Ensembl Fungi and JGI: Community-driven manual gene curation

Introduction

With easy access to affordable sequencing technologies, the volume of genomic data (particularly, for microbial species) is growing exponentially. A majority of these undergo automated gene annotation before entering the public domain. Despite advances in gene prediction algorithms, they still cannot automatically resolve all complexities surrounding the precise location and structure of the genome elements. It is not uncommon, therefore, for there to be significant differences in the quality of these gene annotations compared to carefully manually curated gene sets. It is also the case that for certain key pathogens there are conflicting gene sets circulating in different communities owing to preferred tools and protocols. Given that a high-quality gene set is the key to enabling any further inferences, this is an important problem to address.

Approaches to manual gene model curation

While essential, manual gene curation is a laborious undertaking and often unfunded except for model species. There are several approaches to doing manual gene curation ranging from dedicated teams looking deeply at gene structures to enabling Wikipedia-style community editing.

In this course, we would like to draw your attention to two approaches adopted by the Joint Genome Institute (JGI) and Ensembl Fungi that attempt to make this a more manageable task. You can find specific instructions on how to use both these platforms in Appendix A and B at the end of this tutorial.

JGI MycoCosm: Pre-filtered genes open to editing by collaborating scientists

The JGI Fungal annotation pipeline uses several gene prediction algorithms, including ab-initio, homology, and EST-based gene modelers to produce multiple overlapping gene models for a given locus. A heuristic filtering process then chooses a “best” model at each locus according to specific weights given to each model based on evidence, completeness, homology, presence of known domains and structures, etc. These filtered models are stored in the “FilteredModels” track on the JGI browser. A copy of the FilteredModels is stored as the GeneCatalog. Users with specific privileges (collaborating scientists) can modify, add and remove models from the GeneCatalog using available manual curation tools. These corrected gene models eventually become the reference list of gene models for this organism. More on how to use the MycoCosm platform can be found in Appendix A.
**Ensembl Fungi and Apollo: Distributed among species-specific communities**

Ensembl Fungi facilitates collaborations between research groups interested in the same species to work together to redefine the *de facto* gene set. The process usually begins with an interested community approaching Ensembl Fungi. Ensembl Fungi sets up an Apollo (more below) instance for them, provides training and user support and collates evidence from this community necessary for the curation into Apollo tracks. Ensembl Fungi also helps distribute the work across the members, typically generating gene lists or chromosome regions for each user to examine. Once the curation effort is complete, the gene sets undergo a QC process and get integrated into Ensembl Fungi.

**Apollo** ([http://genomearchitect.github.io/](http://genomearchitect.github.io/)) is a web-based collaborative gene-editing plugin for the JBrowse genome viewer. With it, a team of researchers spread across the globe can work on the same sequences at the same time using just their internet browser. All the gene models in a typical fungal genome can be manually curated in months using this approach. For instance, a recent curation project for *Botrytis cinerea* involved close to 50 members spread across several countries and took around six months, resulting in a completely revised gene set. A similar project was completed for *Blumeria graminis* and one is underway for *Zymoseptoria tritici*. More on Apollo can be found in Appendix B.

**Data to support manual gene curation**

All manual gene curation approaches make use of additional ‘evidence’ to help curators make decisions about gene models. Some supporting data types are listed below - not all of them will exist for all species.

1. Transcriptome data
   a. RNA-Seq coverage (BigWig files)
   b. RNA-Seq read mapping (BAM files)
   c. PacBio Iso-Seq data
   d. Expressed Sequence Tag (EST) and cDNA data
2. Short read intron data
3. Homology and conservation data
4. Polyadenylation site data (polyA seq)
5. Protein sequences
6. Long read and short read sequencing data
7. Other gene sets available for the species
8. Alignments to gene sets and proteins of closely related species
Types of changes made during curation

Most of the discussions in this tutorial focus on **structural curation**. This can involve:

- Verifying the accuracy of **splice junctions**
- Deciding whether (biologically meaningful) **splice variants** can be detected
- Adjusting the transcription start sites (**5′-UTR**) and polyadenylation sites (**3′-UTR**) to the most likely position
- Adjusting transcript boundaries
- Splitting gene models or merging fragments
- Adding new genes
- Removing bad gene models (false positives)
- Choosing the best from a proposed set of gene models for a given locus

There is also **functional curation** that can be done. This can involve:

- Assigning or changing a gene name
- Deciding on the biotype of a gene (based on strong conservation and/or very large ORF)
- Adding Gene Ontology (GO) terms
- Adding or changing the function or description of a gene or product

Examples of structural gene curation

This section contains examples and suggestions for the structural curation tasks listed above. These should be treated as broad guidelines as there will inevitably be subtle species-specific/community-specific differences in the ways that supporting data is interpreted. The screenshots below show Apollo displaying either *Botrytis cinerea* or *Zymoseptoria tritici* data. See Appendix B for a description of the Apollo interface.

