Extending Google Refine for VIVO

Dr. Curtis L. Cole, Dan Dickinson, Kenneth Lee, Eliza Chan

Weill Cornell Medical College

Background

Google Refine (previously Freebase Gridworks) is a freely available open source software package for manipulating datasets. Google Refine strives to provide an easy-to-use toolset to assist in the extraction, review, transformation, and export of datasets. It is advertised as a "power tool for messy data".

One of Google Refine's unique features is tight integration with the Freebase database. Freebase is a community built repository of freely available structured data. Similarly to how VIVO strives to make assertions about faculty and research at a given institution, Freebase provides a graph that connects over twelve million entities on topics ranging from political parties and diseases to amusement parks and types of tea.

Google Refine's integration with Freebase allows users to "reconcile" data in a grid format against entities found within the Freebase graph. Users may specify an object class and additional fields to help identify a particular cell as being an entity within the Freebase graph. Google Refine communicates via an interface to Freebase to either match the cells to entities or receive a set of potential matches. Cells that are matched change to linked data, while cells with multiple potential matches can be reviewed by a user and matched by hand. After a column has been reconciled, Google Refine allows a user to add additional columns based on relationships or properties belonging to the linked entity. For example, a column of reconciled countries can have the gross domestic product or capital added easily into the same data table.

Google Refine also allows a dataset to be aligned to the Freebase schema, to convert grid data into graph data, and then export it into a triple format importable by Freebase. Because of this type integration, Google Refine is a viable, attractive tool for working with datasets that need to be converted for use within semantic web applications.

Goals

Weill Cornell Medical College proposes to enhance both Google Refine and VIVO to allow integration between the two systems, similar to the existing integration with Freebase. Specifically, we intend:

1. To write an extension for Google Refine that
    a. will provide a series of functions to align a dataset to the VIVO core ontologies, assert relationships to help transform data from a grid to a semantic web graph, and export the data into a standardized triple format (such as N3) so that it can be easily ingested by a VIVO installation;
    b. will allow data to be imported into a dataset by referencing VIVO linked data, similar to the "Add columns from Freebase..." function.
2. To implement the Reconciliation Service API within VIVO, providing an interface for Google Refine to reconcile data against a specified VIVO instance.

Documentation

Section I: VIVO servlet - Reconciliation service

The VIVO reconciliation service is a Java HttpServlet that parses requests from Google Refine and returns query results back to Google Refine. With each request, it also returns a callback value in order to correctly map the results to the request that it receives.

To send a request from Google Refine, click on "Reconcile -> Start reconciling" to open the reconcile dialog. Then click on the "Add Standard Service" button to add the VIVO reconciliation service. The URL to VIVO's reconciliation service should have the word "reconcile" at the end, similar to this: http://vivo.mydomain.edu/reconcile.

The VIVO reconciliation service responds to the following requests from Google Refine:

1. Single query
2. Typed single query
3. Typed single query with properties
4. Multiple queries
5. No query

Reconciliation Service API Test Dashboard

The Reconciliation Service API Test Dashboard provides an insight on the backend of the reconciliation service.

In their examples, the JSON queries that appear after clicking on the query text links illustrate the kind of queries that Google Refine's reconciliation service would send to its target reconcile service. Upon receiving the queries, the reconcile service would parse the queries and send the results back to Google Refine.

Here is an example of a query being sent to VIVO from Google Refine:

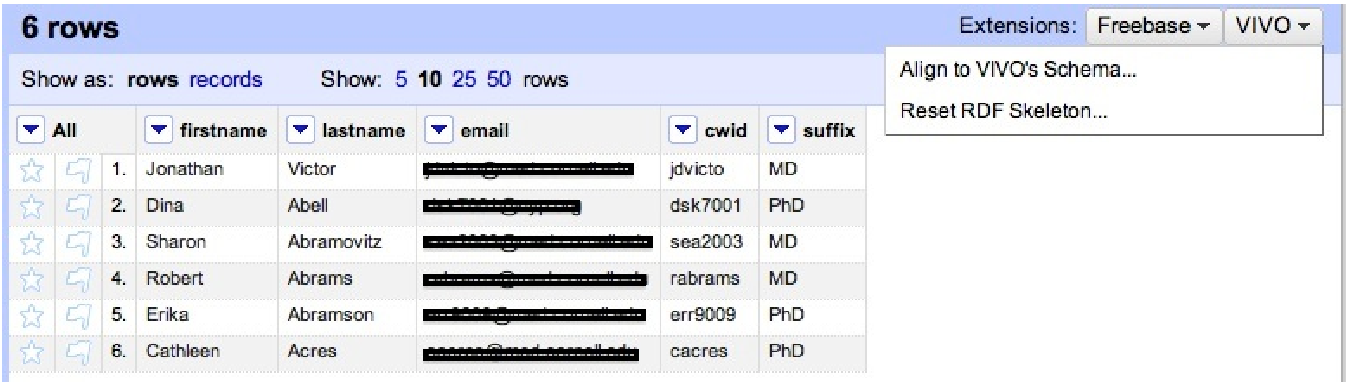
{"q0":{"query":"Cathleen","type":"http://xmlns.com/foaf/0.1/Person","type_strict":"should"}}

Here is an example of the results returned from VIVO to Google Refine:

```
{
  "q0": {
    "result": [
      {
        "id": "http://vivo.med.cornell.edu/individual/cwid-clr9010",
        "name": "Raggio, Cathleen L.",
        "score": 0.15898549556732178,
        "match": "false",
        "type": [
          {
            "id": "http://xmlns.com/foaf/0.1/Agent",
            "name": "Agent"
          },
          {
            "id": "http://vivoweb.org/ontology/core#FacultyMember",
            "name": "core:FacultyMember"
          },
          {
            "id": "http://xmlns.com/foaf/0.1/Person",
            "name": "Person"
          }
        ]
      },
      {
        "id": "http://vivo.med.cornell.edu/individual/cwid-cacres",
        "name": "Acres, Cathleen",
        "score": 0.15898549556732178,
        "match": "false",
        "type": [
          {
            "id": "http://xmlns.com/foaf/0.1/Agent",
            "name": "Agent"
          },
          {
            "id": "http://vivoweb.org/ontology/core#NonFacultyAcademic",
            "name": "core:NonFacultyAcademic"
          },
          {
            "id": "http://xmlns.com/foaf/0.1/Person",
            "name": "Person"
          }
        ]
      },
      {
        "id": "http://vivo.med.cornell.edu/individual/cwid-cal9048",
        "name": "London, Cathleen",
        "score": 0.15898549556732178,
        "match": "false",
        "type": [
          {
            "id": "http://xmlns.com/foaf/0.1/Agent",
            "name": "Agent"
          },
          {
            "id": "http://vivoweb.org/ontology/core#FacultyMember",
            "name": "core:FacultyMember"
          },
          {
            "id": "http://xmlns.com/foaf/0.1/Person",
            "name": "Person"
          }
        ]
      }
    ]
  }
}
```

