Data-mining with InterMine

InterMine is a data analysis platform designed for use with in the Life Sciences. It is in use at many institutions hosting a diverse collection of datasets, including SGD (Yeast), RGD (Rat), MGI (Mouse), ZFin (Zebra fish), Wormbase (C. elegans) and FlyMine (Fruitfly). This workshop covers the use of the web interface, and how to automate access to the same features.

All the tasks in this workshop will relate to FlyMine (which is maintained by us in Cambridge). URLs for several sites are provided below

Overview

The tasks in this workshop will cover:

- Identifier Resolution
- User lists
- Queries
- Results and Reports
- Enrichment queries
- Exporting Data
- Progammable Interface

Some InterMine installations:

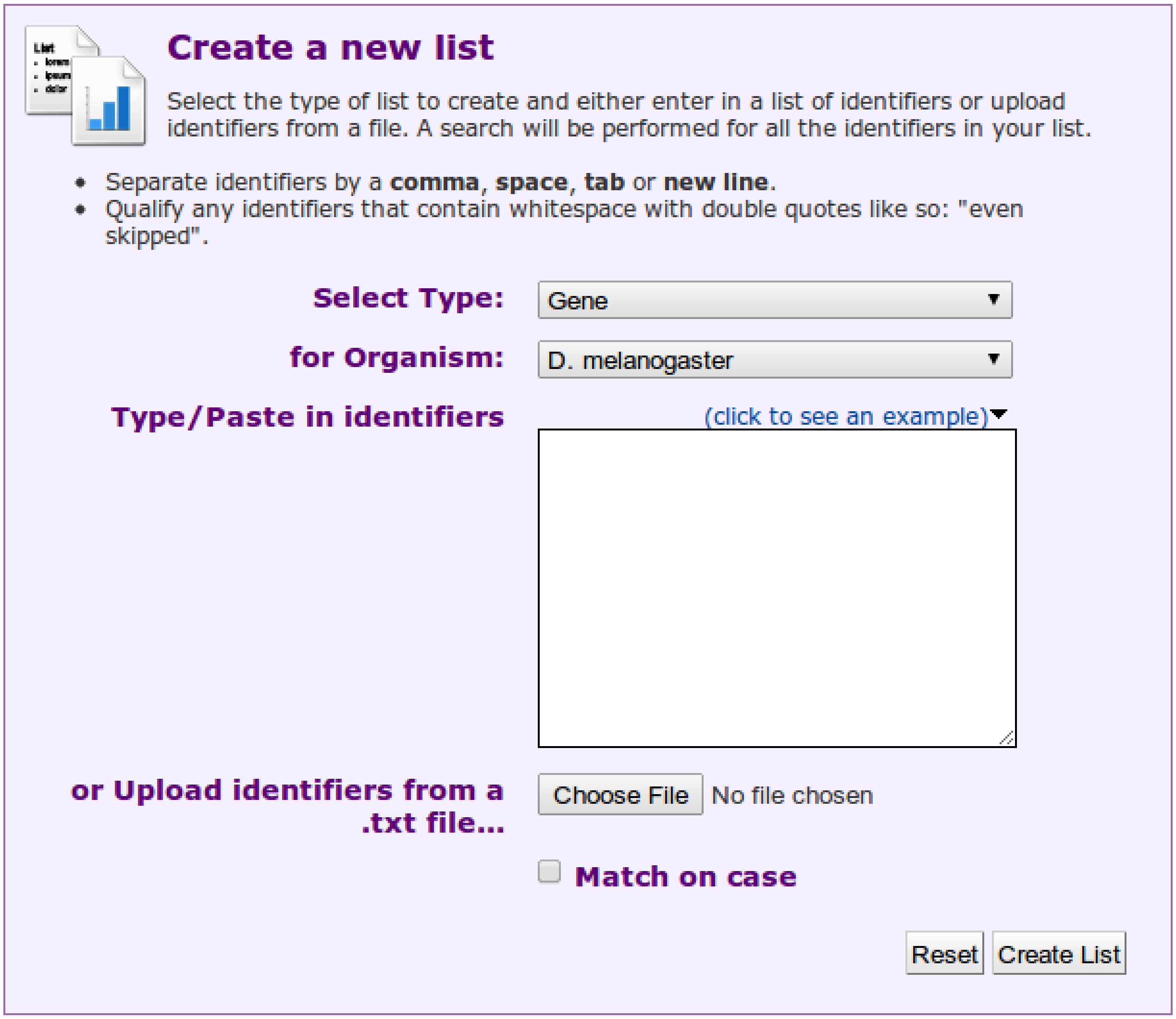
- FlyMine http://www.flymine.org
- YeastMIne (SGD) http://yeastmine.yeastgenome.org/yeastmine
- RatMine (RGD) http://ratmine.mcw.edu/ratmine
- MouseMine (MGI) http://www.mousemine.org/mousemine
- ZebrafishMine (ZFIN) http://zmine.zfin.org
- WormMine (Wormbase) http://www.wormbase.org/tools/wormmine

