). IEA annotations are deleted one year after their generation[4]. You can consider them as a to-do list of sorts for UniProt biocurators. Hence, the idea is to properly annotate the *Enterobacteria phage lambda* gene *G* ourselves, so that we can then transfer the manual annotation. The first place to look is the cited reference (Levin et al. 1993), but this paper is not freely accessible. A PubMed search for it, however, will return also related papers, which we can check.

[4] You may find in QuickPro that a gene you are about to annotate already has several IEA annotations. These can give you pointers as to what you can expect to annotate for that gene, and the WITH record may provide you with some clues as to where the annotation comes from. Remember, however, that these are low quality, automatically generated annotations. You can not use them (i.e. enter them in CACAO and count them as annotations if they are not present there), and the sources they are based on (conserved domains or keywords) are unlikely to be of any use for performing a regular or transfer annotation. In other words, for intents and purposes of this work you should proceed as if there were not IEA annotations.

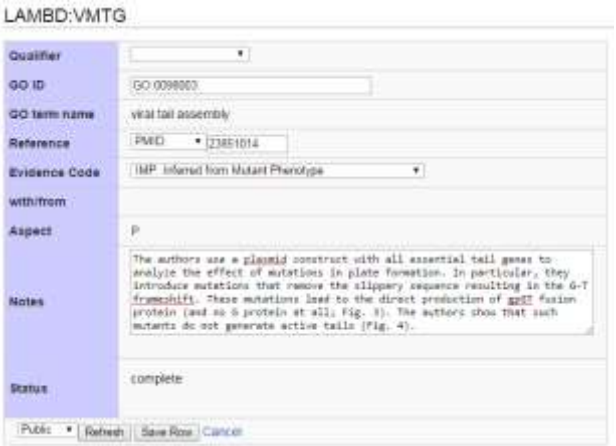
Figure 8 – Following reference (Levin et al. 1993).

On first inspection, one of them seems to provide enough information for an annotation (Xu *et al.* "A Balanced Ratio of Proteins from Gene G and Frameshift-Extended Gene GT Is Required for Phage Lambda Tail Assembly" [PMID:23851014]). The paper clearly demonstrates that the G protein (and its frameshited GT version) is required for tail assembly. Tail assembly (as QuickGO will tell us) is a well-defined biological process with GO term "GO:0098003 - viral tail assembly". This term has two children (fiber and baseplate assembly) but the paper does not specify the role of the *G* gene to this level of detail.

Hence, we'll create an annotation in CACAO for *Enterobacteria phage lambda* gene *G*. This annotation will use the IMP (Inferred from Mutant Phenotype) as the evidence code, and use Figures 3 and 4 of the paper as the main source of evidence[5]. The annotation note will read something like:

"The authors use a plasmid construct with all essential tail genes to analyze the effect of mutations in plate formation. In particular, they introduce mutations that remove the slippery sequence resulting in the G-T frameshift. These mutations lead to the direct production of gpGT fusion protein (and no G protein at all; Fig. 3). The authors show that such mutants do not generate active tails (Fig. 4)"

The page for *Enterobacteria* phage lambda gene *G* already exists in CACAO/GONUTS (LAMBD:VMTG) and contains the three IEA annotations we saw in QuickGO, so we will just add ours to the table.



[5] An annotation on a published article cannot be made based on the information provided in the abstract. You should read the manuscript and identify (and name in your Notes) the specific figures/tables/paragraphs of the paper that provide the experimental evidence that you are using for determining the GO term and the evidence code of your annotation.

Figure 9 – Making a "regular" annotation in CACAO.

Making a transfer annotation

Now that we have a nice and tidy (one hopes) "regular" annotation for lambda phage gene *G*, we must figure out how to "transfer" the annotation to our *Troll* gene (TROLL_93, YP_008430876.1, S5YQ92). Ideally, we would rely on a BLASTP or HHpred hit. The paper by Cadungog *et al.* [PMID:25593264] asserts that there is homology between Bacillus phage Moonbeam CPT_Moonbeam72 (A0A0A0RPE2) tail assembly chaperone and those studied by Xu *et al.* [PMID:15469818]. Using targeted BLASTP against the genomes of several phages listed by Xu *et al.* in their supplementary information (e.g. Bacteriophage PSA), our Troll protein consistently matches the tail assembly chaperone in several of them, but never below the threshold e-value listed in the CACAO GO_REF instructions. And BLASTing directly against the Enterobacteria phage lambda genome does not generate any valid hits.

Hence, a nice and tidy homology-based annotation is not possible. That, however, does not mean that a transfer annotation is impossible.