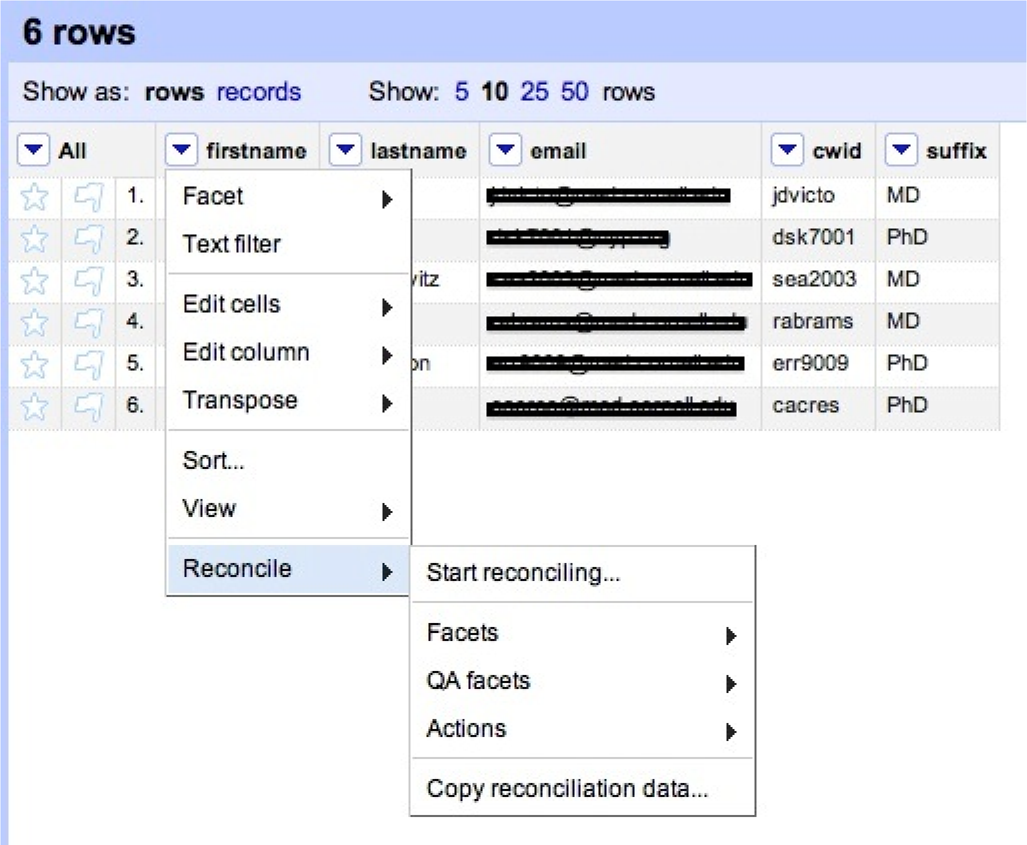
VIVO Reconciliation Service Work Case Scenario

1) In this example, the user wanted to find out if any of the names listed could be reconciled (or matched) with those in a VIVO instance



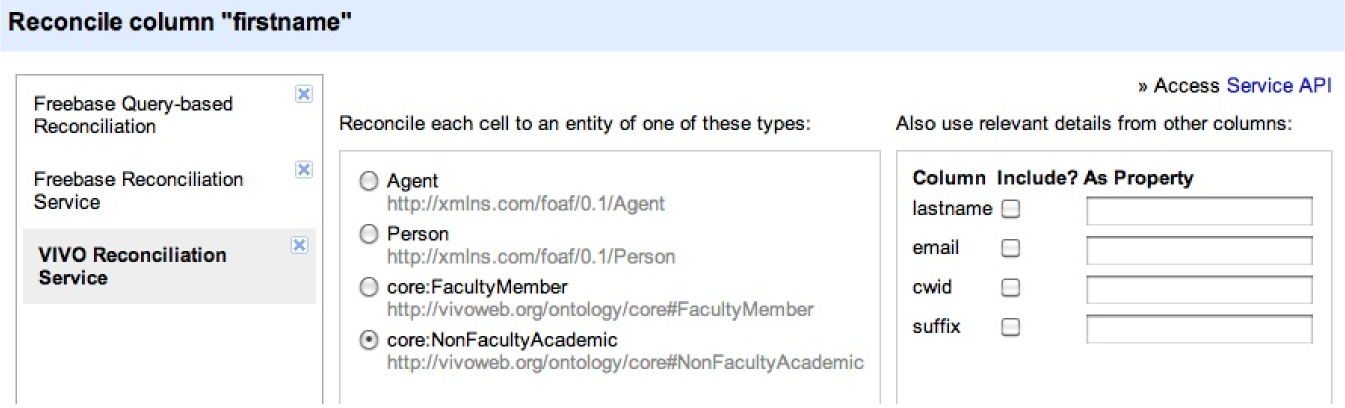
All	firstname	lastname	email	cwid	suffix
1.	Jonathan	Victor	[redacted]	jdvicto	MD
2.	Dina	Abell	[redacted]	dsk7001	PhD
3.	Sharon	Abramovitz	[redacted]	sea2003	MD
4.	Robert	Abrams	[redacted]	rabrams	MD
5.	Erika	Abramson	[redacted]	err9009	PhD
6.	Cathleen	Acres	[redacted]	cacres	PhD

2) To open the reconcile dialog, select the menu item "Start reconciling ..."



All	firstname	lastname	email	cwid	suffix
1.			[redacted]	jdvicto	MD
2.			[redacted]	dsk7001	PhD
3.		itz	[redacted]	sea2003	MD
4.			[redacted]	rabrams	MD
5.		on	[redacted]	err9009	PhD
6.			[redacted]	cacres	PhD

3) When the reconcile dialog opens, click on the button "Add Standard Service ..." (not shown in diagram) to add the service's URL. The URL is typically http://your-vivo-site/reconcile. After adding the service, click on the service item and the types that match with the column would appear (see diagram below). In this case, the type core:NonFacultyAcademic was selected.



