Supplementary exercises for Jalview 2.9 alpha

Cambridge Jalview Training Course – May 2015

cDNA and Protein alignment Split Frame Exercises – setup

Launch Jalview using this URL (download and open the JNLP file):
http://www.jalview.org/services/launchApp?version=Next&jvm-max-heap=15G

Split Frame Exercise 1 – Get Cross-References

1. For now, *uncheck* 'Show annotations' in your Jalview preferences (**Tools => Preferences** and the 'Visual' tab)
2. Choose menu option **Desktop => File => Fetch Sequences…**
3. Select EMBL and fetch these accession ids (copy and paste from here):

X04752 ; U07177 ; AF070998 ; U95378 ; U13680 ; U07178 ; X02152 ; M22585; U13687 ; X01964 ; X53828 ; U28410

4. In the new alignment window, select option **View => Sequence ID Tooltip => Show Database Refs**[1]
5. Mouse over the sequence ids – notice the PDB and/or Uniprot cross-references
6. Check option **View => Show Sequence Features** is selected
7. Mouse over the sequences; note that the coding (exon) regions are annotated as a CDS feature (in grey)
8. Select UNIPROT from the **Calculate => Get Cross-References** submenu

(Jalview is able to retrieve some (but not all) cross-references, including those to PDB and Uniprot)

9. A 'split frame' window is displayed, showing protein sequences below their coding DNA sequences; note that any non-coding regions are omitted here
10. Mouse over the sequences – notice the highlight following in the other pane, and also in the exon regions of the EMBL sequences
11. Scroll the alignments (using scrollbar or trackpad); they should scroll together[2]
12. Select rows, columns or regions of one alignment – the corresponding region of the other alignment should be selected
13. Select a group of sequences in one pane and move them up or down using the cursor keys; what happens in the other pane?
14. Try one of the **Calculate => Sort** options in one pane – what happens in the other pane?
15. Manually enter gaps in the protein sequences. Equivalent gaps (3 bases) should appear in the cDNA; try **Edit => Undo** – what happens?
    (note linked editing currently operates protein-to-cDNA only, not the reverse)
16. In the cDNA pane:
    a. select **Edit => Pad Gaps** (to make all sequences equal length)
    b. select **Calculate => Calculate Tree => Average Distance Using DNA**

---

[1] tip: you can also turn this on or off under **Tools => Preferences | Visual**
[2] mouseover and scroll tracking can be turned on or off by **View | Automatic**
[2] mouseover and scroll tracking can be turned on or off by **View | Automatic Scrolling**

c. Click in the tree view to partition the sequences into groups. Note that corresponding groups are made in both cDNA and protein views.
d. in the Tree panel, select **View => Sort Alignment By Tree**; notice that both panes' sequences are sorted

Discussion point: is it more informative to calculate trees on cDNA or on protein sequences?

17. In either panel, open the Font dialog (**Format => Font**), and try checking and unchecking the option *'Scale protein residues to codons'*; what happens?
18. Turn option **View => Protein** in the cDNA pane off and on to hide or show the protein alignment; similarly option **View => Nucleotide** in the protein pane
19. In either panel, select **View => New View**; you can create multiple split views of the same alignments, and format and colour each one independently (as for single alignments); also try View options **Expand Views** and **Gather Views**
20. In the Protein alignment, select (turn on) option **Annotations => Show annotations**
21. Notice the annotation row for **cDNA Consensus**; this shows the consensus scores for codon triplets, for each column in the peptide sequence
22. Find a column in the peptide alignment with 100% consensus in the protein, but not in the cDNA (I could find four!).

Discussion point: does variation in cDNA hold clues to phylogeny of homologous proteins?

Split Frame Exercise 2 – Translate DNA

1. Close the split window from Exercise 1
2. Re-fetch (if not still displayed) the sequences as at Exercise 1, step 3
3. Select option **Calculate => Translate as cDNA**
4. A split frame view shows DNA above and its peptide translation below
5. Try selecting rows, columns, or a sub-region to see just the selected area 'translated'

Note this differs from Exercise 1 – all of the DNA sequences are translated, including any intron and exon regions. This generates 'real' peptides provided only coding DNA is loaded. Or you could treat it as a simple 'scratchpad' tool to inspect possible coding frame translations, by selecting different nucleotide regions.

Split Frame Exercise 3 – Load cDNA

1. You can also see a split frame view of protein and cDNA sequences loaded from files
2. Close all windows
3. Select **File => Input Alignment from URL** and enter (copy and paste) http://www.jalview.org/builds/nextrel/examples/estrogenReceptorCdna.fa
4. In the alignment window, select **File => Add Sequences => From URL** and paste

http://www.jalview.org/builds/nextrel/examples/estrogenReceptorProtein.fa

5. A dialog will be shown. Accept the option to open in a Split Window
6. The protein alignment is displayed below, and linked to, its coding DNA

You can add peptide sequences to cDNA (as above), or vice versa.