Webservice Boilerplate

```
from intermine.webservice import Service
service = Service(“www.flymine.org/query”, token = None)
```

Task 1: Identifier Resolution and Lists

InterMine is built with the concept of data-integration at its core, so identifier resolution is very important for our users. Users can come to the system with a range of identifier types (MOD ids, Ensembl ids, Uniprot accessions, outdated synonyms, etc), and we try to match those to good, clean, consistent and unique objects we know about. We keep track of this through the list mechanism.



Create a new list

Select the type of list to create and either enter in a list of identifiers or upload identifiers from a file. A search will be performed for all the identifiers in your list.

- Separate identifiers by a **comma**, **space**, **tab** or **new line**.
- Qualify any identifiers that contain whitespace with double quotes like so: "even skipped".

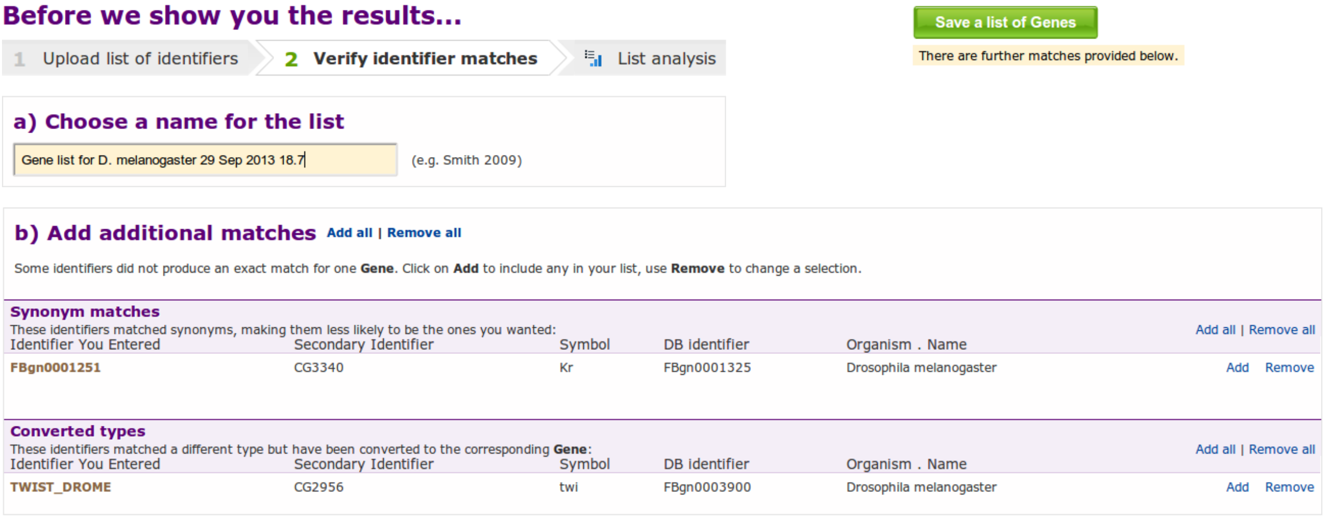
Select Type: Gene

for Organism: D. melanogaster

Type/Paste in identifiers
(click to see an example)
or Upload identifiers from a .txt file...
Choose File
No file chosen
☐ Match on case
Reset
Create List

fig 1: http://www.flymine.org/query/**bag.do**

If there are any issues with the identifiers you need to decide what to do:



Before we show you the results...