Reconcile column "firstname"

» Access Service API

- Freebase Query-based Reconciliation ☒
- Freebase Reconciliation Service ☒
- **VIVO Reconciliation Service** ☒

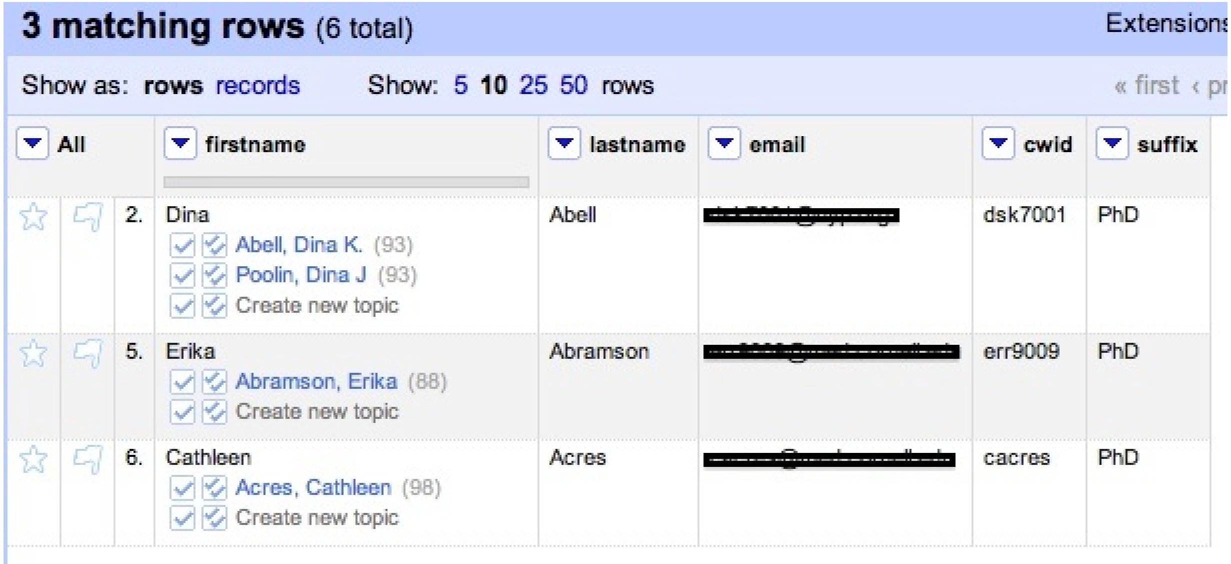
Reconcile each cell to an entity of one of these types:

○ Agent
http://xmlns.com/foaf/0.1/Agent
○ Person
http://xmlns.com/foaf/0.1/Person
○ core:FacultyMember
http://vivoweb.org/ontology/core#FacultyMember
● core:NonFacultyAcademic
http://vivoweb.org/ontology/core#NonFacultyAcademic

Also use relevant details from other columns:

Column	Include?	As Property
lastname	□	
email	□	
cwid	□	
suffix	□	

4) In this example, matches were found for three names, and the suggested matches were imported from the VIVO instance. To select the matched names, click on the check boxes next to the names.



3 matching rows (6 total)

Extensions

Show as: **rows** records   Show: 5 **10** 25 50 rows

« first ‹ pr

All			firstname	lastname	email	cwid	suffix
☆		2.	Dina ☑ ☑ Abell, Dina K. (93) ☑ ☑ Poolin, Dina J (93) ☑ ☑ Create new topic	Abell	[redacted]	dsk7001	PhD
☆		5.	Erika ☑ ☑ Abramson, Erika (88) ☑ ☑ Create new topic	Abramson	[redacted]	err9009	PhD
☆		6.	Cathleen ☑ ☑ Acres, Cathleen (98) ☑ ☑ Create new topic	Acres	[redacted]	cacres	PhD

Summary

The VIVO reconciliation service is a Java HttpServlet that responds to requests from Google Refine, and returns results in JSON format in accordance to Google Refine's Reconciliation Service API.

Section II: VIVO servlet - Property listings

The VIVO property listings service is a Java HttpServlet that receives requests for data properties for given classes from Google Refine and returns query results back to Google Refine.

The path to the property listings service is /get_properties_of_type, and the full path is similar to this: http://vivo.mydomain.edu/get_properties_of_type.

Two parameters are passed to the servlet, type and callback. For instance, to request for data properties associated with the class foaf:Person, the query string would be similar to the following:

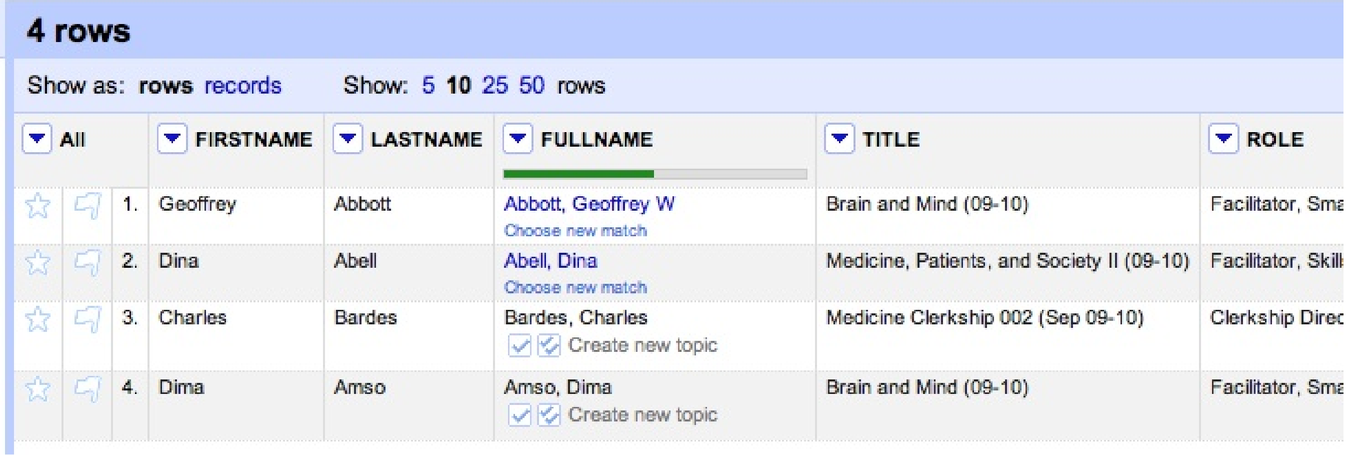
http://vivo.mydomain.edu/get_properties_of_type?type=http%3A%2F%2Fxmlns.com%2Ffoaf%2F0.1%2FPerson&callback=jsonp1313077460641

See below an example of the response in json format:

```
jsonp1313077460641 (
  {
    "properties":
    [
      {
        "id":"http://vivoweb.org/ontology/core#faxNumber",
        "schema":{"id":"http://xmlns.com/foaf/0.1/Agent","alias":[],"name":"Agent"},
        "alias":[],"name":"fax",
        "expects":{"id":"http://vivoweb.org/ontology/core#faxNumber","alias":[],"name":"fax"}
      },
      {
        "id":"http://vivoweb.org/ontology/core#overview",
        "schema":{"id":"http://xmlns.com/foaf/0.1/Agent","alias":[],"name":"Agent"},
        "alias":[],"name":"overview",
        "expects":{"id":"http://vivoweb.org/ontology/core#overview","alias":[],"name":"overview"}
      }
    ]
  }
)
```

VIVO Property Listings Work Case Scenario

1) In this example, names have been reconciled with a VIVO instance.