This method will work with your own data sets, provided certain conditions are met:

- the DNA sequences must translate exactly to their corresponding peptide sequences (but may include one start and/or stop codon)
- the two sets of sequences do not have to be in the same order, but Jalview will 'pair them off' in order if there is ambiguity (if one sequence translates to / from more than one sequence)

Split Frame Exercise 4 – Realign Split Frame

1. If you realign one half of a split frame, Jalview will insert gaps in the other half to match
2. Repeat (or continue) Exercise 3, to show a split view of nine Estrogen Receptor proteins and their cDNA
3. In the Protein alignment, select **Web Service => Alignment** and an alignment method of your choice; the alignment should take less than a minute
4. A new split view is opened showing the realigned protein sequences below, and in the top pane the set of cDNA sequences with gaps inserted to match those in the peptide sequences
   Note: these are gaps added by Jalview to 'match' the protein alignment, *not* an algorithmic realignment of the cDNA sequences
5. You can repeat this exercise, aligning the cDNA instead of the protein; Jalview will insert gaps in the protein based on the alignment of their codons in the cDNA alignment
   Note: this can result in a sparse (heavily gapped) protein 'alignment'

Discussion point: is it more informative to perform alignment on cDNA or on protein sequences?

Calculating T-COFFEE Reliability scores

1. Load the tcoffee_scores.fasta alignment into Jalview
2. Calculate a T-COFFEE score file for the alignment using the T-COFFEE COREX score server at http://tcoffee.crg.cat/apps/tcoffee/do:core
3. Locate the ascii score file on the results page – it's called 'sequence alignment in ascii format' (url should end in result.score_ascii)
    a. Save this to your desktop
    b. drag the file onto the alignment in Jalview
4. Toggle the T-COFFEE score colouring
5. Experiment with the Colour By Annotation dialog and export Annotation open in the file menu to see how Jalview handles T-COFFEE scores
6. Try uploading another file to the score server to calculate scores for your own sequences

Groovy Scripting and Jalview

1. Open the groovy console from the Desktop's Tools menu.

Load and experiment with the following scripts using the example alignment:

2. Printing the alignment's title
http://www.jalview.org/examples/groovy/printtitle.groovy

3. Generating CSV from annotation
http://www.jalview.org/examples/groovy/annotationsascsv.groovy

4. Parsing description strings as alignment annotation (using the 'Extract scores' function)
Example file:
http://www.compbio.dundee.ac.uk/user/ws-dev1/examples/scanps_out.blc
Script:
http://www.jalview.org/examples/groovy/parseproperties.groovy

6. Manipulating features programmatically
http://www.jalview.org/examples/groovy/removeFeaturesByGroup.groovy

7. Manipulating sequence IDs in alignment
Load some sequences from Uniprot using the sequence fetcher.
Try this script http://www.jalview.org/examples/groovy/stripUniprotPrefixes.groovy

Setting up the Jalview Yoxos Eclipse development environment – if you want to add to Jalview!

1. Download and double click the jalviewDeveloper-www profile (on tutorial page)
2. wait a few minutes then click the 'I accept button' and OK
3. wait a few minutes more for the eclipse download to happen
4. If you want to do commits, then you'll need to get a username and password over at issues.jalview.org
5. Press ok for default eclipse workspace paths.
6. use your the username and password to log in to Jalview's git repository at source.jalview.org if you have one, otherwise you can hit cancel when it prompts you for username/password (If you hit ok then you'll need to set master password details .. don't worry - these machines will be wiped at the end of the day).
7. Wait a bit more - Jalview's git repository will be cloned automatically
8. Open git repository browser and Switch branches to Release_2_8_Branch to get the latest patched release branch in your desired repository.
9. Choose Import ..-> From Git -> Import Existing project -> From Local Repository -> pick the public or personal repository

10. Fix paths (some of the below may not be necessary)
i. Open builders - there's one invalid builder that you'll need to get rid of
ii. Open build path - VARNAv3.9-dev.jar reports being not found:
 * first remove from project path.
 * then add it again by right clicking after browsing to the 'lib' directory
iii. Fix the plugin.jar user library
* select plugin.jar in the build path and hit 'Edit ...'
* select 'user libraries ...' button.
* create a new user library called plugin.jar that links to the jalview 'plugin.jar' classpath entry
* select 'Add external jar' and locate plugin.jar on your system "Program files (x86)/Java then search for plugin.jar
* hit ok then finish then ok to finish editing the classpath.

11. source should be ready to hack ! test you can launch Jalview by :
i. right click on project and pick Run As -> External Application
ii. once main types are found look for jalview.bin.Jalview
iii. let the program launch.
iv. to double check - make sure you can import the example alignment at http://www.jalview.org/examples/exampleFile_2_7.jar

12. Building the applet is tricker (need to use the makeapplet target to build the jalviewApplet.jar), but it should launch via the appletviewer using as 'launch java applet' runtime profile. Use parameters from one of the examples in the examples/applets.html page.