There are further matches provided below.

a) Choose a name for the list

Gene list for D. melanogaster 29 Sep 2013 18.7 (e.g. Smith 2009)

b) Add additional matches

Some identifiers did not produce an exact match for one **Gene**. Click on **Add** to include any in your list, use **Remove** to change a selection.

Synonym matches

These identifiers matched synonyms, making them less likely to be the ones you wanted:

Identifier You Entered	Secondary Identifier	Symbol	DB identifier	Organism . Name	Add all | Remove all
FBgn0001251	CG3340	Kr	FBgn0001325	Drosophila melanogaster	Add Remove

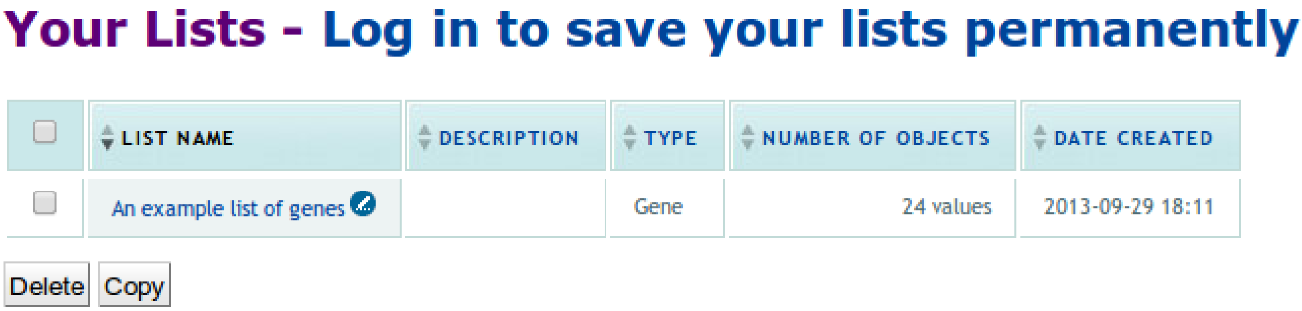
Converted types

These identifiers matched a different type but have been converted to the corresponding **Gene**:

Identifier You Entered	Secondary Identifier	Symbol	DB identifier	Organism . Name	Add all | Remove all
TWIST_DROME	CG2956	twi	FBgn0003900	Drosophila melanogaster	Add Remove

fig 2: http://www.flymine.org/query/buildBag.do

Once you have created a list, you can see that you have access to it in the “mymine” section of the site:



Your Lists - Log in to save your lists permanently

☐	LIST NAME	DESCRIPTION	TYPE	NUMBER OF OBJECTS	DATE CREATED
☐	An example list of genes		Gene	24 values	2013-09-29 18:11

fig 3:http://www.flymine.org/query/mymine.do

These features can all be accessed through webservices:

Creating a list:

```
POST $BASE/service/lists?token=$TOKEN&name=$NAME&type=$TYPE
[YOUR IDS]
```

In Python:

```
name = “My List”
type = “Gene”
src  = “some/path/to/a/file/with/ids.txt”
list = service.create_list(src, name = name, list_type = type)
```

Getting information about your lists:

```
GET $BASE/service/lists?token=$TOKEN
```

In Python

```
service.get_all_lists()
```

Lists can be combined using set operations (union, intersection, difference), edited and deleted.