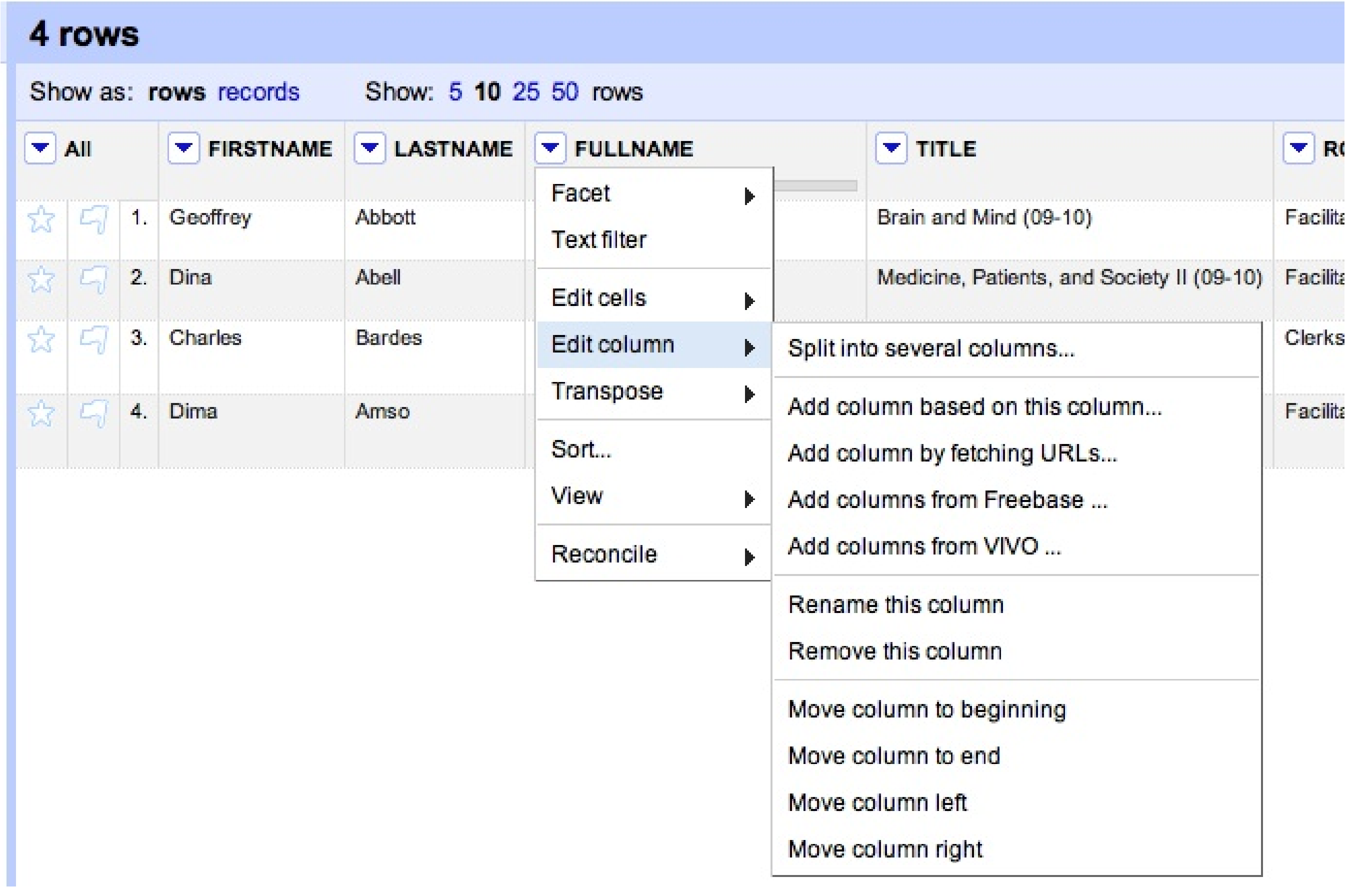


4 rows

Show as: **rows** records   Show: 5 **10** 25 50 rows

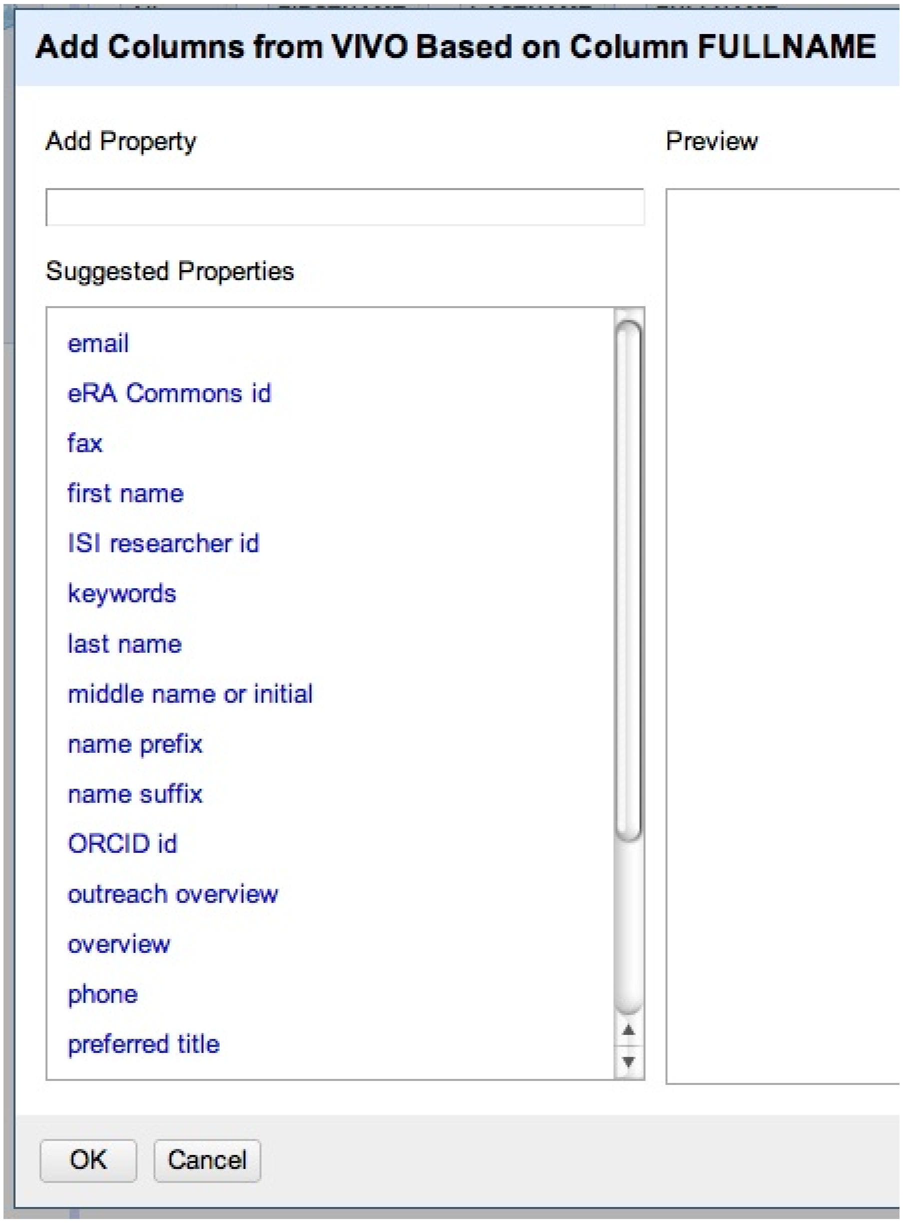
All	FIRSTNAME	LASTNAME	FULLNAME	TITLE	ROLE
1.	Geoffrey	Abbott	Abbott, Geoffrey W Choose new match	Brain and Mind (09-10)	Facilitator, Sma
2.	Dina	Abell	Abell, Dina Choose new match	Medicine, Patients, and Society II (09-10)	Facilitator, Skill
3.	Charles	Bardes	Bardes, Charles ✓ ☑ Create new topic	Medicine Clerkship 002 (Sep 09-10)	Clerkship Direc
4.	Dima	Amso	Amso, Dima ✓ ☑ Create new topic	Brain and Mind (09-10)	Facilitator, Sma

