GEORGETOWN UNIVERSITY

BEE GENE MODEL ANNOTATION USING APOLLO

Prepared by Monica C. Munoz-Torres, Justin T. Reese, Jaideep P. Sundaram and Chris Elsik.
Elsik Laboratory.
Department of Biology, Georgetown University, Washington, DC.

Last Update: May 12, 2012.

Goal:

The goal of this community effort on manual annotations using Apollo is to improve the Official Gene Sets for *Apis mellifera, A. florea*, *A. cerana*, *Bombus terrestris* and *B. impatiens* by viewing and manually editing gene models, and adding gene models that are not present in the official gene sets. Hereafter, we will refer to the species of interest as 'bee'.

The official gene sets may include gene predictions from any combination of the following sources: NCBI RefSeq, Fgenesh, AUGUSTUS, Twinscan/N-SCAN, Maker and GLEAN. Development of an improved official gene set will involve replacing some of the official gene models with your manually annotated gene models, and adding new gene models that you might create.

A few important things to keep in mind:

- When editing an existing gene model, if it applies, please be sure to identify the corresponding RefSeq model that you are planning to replace with your manual annotation (RefSeq ID: XP, NP for protein sequences; XM, NM for mRNAs). RefSeq entries constitute a high-quality gene prediction set and because of this, in many cases a RefSeq gene model can be the starting point for your manual annotation efforts.
- In a number of cases you may choose not to edit a gene model; even so, it is important that you check the corresponding RefSeq gene model (if available). In these cases you may also write back ("save your work") to the server, even if no changes were made.
- There may be more than one transcript per gene model, so please check them all. You may also wish to annotate additional transcripts rather than replacing a RefSeq or an official transcript. In some cases, annotation may involve adding UTRs without modifying the CDS.

Using our current version of Apollo does not require knowledge of programming language; Apollo will be automatically downloaded and installed on your desktop. Users do not have to worry themselves with configuration files. The Hymenoptera Genome Database (HGD) has set up Apollo for the bee genomes so that annotators may easily access, edit and save their work on their regions of interest.

General Process of Annotating Bee Genes Using Apollo

I. Find a protein homolog of your gene of interest in GenBank, Ensembl or Uniprot. Alternatively, you may start with your own bee sequence; i.e: genomic, amino acid, EST/cDNA.

II. Determine the bee predicted gene model that is homologous to your gene of interest, or determine the scaffold to which the sequence of your protein homolog or bee EST/cDNA of interest was aligned using BLAST at BeeBase or NCBI.
III. Register for an *Apollo Account* at BeeBase.
IV. Access Apollo's automatic download page on BeeBase, and connect to BeeBase to start Apollo entering the predicted gene identifier or scaffold number and coordinates. Use Apollo to create an annotation.
V. Enter additional information to your annotation.
VI. Save the annotation by writing back to the BeeBase server. Alternatively, save your annotation onto your hard drive as a Chado-XML file.
VII. Additional Information.
VIII. Troubleshooting.

I. Find a homolog to start the annotation process.

Why? In the absence of a partial or full-length bee sequence of your gene of interest, you will need to look for a homolog of your gene of interest in other species, to most efficiently retrieve this sequence in the bee genome you are inspecting.

It is best to use protein sequences to query the databases when looking for a homolog in other species. Protein sequences diverge more slowly than DNA sequences, allowing for more sensitive blast searches. The human and drosophilids genomes are the best species to start with because they have the best annotated protein sets to date, thanks to large full-length cDNA projects and exhaustive curation. Additionally, draft genome sequences are available for six ant species and one parasitoid wasp. In most cases, manual annotation efforts were also conducted. Swissprot (part of UniProt) is one of the best starting places to search protein sequences, thanks to their stringent curation parameters. Other databases include Ensembl and NCBI. In the next three examples, we will retrieve the insect protein Antennapedia (Antp).

I. A. Find a homolog using UniProt

Go to the UniProt website (http://www.uniprot.org/) and enter *Antp* in the search box. Currently 235 reviewed UniProtKB/Swiss-Prot (marked with a gold star) records and 1,280 un-reviewed ones are retrieved. Scroll down to find the fruitfly homolog *ANTP_DROME*, and click on its ID (Accession = P02833). The ID format of this entry (gene name, underscore, organism name) indicates that this is a curated Swissprot sequence (starred). In the absence of a Swissprot sequence you may choose a Trembl homolog (with the ID format: accession number, underscore, organism name) such as *Q9GYV0_APIME*. These are, usually, predicted genes that have not yet been manually curated, so are less reliable than Swissprot entries.

Scroll down to find the 'Sequences' section and click on the word 'FASTA' to copy/download it and paste it onto your BLAST page of choice.

I. B. Find a homolog using Ensembl

Go to the Ensembl website (http://www.ensembl.org/) and type *Antp* in the search box and click *Go*. Results are organized by *Feature type* or by *Species*. There are 1 *Gene* and 11 *Transcript* entries for *Fruitfly*. Click on the *Gene* feature for *Fruitfly*, and choose the first record labeled with the FlyBase GeneId (*Antp*), (you can alternatively choose the first available gene labeled with the database cross-reference identifier from UniProtKB/Swissprot -- as your homolog). In the absence of a Swissprot sequence, you may choose one of the other predicted transcript models.

There may be more than one transcript per gene. In this example, 11 transcripts are listed; choose the top one (*Antp-RN*, in this case) and click on it, and then go to the left column to click on the *Export Data* tab. When prompted, choose the *Peptide* sequence from the list and click on the word *Next* to receive the peptide sequence in FASTA format. Remember to click on the transcript of choice before exporting the data, so that you may easily retrieve a sequence of amino acid residues. Copy/Download it and paste it onto your BLAST page of choice.

Our *Antp* peptide sequence downloaded in FASTA format looks like this:

```
>FBtr0081651 peptide: FBpp0089086 pep:KNOWN_protein_coding
MTMSTNNCESMTSYFTNSYMGADMHHGHYPGNGVTDLDAQQMHHYSQNANHQGNMPYPRF
PPYDRMPYYNGQGMDQQQQHQVYSRPDSPSSQVGGVMPQAQTNGQLGVPQQQQQQQQQPS
QNQQQQQAQQAPQQLQQQLPQVTQQVTHPQQQQQQPVVYASCKLQAAVGGLGMVPEGGSP
PLVDQMSGHHMNAQMTLPHHMGHPQAQLGYTDVGVPDVTEVHQNHHNMGMYQQQSGVPPV
GAPPQGMMHQGQGPPQMHQGHPGQHTPPSQNPNSQSSGMPSPLYPWMRSQFERKRGRQTY
TRYQTLELEKEFHFNRYLTRRRRIEIAHALCLTERQIKIWFQNRRMKWKKENKTKGEPGS
GGEGDEITPPNSPQ
```

You may choose to change the name as you see more appropriate for your records.

