An Introduction to Taverna

Basic Features of Taverna – A Practical

1. Installing the Workbench

Exercise 1 - Running Taverna

- Download Taverna from http://taverna.sourceforge.net
  - **Windows or linux**

If you are using either a modern version of Windows (Win2k or WinXP, with XP preferred) or any form of linux, solaris etc. you should download the workbench zip file. For windows users, Taverna can be unzipped and used, for linux you will also need to install GraphViz (http://www.graphviz.org/ the appropriate rpm for your platform)

  - **Mac OSX**

If you are using Mac OSX you should download the .dmg workbench file. Double-click to open the disk image and copy both components (Taverna and GraphViz) onto your hard-disk to run the application

- YOU WILL ALSO NEED a modern Java Runtime Environment (JRE) or Java Software Development Kit (SDK) from http://java.sun.com Java 5 or above

Workbench Layout

- AME – Advanced Model Explorer

  The Advanced Model Explorer (AME) is the primary editing component within Taverna. Through it you can load, save and edit any property of a workflow.

  - enables
    - building
    - loading
    - editing
    - saving workflows

Workflow Diagram Window

Visual representation of workflow

- Shows inputs / outputs, services and control flows
- Enables saving of workflow diagrams for publishing and sharing

Available Services Panel

Lists services available by default in Taverna

- ~ 3000 services
  - Local java services
  - Simple web services
  - Soaplab services – legacy command-line application
  - Gowlab services
  - BioMart database services
  - BioMoby services

  Allows the user to add new services or workflows from the web or from file systems

Installing Plugins

- Additional plugins such as Feta and Logbook are available for Taverna. These can be added via the Tools menu.

- Go to the 'Tools' menu at the top of the workbench and select the *Plugin manager*
- Select *find new plugins*
- Tick the box for *Feta* and install this plugin
- Two more options '*Discover*' will now have appeared at the top of your screen
- Feta is now available through the Discover tab

2. Adding New Services

Exercise 2 - Adding a New Service

New services can be gathered from anywhere on the web

1. Go to http://taverna.sourceforge.net/ and go to the documentation page.
2. Select '(partial) service list'

   These services are not all included by default when Taverna opens.
3. Scroll down the page to DDBJ services. You will see a list of available DDBJ services.
4. Click on the DDBJ blast service (http://xml.nig.ac.jp/wsdl/Blast.wsdl) and copy the web page address

Exercise 2 - Adding a New Service

5. Go to the '*Available services*' panel and right-click on 'Available Processors'.

   For each type of service, you are given the option to add a new service, or set of services.
6. Select '*Add new wsdl scavenger*'.

   A window will pop-up asking for a web address
7. Enter the Blast Web service address
8. Scroll down to the bottom of the '*Available Services*' panel and look at the new DDBJ service that is now included.

3. Finding and Invoking a Service

Exercise 3.1 - Finding a Service

1. Go to the 'Available Services' Panel
2. Search for *Fasta* in the '*search list*' box at the top of the panel (we will start with simple sequence retrieval)
3. You will see several services highlighted in red
4. Scroll down to '*Get Protein FASTA*'
   This service returns a Fasta sequence from a database if you supply it with a sequence identifier

Exercise 3.2 - Invoking a Single Service

5. Right click on the '*Get Protein FASTA*' service and select '*Invoke service*'
6. In the pop-up 'Run workflow' window add a protein sequence GI by selecting ID and right-clicking. Select 'new input value' and enter a value (e.g., 1220173) in the box on the right

   Note: GI is a genbank gene identifier (you don't need the gi: just the number, for example, the MAP kinase phosphatase sequence 'GI:1220173' would be entered as '1220173'
7. Click '*Run workflow*' and the service is invoked

Exercise 3.3 - Viewing Results

8. Click on '*Results*'
   The fasta sequence is displayed on right when you select *click to view*
9. Click on '*Process Report*'
   Look at processes.
   This shows the experiment provenance – *where* and *when* processes were run
10. Click on '*Status*'
    Look at options
    As workflows run, you can monitor their progress here.