2) Click on the menu item "Add columns from VIVO" to search the VIVO instance for data properties that are associated with the reconciled names.



All			FIRSTNAME	LASTNAME	FULLNAME	TITLE	RO
☆		1.	Geoffrey	Abbott		Brain and Mind (09-10)	Facilita
☆		2.	Dina	Abell		Medicine, Patients, and Society II (09-10)	Facilita
☆		3.	Charles	Bardes			Clerks
☆		4.	Dima	Amso			Facilita

3) In this case, the class foaf:Person was used for the reconciliation, and its associated data properties are listed in the Suggested Properties box.



Add Columns from VIVO Based on Column FULLNAME

Add Property

Preview

Suggested Properties

email
eRA Commons id
fax
first name
ISI researcher id
keywords
last name
middle name or initial
name prefix
name suffix
ORCID id
outreach overview
overview
phone
preferred title

Section III: VIVO servlet - MQL read

The VIVO MQL read service is a Java HttpServlet that receives requests for values of data properties from Google Refine and returns query results back to Google Refine.

The path to the MQL read service is /grefineMqlread, and the full path is similar to this: http://vivo.mydomain.edu/grefineMqlread.

The requests and response are in json format that comply with the mqlread API.

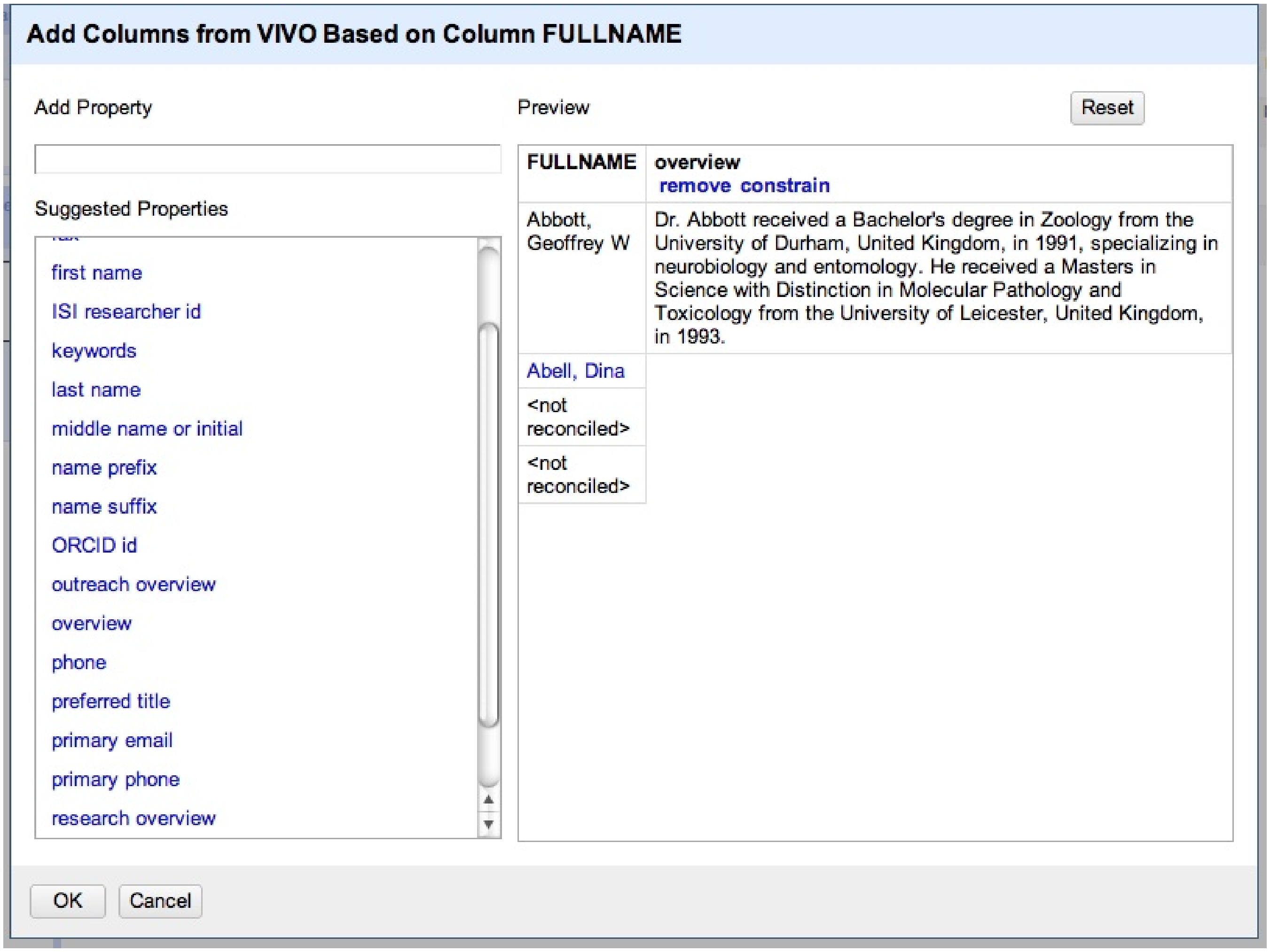
See below an example of a query and a response:

```
{"query":
  [
    {
      "id":null,
      "id|=":["http://vivo.med.cornell.edu/individual/n3704","http://vivo.med.cornell.edu/individual/n3526"],
      "http://xmlns.com/foaf/0.1/firstName":  [{"optional":true,"limit":10,"name":null,"id":null,"type":[]}]
    }
  ]
}

{"result":
  [
    {"id":"http://vivo.med.cornell.edu/individual/n3704","http://xmlns.com/foaf/0.1/firstName":["Geoffrey"]},
    {"id":"http://vivo.med.cornell.edu/individual/n3526","http://xmlns.com/foaf/0.1/firstName":["Dina"]}
  ]
}
```