I. C. Find a homolog using NCBI

On the *Search* section at the NCBI website (http://www.ncbi.nlm.nih.gov/) select the *All Databases* database from the drop-down menu. Enter the query "`Drosophila melanogaster[organism] AND Antp`" in the search box (without quotations) to limit this search only to Fruitfly records; currently, 51 records are available on the *Protein* database, and 29 on the *Gene* database. If available, records from the *Gene* database offer the 'safest' path to the most curated RefSeq entry for any given model. In this example, the first record in the *Gene* database results is the *Antp* gene. Click on this record to display the *Gene* page, and scroll down to find the *NCBI Reference Sequences* (RefSeq) section. Choose the protein record to display this entry in the *Protein* Database. Although not necessary in this example, please remember to **always read the description carefully** to ensure that you are **not** choosing a paralog of your desired protein. Also, check that the correct gene symbol is listed next to the word */gene=* in the *CDS* feature area. Click on the word 'FASTA' located under the identifier, to display, copy/download it and paste it onto your BLAST page of choice.

II. Locating your gene in a bee genome assembly

II. A. Using BeeBase BLAST to determine the existence of an official gene model homologous to your gene of interest.

You may wish to both identify a predicted gene, and identify the scaffold region containing the bee genes of interest, to check if there are paralogs not included in the predicted gene sets.

On the BeeBase BLAST site http://hymenopteragenome.org/beebase/?q=blast choose the BLAST method and database to query accordingly from the drop down menus, paste your protein sequence into the search box, and click the *Search* button. For example, you could use your protein sequence to search the nucleotide database with all scaffolds in the *Apis mellifera* Assembly version 4.5 → *Amel_4.5_Scaffolds* using *tblastn*.

We have divided all BLAST databases per species, thus separate choices must be made in order to BLAST each genome on either the *A. mellifera* site, or the *B. terrestris* site, etc. The standard BLAST output includes a summary of each hit in the database or *Subject* sequence followed by alignments for each hit. The output details the gene model identifier, the alignment *Score* and *Expect* (E) value. Be sure to look both at scores and alignments to make the most informed decision about your termite homolog.

II. B. Using BeeBase BLAST to identify location on the bee assembly

On the BeeBase BLAST site http://hymenopteragenome.org/beebase/?q=blast choose the BLAST method and database to query accordingly from the drop down menus, paste your sequence into the search box, and click the *Search* button. As mentioned above, you can search the assembled scaffolds from *Apis mellifera* v4.5 → Amel_4.5_Scaffolds using a protein sequence and the *tblastn* option.

A link to the right of each alignment on the BLAST output indicates that you may *See complete hit in GBrowse*. If the link is not active, visit the GBrowse pages for your organism of interest and enter the gene model identifier as displayed on the BLAST results page. The most up to date links for GBrowsers on BeeBase can be found at: http://hymenopteragenome.org/beebase/?q=genome_resources.

II. C. Using NCBI to identify the location of your candidate homolog on a bee assembly.

On the NCBI BLAST site, http://www.ncbi.nlm.nih.gov/BLAST/ click on *List all genomic BLAST databases*. Select your species by clicking on *Invertebrates*, *Insects* and then the name of your species, if available. The NCBI BLAST page for the honey bee genome sequence is located at http://www.ncbi.nlm.nih.gov/genome/seq/BlastGen/BlastGen.cgi?taxid=7460.

Paste your protein sequence in FASTA format on the designated space and select the *genome (reference only)* database from the drop-down menu; select *tblastn* as the program to use and click the *Begin Search* button. A new window will open up. **Click** on *View report* to browse through the results.

III. Register for a 'Bee Apollo Account' at BeeBase.

Obtain an *Apis mellifera* Apollo Account on BeeBase by going to:
http://hymenopteragenome.org/beebase/?q=registration_apollo_amel

Obtain a *Bombus terrestris* Apollo Account on BeeBase by going to:
http://hymenopteragenome.org/beebase/?q=registration_apollo_bter

When your account has been activated, you are ready to start using our Apollo Annotation Tool.

Keep in mind!

- You must register for each species separately.
- If you are an Apollo registered user in more than one bee species, you need to check that you are using the correct file every time you open Apollo!

IV. Access the Apollo automatic download page on BeeBase and run the Apollo Annotation Tool for Bee genomes.

In the following lines we describe how to use Apollo to annotate a gene model from the Honey Bee *Apis mellifera* assembly version 4.5 and the *Bombus terrestris* assembly version 1.0.

Please Visit the Bee Apollo Annotations Page on BeeBase for instructions on downloading the Apollo Annotation tool for your genome of interest.
http://hymenopteragenome.org/beebase/?q=bee_apollo_annotations

When prompted, grant Apollo permission to access your computer by giving a click on the *Allow* button. Our server is located at genomes.arc.georgetown.edu. After this, you will be able to access the Bee Apollo databases with the links on your desktop.

First, choose the *Chado database* as your data source.

Second, choose the corresponding *Chado database* from the drop down menu:
Apis mellifera assembly 4.5 if you are working on the honey bee genome, or *Bterrestris* if working on the European bumble bee genome. You will ONLY be able to access the species database you are calling with the link on your desktop, so please double-check every time.

Third, use the *Login* and *Password* you received when you registered for an *Apollo Account* on BeeBase.

You may enter the scaffold ID, indicating *start* and *end* coordinates. Please keep in mind that there are thousands of scaffolds to choose from and scaffolds are organized non-alphanumerically but by the first digit after the last character and successively upwards.

Lastly, on the *Select a region to display* box, choose:
Type of region: scaffold
Choose Chromosome: choose the scaffold number from the pull-down menu
Enter *start* and *end* coordinates on the right.
Click *Ok*.

V. Annotating a gene model using Apollo

While annotating your gene please keep the following ideas in mind: what you are looking at is a 'frozen photograph' of what the genome assembly looks like. You will not be able to change that photograph; instead you will build a layer of information that may be placed on top, reflecting the appropriate changes, much like photo-editing software would do to a picture. Thus, if you identify insertions or deletions you will NOT be able to modify the assembly to reflect the necessary change; instead, your observations will be contributed to our efforts to improve the official gene set and the changes will be included in the updated OGS, where your manual annotation will replace the official gene set annotation. We will then make every effort to incorporate these changes onto newer versions of the assembly.

Because your annotation will replace the official gene set annotation and the NCBI/RefSeq corresponding entry, you **MUST** capture the RefSeq ID number as a synonym. If you have not yet determined the corresponding RefSeq prediction(s), you may access this information from within Apollo.

V. A The Apollo View: initial reconnaissance and adjustments.

Suggestions: To increase the vertical space available to view data, move the evidence information panel to the vertical option by choosing the *Make evidence panel vertical* option from the *View* menu; re-size the window and re-position splitters to increase the area of the evidence tiers (the black areas). The blue area in the middle, marked by red arrowheads, is your 'working area'. You will 'drag' the gene model/exons/regions you want to modify into this area, and all modifications to a gene model will be done while working on this 'temporary' models.

The genomic region containing your gene should be approximately located in the center of the sequence extent viewed in Apollo and your gene may appear on the top ('positive') or bottom ('negative') strand. The RefSeq predictions are labeled with their genomic extents, rather than with their ID's. You can retrieve a RefSeq ID by clicking on the RefSeq prediction to select it (clicking on an intron or double-clicking on an exon selects the entire prediction), then holding down the right mouse button. From the right-click or apple-click menu, select *Get info about this feature via Web*. Confirm the identity of the RefSeq prediction that corresponds to your gene.

