XPath service
Getting data out of XML

Alan R Williams
materials by Katy Wolstencroft
University of Manchester

The Basics of XML

- XML – e**X**tensible **M**arkup **L**anguage
- Designed for the storage and transport of data
  - This includes passing data between services or retrieving data from a Web page
- Provides a machine readable dataset
- Many service providers export data in XML

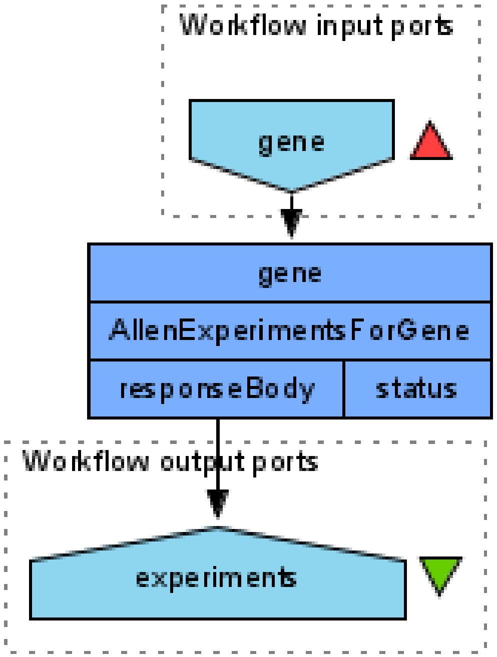
XML Example

```
<?xml version="1.0"?>
<note>
   <to>Katy</to>
   <from>Helen</from>
   <heading>Reminder</heading>
   <body>Don't forget about Bonn Trip!</body>
</note>
```

- The following website has lots of information about XML and tutorials: http://www.w3schools.com/xml

Extracting data from experiments

- Open the *experiments_for_gene* workflow
- It can be downloaded from http://www.myexperiment.org/packs/641.html



Extracting data from experiments

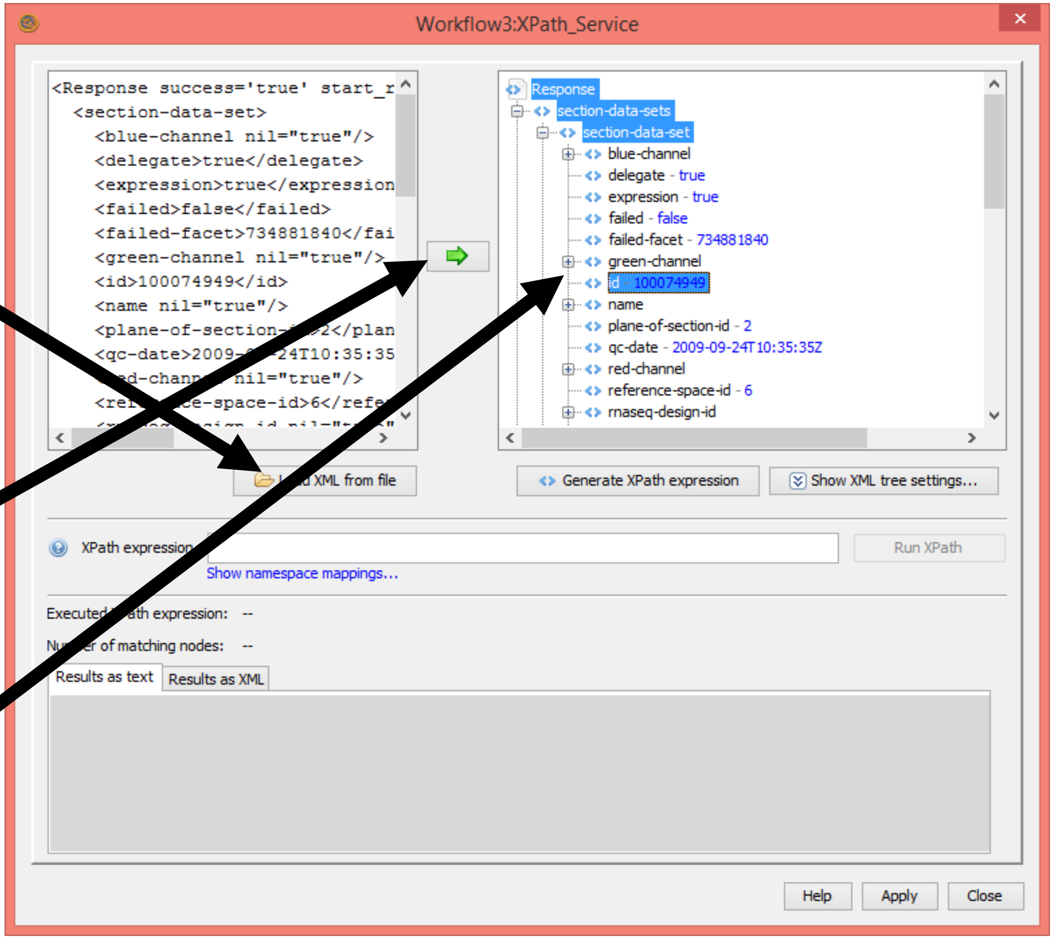
- As the input value enter ***Adora2a***
- Run the workflow
- Look at the output – it is XML data
- Save the output as experiments.xml