TASKS

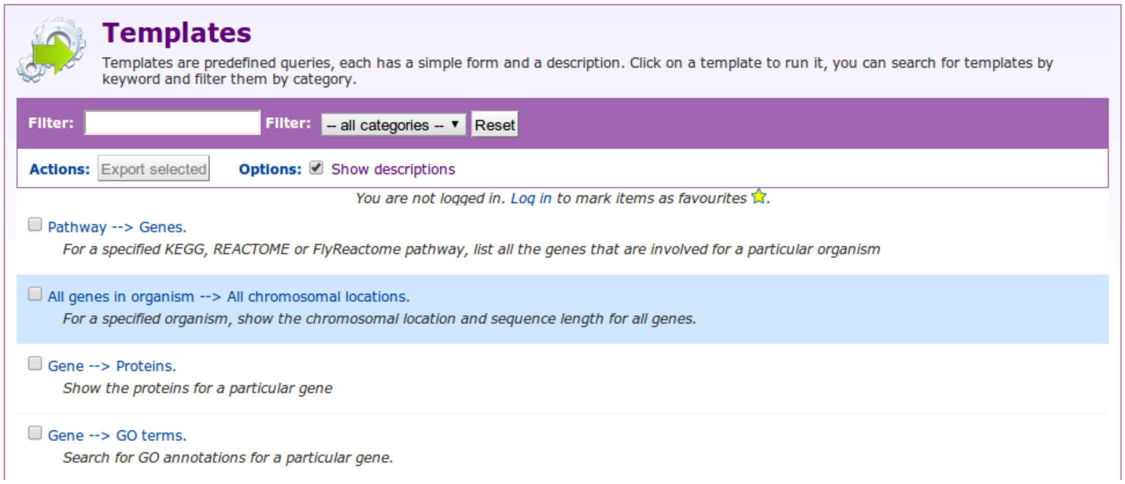
enter a set of identifiers and find the genes they resolve to
- Use the identifiers provided in the example.
- and/or Use the example file provided.

Resolve any issues to your satisfaction

With webservice:
Create a list using the web services.
Change its name.
Find the public lists on flymine that share items with your list.
Save a list made by removing those items from the public set.
Delete one of your new lists.

Queries and Results

Identifier resolution is a useful tool, but imprecise - it accepts a wide range of input and seeks only to find items of a certain type. More specific queries can be run over all the data in an InterMine instance, and the results can be inspected, revised and exported. Most instances of InterMine have a wide range of commonly used queries predefined and able to be used with your input, known as templates:



Templates are predefined queries, each has a simple form and a description. Click on a template to run it, you can search for templates by keyword and filter them by category.

You are not logged in. Log in to mark items as favourites.

☐ Pathway --> Genes.
For a specified KEGG, REACTOME or FlyReactome pathway, list all the genes that are involved for a particular organism

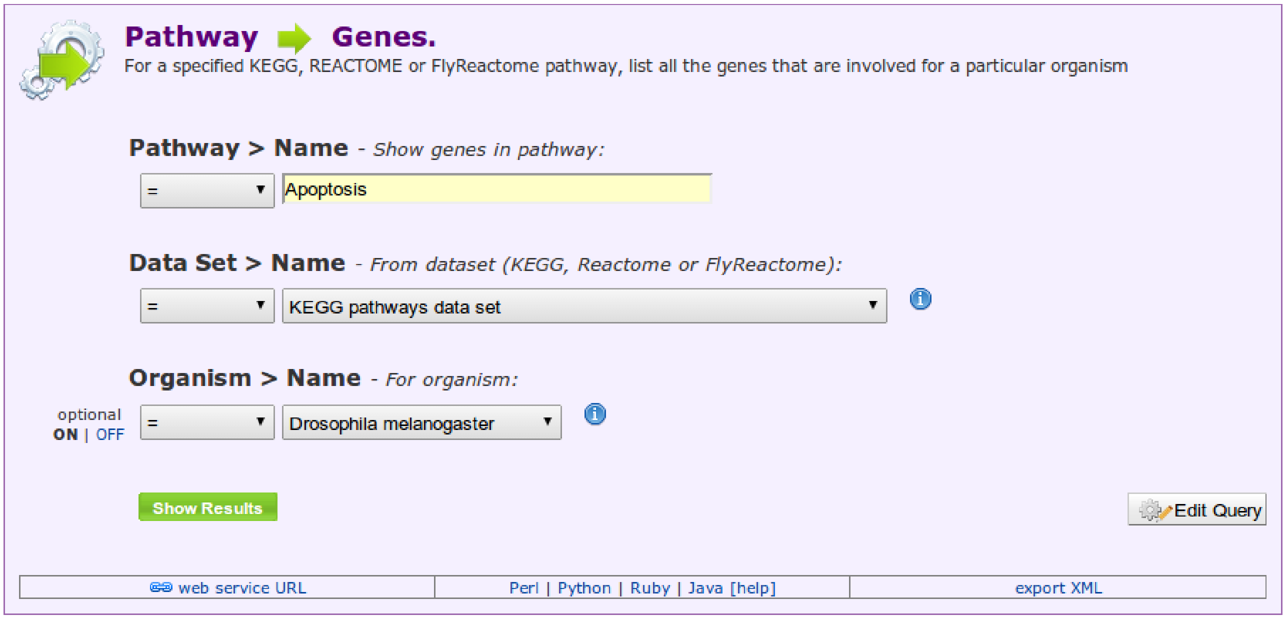
☐ All genes in organism --> All chromosomal locations.
For a specified organism, show the chromosomal location and sequence length for all genes.