If you are not able to identify a corresponding prediction when you open Apollo, find the correct region using the *Find* option in the *Edit* menu on the upper left corner of the menu bar. This *Find* feature accepts genomic sequences and will search both strands. Use a short sequence (40 bp are enough) and be sure this extent does not cross an intron. This sequence search option finds only exact matches without gaps.

Check the data aligned to the opposite strand in the region of interest. Do any of them (especially ESTs!) look like they belong on the working strand? If so, select the offending data (rubber-band holding down the middle button or alt/option button (Mac) while you select, or double click on the model to select) and use the right-button / apple-click menu option *Move to other strand*. You may wish to adjust the view by hiding the non-working strand and/or choosing *reverse complement*; both of these are options in the *View* menu.

Scroll through the aligned data, e.g: protein alignments, predictions, cDNAs: do any of them extend beyond your selected annotation? Do you see any significantly longer cDNAs? If so, these are likely to be coding sequences as well. If this is not the case, you may follow directions below for **Simple Cases** (section V. B). If the answer is yes, then follow directions for **More Complex Cases** (section V. D).

Keep in Mind!

Protein alignments do not necessarily reflect the entire length of the similar protein; non-conserved regions simply do not show up. This may result in a short protein alignment or one with missing internal exons. Protein alignments may also do funny things in regions with tandem closely related genes; for example, aligning in part to one gene and then skipping over to align the rest to a second gene.

V. B. Simple Cases:

One of the predictions is correct (e.g: you chose RefSeq), or nearly correct (assuming the answer above was "no", i.e: no aligned data extends beyond the predicted model and if so, it is not likely to be coding sequence, and/or the selected prediction matches what you know about the gene):

a. Can you add UTRs?

If available, the RefSeq annotation may include UTR's. If this is the case, you will probably wish to start your annotation by dragging and dropping the NCBI prediction down into the light blue annotation zone.

If EST data are available and they extend beyond your original annotation, extend or add UTRs by using the *Set as 5' end* or *Set as 3' end* functions found in the right-button / apple-click menu. To do this, press the shift key, then select the appropriate annotation exon and the appropriate EST exon; release the shift key and use the right/apple-click to bring up the menu.

b. Check exon structures:

Zoom in sufficiently so that you may clearly resolve each exon as a distinct rectangle. Use the edge-matching function by either selecting the whole annotation or selecting one exon at a time. Scrolling along the length of the annotation, check the edges of the annotation match with all available EST data. Also note if there are ESTs that lack one or more the annotated exons or include additional exons.

If you detect differences, note whether all the data disagree with the annotation, in which case you need to make corrections; or whether some of the data support the annotation and some of the data support one or more alternative transcripts. Annotate as many alternatives transcripts as are well supported by the data. Erring on the side of caution, anything unusual should be supported by at least two pieces of data.

Again, to change an exon boundary that needs to be corrected to match data in the evidence tier: press the shift key, then select the appropriate annotation exon and the appropriate EST exon; release the shift key and use the right mouse button to bring up the menu and select *Set as 5' end* or *Set as 3' end* as necessary.

Spliced ESTs are more reliable than non-spliced ESTs. Atypical EST data often appear to be spliced, but do not align to canonical splice sites; these should be considered with suspicion.

c. Check splice sites:

If there are any non-canonical splices (other than GT/AG), these will be indicated by a yellow arrow-like image (GIF) superimposed on the end of the exon. However, Apollo does not add these indicators if the exon representation is too small, so it is convenient to check for these places when checking for correspondence within edges, scrolling through a zoomed-in view. If you have added alternative transcripts, you may have to make a second pass.

If a non-canonical splice is present, open up the *Exon detail editor* (select the annotation, open the right/apple-click menu and click on *Exon detail editor*) to view the splice at the sequence level. Very frequently a GC splice donor is causing the 'problem'; Apollo does not understand that a GC donor is valid, so it will flag them. These do not need to be corrected, but should be flagged with the appropriate comment. (*Adding Comments* is addressed in Section VI.)

A GC splice donor may have caused the problem in a different way: the prediction may have ignored the GC donor and picked a non-canonical splice instead. If there is an in-frame GC close by, that is likely to be the correct splice; make this adjustment in the *Exon detail editor*.

If there does not appear to be any way to resolve the non-canonical splice, leave it and add a comment.

d. Check the 'start' and 'stop' sites:

Generally speaking the start codon will be in the first or second exon (5' exons that are entirely UTR are common) and the stop codon will be located within the last exon. If this is not the case, try to determine if there is a problem and correct it.

The start codon can be set manually using the *Exon detail editor*: using right/apple-click, click on the first nucleotide of the start codon. Click on the option *Set start of translation* from the menu. Apollo determines the stop codon automatically. In special cases (e.g: selenocysteine read-through) it can be overridden using a toggle in the *Annotation info editor*.

Apollo will flag an exon that appears to have invalid starts or stops with an arrow-like GIF (as for non-canonical splices). If a terminal exon is marked with such a flag, it means there is no stop within that exon and the protein sequence runs to the end of the exon. Check to see if there are data supporting additional 3' exons; if not, go to the *Exon detail editor* window and pull out the exon to the first in-frame stop.

If the 5' exon is flagged, it may mean one of three things:

(1) There is no ATG start in the correct frame: In this case, there may be an in-frame start immediately upstream and you can extend the exon to incorporate it. Please note that the start codon may also be in an exon further up in the 5' direction. If you cannot identify that exon, add the appropriate comment (using the transcript comment section in the *Annotation info editor*).

(2) There is an ATG start in-frame, but it is not the first one in frame for this model (there is no in-frame stop preceding the ATG): In other words, Apollo is flagging the fact that this entire exon may be a "coding exon" and the ATG may not be the correct start codon. In this case check if the true stop appears to be upstream. If not, you can override Apollo by setting the translation start manually (see above).

(3) A non-canonical start codon has been predicted: in this case the codon will be also highlighted in purple instead of green). In this case, check whether a non-canonical (i.e: non-ATG) start codon is usually present in homologs of this model, and/or check whether this is a likely occurrence in this genome. If you cannot reach a satisfactory answer to this question, you may override Apollo by setting the translation start manually to a canonical ATG start codon (see above), choosing the one that both, most closely reflects what you know about the model, and has the best support from the biological evidence (e.g: EST alignments, etc).

e. **Check the predicted protein product(s):**

If any of your manipulations have thrown an exon out of frame or caused any other drastic changes that will immediately show up in Apollo, then the blue protein-coding extent will appear truncated (empty rectangle). This requires correction!

If the annotation looks good, pull up the protein sequence by selecting the annotation and using the right/apple-click menu option *Sequence*. Copy this into a protein BLAST (e.g: NCBI BLAST) and BLAST against the all organisms nr database. If the alignment looks good, note also whether the length of your protein is an approximate match to the best hits. **Please remember**: the first hit may be an NCBI bee prediction – this one does not count! Hits labeled as "Predictions" are less reliable; try to find a hit that is not marked as a prediction or as hypothetical model, but is a verified RefSeq model.

While searching on NCBI you may come across a hit to a bee cDNA contributed by an individual lab. If this is the case, you should treat this as additional evidence data and should compare it to the entire annotation, including UTR's.