Extracting XML Elements

- Often, we need to extract particular elements of the XML to feed into the next services in the workflow
- Taverna has a dedicated service to help you do this
- Go to the **Service panel** and find **XPath Service** in the **Service Templates** section.
- We will use this service to explore the results of the AllenExperimentsForGene service

Extracting data from experiments

- Now add an **XPath Service** to the workflow
- In the pop-up window, click **Load XML from file** and browse to the experiments file and click **Open**
- Click the green arrow to load it into the service
- By expanding the + icons, you can explore the file
- Expand the XML under *section-data-sets*, until you reach *id*
- Click *id*



```
<Response success='true' start_r
  <section-data-set>
    <blue-channel nil="true"/>
    <delegate>true</delegate>
    <expression>true</expression
    <failed>false</failed>
    <failed-facet>734881840</fai
    <green-channel nil="true"/>
    <id>100074949</id>
    <name nil="true"/>
    <plane-of-section-id>2</plan
    <qc-date>2009-09-24T10:35:35
    <red-channel nil="true"/>
    <reference-space-id>6</refe
    <rnaseq-design-id nil="true"
```

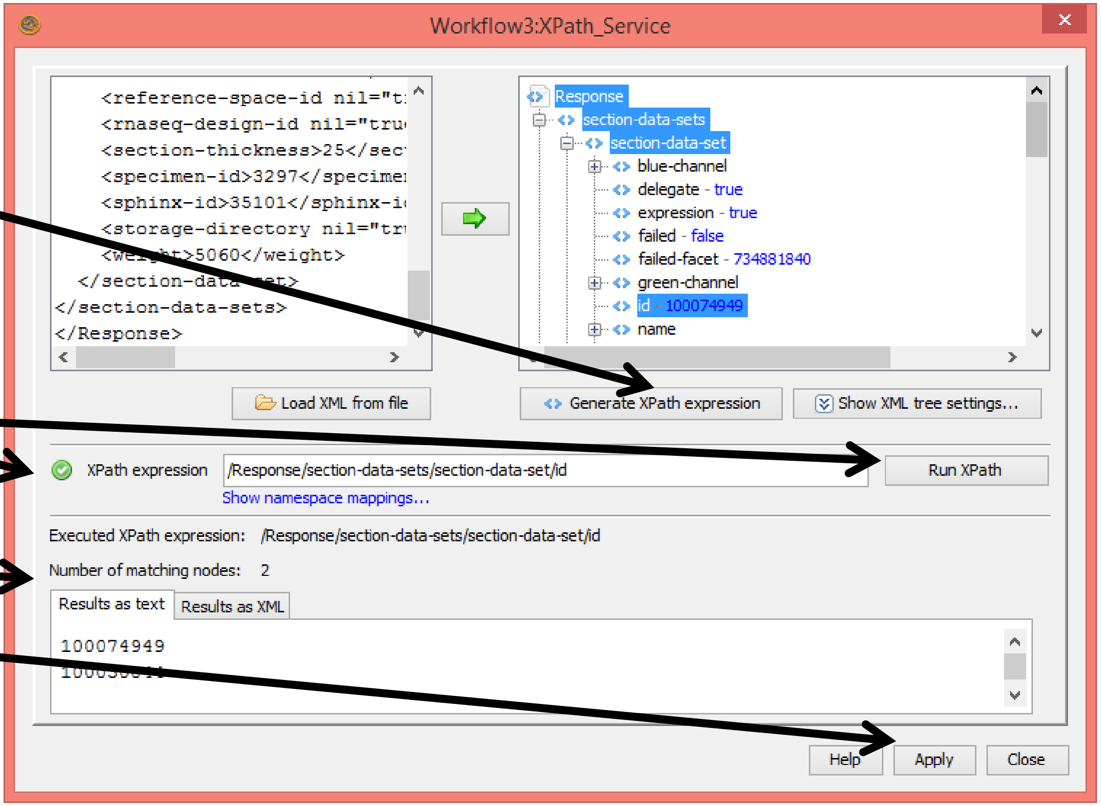
```
Response
  section-data-sets
    section-data-set
      blue-channel
      delegate - true
      expression - true
      failed - false
      failed-facet - 734881840
      green-channel
      id   100074949
      name
      plane-of-section-id - 2
      qc-date - 2009-09-24T10:35:35Z
      red-channel
      reference-space-id - 6
      rnaseq-design-id
```