☐ Gene --> Proteins.
Show the proteins for a particular gene

☐ Gene --> GO terms.
Search for GO annotations for a particular gene.

fig 4: http://www.flymine.org/query/templates.do

These queries have a small set of configurable parameters and can be run to produce results:



For a specified KEGG, REACTOME or FlyReactome pathway, list all the genes that are involved for a particular organism

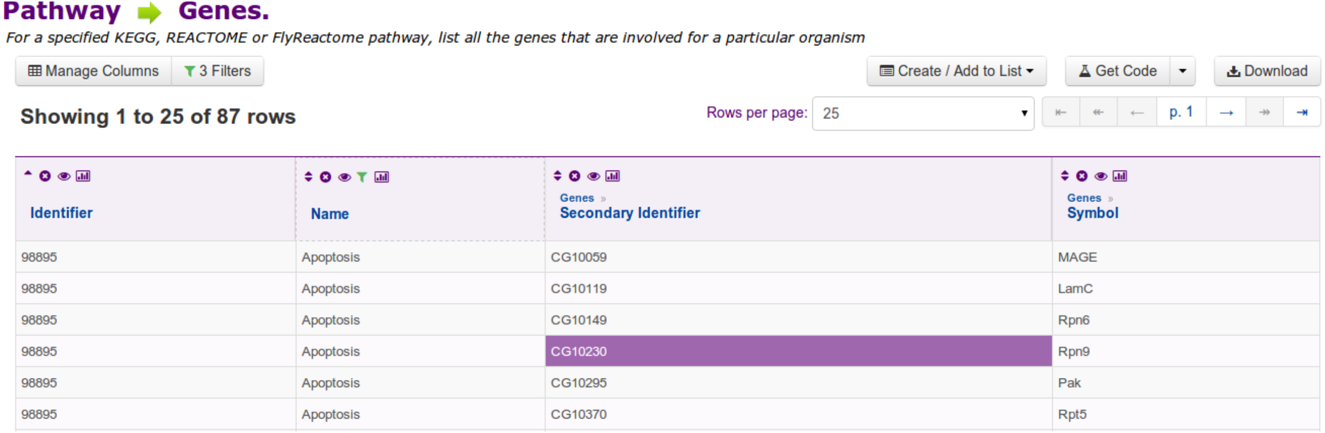
Pathway > Name - *Show genes in pathway:*
= Apoptosis

Data Set > Name - *From dataset (KEGG, Reactome or FlyReactome):*
= KEGG pathways data set

Organism > Name - *For organism:*
= Drosophila melanogaster

fig 5: http://www.flymine.org/query/template.do?name=Pathway_Genes

The results are presented in a table:


Pathway ➡ Genes.

For a specified KEGG, REACTOME or FlyReactome pathway, list all the genes that are involved for a particular organism