If the protein product still does not look correct, please consult the **More Complex Cases** section below. If the protein looks good… **you may Proceed to Part VI. (Hooray!)**

V. C. Additional functionality you may need:

a. Search for a specific sequence: Apollo provides two different options to search for a specific sequence; both require an exact match. (1) In the (upper left) *Edit* menu select *Find*; this function performs only nucleotide queries and can search both strands. (2) In the *Exon detail editor*, in the lower left there is a *Find sequence* button, which allows nucleotide or amino acid searches, but searches only the strand from which the *Exon detail editor* was launched. In both cases, clicking on the hit (genomic range) will take you to that position within the main Apollo window and Exon *detail editor* display.

b. Get genomic sequence: by rubber-banding (middle-button, alt/option button in Mac) any region along the sequence coordinates scale you may obtain the sequence (for one strand) of that region; it will appear in a box similar to the *Sequence* box specific to annotations.

c. Merge exons: select each exon while holding down the shift key, open the right/apple-click menu and select *Merge exons*.

d. Merge transcripts: select the terminal exon of each transcript (hold down the shift key), open the right/apple-click menu, and select *Merge transcripts*.

e. Add an exon: Normally Apollo allows you to drag an exon from the evidence tier and add it directly to an annotated transcript (by dropping it in an overlapping position; the shift key allows you to move laterally in order to overlap). However, if this function is not working, you can do this as a merge: drag down the exon into the annotation tier and create a new annotation. Then, using the shift key, select the new single-exon annotation, select the recipient transcript, and from the right-button menu select *Merge transcripts*. Check that adding the exon did not throw the downstream portion of the annotation out of frame.

f. Delete an exon: Select the exon using a single click (a double click will select the whole transcript). Bring up the right/apple-click menu and click *Delete selection*. Check that deleting the exon did not throw the downstream portion of the annotation out of frame.

g. Create an exon de novo (within an intron): In the Exon detail editor, move to the appropriate region and right/apple-click on a nucleotide. Apollo will create a 1-base exon; pull the ends out to the appropriate extent.

h. Create an exon de novo (outside existing annotations): Use the *Find* tool from the *Edit* menu. Insert a sequence string and then click on the coordinates of the hit; Apollo should move you to that region, with your hit highlighted. Using the right/apple-click you should be given the option *Create new annotation*. Open the *Exon detail editor* to adjust the extent of the new exon/annotation; if appropriate, use the *Merge transcripts* option to add the new exon to an existing annotation.

V. D. Complex Cases:

a. Incomplete annotation: protein check indicates gaps, missing 5' sequences or missing 3' sequences.

The best way to find a bit of missing protein sequence may be to perform a TBLASTN search, using the homologous bit of protein sequence from a close BLAST hit for the protein component of the query. Using an *Align two sequences* option, the genomic nucleotide portion of the query can be a small section of genomic sequence you have selected (use the middle button or alt/option key to rubber-band that portion of the sequence coordinate axis in Apollo). If this is not successful, the query can be searched against the entire bee genome at NCBI or BeeBase. **Note**: with a "small" protein query (~10-50 aa), TBLASTN may not yield results unless the expect value is bumped up to 100 or 1000.

If you are successful in identifying the missing regions, you can use the termite sequence hit in the *Find sequence* box from the *Exon detail editor* to find the sequence in the Apollo view. Always remember that this tool does not tolerate mismatches or gaps.

If you add or extend exons, the resulting transcript should be reassessed using the steps in the **Simple Cases** section above. If the new exons are part of a second gene prediction, this becomes a merge (see below).

If you cannot find the missing bits, check to see if there is a sequencing gap that may be responsible. When zoomed into to a moderate level, a sequencing gap appears as a thicker red bar along the DNA coordinate line. Zoomed in further, it resolves as a zigzagged line; zoomed into the sequence level it appears as a string of "N"s.

If you cannot complete the annotation, report this in the *Annotation info editor*, using appropriate canned comments as described in Part V.

b. Merge of 2 RefSeq predictions on same scaffold.

If there is an EST or prediction supporting the merge, start your annotation by dragging it into the blue annotation section. A protein alignment is not usually a convenient starting point, because many of the splices will be incorrect, and non-conserved regions will not be represented.

Alternatively, you can start by dragging down the two component RefSeq predictions and merging the resulting annotations (see above). Do not be concerned if there are several annotations between the two components (which end up within an intron after the merge); such nested genes are very common. The resulting transcript should be reassessed using the steps in the **Simple Cases** section above. Be sure to capture both RefSeq IDs in Part V, below. Using the appropriate canned comment, record in the *Annotation info editor* that this annotation is the result of a merge.

c. Merge of 2 RefSeq predictions on different scaffolds.

A merge of annotations across scaffolds cannot be done at this time. Document the fact that the data support a merge in annotation comments for both components. Use the following two canned comments:

"RESULT OF: merging two or more gene models across scaffolds"
"RESULT OF: merging two or more gene models. Gene models involved in merge:"

d. Split of a RefSeq prediction.

A split is indicated if the predicted protein aligns to two entirely different protein groups. There may also be EST data that support a split (in this case, be sure it is not a case of alternative transcripts). A split can be created in one of two ways: you may either 1) select the flanking exons using the right/apple-click menu option *Split transcript*, or 2) annotating each resulting fragment independently. Each resulting component should be assessed using the steps in the **Simple Cases** section above. Be sure to capture the original RefSeq ID for both annotations in Part V, below.

e. Frameshifts, single-base errors, selenocysteine and other inconvenient phenomena

Apollo allows you to make single base modifications or frameshifts that are reflected in the sequence and structure of any transcripts overlapping the modification. Note that these manipulations do NOT change the underlying genomic sequence. Changes are made in the *Exon detail editor* by clicking with the right/apple button on a single nucleotide; this will bring up a menu that gives you several options for introducing sequence changes. The resulting transcript should be reassessed using the steps in the **Simple Cases** section above.

The corrected transcript and protein sequence will appear when the right/apple-click menu *Sequence* option is used. Since the underlying genomic sequence has not changed, the sequences of the annotated products cannot be independently derived and/or verified using automated scripts, so use comments in the *Annotation info editor* to alert the HGD staff and report the CDS edits.

New Data, New Challenges:
The Official Gene Set (OGS) for the Honey Bee genome was generated using GLEAN; this set still has a lower number of genes compared to other methods, in part due to the frameshifts generated by the inclusion of 454 data. This is why, independently of the number of iterations of GLEAN that we can run, the dataset does not seem to improve significantly.

The *B. terrestris* assembly (Bter_1.0) also includes 454 (transcriptome) data, and the RefSeq set is the OGS for this species. Contrary to GLEAN and RefSeq, the algorithms behind MAKER are forgiving with this kind of data. Users may view the *B. terrestris* gene set produced using MAKER as the second best gene set. Given the challenges of assembling NGS data, transcript-based assemblies include a number of inherent errors, which MAKER does not necessarily account for, thus possibly incorporating a higher number of errors such as frameshifts. There are approximately 500 genes with frameshifts, which are included in the RefSeq_freameshifts and Augustus_frameshifts tracks, and visible in both Apollo and GBrowse for *B. terrestris*.

Due to these errors, it is important to keep in mind that when searching the assembly using a homolog from a related species to identify all known members of a gene family of interest, not all members of the family will necessarily be included in the results.