Executed XPath expression: --

Number of matching nodes: --

Extracting data from experiments

- Click on **Generate Xpath expression**
- You can check and edit the XPath expression if you want
- Then click on **Run Xpath**
- Look at the predicted results
- Click **Apply** and **Close**
- This service will now pull out all the ids for each experiment



Workflow3:XPath_Service

```
    <reference-space-id nil="t
    <rnaseq-design-id nil="tru
    <section-thickness>25</sec
    <specimen-id>3297</specime
    <sphinx-id>35101</sphinx-i
    <storage-directory nil="tr
    <weight>5060</weight>
  </section-data-set>
</section-data-sets>
</Response>
```

XPath expression: /Response/section-data-sets/section-data-set/id

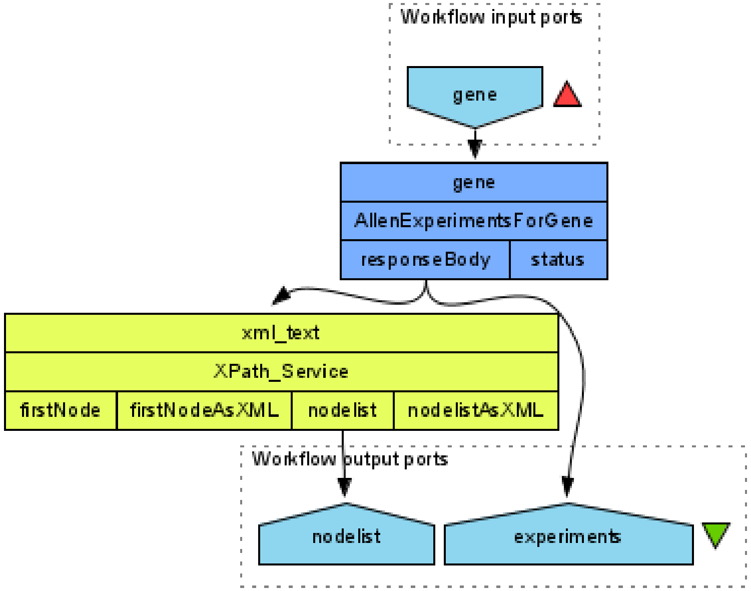
Executed XPath expression: /Response/section-data-sets/section-data-set/id

Number of matching nodes: 2

```
100074949
100030044
```

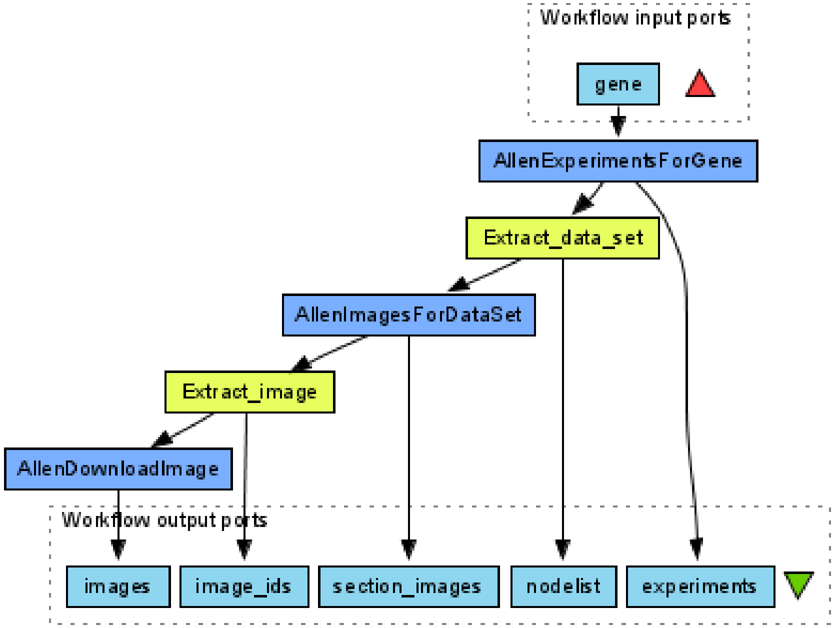
Extracting data from experiments

- Connect the input of the *XPath Service* to the *responseBody* output of *AllenExperimentsByGene*
- Add an output port to the workflow
- Connect the workflow output port to the *nodelist* port of the *XPath Service*
- Save and run the workflow again



Multiple REST and XPath

- Open the “*Download images for gene*” workflow from the workshop myExperiment pack
- Run the workflow with Adora2a as the input value
- It may take some time for the images to download
- Look at the configuration of the XPath and REST services – in the **Design** view, select the service and click **Details**



Summary

- You can now configure XPath services from example XML
- You can select parts of an XML to extract
- You understand how multiple REST and XPath services can be combined