Browsing the literature (starting with Xu *et al.* [PMID:15469818] and following references and cross-listed papers in PubMed), we can identify several instances that remark on the conservation of the *Enterobacteria phage lambda* G-T-H gene arrangement. For instance, Schuch* and Fischetti (2006) remark on their "Detailed Genomic Analysis of the Wβ and γ Phages Infecting Bacillus anthracis: Implications for Evolution of Environmental Fitness and Antibiotic Resistance" [PMID:16585764]:

> Recently, a highly conserved programmed translational −1 frameshift was found to be common among the tail assembly genes of most double-stranded DNA phage (70) ... Analysis of the γ and Wβ sequences did identify two putative orthologs of *G* and *T*, *orf11* and *orf12*, which are of the appropriate size and, like *G* an *T*, are encoded between a major tail protein (*orf10*) and a tape measure protein (*orf13*). Unlike, *G* and *T*, however, the *orf11* and *orf12* loci do not overlap and appear to lack a conventional slippery sequence. Despite this, a nonconventional slippery sequence, providing either a −2 or a +1 frameshift, could fuse the *orf11* and *orf12* products and is worthy of further investigation.

By combining the weak but consistent similarity of TROLL_93 with many other reported tail assembly chaperones and the multiple statements about synteny of the frame-shifting tail chaperones [TROLL_93-TROLL_92] preceding the tapemeasure gene (which is reasonably well-annotated in Troll: TROLL_94), we have solid grounds to postulate that TROLL_93 is a distant homolog of *Enterobacteria phage lambda* gene *G* (with TROLL_92 playing the part of *T*), and hence transfer the involvement of this putative tail chaperone in tail assembly from *Enterobacteria phage lambda* to *Bacillus phage Troll*.

We will do this by means of an *IGC – Inferred by Genomic Context* evidence code, using the *Enterobacteria* phage lambda gene *G* and several other homologs with known synteny conservation in the *WITH* field. Our reference will be the CACAO GO_REF. The note will read as follows:

"BLASTP shows that the protein coded by TROLL_93 is a homolog of the "tail assembly chaperone" AIW03469 (coded by CPT_Moonbeam72), which is known to be homologous to several phage tail assembly chaperones displaying a conserved frameshift and preserved gene organization consisting of the two frameshifted chaperones (TROLL_93 and TROLL_92) upstream of the tapemeasure gene (TROLL_94) [PMID:25593264, PMID:15469818, PMID:16585764]. Examples of these include the chaperones coded by *Listeria* Bacteriophage PSA *ORF11* (CAC85567), and *Enterobacteria* phage lambda gene *G* (AAA96546) or *Streptococcus thermophilus* bacteriophage Sfi19 *orf117* (AAC39294). Given the strong synteny conservation and the specific nature of the genes involved, the TROLL_93

gene product can be assumed to have a similar chaperone role in tail assembly to its Enterobacteria phage lambda counterpart."

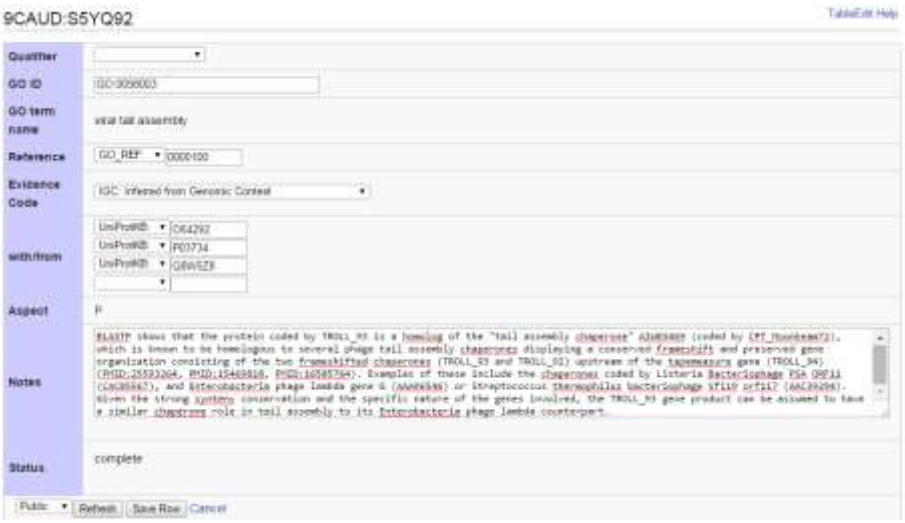


Figure 10 – Making a "transfer" annotation in CACAO.

And hence, by means of a "regular" and a "transfer" GO annotation we have managed to annotate the product of gene TROLL_93 to a specific biological process ("GO:0098003 - viral tail assembly) as experimentally established in *Enterobacteria phage lambda*. So on to the next gene/annotation…