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 section of BIOL 316L Phage Hunters Genome Analysis we will be making Gene Ontology (GO) annotations of phage genes. 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 (such as "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 Phage Hunters 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.

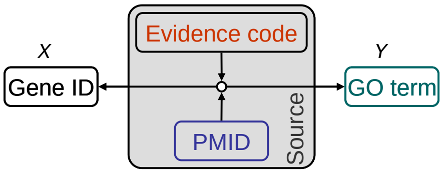


Figure 1 – Schematic diagram for "regular" GO annotations. A gene (X) is annotated as having GO term (Y), which specifies a well-defined function/process/component, using an evidence code and the PubMed ID (PMID) of a scientific article as the *source* for the annotation.

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].

Phage Hunters annotations

In Phage Hunters Genome Analysis we are sequencing a completely novel organism. This means that no experimental work has been done on our phage and, therefore, we cannot perform "regular" GO annotations. 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 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 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 [2].

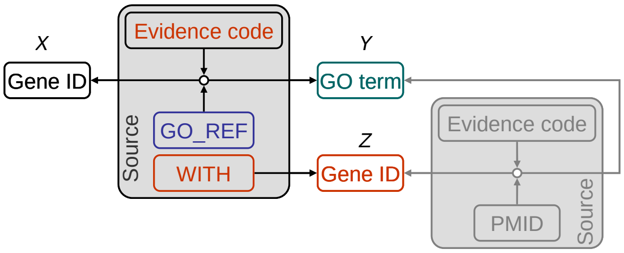


Figure 2 – Schematic diagram for Phage Hunters "transfer" GO annotations. A gene (X) is annotated as having GO term (Y), by establishing that gene X is homologous to gene Z, where the annotated function/process/component (Y) has been established through experimental means. The source for the annotation 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.

Our annotation process is, therefore, slightly different from the regular case. In the case of Phage Hunters annotations, we will be stating that gene X has function/process/component Y, based on its

[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 the instructors.

[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. In the case of phages, you should attempt to explain the possible role of the protein in phage biology when transferring annotations from non-phage homologs.

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. In our case, we will use our very own GO_REF (GO_REF:0000100; Gene Ontology annotation by SEA-PHAGE biocurators), the description of which you can find here (http://www.geneontology.org/doc/GO.references). Crucially, the *source* includes also another element, using the *WITH* field of GO annotations. This is the identifier for the gene (Z) that we are transferring the annotation from.

The Phage Hunters 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 Phage Hunters is fundamentally reversed. Here we start with the genes of our phage, 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 GO annotations to make our annotations (i.e. transferring the homolog annotation to our phage gene)

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

Annotations using the CACAO interface

To facilitate and standardize the annotation process, we will be using the CACAO/GONUTS interface, developed by Jim Hu at Texas A&M University. CACAO is an intercampus annotation competition, but we will be using a private subspace of CACAO to perform Phage Hunters annotations. 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)