Verifying the accuracy of splice junctions

Errors in splice junction prediction can occur when the splice donor site is GC instead of GT; this non-canonical splice site is not easily detected by gene prediction tools. Occasionally you will see a GT splice junction in a predicted gene, whereas a GC junction nearby is the correct one. Another error that sometimes occurs is the wrong prediction of the translation start site if an in-frame start codon is nearby. Sometimes the gene prediction then calls the second ATG as the likely start codon, while the first (upstream) ATG codon is correct.
To assist in the decision to modify a splice site, you can also download the translated sequences (menu option available in Apollo) and use them to search well-curated protein databases, such as UniProt, to see if you can resolve the question using protein alignments. Incorrect splice sites would likely cause gaps in the alignments. Keep in mind that the best alignment may be the exact prediction from which you initiated your annotation; you should not consider the identical protein from your organism as external evidence supporting the annotation. Instead, look at alignments to proteins from other organisms. If there does not appear to be any way to resolve the non-canonical splice, leave it as is and add a comment.

**Detecting new splice variants**

Another important thing to look out for is splice variants. Sometimes you can see from the read mapping track that in a certain region, a proportion of reads is spliced whereas another proportion is unspliced. This is not always biologically relevant, such as below:
The reads that map across the intron above are most likely derived from unspliced or partially spliced RNAs. If the intron is not spliced, merging the two flanking exons leads to frameshifts and stop codons, and would never result in a functional protein. The proportion of unspliced or partially spliced RNAs can in some cases be high, up to 10%, as compared to the real exons. In the particular case illustrated in the figure, the *Botrytis cinerea* annotators decided that this was not alternative splicing. In other cases, alternative splicing may be meaningful, but it is up to your personal judgement to decide this as there are no general rules.

If alternative splicing occurs, and only one gene model is provided, you can duplicate the gene model (in Apollo by a double click) and make the adjustment to the splice junctions in the duplicate to create the splice variant. The splice variants must be numbered as different mRNAs (usually the mRNA gets the gene name followed by a suffix such as .1, .2, .3 and so on for different splice variants). An example of alternative splicing can be seen below in the hydrophobin gene Bhp1 (gene is in right to left orientation) in *Botrytis cinerea*. A small proportion of the transcripts have a shorter second intron, leading to an insertion of 13 amino acids in the third exon. The ratio of the major variant and minor variant is ~20:1; nevertheless this may be biologically meaningful.
Adjusting untranslated regions (UTRs)

In some cases, UTRs may not have been predicted by the gene prediction software that was used and, therefore, most gene models will not have any UTR information. Even if the gene models you are looking at contain UTR information, this is still worth checking. RNA-Seq coverage can be used to identify the start and end of transcription. Sometimes the place is fairly obvious, such as in the image below. The UTRs in the figure below can be extended to the approximate positions where the BigWig coverage track reaches the baseline.
Here are two rules to be considered for UTRs:

- When you adjust the 5’-UTR, make sure to always set it at a G. Many fungal transcription start sites are at the end of C/T rich regions and start with a G (preferentially with GA or GG). So, choose an appropriate G that follows a C/T rich stretch.
- A 3’-UTR ends at the site of polyadenylation, which is ~10-30 nucleotides downstream of a polyadenylation signal. In mammals, the polyadenylation signal is very easy and straightforward: AATAAA. In fungi, it is not conserved so well. Often the AAT is present, but the next three nucleotides have many options, some of which are preferred. Frequently, motifs around putative polyadenylation sites are AATAAT, AATATT, AATATC and AATACT (Gs in the last three nucleotides of the motif are less frequent). Be aware that the polyadenylation itself occurs 10-30 bases downstream, and there is a slight preference for the motif CA, where the poly(A)-tail is added to the A. That is a good place to end the 3’-UTR.

The gene model shown below is an example from the *Botrytis cinerea* annotation project. The users adjusted the 5’-UTR (gene is in inverse orientation) to a site where the RNA-Seq coverage drops <500 (the peak around the start codon has coverage ~100,000). The transcript starts with GACCTCG.

![Figure 5 - 5’-UTR adjustment](image)

The 3’-UTR of the same gene was adjusted as follows: the sequence motifs AATGAT and AATCTA occur shortly after one another. About 10 nucleotides downstream there is a sequence GCTA where coverage drops to <500. This is where the curators considered was the...
most likely place for the polyadenylation site and the end of the 3’-UTR. For such highly expressed genes, coverage is unlikely to drop entirely to zero.