Showing 1 to 25 of 87 rows

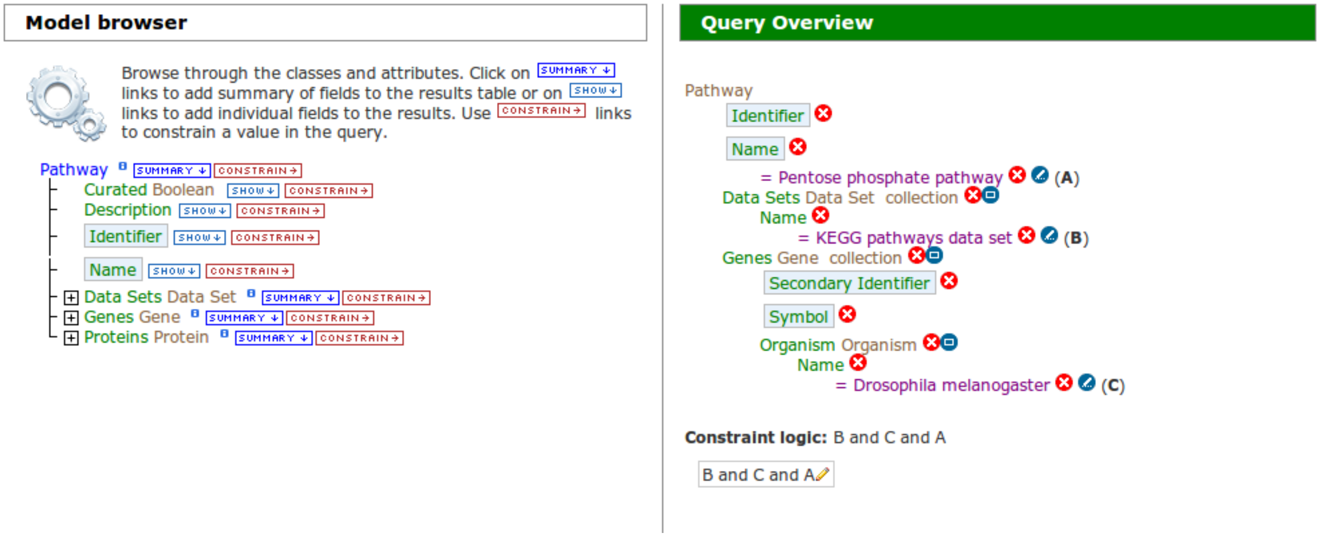
Rows per page: 25

Identifier	Name	Genes » Secondary Identifier	Genes » Symbol
98895	Apoptosis	CG10059	MAGE
98895	Apoptosis	CG10119	LamC
98895	Apoptosis	CG10149	Rpn6
98895	Apoptosis	CG10230	Rpn9
98895	Apoptosis	CG10295	Pak
98895	Apoptosis	CG10370	Rpt5

This table allows the user to:

- page through results
- sort the rows
- add and remove columns
- add and remove filters
- inspect individual items
- download the results (or just a column of the results)
- get a URI, script or XML that represents this query
- create a list from the results

Templates are just a convenient short-cut for the underlying query (the particular combination of columns and filters that makes up this view over the data). You can see this by editing a template:


Model browser

Browse through the classes and attributes. Click on SUMMARY links to add summary of fields to the results table or on SHOW links to add individual fields to the results. Use CONSTRAIN links to constrain a value in the query.

Pathway SUMMARY CONSTRAIN
Curated Boolean SHOW CONSTRAIN
Description SHOW CONSTRAIN
Identifier SHOW CONSTRAIN
Name SHOW CONSTRAIN
Data Sets Data Set SUMMARY CONSTRAIN
Genes Gene SUMMARY CONSTRAIN
Proteins Protein SUMMARY CONSTRAIN

Query Overview

Pathway
Identifier
Name
= Pentose phosphate pathway (A)
Data Sets Data Set collection
Name
= KEGG pathways data set (B)
Genes Gene collection
Secondary Identifier
Symbol
Organism Organism
Name
= Drosophila melanogaster (C)

Constraint logic: B and C and A

B and C and A

In time, if your needs are specific, you may wish to build up your own library of saved queries and templates - any user can create and edit queries.