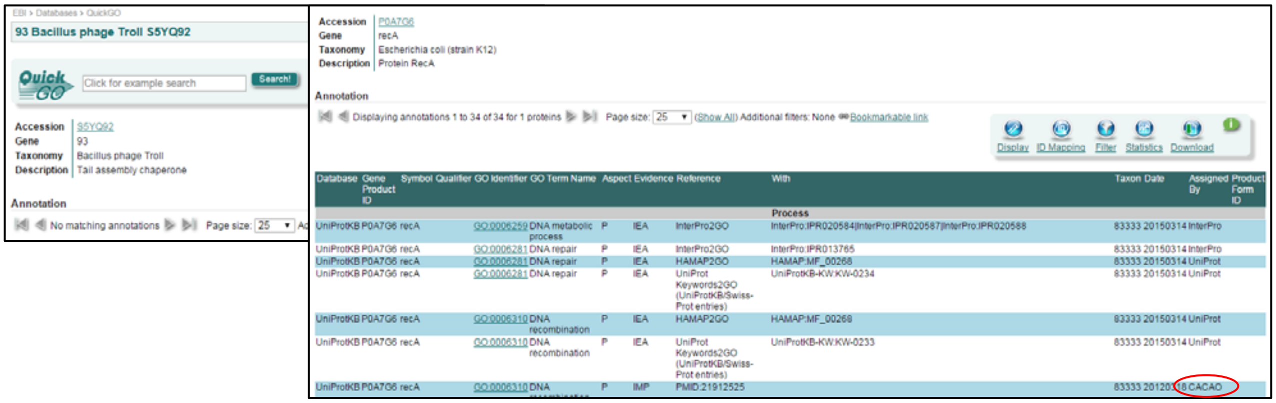
Getting phage genes

The first step is to get phage genes to annotate. In order for our annotations to be incorporated into GO, we must use genes with an assigned UniProt accession. Genes get UniProt accessions after the genome sequence has been uploaded to GenBank (which will happen at the end of the semester), so we will be performing GO annotations on a UMBC phage from a previous year and use the experience to hand craft our own GO annotations for this year's phage, so that we can upload them directly in our GenBank submission.

The first step is obviously to get a gene to annotate. Browse the *Bacillus phage Troll* genome and look for possible candidates for annotation (try first genes with some annotation other than "hypothetical protein"). Once you have identified a gene, get its protein accession (e.g. YP_008430876.1 for TROLL_93 "tail assembly chaperone"). Go to the UniProt website and search with this accession. You will get as a result an item with a UniProt accession (S5YQ92).

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. In the case of S5YQ92 there are no previous annotations, but this is not necessarily the case for other genes, such as *Escherichia coli recA* (P0A7G6).



EBI > Databases > QuickGO

93 Bacillus phage Troll S5YQ92

Accession	S5YQ92
Gene	93
Taxonomy	Bacillus phage Troll
Description	Tail assembly chaperone

Annotation

No matching annotations Page size: 25

Accession	P0A7G6
Gene	recA
Taxonomy	Escherichia coli (strain K12)
Description	Protein RecA

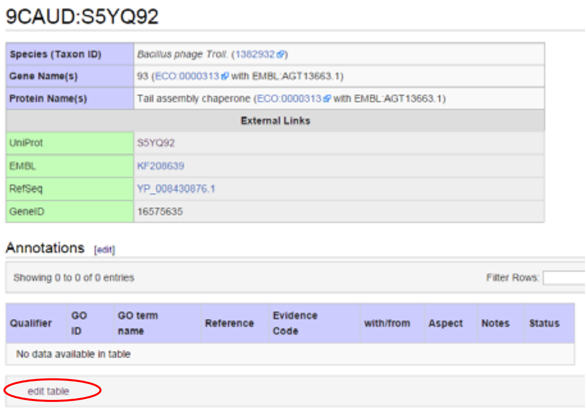
Annotation

Displaying annotations 1 to 34 of 34 for 1 proteins Page size: 25 (Show All) Additional filters: None Bookmarkable link

Database	Gene Product ID	Symbol	Qualifier	GO Identifier	GO Term Name	Aspect	Evidence	Reference	With	Taxon	Date	Assigned By	Product Form ID
									Process				
UniProtKB	P0A7G6	recA		GO:0006259	DNA metabolic process	P	IEA	InterPro2GO	InterPro:IPR020584|InterPro:IPR020587|InterPro:IPR020588	83333	20150314	InterPro	
UniProtKB	P0A7G6	recA		GO:0006281	DNA repair	P	IEA	InterPro2GO	InterPro:IPR013765	83333	20150314	InterPro	
UniProtKB	P0A7G6	recA		GO:0006281	DNA repair	P	IEA	HAMAP2GO	HAMAP:MF_00268	83333	20150314	UniProt	
UniProtKB	P0A7G6	recA		GO:0006281	DNA repair	P	IEA	UniProt Keywords2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-KW:KW-0234	83333	20150314	UniProt	
UniProtKB	P0A7G6	recA		GO:0006310	DNA recombination	P	IEA	HAMAP2GO	HAMAP:MF_00268	83333	20150314	UniProt	
UniProtKB	P0A7G6	recA		GO:0006310	DNA recombination	P	IEA	UniProt Keywords2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-KW:KW-0233	83333	20150314	UniProt	
UniProtKB	P0A7G6	recA		GO:0006310	DNA	P	IMP	PMID:21912525		83333	20120[illegible]18	CACAO	

Figure 3 – Screenshot of the QuickGo pages for *Bacillus phage Troll* TROLL_93 "tail assembly chaperone" and *Escherichia coli recA*.