Observing in detail your BLAST hits in GBrowse, you will be able to see a "frameshift" version of the RefSeq and Augustus tracks, which will allow you to chose the best model representing the homolog you used to query the *B. terrestris* genome, even if this model does not make part of the OGS. If you come across a pseudogene, then you can probably try to use a frameshifted version of the gene (if available) to allow you to resolve the correctly annotated version of the gene.

One example of how to spot and deal with frameshift data in the *B. terrestris* gene set is presented below.

Imagine that you are a researcher who has been studying a family of eukaryotic proteins containing ankyrin repeat domains, and you have a set from the parasitoid wasp *Nasonia vitripennis*. The amino acid sequence of one of the members of the family is pasted below:

```
>gi|345491341|ref|XP_001604053.2|  PREDICTED:  ankyrin  repeat  domain-
containing protein 13C-like isoform 1 [Nasonia vitripennis]
MNQRQAGRLKSQSTQAIMAETEQYPLHKCIFEGDVKALSSLIRVHDITAKDKQGNTPLHLAVMLGRKECV
QLLLAHGVPVKVKNLAGWSPLAEAISYGDRQTISSLVRKLKQQAREQMEERRPNLVAALRQMGDFYMELK
WDFQSWVPLVSRILPSDICRIHKSGASIRMDTTLVDFNDMRWERGDISFIFNGDQKPSKSLTVLDNKAKL
FQRVRYEETELEIEDEVDILMSSDIMAAQMSTKGITFSRAQTGWIFKEDKREMVGPFNADFYQINGMVLE
SRKRREHLSEEDLQKNKAIMESLTKGSSQGFQNGEPPMRRASLNPPPDSNITWEDYLSAPSGQCPLLGRS
LVYKESSKSFKATVAMSPDFPLTVDMLLNVLEVIAPFKHFNKLREFVLMKLPPGFPVKIDIPILPTVTAK
ITFQEFAFKNDISPDLFQVPLDYFEDPMRFPDL
```

You used this known *N. vitripennis* protein to query the NCBI-RefSeq gene set of the *B. terrestris* genome using BLAST on Beebase, and obtained the following hit at the top. Take a look at the alignment and pay special attention to the region highlighted in gray, an alignment gap:

```
>gi|340708501|ref|XP_003392864.1| PREDICTED: LOW QUALITY PROTEIN:
           ankyrin repeat domain-containing protein 13C-like
           [Bombus terrestris]
          Length = 434

 Score =  719 bits (1857), Expect = 0.0
 Identities = 358/434 (82%), Positives = 387/434 (89%), Gaps = 7/434 (1%)

Query: 20  ETEQYPLHKCIFEGDVKALSSLIRVHDITAKDKQGNTPLHLAVMLGRKECVQLLLAHGVP 79
           E E+YPLHKCIF+GD+KALSSLIR+++I  KD QGNTPLHLAVMLGRKE +QLLLAHG P
Sbjct: 8   ENEKYPLHKCIFQGDIKALSSLIRIYNIAEKDTQGNTPLHLAVMLGRKESIQLLLAHGAP 67

Query: 80  VKVKNLAGWSPLAEAISYGDRQTISSLVRKLKQQAREQMEERRPNLVAALRQMGDFYMEL 139
```

```
           VKVKNLAGWSPLAEAISYGDRQTISSLVRKL QQAREQMEERRPNL+A LRQMGDFYMEL
Sbjct: 68  VKVKNLAGWSPLAEAISYGDRQTISSLVRKLNQQAREQMEERRPNLLATLRQMGDFYMEL 127

Query: 140 KWDFQSWVPLVSRILPSDICRIHKSGASIRMDTTLVDFNDMRWERGDISFIFNGDQKPSK 199
           KWDFQSWVPLVSR+LPSD+C+IHK GASIRMDTTL        ERGDISFIFNGD+KPSK
Sbjct: 128 KWDFQSWVPLVSRVLPSDVCKIHKRGASIRMDTTLX-------ERGDISFIFNGDKKPSK 180

Query: 200 SLTVLDNKAKLFQRVRYXXXXXXXXXXXXXXMSSDIMAAQMSTKGITFSRAQTGWIFKED 259
           SL VLDN AK +Q VRY              MSSDIMAAQMSTKGIT S+AQTGWIF+ED
Sbjct: 181 SLAVLDNLAKRYQTVRYEETELEIQDEVDILMSSDIMAAQMSTKGITLSKAQTGWIFRED 240

Query: 260 KREMVGPFNADFYQINGMVLESRKRREHLSEEDLQKNKAIMESLTKGSSQGFQNGEPPMR 319
           KREMVGPF+A+ YQI+GMVLES+KRREHLSEEDLQKNKAIMESLTKGSSQGF N +P +R
Sbjct: 241 KREMVGPFHAECYQIDGMVLESKKRREHLSEEDLQKNKAIMESLTKGSSQGFANEKPQVR 300

Query: 320 RASLNPPPDSNITWEDYLSAPSGQCPLLGRSLVYKESSKSFKATVAMSPDFPLTVDMLLN 379
           RASLNPPP+SNITWE+Y+ AP GQ PLLGR+LVYKESSKSFKATVAMSPDFPLTVDMLLN
Sbjct: 301 RASLNPPPESNITWEEYIMAPPGQSPLLGRNLVYKESSKSFKATVAMSPDFPLTVDMLLN 360

Query: 380 VLEVIAPFKHFNKLREFVLMKLPPGFPVKIDIPILPTVTAKITFQEFAFKNDISPDLFQV 439
           VLEVIAPFKHF+KLREFVLMKLPPGFPVKIDIPILPTVTAKITFQEFAF+NDI P LF++
Sbjct: 361 VLEVIAPFKHFSKLREFVLMKLPPGFPVKIDIPILPTVTAKITFQEFAFRNDIDPKLFKI 420

Query: 440 PLDYFEDPMRFPDL 453
           P DY EDPMRFPDL
Sbjct: 421 PPDYLEDPMRFPDL 434
```

Searching this model on GBrowse for *B. terrestris* shows that it is located in Group1.1, between coordinates 524000 and 530000. Open this region in Apollo WebStart and drag the RefSeq_frameshift model into the annotation region (light blue). Notice that Apollo has done an unfortunate job at calculating the best ORF for this protein, most likely due to sequencing errors and/or a region of low quality sequence.

Bring out the *Exon Detail Editor* for this model, and search for the string of 7 amino acid residues immediately before the alignment gap: `SIRMDTT`. Then find the string of 7 amino acids right after the gap: `RGDISFI`. Did you notice anything different? It is now evident that the prediction method "jumped" one position on the reading frame, in order to produce the amino acid sequence labeled as "LOW QUALITY PROTEIN" XP_003392864.1

Become aware of these changes and annotate the model accordingly to indicate their existence. Apollo cannot correct the transcript data already saved at HGD's databases, even if you are able to demonstrate that it must be corrected and have done so using the annotation tools; in other words, Apollo cannot save these changes on top of the reference assembly, but users can (and should!) save their work back to the database, and on their desktops AND comment on each transcript's *Annotation Info Editor*. In this case use the following Apollo features to annotate this gene:

1 - *Set start of translation* at the beginning of the protein according to the RefSeq model (on coordinate/base pair 528,000)

2 - Search for the first aligned amino acid after the gap (E, nucleotides GAA, position 527026) and use the *Set -1 translational frameshift* feature to correct Apollo on how to read the amino acid sequence starting on that coordinate.
3 - Open the *Annotation Info Editor* and add a comment to the transcript, use the pull-down the Menu of canned comments and choose the *FRAMESHIFT* comment.