These features can also be accessed through the webservices. First, looking at templates:

Get the list of available templates:

```
names = service.templates.keys()
```

Get a specific template, and run it:

```
q = service.get_template(“Pathway_Genes”)
filter = {“value”: “Apoptosis”}
symbols = [g.symbol for g in q.results(A = filter)]
print(symbols)
```

It can be much cleaner in code to use the raw underlying query (templates are very much a UI convenience). The full query API is powerful and complex, but thankfully queries can be generated in the web interface, eg:

```
query = service.new_query("Pathway")
query.add_view("identifier", "name", "genes.secondaryIdentifier",
"genes.symbol")
query.add_constraint("name", "=", "Pentose phosphate pathway", "A")
query.add_constraint("dataSets.name", "=", "KEGG pathways data set",
"B")
query.add_constraint("genes.organism.name", "=", "Drosophila
melanogaster", "C")

for row in query.rows():
    print row["identifier"], row["name"] \
          row["genes.secondaryIdentifier"], row["genes.symbol"]
```

TASKS

1. Use templates to inspect the interactions and orthologues of a gene or a set of genes.
  - Which genes have a particularly large number of interactions with your set of genes? What if you are only interested in “suppression” interactions?
  - Which orthologue data set has the most orthologues for genes in your list? What if you only look at the orthologues in mosquito?

2. Add a column to your results:
  - An attribute of one of the items in the table
  - An attribute of an item not in the table (a new connection).

3. Filter the table by:
  - Using the column-summaries
  - By using the filter dialogue

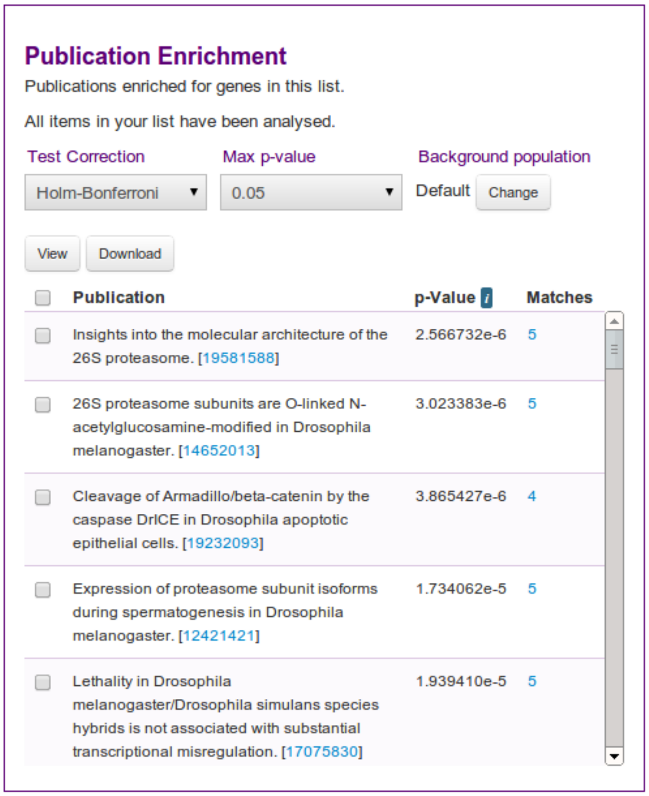
4. Export the results:
  - As a spreadsheet file (TSV, CSV)
  - As a machine readable file (XML, JSON)
  - To Galaxy or Genomespace

5. Create a new list of genes from these results (ie. the list of genes your genes interact with) - see: http://pythonhosted.org/intermine/intermine.lists.listmanager.ListManager-class.html#create_list.

6. Run one of your queries using python, and print out the results. (If you have time, try adding a constraint, as in 1, a column as in 2, or creating a list from the results, as in 3), entirely using the web services.

Analysing Lists

Lists can be using another specialised type of query: enrichment queries. These are performed automatically on list analysis pages:



Publication Enrichment

Publications enriched for genes in this list.