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 (`S5YQ92`; that is: 9CAUD:S5YQ92) in the CACAO interface.



9CAUD:S5YQ92

Species (Taxon ID)	*Bacillus phage Troll.* (1382932)
Gene Name(s)	93 (ECO:0000313 with EMBL:AGT13663.1)
Protein Name(s)	Tail assembly chaperone (ECO:0000313 with EMBL:AGT13663.1)
External Links	
UniProt	S5YQ92
EMBL	KF208639
RefSeq	YP_008430876.1
GeneID	16575635

Annotations [edit]

Showing 0 to 0 of 0 entries

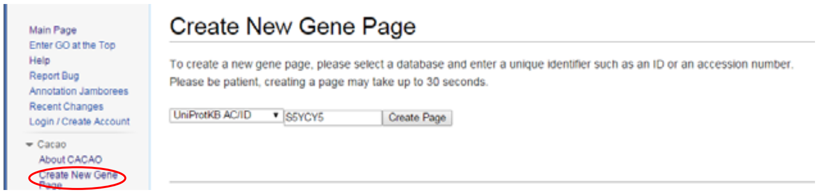
Filter Rows:

Qualifier	GO ID	GO term name	Reference	Evidence Code	with/from	Aspect	Notes	Status
No data available in table								

Figure 4 – Screenshot of the CACAO page for *Bacillus phage Troll* TROLL_93 "tail assembly chaperone".

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 and paste your UniProt accession into it, then hit *Create Page*.

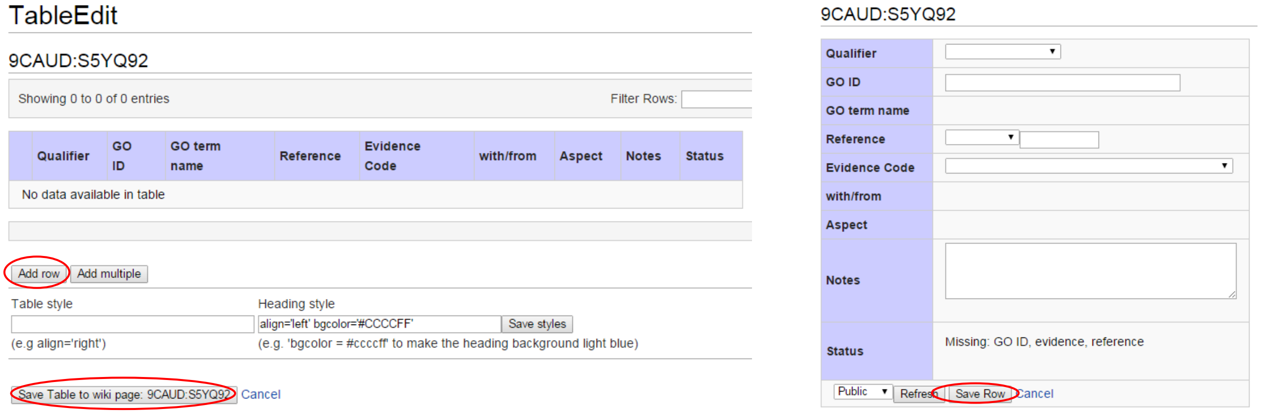


Create New Gene Page

To create a new gene page, please select a database and enter a unique identifier such as an ID or an accession number. Please be patient, creating a page may take up to 30 seconds.

UniProtKB AC/ID ▾ S5YCY5 Create Page

Figure 5 – Gene page creation in CACAO.



TableEdit

9CAUD:S5YQ92

Showing 0 to 0 of 0 entries

Filter Rows:

Qualifier	GO ID	GO term name	Reference	Evidence Code	with/from	Aspect	Notes	Status
No data available in table								

Table style: (e.g align='right')

Heading style: align='left' bgcolor='#CCCCFF' (e.g. 'bgcolor = #ccccff' to make the heading background light blue)