Exercise 3 - Conclusion

The processes for running and invoking a single service are the basics for any workflow and the tracking of processes and generation of results are the same however complicated a workflow becomes

In the next few exercises, we will look at some example workflows and build some of our own from scratch

4. Finding and Using Workflows

Exercise 4.1 - Finding and Using Workflows

1. Select 'Load Workflow' from the File menu at the top of the workbench.
   You will see a selection of .xml files in an examples directory. These are workflow definition files
2. Select '*ConvertedEMBOSSTutorial.xml*' and a pre-defined workflow will be loaded
3. View the workflow diagram - you will see services of in different colours

Exercise 4.2 - Workflow Documentation

4. Find out what the workflow does by reading the workflow metadata
5. In the AME – click on the name of the workflow – in this case '*A workflow version of the EMBOSS tutorial*' and then select the '*workflow metadata*' tab at the top of the AME.

   You will see a text description of the workflow, its author and its unique LSID. When publishing workflows for others, this annotation is useful information and allows the acknowledgement of intellectual property

Exercise 4.3 - Workflow Features

6. Run the workflow by selecting 'Run workflow' from the file menu
7. Watch the progress of the workflow in the '*enactor invocation*' window. As services complete, the enactor reports the events. If a service fails, the enactor reports this also.

Loading workflows from the Web

1. Go to the webpage http://www.cs.man.ac.uk/~ytanoh
2. Select '*ConditionaBranchChoice.xml*' and copy the web address
3. Go back to the taverna workbench and select 'Open from web'
4. Run the workflow .

You will see at least one of the services fail. These are conditionals – fail of false or fail if true. These are useful operators for controlling progress of your workflow based on intermediate results

You will see black arrows and white circles – black arrows show the flow of the data and white circles are *control links*.

5&6. Building a Simple Workflow

Exercise 5.1 - Adding a Service

1. Import the '*Get Protein FASTA*' service into a new workflow model

   First, you will need to close the current workflow from the file menu, then find the '*Get Protein Fasta*' service again in the '*Available services*' panel.
2. Right-click on 'Get Protein Fasta' and import it into the workbench by selecting '*Add to Model*'
3. Go to the AME and expand the [+] next to the newly imported *'Get Protein Fasta'* service. You will see:
   - 1 input (green arrow pointing up)
   - 1 output (purple arrow pointing down)

Exercise 5.2 - Adding Input

1. Define a new workflow input by right-clicking on '*Workflow Input*' and selecting 'create new Input'
2. Supply a suitable name *e.g.*, '*geneIdentifier*'
3. Connect this new input to the '*Get Protein Fasta*' service by right-clicking on '*geneIdentifier*' and selecting '*getFasta ->id*'

You always build workflows with the flow of data

Exercise 5.3 - Adding Output

4. Define a new workflow output by right-clicking on '*workflow output*' and selecting 'create new output'
5. Supply a suitable name e.g. '*fastaSequence*'
6. Connect this new output to the '*Get Protein Fasta*' service. remembering to build with the flow of data
   You have now built a simple workflow from scratch!
7. Run the workflow by selecting 'run workflow' from the '*Tools and Workflow Invocation*' menu at the very top of the workbench
   You will again need to supply a GI – for later exercises, please use a protein GI – e.g. 1220173

Exercise 6 - Stringing Services Together

We have used *'Get Protein Fasta'* to retrieve a sequence from the Genbank database. What can we do with a sequence?

- Blast it?
- Find protein motifs?
- Find secondary structure elements, such as alpha helices?
- Find GO annotations?

Blast it?

The first thing you need to do is find a service which performs a blast. For this, we are going to use the ***Feta Semantic Discovery*** Tool

Feta is a tool to semantically describe services. Instead of the user needing to know exactly what a service provider has called their services, the user can search by the biological tasks that are performed by the services, or by properties of the service, for example, the types of inputs it requires/outputs it produces

Exercise 6.1 - Finding Blast using Feta

1. Select the *'Discover'* tab and select '*uses method*' from the first drop down menu
2. When you select it, '*bioinformatics algorithm*' will appear in the adjoining box. Scroll down this list to find '*Similarity search algorithm*', and then the subclass of this, '*BLAST*'
3. Select 'BLAST' and click '*Find Service*'
   The results are all the annotated services that perform BLAST analyses (there may be more un-annotated ones!)

