Hands-on session SABIO-RK

de.NBI Training - April 26th, 2018 Magdeburg

SABIO-RK user interface: http://sabio.h-its.org

1. Search for data about human liver and compare the number of results from free text (2976) and advanced search (tissue: liver, organism: Homo sapiens) (2557)
    a. Filter for wildtype information only (2282)

2. Search for data in mammalian blood cells by using ontology based search (tissue: blood (BTO), organism: mammalia (NCBI)) (1154)
    a. Check the number of organisms (8), the different blood cell lines and kinetic parameters in the *Visual Search*
    b. Filter for entries containing a kinetic rate equation (459)
    c. Filter for data at 37°C and pH 6-8 (115)

3. Search for pathway glycolysis (“glycolysis/gluconeogenesis”) in rat heart (22)
    a. Check the different reactions catalyzed by the enzyme in *Reaction View* (6)

4. Search for enzyme hexokinase (746) (EC 2.7.1.1) (750) and restrict the search for data in Saccharomyces cerevisiae (Baker’s yeast) (72) using *Visual Search*

5. Search the UniprotKB Database (https://www.uniprot.org) for hexokinase (EC 2.7.1.1) in Saccharomyces cerevisiae
    a. Find the isozyme which has most kinetic data in SABIO-RK (use the cross-references to SABIO-RK) (P04807 – 39 entries, P04806 – 13 entries)

6. Search for reactions which are inhibited by ATP (155) and for which a Ki value is available (100)
    a. Check the *Reaction View*, find the corresponding KEGG ReactionID for the first reaction (SABIO ReactionID: 141) (R00214) and click on the link to KEGG to view the reaction in KEGG
    b. Use the SABIO-RK link on the KEGG page to go back and see how many entries are available for the reaction in general (150)

7. Search for model from Pritchard (use free text and filter for BioModel) (19)

Export functionality:

8. Export the results (all entries) from query in 7.
    a. as SBML file (store on your local machine)
    b. as table including parameters and rate equations (store on your local machine)

<u>SABIO-RK web services:</u> **http://sabio.h-its.org/layouts/content/webservices.gsp**

9. Use the web services to define the query in 3. (create a SBML file)

   http://sabiork.h-its.org/sabioRestWebServices/searchKineticLaws/sbml?q=Tissue:”heart” AND Organism:”Rattus norvegicus” AND Pathway:”glycolysis/gluconeogenesis”

10. Use the web services to define a query for hexokinase (EC 2.7.1.1) in Saccharomyces cerevisiae (create a list of entryIDs)

    http://sabiork.h-its.org/sabioRestWebServices/searchKineticLaws/entryIDs?q=Enzymename:”hexokinase” AND Organism:”Saccharomyces cerevisiae”

    http://sabiork.h-its.org/sabioRestWebServices/searchKineticLaws/entryIDs?q=ECNumber:”2.7.1.1” AND Organism:”Saccharomyces cerevisiae”

11. Use Python script 1 to define the query in 3. (create a table containing the columns EntryID, Organism, UniprotID, ECNumber, Parameter, Temperature, and pH)

```
import requests

ENTRYID_QUERY_URL = 'http://sabiork.h-its.org/sabioRestWebServices/searchKineticLaws/entryIDs'
PARAM_QUERY_URL = 'http://sabiork.h-its.org/entry/exportToExcelCustomizable'


# ask SABIO-RK for all EntryIDs matching a query
query_dict = {"Organism":'"Rattus norvegicus"',"Tissue":"heart","Pathway":"glycolysis/gluconeogenesis"}
query_string = ' AND '.join(['%s:%s' % (k,v) for k,v in query_dict.items()])
query = {'format':'txt', 'q':query_string}


# make GET request
request = requests.get(ENTRYID_QUERY_URL, params = query)
request.raise_for_status() # raise if 404 error


# each entry is reported on a new line
entryIDs = [int(x) for x in request.text.strip().split('\n')]
print('%d matching entries found.' % len(entryIDs))


# encode next request, for parameter data given entry IDs
data_field = {'entryIDs[]': entryIDs}
query = {'format':'tsv', 'fields[]':['EntryID', 'Organism', 'UniprotID','ECNumber', 'Parameter','Temperature','pH']}


# make POST request
request = requests.post(PARAM_QUERY_URL, params=query, data=data_field)
request.raise_for_status()


# results
print(request.text)
```