9CAUD:S5YQ92

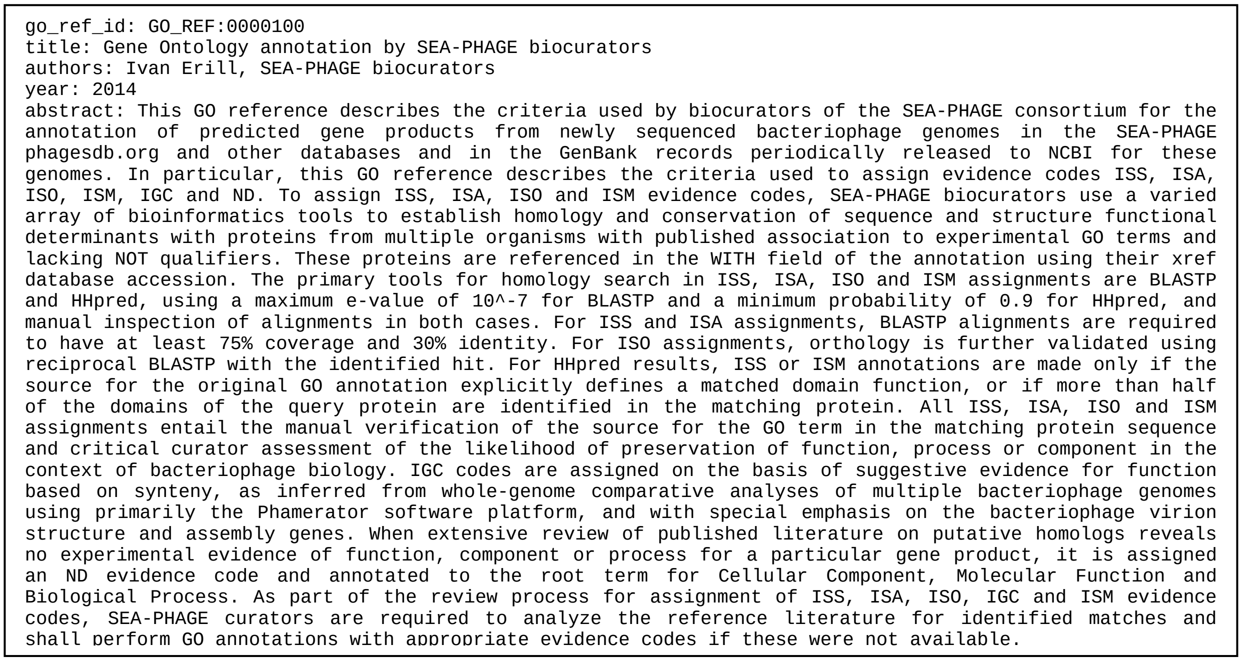
Qualifier	▾
GO ID	
GO term name	
Reference	▾
Evidence Code	▾
with/from	
Aspect	
Notes	
Status	Missing: GO ID, evidence, reference

Figure 6 – Making an annotation in CACAO. Adding a row to the annotation table (left) and making the annotation (right).

Making an annotation

Once you have created a gene page (or using an existing one), you can make annotations using the *edit table* link shown in Figure 4 and then clicking on the *Add row* button. This will bring up a form to create the annotation. 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 Phage Hunters 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). See examples in any CACAO annotation, such as this one. Take also a look at the instructions on the GO REF. You should also note here the rationale for transferring the function/process/component from the homolog to the phage gene, especially when the homolog is not a phage gene (that is, why you think the same function/process/component applies to the phage gene).



```
go_ref_id: GO_REF:0000100
title: Gene Ontology annotation by SEA-PHAGE biocurators
authors: Ivan Erill, SEA-PHAGE biocurators
year: 2014
abstract: This GO reference describes the criteria used by biocurators of the SEA-PHAGE consortium for the
annotation of predicted gene products from newly sequenced bacteriophage genomes in the SEA-PHAGE
phagesdb.org and other databases and in the GenBank records periodically released to NCBI for these
genomes. In particular, this GO reference describes the criteria used to assign evidence codes ISS, ISA,
ISO, ISM, IGC and ND. To assign ISS, ISA, ISO and ISM evidence codes, SEA-PHAGE biocurators use a varied
array of bioinformatics tools to establish homology and conservation of sequence and structure functional
determinants with proteins from multiple organisms with published association to experimental GO terms and
lacking NOT qualifiers. These proteins are referenced in the WITH field of the annotation using their xref
database accession. The primary tools for homology search in ISS, ISA, ISO and ISM assignments are BLASTP
and HHpred, using a maximum e-value of 10^-7 for BLASTP and a minimum probability of 0.9 for HHpred, and
manual inspection of alignments in both cases. For ISS and ISA assignments, BLASTP alignments are required
to have at least 75% coverage and 30% identity. For ISO assignments, orthology is further validated using
reciprocal BLASTP with the identified hit. For HHpred results, ISS or ISM annotations are made only if the
source for the original GO annotation explicitly defines a matched domain function, or if more than half
of the domains of the query protein are identified in the matching protein. All ISS, ISA, ISO and ISM
assignments entail the manual verification of the source for the GO term in the matching protein sequence
and critical curator assessment of the likelihood of preservation of function, process or component in the
context of bacteriophage biology. IGC codes are assigned on the basis of suggestive evidence for function
based on synteny, as inferred from whole-genome comparative analyses of multiple bacteriophage genomes
using primarily the Phamerator software platform, and with special emphasis on the bacteriophage virion
structure and assembly genes. When extensive review of published literature on putative homologs reveals
no experimental evidence of function, component or process for a particular gene product, it is assigned
an ND evidence code and annotated to the root term for Cellular Component, Molecular Function and
Biological Process. As part of the review process for assignment of ISS, ISA, ISO, IGC and ISM evidence
codes, SEA-PHAGE curators are required to analyze the reference literature for identified matches and
shall perform GO annotations with appropriate evidence codes if these were not available.
```