When appropriate, you may alternatively use one of these canned comments:

- "CDS edit: stop codon readthrough due to selenocysteine"
- "CDS edit: sequencing error"

You may also type more specific information about the sequencing error by entering your own comment.

VI. Adding important project information to your annotations

Before saving your annotations, we kindly request that you include additional information about each of the gene models you have annotated. For each gene model, if there is a RefSeq, NCBI *ab initio*, or Fgeneshpp gene model in the region overlapping your new annotation (and on the same strand), determine the ID for the gene model. To do this, right/apple-click on feature to get feature ID and additional information. For each of your annotated transcripts: click the annotation, then right/apple click and select *Annotation info editor*.

If you know the NCBI/Flybase/HGD/etc gene symbol for the homolog of your annotation, or if you find an overlapping RefSeq gene, add these names as synonyms for the gene by clicking *add* on the *Synonym* box on the left side of the *Annotation info editor* panel. This is necessary to replace the old gene model from the OGS with your revised gene model.

To add comments about the gene, click the top box that on the right panel that starts with *Edit Amel:…* or *Edit Bter:…* The box underneath is for adding comments about individual transcripts.

In each of the *Edit* panels (for genes and transcripts), you may add comments. First click *Add* to add a new comment. Then select a comment from the pull down menu, or simply start typing your own. Some very useful comments include:

- Type of annotation (for example whether or not gene model was changed)
- Datasource (for example if the Fgeneshpp predicted gene was the starting point for your annotation)
- The kinds of changes you made to the OGS gene model, such as split or merge
- Functional assignment description
- Whether you would like for a BeeBase curator to check the splice sites
- Whether part of your gene is on a different scaffold

It is helpful if you use canned comments whenever possible, but enter a free text comment if none of the canned comments seem to work. And please, don’t be shy. You are the expert, and the curators at BeeBase would like to pick your brain as much as possible about each of your annotations, so include every bit of information you deem important.

VII. <u>Saving your Annotations</u>

VII. A. Saving and reopening your work, before it is completed.

One of the beauties of the new version of Apollo is that you may now save incomplete work by writing back to the server. On the same token, please make every possible effort to ensure that your annotations are accurate before writing them back to the server. If you have to interrupt your work and disconnect from the server / shut down Apollo, please write back your incomplete work as described in the following section.

Please note: saving unfinished work on your hard drive and resuming later has caused intermittent problems. This problem may be platform-specific, but we are not sure. Please refrain from saving your unfinished work and supporting evidence in GAME-XML format and then reopening to resume working; your data may be lost. The safest thing to do is to complete your annotation and save it using the instructions in VII. B. during a single Apollo session.

VII. B. Saving your completed annotation: writing back to BeeBase.

Saving your work should not be very complicated. It is as easy as saving your work in most computer programs you may be familiar with (with a GUI). Each file that you save and submit may have one or more annotations on the same scaffold.

Click *File* → *Save as*.

A new dialog box will pop up and the default settings will be the last ones used (when you opened Apollo): select *Chado Database* as your data source and *Apis mellifera Assembly 4.5* as your *Chado database*. In this way you will "write back" your annotations to the server. And you are done and done! (Hooray again!).

We recommend that you keep a copy of your completed annotation (without supporting evidence) as a **Chado-XML** file on your desktop:

Click *File* → *Save as*. A new dialog box will pop up.
For *Choose data source*, select *Chado XML format*.
For *Chado XML format* enter the name of the file. It is useful to include the word "chado" in the name of the file. Make sure the file has a .xml extension.
Make sure the box *Save annotations* is **checked**, and the *Save evidence* box is **unchecked**.
Click *Ok*.

Now your annotation file (annotation only, without evidence) is saved on your hard drive. Pay attention to which directory it is in. By default, Apollo will save your files in the data directory within the Apollo folder. It is a good idea to look in your directory to check the filename for the .xml extension again after it has been saved. You should not use Chado-XML format for saving your unfinished data, because the *Save evidence* feature does not save the EST and cDNA evidence in this case (it's an Apollo bug).

VIII. Additional Information

VIII. A. Right clicking or apple-clicking on a feature to get feature ID and additional information

For all features except repeats, you may right/apple-click in the graphical viewer and select *Get info about this feature via web* to view additional information about the feature. Some features are linked directly to their source database page at NCBI, or UniProt. Other features are linked to an internal BeeBase page that provides additional information.

VIII. B. Looking up homolog descriptions

You may click on a protein homolog feature to go to the accession web page at UniProt/Swissprot.

VIII. C. If Apollo keeps crashing

Apollo is an open-source project and there are still a few bugs being addressed. If Apollo consistently crashes, please check the Troubleshooting section, and if you still find no answers, please contact us at hymenopterabase at gmail dot com, and we will do our best to help you fix it.

IX. Troubleshooting

IX. A. Installing Java

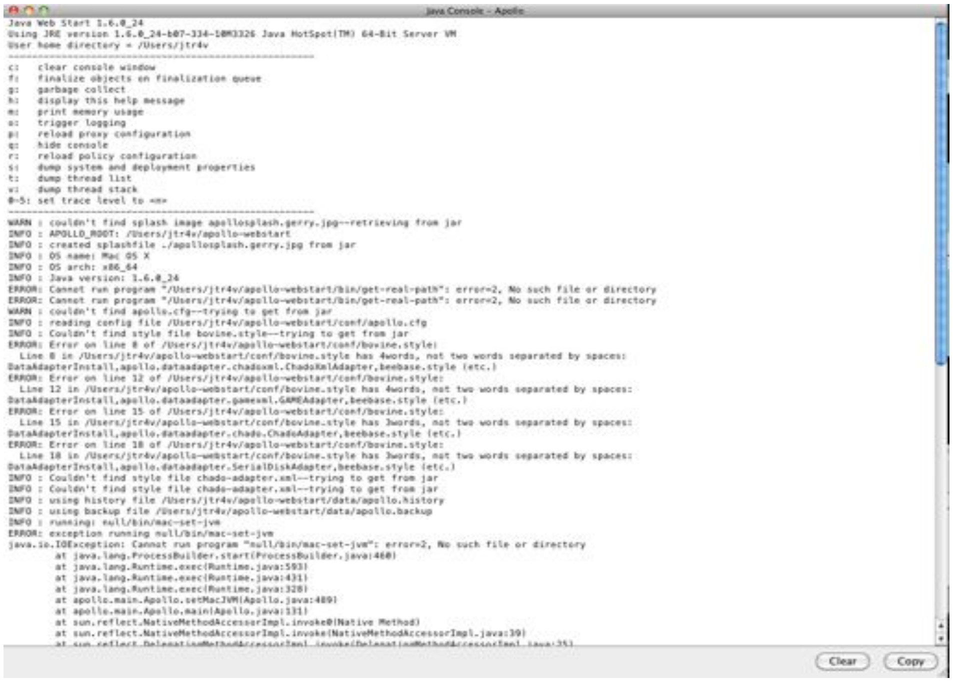
Running Apollo requires that Java be installed in your operating system. If you click on the Apollo JNLP file and Apollo does not start, it is possible that Java is not installed on your operating system. To install java, follow the instructions here:
http://www.java.com/en/download/help/download_options.xml

IX. B. Enabling the java console

If Apollo freezes for several minutes or more while executing an operation (e.g. opening a connection to a Chado database, saving a feature), it is likely that it has encountered an error.