VIVO MQL Read Service Work Case Scenario

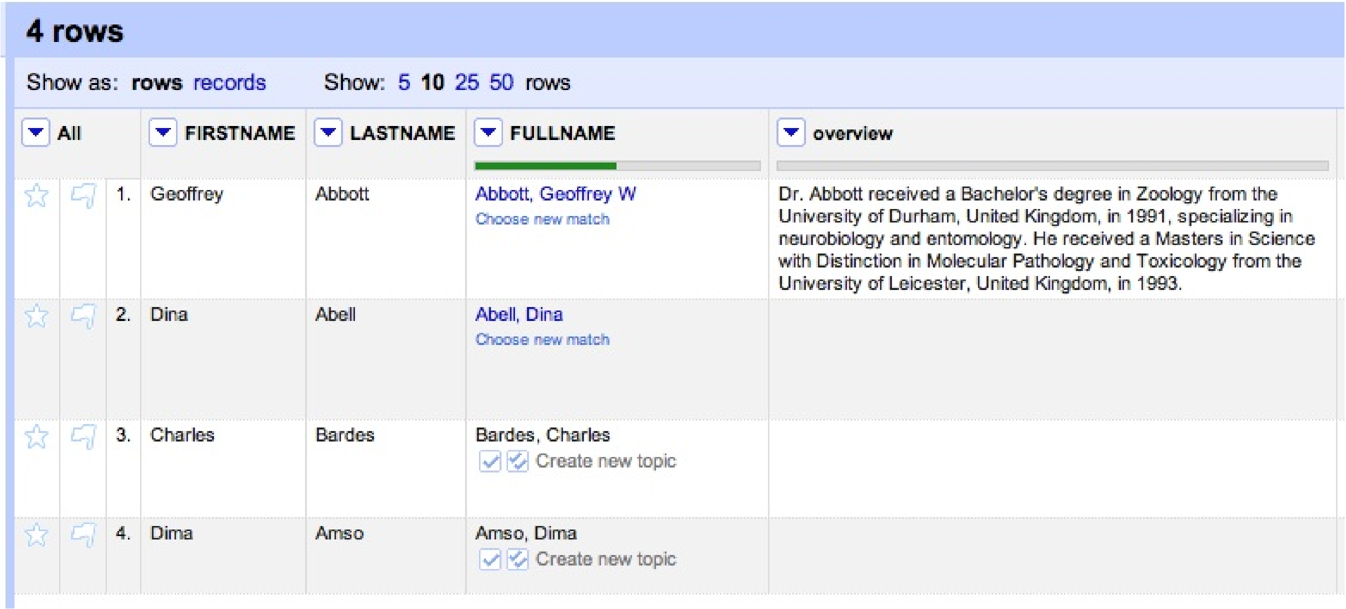
1) In this example, the overview value associated with the person was displayed in the Preview panel



Add Columns from VIVO Based on Column FULLNAME

FULLNAME	overview remove constrain
Abbott, Geoffrey W	Dr. Abbott received a Bachelor's degree in Zoology from the University of Durham, United Kingdom, in 1991, specializing in neurobiology and entomology. He received a Masters in Science with Distinction in Molecular Pathology and Toxicology from the University of Leicester, United Kingdom, in 1993.
Abell, Dina	
<not reconciled>	
<not reconciled>	

2) An overview column was added to Google Refine



4 rows

Show as: **rows** records Show: 5 **10** 25 50 rows

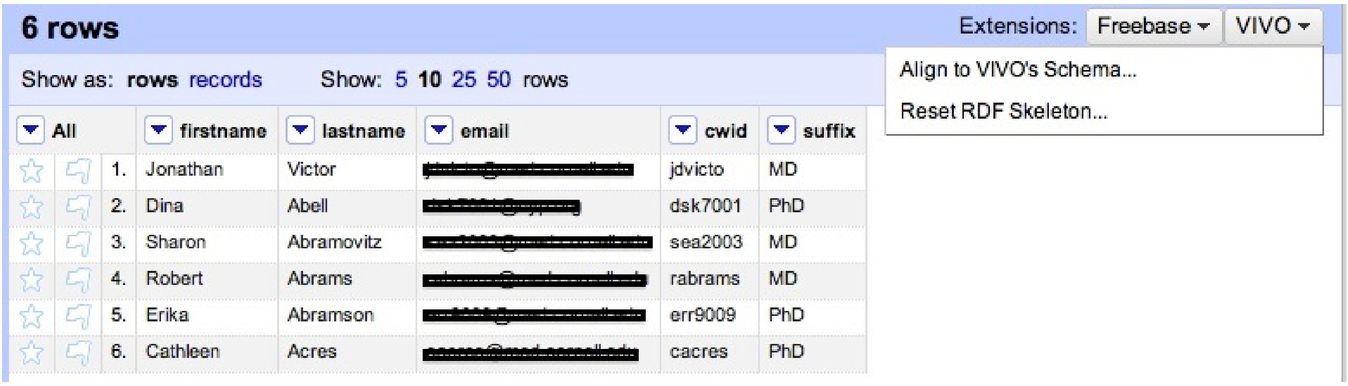
All			FIRSTNAME	LASTNAME	FULLNAME	overview
☆		1.	Geoffrey	Abbott	Abbott, Geoffrey W Choose new match	Dr. Abbott received a Bachelor's degree in Zoology from the University of Durham, United Kingdom, in 1991, specializing in neurobiology and entomology. He received a Masters in Science with Distinction in Molecular Pathology and Toxicology from the University of Leicester, United Kingdom, in 1993.
☆		2.	Dina	Abell	Abell, Dina Choose new match	
☆		3.	Charles	Bardes	Bardes, Charles ☑ ☑ Create new topic	
☆		4.	Dima	Amso	Amso, Dima ☑ ☑ Create new topic	

Section IV: Google Refine VIVO extension - Align to VIVO's schema

Google Refine has a documentation page about writing an extension. In addition, the "RDF Extension" project provides an excellent demonstration of a functional Google Refine extension.