Figure 7 – The Gene Ontology annotation by SEA-PHAGE biocurators GO_REF:0000100.

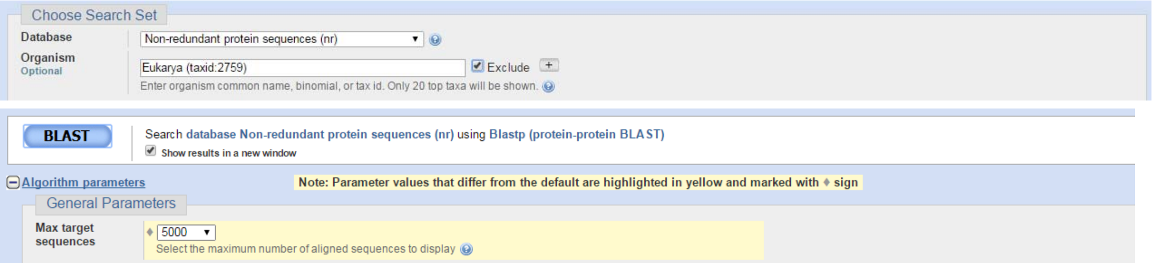
Once you have filled up the required fields, click *Save Row* and then, on the *TableEdit* page don't forget to click on the *Save Table to wiki page* button (Figure 6). Otherwise your annotation will NOT be saved.

Making a GO annotation for a phage gene

Making "transfer" annotations is not easy. First, and foremost, you must not rely on the assigned function (if any) in the genome annotation. This was done by your peers, but not with the intention of providing a permanent and validated functional association for the gene as we will do now with the GO annotation. The following describes the process of making an annotation for the YP_008430876.1 - TROLL_93 "tail assembly chaperone" gene. It is intended to be an illustration for the process, not a direct template you should follow in your annotations.

BLASTP and HHpred

The first thing to do is to run a search with the main programs we use in class: 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 phages (usually poorly annotated) 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 phage gene).



Choose Search Set

Database: Non-redundant protein sequences (nr)

Organism (Optional): Eukarya (taxid:2759) ☑ Exclude

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.

BLAST — Search database Non-redundant protein sequences (nr) using Blastp (protein-protein BLAST)

☑ Show results in a new window

Algorithm parameters

Note: Parameter values that differ from the default are highlighted in yellow and marked with ♦ sign

General Parameters

Max target sequences: ♦ 5000

Select the maximum number of aligned sequences to display

Figure 8 – 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 phages. Most experimental work and serious annotation on phages has been done on a handful of them. These include *Enterobacteria* phage lambda, *Enterobacteria* phage T4 and *Enterobacteria* phage T7. Closer to home, 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). In fact, only a few entries in the BLASTP result list hit genes annotated as "tail chaperones" [3]. The first one comes from 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

[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*.

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).