Exercise 6.1 - Finding Blast

4. Select '*searchSimple*' from the list and look at the details
5. Look at the service description
   This tells you what the service does and what each input/output is expecting/produces. It also tells you where the service comes from. For this example, we are using BLAST from the DNA Databank in Japan
6. Right-click on '*searchSimple*' in the Feta results list and select 'add to model'
   This adds the service to your current workflow in the 'Design Window'
7. Before you go back to the Design window, go back to search services and experiment with other ways of finding services – e.g., by task, input/output, resource etc

Exercise 6.2 - Adding Blast

1. Go back to the Design window. '*Search simple*' will have been imported into your model
2. In the AME expand the [+] for the '*search simple*' service and view the input/output parameters

   This time, you will see three inputs and two outputs.
   For the workflow to run, each input must be defined. If there are multiple outputs, a workflow will usually run if at least one output is defined.

Exercise 6.2 - Adding Blast

3. Create an **output** called '*blast_report*' (in the same way we did before)
4. The sequence input for the Blast service will be the output from the '*Get Protein Fasta*' service. Connect the two together, from '*fastaSequence*' (output from *GetProteinSequence* service) to '*search simple query*'
5. Create two more inputs called '*database*' and '*program*' and connect them to the '*database*' and '*program*' inputs on '*search simple*' service

Exercise 6.3 - Running the 'Blast it' Workflow

1. Once more select '*run workflow*' from the '*Tools and Workflow Invocation*' menu. You will see a run workflow window asking for 3 input values
2. Insert a GI (*e.g.,* 1220173), a program (blastp for protein-protein blast), and a database (SWISS for swissprot)
3. Click 'run workflow'. This time you will see a blast report and a fasta sequence as a result

Exercise 6.4 - Changing Parameters

For parameters that do not change often, you will not wish to always type them in as input. In this example, the database and blast program may only change occasionally, so there is an alternative way of defining them

1. Go back to the AME and remove the '*database*' and '*program*' inputs by right-clicking and selecting '*remove from model*'

Exercise 6.5: String Constants

1. Select '*string constant' from 'Available Services'*
2. Right-click and select 'add to model with name…'
3. Insert '*program*' in the pop-up window
4. Select '*string constant*' for a second time and repeat for a string constant named '*database*'
5. In the AME, right-click on '*program*' and select '*edit me*'
6. Edit the text to 'blastp'. Repeat for 'database' and enter 'SWISS' for the swissprot database
7. Run the workflow – it runs in the same way
8. Save the workflow by selecting the 'save' icon at the top of the AME.

7. Protein Annotation Workflow

Exercise 7.1 - Adding More Services

How can we use Taverna to annotate our protein with function descriptions?

1. In the 'available services' panel, find the emboss soaplab services and find the '*protein_motifs*' section

   Hint: use the simple text search at the top of the panel

2. Find out which of these services enable searching of the Prosite and Prints databases by fetching the service descriptions. To do this *right-click* on '*protein_motifs*' and select '*fetch descriptions*'

3. Import both services into the workflow model.

Exercise 7.1: Adding More Services

5. Connect these services up to the workflow so that you can find Prints and Prosite matches in the query sequence returned from *'Get Protein Fasta'* – you will see that soaplab services have many input values

   Soaplab services have many input parameters, but many have default values so may not always need to be altered. In this case, you can run the services by simply adding the query sequence. Go to the EMBOSS home page to find out which input(s) relate to the query sequence.

   This extra searching is impractical – the Feta Semantic Discovery tool is designed to combat this problem

Exercise 7.2: Running the Protein Annotation Workflow

1. Run the workflow – now you have blast results and protein domain/motif matches

   How else can you annotate your protein? You might want to search for other ways of characterising your sequence. We suggest annotating the sequence with secondary structural elements (e.g., beta sheets) and with Gene Ontology (GO) annotations.

Exercise 8 - Saving Results

Taverna provides several options for saving data.

a) Individual data items can be saved by right-clicking on them
b) All data can be saved to disk
c) Textual/tabular data can be saved to excel

Save all the data from your workflow