The following work case scenario demonstrates how grid-shaped data can be aligned to VIVO's schema and transformed into RDF.

1) In this example, the VIVO instance was missing the email and suffix for the people listed. The goal was to align the missing data to the VIVO schema so that it could be imported to a VIVO instance. To open the alignment dialog, click on the menu item “Align to VIVO’s schema…”.



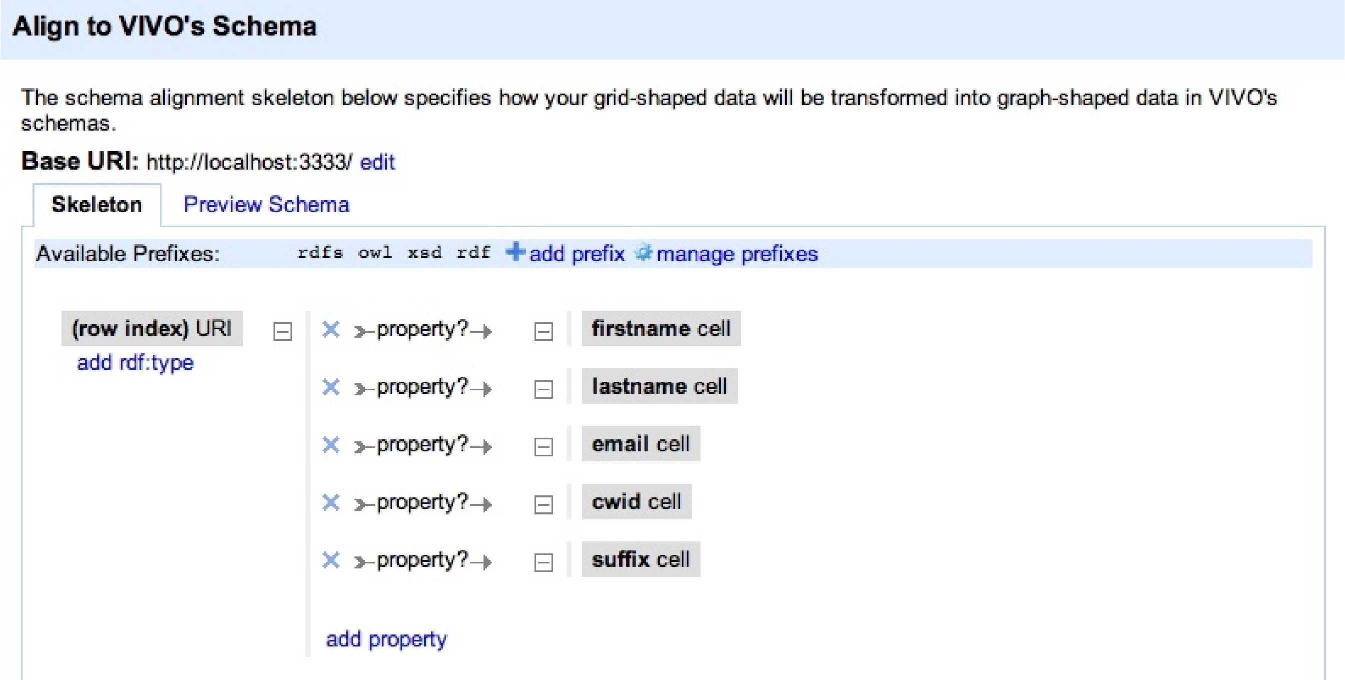
6 rows

Show as: **rows** records Show: 5 **10** 25 50 rows

All			firstname	lastname	email	cwid	suffix
☆		1.	Jonathan	Victor	[redacted]	jdvicto	MD
☆		2.	Dina	Abell	[redacted]	dsk7001	PhD
☆		3.	Sharon	Abramovitz	[redacted]	sea2003	MD
☆		4.	Robert	Abrams	[redacted]	rabrams	MD
☆		5.	Erika	Abramson	[redacted]	err9009	PhD
☆		6.	Cathleen	Acres	[redacted]	cacres	PhD

2) Align the properties to the VIVO schema

The initial dialog shows a “skeleton” that the user would need to specify which columns were to be aligned with which VIVO properties.



Align to VIVO's Schema

The schema alignment skeleton below specifies how your grid-shaped data will be transformed into graph-shaped data in VIVO's schemas.

Base URI: http://localhost:3333/ edit

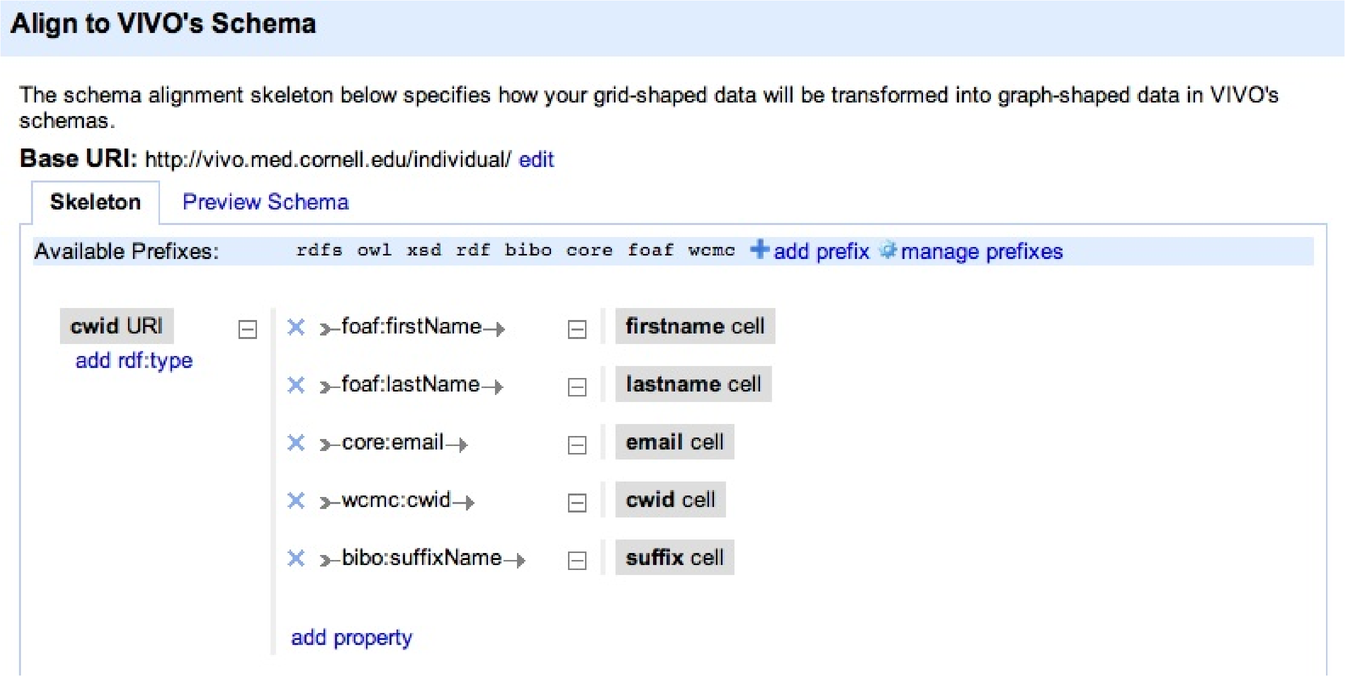
Available Prefixes: rdfs owl xsd rdf +add prefix manage prefixes