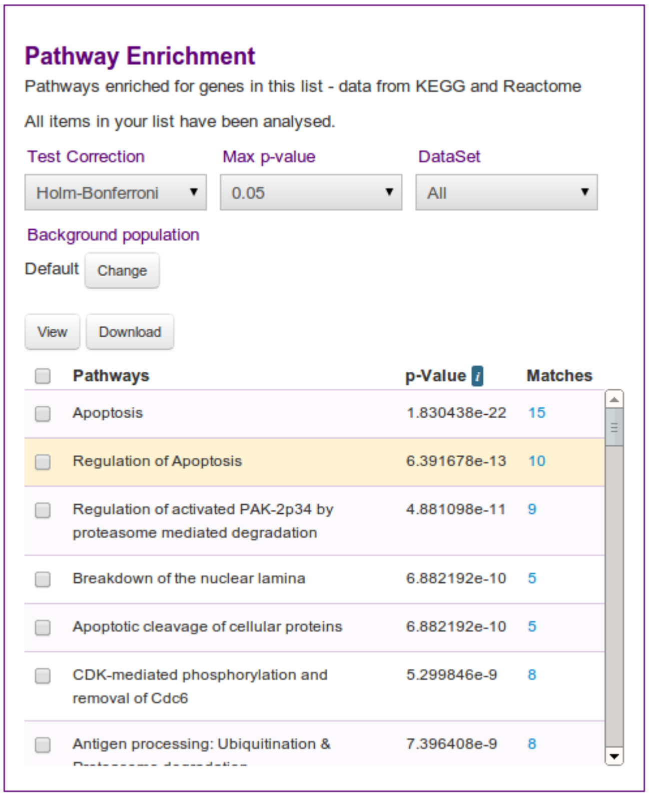
All items in your list have been analysed.

Test Correction: Holm-Bonferroni

Max p-value: 0.05

Background population: Default

☐	Publication	p-Value	Matches
☐	Insights into the molecular architecture of the 26S proteasome. [19581588]	2.566732e-6	5
☐	26S proteasome subunits are O-linked N-acetylglucosamine-modified in Drosophila melanogaster. [14652013]	3.023383e-6	5
☐	Cleavage of Armadillo/beta-catenin by the caspase DrICE in Drosophila apoptotic epithelial cells. [19232093]	3.865427e-6	4
☐	Expression of proteasome subunit isoforms during spermatogenesis in Drosophila melanogaster. [12421421]	1.734062e-5	5
☐	Lethality in Drosophila melanogaster/Drosophila simulans species hybrids is not associated with substantial transcriptional misregulation. [17075830]	1.939410e-5	5



Pathway Enrichment

Pathways enriched for genes in this list - data from KEGG and Reactome

All items in your list have been analysed.

Test Correction: Holm-Bonferroni

Max p-value: 0.05

DataSet: All

Background population: Default

☐	Pathways	p-Value	Matches
☐	Apoptosis	1.830438e-22	15
☐	Regulation of Apoptosis	6.391678e-13	10
☐	Regulation of activated PAK-2p34 by proteasome mediated degradation	4.881098e-11	9
☐	Breakdown of the nuclear lamina	6.882192e-10	5
☐	Apoptotic cleavage of cellular proteins	6.882192e-10	5
☐	CDK-mediated phosphorylation and removal of Cdc6	5.299846e-9	8
☐	Antigen processing: Ubiquitination &	7.396408e-9	8

These queries attempt to assess how significant it is that items in your list are associated with certain other items in the data warehouse (see http://intermine.readthedocs.org/en/latest/embedding/list-widgets/enrichment-widgets for method and discussion). You can see which items in your list matched, and also inspect the related item (such as visiting the NCBI page for the publication), as well as modifying the results by adjusting the correction algorithm and p-value cut-off threshold.

Because the enrichment results are based on a statistical calculation, the results can differ depending on the background population and whether or not certain normalisations are made (such as for gene length).

These queries can also be run using webservices:

```
list = service.get_list(“my list name”)
for item in list.calculate_enrichment(“pathway_enrichment”):
  print item.identifier, item.p_value
```

Information on the widgets themselves is also available:

```
for w in service.widgets.values():
  print("{name} - {description}".format(**w))
```

TASKS

1. Which publications/GO Terms are enriched for your list?
2. How to the results change when the following parameters are adjusted:
    1. Correction algorithm
    2. Background population
    3. Normalisation for gene length
3. Download the results as a spreadsheet file (TSV)
4. Run an enrichment query using the web services.

Useful Links and Addresses

www.intermine.org Web UI Documentation and handbook
iodocs.labs.intermine.org/flymine HTTP API documentation
pythonhosted.org/intermine Python API documentation

dev@intermine.org The InterMine users mailing list (for technical queries)
help@intermine.org For general usage questions.

Each mine will also have its own help desk.