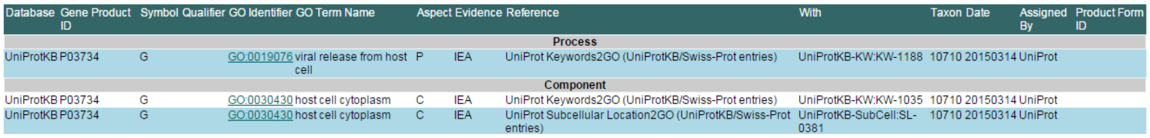
This leads us to reference 8: Xu *et al.* "Conserved translational frameshift in dsDNA bacteriophage tail assembly genes." [PMID:15469818]. Here we find this:

> 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).



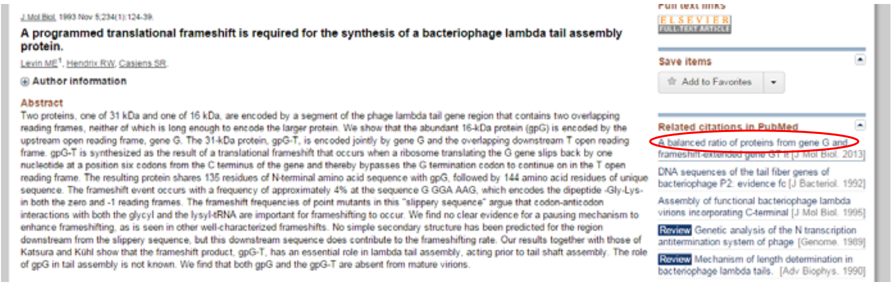
Database	Gene Product ID	Symbol	Qualifier	GO Identifier	GO Term Name	Aspect	Evidence	Reference	With	Taxon	Date	Assigned By	Product Form ID
							Process						
UniProtKB	P03734	G		GO:0019076	viral release from host cell	P	IEA	UniProt Keywords2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-KW:KW-1188	10710	20150314	UniProt	
							Component						
UniProtKB	P03734	G		GO:0030430	host cell cytoplasm	C	IEA	UniProt Keywords2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-KW:KW-1035	10710	20150314	UniProt	
UniProtKB	P03734	G		GO:0030430	host cell cytoplasm	C	IEA	UniProt Subcellular Location2GO (UniProtKB/Swiss-Prot entries)	UniProtKB-SubCell:SL-0381	10710	20150314	UniProt	

Figure 9 – 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 a 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 Troll 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 genes, 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.



J Mol Biol. 1993 Nov 5;234(1):124-39.

A programmed translational frameshift is required for the synthesis of a bacteriophage lambda tail assembly protein.

Levin ME[1], Hendrix RW, Casjens SR.

⊕ Author information

Abstract

Two proteins, one of 31 kDa and one of 16 kDa, are encoded by a segment of the phage lambda tail gene region that contains two overlapping reading frames, neither of which is long enough to encode the larger protein. We show that the abundant 16-kDa protein (gpG) is encoded by the upstream open reading frame, gene G. The 31-kDa protein, gpG-T, is encoded jointly by gene G and the overlapping downstream T open reading frame. gpG-T is synthesized as the result of a translational frameshift that occurs when a ribosome translating the G gene slips back by one nucleotide at a position six codons from the C terminus of the gene and thereby bypasses the G termination codon to continue on in the T open reading frame. The resulting protein shares 135 residues of N-terminal amino acid sequence with gpG, followed by 144 amino acid residues of unique sequence. The frameshift event occurs with a frequency of approximately 4% at the sequence G GGA AAG, which encodes the dipeptide -Gly-Lys- in both the zero and -1 reading frames. The frameshift frequencies of point mutants in this "slippery sequence" argue that codon-anticodon interactions with both the glycyl and the lysyl-tRNA are important for frameshifting to occur. We find no clear evidence for a pausing mechanism to enhance frameshifting, as is seen in other well-characterized frameshifts. No simple secondary structure has been predicted for the region downstream from the slippery sequence, but this downstream sequence does contribute to the frameshifting rate. Our results together with those of Katsura and Kühl show that the frameshift product, gpG-T, has an essential role in lambda tail assembly, acting prior to tail shaft assembly. The role of gpG in tail assembly is not known. We find that both gpG and the gpG-T are absent from mature virions.