Figure 6 - 3’-UTR adjustment

Adjusting transcript boundaries
It is not always clear where a transcript ends, as shown below (gene going from right to left). The coding sequence ends in the middle of the image, yet there is RNA-Seq coverage all the way to the left, suggesting a 3’-UTR that would be at least 1.5 kb in length! Also, the 3’-UTR would have an intron around position 1,198,300 which seems an unlikely situation. In this case, it may be a long non-coding RNA. In this example, the *Botrytis cinerea* curators decided to end the 3’-UTR at a position where a reasonable polyadenylation consensus is detected, and the length of the 3’-UTR is in the range of 50-200 nt. It would be meaningless (and probably misleading) to extend it to 1.5 kb.

Figure 7 - Confusing transcript end (Gene Bcin03g03530 in *Botrytis cinerea*)
Splitting gene models and merging fragments

For neighbouring genes with short intergenic regions, it is not always obvious where one gene ends and the next gene begins (as shown in the conflicting gene annotations below). The transcripts may even overlap if the neighbouring genes are transcribed convergently towards each other making it hard to determine intron and UTR limits. Aligning stranded data (in the evidence tracks) can help resolve this.

Furthermore, for compact genomes like *Zymoseptoria tritici* data from polyA studies can greatly help clear up confusion over merged gene models and rare read throughs.

![Image](image_url)

**Figure 8** - Two neighbouring genes wrongly merge together in the top pink gene model. These are likely to be two genes expressed differently (shown by RNA-Seq coverage)

In contrast, sometimes gene models that have been predicted as being separate might need merging. The image below shows an example of two gene models from a predicted set being merged together based on RNA-Seq data and models of closely related species.
Figure 9: The annotation in pink shows two genes but, based on alignments to other species, the curators have decided to merge these two genes into one model (Bcin14g05150, *Botrytis cinerea*)
Adding a new gene

Figure 10: Adding a new gene

The screenshot above shows a new gene that was added in *Botrytis cinerea* based on RNA-Seq data and matches to a closely related species. See section below on the ‘Absence of a gene’ for situations where RNA-Seq alignments alone may not always inform the presence of a new gene.

Choosing the best from a proposed set of gene models

If there are multiple gene predictions for a species, it is likely that there will be conflicting genes for a given locus. Often the curator’s job will be to choose the most representative and accurate model from the set as shown below.
Figure 11: The figure shows four alternative gene models at the top, followed by Illumina RNA-Seq and PacBio Iso-Seq data underneath. Two of the gene predictions contain a gene model merge even though the evidence points to two separate genes. In this instance, the curators have chosen the first model as the best representation for this locus.

Other interesting observations

Very small exons

If you examine the picture below (*Botrytis cinerea*), you will see that the third tiny exon appears not to be supported by RNA-Seq data. This exon is only 5 nucleotides in length and it is difficult/impossible to map RNA-Seq reads if the corresponding match is too short. This explains the gap in the read coverage. The gene model, which encodes a GH10 endoxylanase, is perfect as it is here, but required quite a bit of tweaking. There are several other mind-boggling gene models that have been seen (for instance, exons of 2 nucleotides).
Absence of a gene
It is entirely possible for there to be regions in the genome where you will see RNA aligned to a locus (coverage sometimes high) but with no gene model predicted. This can happen for many reasons. For instance, the RNA could be derived from transposons or pseudogenes. The assembly itself may contain errors or the RNA alignments could be wrong or misleading specially over highly-similar loci such as repeats. It is not advisable to insert entirely new gene models if uncertain and if there is no other clear evidence (in addition to RNA) to support it (for instance, strong conservation).

Spliced antisense transcripts
There are also examples of spliced, antisense transcripts where the read mapping suggests that an intron is spliced, but the gene model does not show it. In some cases, inspection of the splice junction will show you that in the orientation of the gene model, the splice junctions read CT……AG, totally against all consensus rules. In reality, such a situation reflects an intron in an antisense transcript with splice junctions GT…AG, which is perfectly normal. These antisense transcripts may be non-coding.