When troubleshooting and reporting problems you are experiencing with Apollo, it is often useful to be able to read these errors. To do this, you must enable the java console, which is a log file that records low-level information about what Apollo is doing. **When requesting help or reporting a bug or problem to BeeBase, please include the output from the java console whenever possible.**

The java console is simply a window that shows lower-level information about what Apollo is doing. This information is often useful when debugging problems with Apollo. It looks something like this:



```
Java Web Start 1.6.0_24
Using JRE version 1.6.0_24-b07-334-10M3326 Java HotSpot(TM) 64-Bit Server VM
User home directory = /Users/jtr4v
----------------------------------------------------
c:   clear console window
f:   finalize objects on finalization queue
g:   garbage collect
h:   display this help message
m:   print memory usage
o:   trigger logging
p:   reload proxy configuration
q:   hide console
r:   reload policy configuration
s:   dump system and deployment properties
t:   dump thread list
v:   dump thread stack
0-5: set trace level to <n>
----------------------------------------------------
WARN : couldn't find splash image apollosplash.gerry.jpg--retrieving from jar
INFO : APOLLO_ROOT: /Users/jtr4v/apollo-webstart
INFO : created splashfile ./apollosplash.gerry.jpg from jar
INFO : OS name: Mac OS X
INFO : OS arch: x86_64
INFO : Java version: 1.6.0_24
ERROR: Cannot run program "/Users/jtr4v/apollo-webstart/bin/get-real-path": error=2, No such file or directory
ERROR: Cannot run program "/Users/jtr4v/apollo-webstart/bin/get-real-path": error=2, No such file or directory
WARN : couldn't find apollo.cfg--trying to get from jar
INFO : reading config file /Users/jtr4v/apollo-webstart/conf/apollo.cfg
INFO : Couldn't find style file bovine.style--trying to get from jar
ERROR: Error on line 8 of /Users/jtr4v/apollo-webstart/conf/bovine.style:
  Line 8 in /Users/jtr4v/apollo-webstart/conf/bovine.style has 4words, not two words separated by spaces:
DataAdapterInstall,apollo.dataadapter.chadoxml.ChadoXmlAdapter,beebase.style (etc.)
ERROR: Error on line 12 of /Users/jtr4v/apollo-webstart/conf/bovine.style:
  Line 12 in /Users/jtr4v/apollo-webstart/conf/bovine.style has 4words, not two words separated by spaces:
DataAdapterInstall,apollo.dataadapter.gamexml.GAMEAdapter,beebase.style (etc.)
ERROR: Error on line 15 of /Users/jtr4v/apollo-webstart/conf/bovine.style:
  Line 15 in /Users/jtr4v/apollo-webstart/conf/bovine.style has 3words, not two words separated by spaces:
DataAdapterInstall,apollo.dataadapter.chado.ChadoAdapter,beebase.style (etc.)
ERROR: Error on line 18 of /Users/jtr4v/apollo-webstart/conf/bovine.style:
  Line 18 in /Users/jtr4v/apollo-webstart/conf/bovine.style has 3words, not two words separated by spaces:
DataAdapterInstall,apollo.dataadapter.SerialDiskAdapter,beebase.style (etc.)
INFO : Couldn't find style file chado-adapter.xml--trying to get from jar
INFO : Couldn't find style file chado-adapter.xml--trying to get from jar
INFO : using history file /Users/jtr4v/apollo-webstart/data/apollo.history
INFO : using backup file /Users/jtr4v/apollo-webstart/data/apollo.backup
INFO : running: null/bin/mac-set-jvm
ERROR: exception running null/bin/mac-set-jvm
java.io.IOException: Cannot run program "null/bin/mac-set-jvm": error=2, No such file or directory
        at java.lang.ProcessBuilder.start(ProcessBuilder.java:460)
        at java.lang.Runtime.exec(Runtime.java:593)
        at java.lang.Runtime.exec(Runtime.java:431)
        at java.lang.Runtime.exec(Runtime.java:328)
        at apollo.main.Apollo.setMacJVM(Apollo.java:489)
        at apollo.main.Apollo.main(Apollo.java:131)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
```

Fig 1. Java Console.

Enabling the java console depends on the operating system you use. Below is a guide to enabling the java console in Mac Os X, Windows and Linux:

Os X

1. In spotlight in the upper right hand corner of your screen, search for **java preferences**, and click on "java preferences".

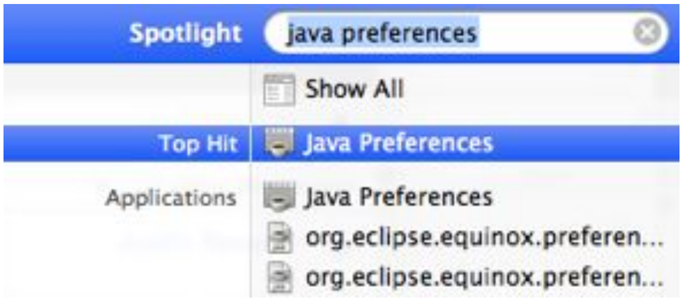
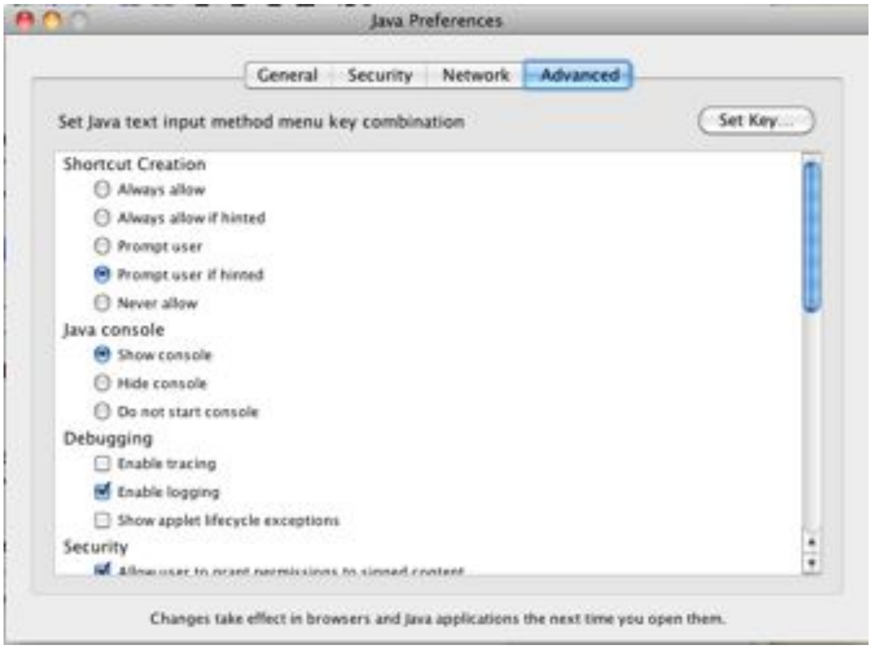


Fig 2. Spotlight Search Window; Apple Inc.