Full text links

ELSEVIER FULL-TEXT ARTICLE

Save items

☆ Add to Favorites ▾

Related citations in PubMed

A balanced ratio of proteins from gene G and frameshift-extended gene GT is [J Mol Biol. 2013]

DNA sequences of the tail fiber genes of bacteriophage P2: evidence fc [J Bacteriol. 1992]

Assembly of functional bacteriophage lambda virions incorporating C-terminal [J Mol Biol. 1995]

Review Genetic analysis of the N transcription antitermination system of phage [Genome. 1989]

Review Mechanism of length determination in bacteriophage lambda tails. [Adv Biophys. 1990]

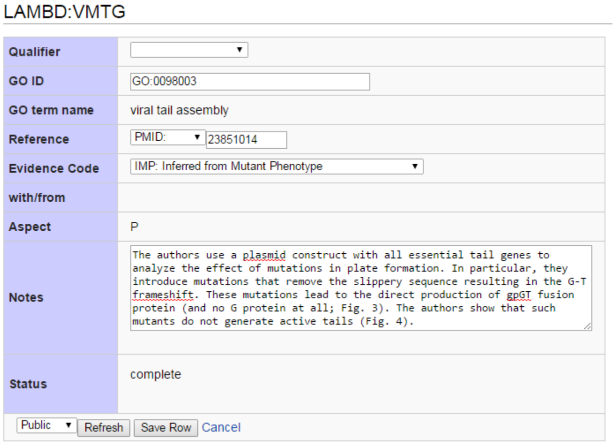
Figure 10 – Following reference (Levin et al. 1993).

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 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.



LAMBD:VMTG

Qualifier	▾
GO ID	GO:0098003
GO term name	viral tail assembly
Reference	PMID: ▾ 23851014
Evidence Code	IMP: Inferred from Mutant Phenotype ▾
with/from	
Aspect	P
Notes	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).
Status	complete

Figure 11 – Making a "regular" annotation in CACAO.

[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.

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. 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.

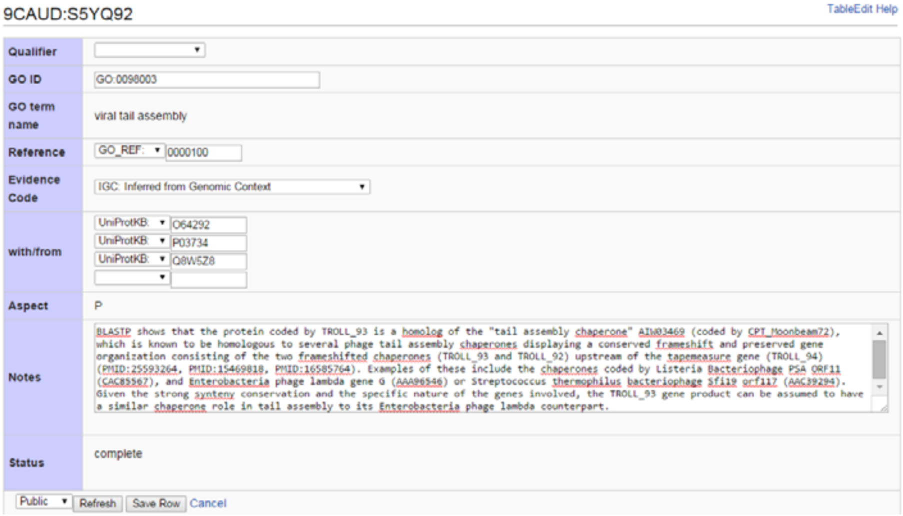
Browsing the literature (starting with Xu *et al.* [PMID:15469818] and following references and cross-listed papers in PubMed), we can identify several instances that remark on the conservation of the Enterobacteria phage lambda G-T-H gene arrangement. For instance, Schuch* and Fischetti (2006) remark 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 SEA-PHAGES 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.”



9CAUD:S5YQ92

Qualifier	
GO ID	GO:0098003
GO term name	viral tail assembly
Reference	GO_REF: 0000100
Evidence Code	IGC: Inferred from Genomic Context
with/from	UniProtKB: O64292 UniProtKB: P03734 UniProtKB: Q8W5Z8
Aspect	P
Notes	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.
Status	complete

Figure 12 – 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…