Surprising variation
More data can occasionally uncover surprising variation. In the example below, showing multiple gene sets and RNA-Seq data for Z. tritici, all the Illumina tracks show a very small intron that is absent in the gene models and RNA from other laboratories. In theory, the strains should be the same but are maintained in different labs and possibly different conditions. One could imagine that mutations could accumulate and lead to such a difference.
Figure 13 - More data shows interesting variation (*Zymoseptoria tritici*): an intro absent in all gene models and RNA from other laboratories

**Comments from community annotation project members**

From Dr Jan van Kan (Wageningen University, Netherlands) who lead the gene editing effort for *Botrytis cinerea*:

“After annotating several thousands of genes, I can tell you it is fun and not (always) difficult. In fact, in most cases the situation is obvious and simple. For most of the genes, you will easily be able to verify whether the exon structure is OK, and you will be able to adjust the UTRs to a reasonable position.”
From Dr Alice Feurtey (Max Planck Institute) currently involved in the *Zymoseptoria tritici* curation project highlighting the need for common guidelines to standardise the annotation process:

“We need common, detailed guidelines that annotators can follow. In our review of the test manual annotations, we have found that different people make different decisions in similar scenarios.”

**Acknowledgements**

We would like to thank the following people for their input in producing this document:

- Dr Jan van Kan at the Wageningen University in the Netherlands for spearheading the *Botrytis cinerea* community annotation project and for producing a very useful manual on Apollo gene editing, many examples and insights from which have been used in this tutorial
- Dr Marc-Henri LeBrun (INRA, France) and Dr Alice Feurtey (Max Planck Institute) for sending many examples from the ongoing *Zymoseptoria tritici* curation project and providing feedback on this tutorial
- Helder Pedro from the Ensembl team for setting and facilitating the community curation projects
- Dr Jane Loveland (Ensembl-Havana annotation project leader at EMBL-EBI) and Dr Alan Kuo (JGI) for valuable feedback and comments
Appendix A: How to use the JGI MycoCosm platform

The Transcript Annotation Page

If you are a registered user, you can annotate a genome with information about the gene you are viewing. This is accomplished via the Transcript Annotation tool, which displays annotation information for the gene, and allows a user to modify several fields, including a model’s Disposition by promotion (or demotion) to (or from) GeneCatalog.

**Name** (GenBank “gene”) provides a unique, organism-specific identifier which should be consistent with community standards.

**Description** (GenBank “note”) provides a place to record information. Can be as detailed as needed, provided that the information is accurate and useful to researchers not familiar with the type of protein.

**Define** (GenBank “product”) provides a precise description of the gene and gene product, and if possible, it should include the gene's main function(s). Very often, the define of a related entry in Swissprot can be used.

**Disposition** provides two options regarding a models inclusion in GeneCatalog:
- “Catalog” for addition
- “Demote” for removal
There are multiple ways of accessing the transcript annotation page for a given gene model:

1. Via the View/Modify manual annotation link on the gene model’s Protein page:

2. Via Advanced Searching directly against annotations
   a. Gene models which match the specified search criteria are returned as a table, sorted by relevance score. The Gene column provides the following links:
      - **Protein id**: Link to the Protein page
      - **Transcript id**: Link to the Transcript Annotation page
      - **Location**: Link to the genome browser, zoomed on the gene model
3. Via the GO/KEGG/KOG functional tools
   a. These utilities provide dynamic lists of gene models which match functional
      search criteria specific to the particular functional category
   b. (GO) For gene models belonging to a particular GO category, the Links
      column contains the following:
      • P: Link to the Protein page
      • A: Link to the Transcript Annotation page
<table>
<thead>
<tr>
<th>GO Term</th>
<th>Gene Models in Chipad1</th>
<th>Total Gene Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>7234</td>
<td>7234</td>
</tr>
<tr>
<td>GO:0008150 biological_process</td>
<td>4785</td>
<td>4785</td>
</tr>
<tr>
<td>GO:0032502 developmental process</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

**Download:**

- Proteins
- Transcripts
- Check All
- Uncheck all

<table>
<thead>
<tr>
<th>Name</th>
<th>ProteinID</th>
<th>Links</th>
<th>JGI DB/Batch</th>
<th>Quality</th>
<th>All Xref</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>fgenes1_lg27_#_029 #_TRINITY_DN7619_c0_g1_i1</em></td>
<td>436711</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR0001164</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>enarios1_lg11_#_587 #_TRINITY_DN8816_c0_g3_i1</em></td>
<td>421445</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR0005713</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>fgenes1_lg24_#_248 #_TRINITY_DN11231_c0_g1_i1</em></td>
<td>421633</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR0001164</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>gpr1.1.8.763.1</em></td>
<td>349621</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR0001164</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>fgenes1_lg13_#_1156 #_TRINITY_DN5620_c0_g1_i1</em></td>
<td>427291</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR00011370</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>fgenes1_lg2_#_591</em></td>
<td>448345</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR00011370</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GO:0032502 developmental process</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>fgenes1_lg3_#_106</em></td>
<td>449656</td>
<td>PA Chipad1:1581 IEA</td>
<td>IPR0003663</td>
<td>IPR0006828</td>
<td>3.6.4.3</td>
</tr>
</tbody>
</table>
c. (KEGG/KOG) For gene models belonging to a particular KEGG metabolic pathway (EC designation) or KOG functional group (KOG id), the Curated? column contains a YES/NO link to the Transcript Annotation page.
Model Promotion