-OR-→ Navigate to the following folder: **Applications** > **Utilities** > **Java** and click on the **java preferences** icon

2. Click the **Advanced** tab
3. Under the Java console section, select the **Show console** radio button.
4. Close and re-open your Apollo. You should see the Java Console appear as a separate window. (The Java console should now appear every time Apollo is opened. To deactivate this behavior, go back to Java Preferences, click the advanced tab and select the **Do not start console** radio button.



Set Java text input method menu key combination

Shortcut Creation

- ○ Always allow
- ○ Always allow if hinted
- ○ Prompt user
- ● Prompt user if hinted
- ○ Never allow

Java console

- ● Show console
- ○ Hide console
- ○ Do not start console

Debugging

- ☐ Enable tracing
- ☑ Enable logging
- ☐ Show applet lifecycle exceptions

Security

- ☑ Allow user to grant permissions to signed content

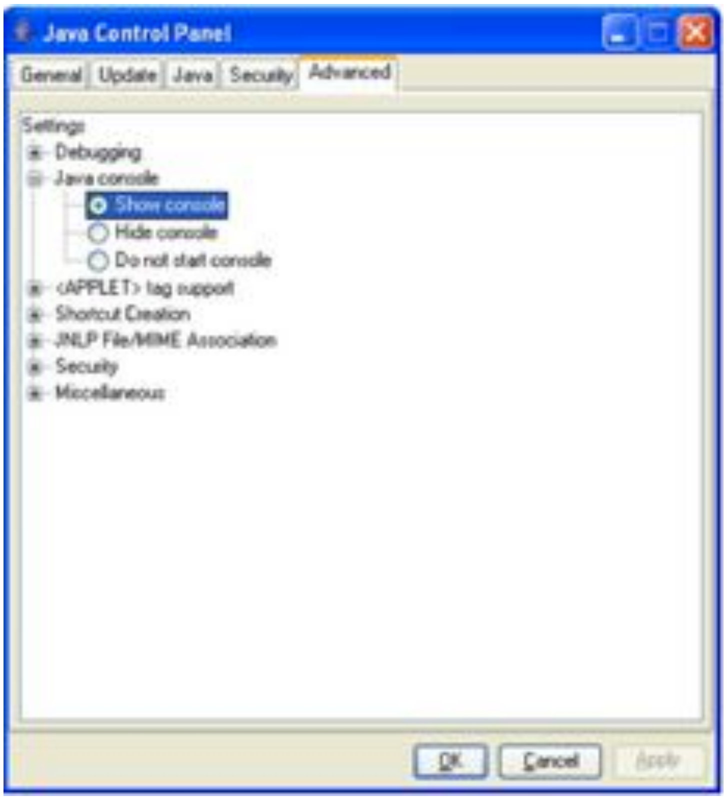
Changes take effect in browsers and Java applications the next time you open them.

Fig 3. Java Preferences Window; Apple Inc.

Windows

See here for details: http://www.java.com/en/download/help/javaconsole.xml

1. Click **Start**
2. Select **Control Panel**.
3. Double click the **Java** icon.
4. Click the **Advance** tab.
5. Click on the ⊞ sign under “Java console”
6. Select **Show Console** and click **Apply**.



Settings
- Debugging
- Java console
  - ● Show console
  - ○ Hide console
  - ○ Do not start console
- <APPLET> tag support
- Shortcut Creation
- JNLP File/MIME Association
- Security
- Miscellaneous

Fig 4. Java Control Panel in MS Windows.

Linux

1. Open a terminal window
2. Go to the directory in which you install the Java Runtime Environement (JRE) and find a program called **ControlPanel**. If you don't know where this is, type:
   ```
   find / -name "ControlPanel"
   ```
   to locate it. This will likely take a few minutes.
3. Open the Java Control Panel by typing:
   ```
   ./ControlPanel
   ```
4. In the Java Control Panel, click the **Basic** tab
5. Select **Show Console** radio button
6. Click the **Apply** button

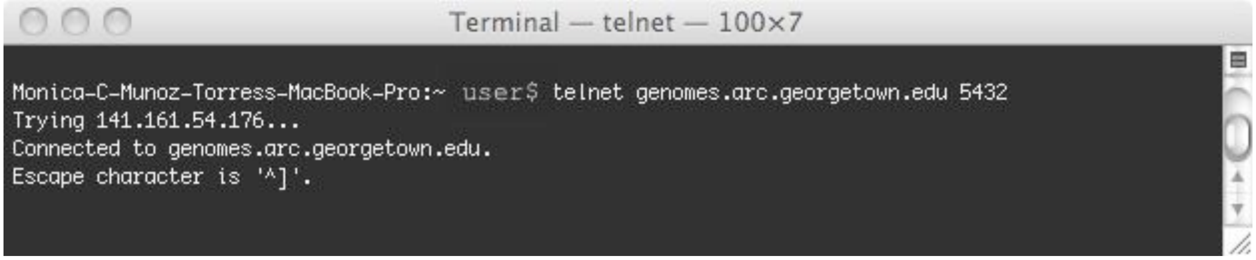
IX. C. Connecting to HGD server at Georgetown University: open port 5432.

In order for Apollo to function correctly, your computer must be allowed to connect to our server at Georgetown on port 5432. If Apollo is unable to connect to our servers, it is possible that a firewall is blocking this port and preventing a connection to our server. If you are comfortable using a command line, you can diagnose this problem by opening a terminal and typing this command:

```
telnet genomes.arc.georgetown.edu 5432
```

If the server replies as in the following example, then port 5432 is open and this is not the problem:

```
Trying 141.161.54.176...
Connected to genomes.arc.georgetown.edu.
Escape character is '^]'.
```



```
Monica-C-Munoz-Torress-MacBook-Pro:~ user$ telnet genomes.arc.georgetown.edu 5432
Trying 141.161.54.176...
Connected to genomes.arc.georgetown.edu.
Escape character is '^]'.
```

Fig 5. Testing whether a connection to HGD server at Georgetown University exists.

If there is no reply and the request times out with a message as shown here:

```
Trying 141.161.54.176...
telnet: connect to address 141.161.54.176: Operation timed out
telnet: Unable to connect to remote host
```

it is possible that this port is being blocked by a firewall at your university/institution. To correct this, you will need to contact the IT help desk at your institution and request that they open this port or use Apollo on a computer outside the firewall. It is also possible that a firewall on your computer itself is blocking this port. To correct this, contact your IT help desk to get instructions about how to change your firewall settings to allow connections to port 5432.

This tutorial was prepared for BeeBase by:
Monica C. Munoz-Torres, Justin T. Reese, Jaideep P. Sundaram and Chris Elsik.
Elsik Laboratory.
Department of Biology, Georgetown University. Washington, DC.
http://genomes.arc.georgetown.edu/elsiklab/

It includes modifications to the instructions prepared for

Bovine Genome Database, by:
Juan Anzola, Texas A&M University
Christopher Childers, Georgetown University
Lynn Crosby, FlyBase Consortium, Harvard University
C. Michael Dickens, Chris Elsik and Justin Reese, Texas A&M University

The Nasonia Genome Sequencing Working Group, by:
Chris Desjardins, University of Rochester
Chris Elsik, Georgetown University