To search for and evaluate alternative models at a given locus, expand all model tracks (red) and EST tracks (green). In many cases, a better model has already been generated by one of the gene predictors but was not promoted to GeneCatalog. For example, below is a view of select tracks displaying a long model covering three short fragment models, with EST and RNA coverage:

If an alternative model exists and is determined to be more accurate than the current model, it should be promoted to GeneCatalog. Use the Disposition field on the Transcript Annotation page to promote a model to GeneCatalog by setting the value to “Catalog”.

Page 21 – Nishadi De Silva
Model Creation
If none of the alternative models are of acceptable quality, it will be necessary to create a model using the Track Editor tool: http://genome.jgi.doe.gov/help/track_editor.html

Using the Track Editor, it is possible to:
- Create a new model by copying an existing model
- Edit a new model
- Add existing exons to a new model
- Create an ab initio model

Once editing is finished, the model should be released in order to initiate protein analysis. However, since releasing a model does not automatically add it to the GeneCatalog, the model’s Disposition must also be set to “Catalog” via the model’s Transcript Annotation page (similar to Model Promotion).

Model Demotion
Regardless of whether an existing model was promoted or a new model was created, the old/incorrect model should be demoted; otherwise it will appear concurrently with the new/correct model. Similar to Model Promotion, use the Disposition field on the Transcript Annotation page to demote a model from GeneCatalog by setting the value to “Demote”. This option does not delete the model or its annotation from the database. It simply removes it from the Catalog track.
Appendix B: How to use the Apollo gene editing tool

Figure B:1 - Anatomy of the Apollo gene editing interface

Apollo (http://genomearchitect.github.io/) allows annotators to modify and refine the precise location and structure of the genome elements that predictive algorithms cannot yet resolve automatically. Using Apollo, annotators may corroborate or modify the structures of coding genes, pseudogenes, repeat regions, transposable elements, and non-coding RNAs (i.e: snRNA, snoRNA, rRNA, tRNA, and miRNA).

In order to make any changes to gene models, you must first move them to the “user annotation track” highlighted in yellow above. You can click on a gene model in the “gene model” track and simply drag the feature up into the “user annotation track” while holding the mouse button. The gene model will appear in the track with different boxes for coding and non-coding sequences. Clicking the right mouse button while the cursor is on a feature opens a z-menu. Clicking the left mouse button while the cursor is on a feature enables you to move the boundaries of this feature (moving the splice junction or the UTR boundary).

You may reveal or hide any of the data tracks listed in tabular form by ticking/unticking the track selection checklist.

The blue bar at the top holds top-level menus with the following functions:

- ‘File’:
  - Allows users to add data files (e.g. GFF3, BAM, BigWig, etc.) by opening sequence and track files, as well as loading tracks via URLs. Apollo automatically suggests tracks to display their contents.
○ It is possible to combine the information from quantitative tracks into a ‘Combination Track’. Data from tracks containing graphs may be compared and combined in an additive, subtractive, or divisive arithmetic operation. The resulting track highlights the differences between the data.
○ The third option allows users to ‘Add sequence search track’. This tool creates tracks showing regions of the reference sequence (or its translations) that match a given string of nucleotides or amino acids residues.

● ‘View’:
  ○ Allows users to colour all exons in display according to CDS frame.
  ○ Users may choose between light and dark options for their working environment by changing the ‘Colour Scheme.’
  ○ Toggle the view of the plus and minus strands, and reveal or hide the labels for each track.
  ○ It is also possible to highlight a region using the ‘Set highlight’ option and marking the region. The highlight option will automatically be turned ‘On’ when inspecting the results from a BLAT search.
  ○ Annotators will also use this menu when resizing the scale of quantitative tracks.

● ‘Tools’ leads users to perform BLAT searches
● The ‘Help’ tab includes links to a list of helpful commands for Apollo, details about the version of Apollo in use and about JBrowse, as well as a link to explore Apollo Web Services options.

● On the upper right corner, a box with the username offers the option to logout. When logged out, the word ‘Login’ will be displayed instead of the username.

The Apollo user guide can be found here: http://genomearchitect.github.io/users-guide/