(row index) URI
add rdf:type

property? → firstname cell
property? → lastname cell
property? → email cell
property? → cwid cell
property? → suffix cell

add property

The following dialog shows the columns that had been aligned to the VIVO properties.



Align to VIVO's Schema

The schema alignment skeleton below specifies how your grid-shaped data will be transformed into graph-shaped data in VIVO's schemas.

Base URI: http://vivo.med.cornell.edu/individual/ edit

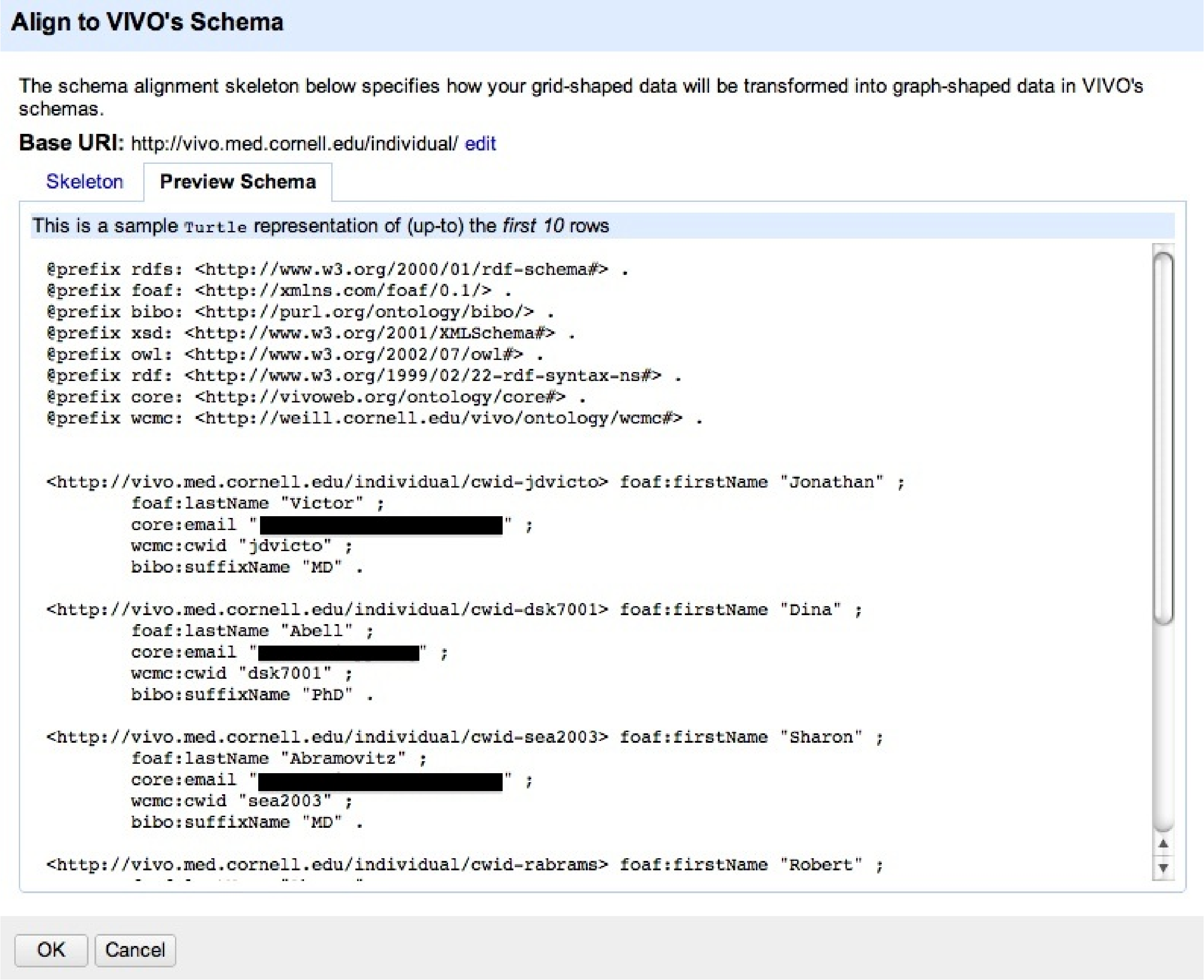
Available Prefixes: rdfs owl xsd rdf bibo core foaf wcmc +add prefix manage prefixes

cwid URI
add rdf:type

foaf:firstName → firstname cell
foaf:lastName → lastname cell
core:email → email cell
wcmc:cwid → cwid cell
bibo:suffixName → suffix cell

add property

3) After completion of the alignment, the Preview Schema tab showed the first 10 rows of the resulting RDF data.



Align to VIVO's Schema

The schema alignment skeleton below specifies how your grid-shaped data will be transformed into graph-shaped data in VIVO's schemas.

Base URI: http://vivo.med.cornell.edu/individual/ edit

This is a sample `Turtle` representation of (up-to) the *first 10* rows

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix core: <http://vivoweb.org/ontology/core#> .
@prefix wcmc: <http://weill.cornell.edu/vivo/ontology/wcmc#> .


<http://vivo.med.cornell.edu/individual/cwid-jdvicto> foaf:firstName "Jonathan" ;
        foaf:lastName "Victor" ;
        core:email "[redacted]" ;
        wcmc:cwid "jdvicto" ;
        bibo:suffixName "MD" .

<http://vivo.med.cornell.edu/individual/cwid-dsk7001> foaf:firstName "Dina" ;
        foaf:lastName "Abell" ;
        core:email "[redacted]" ;
        wcmc:cwid "dsk7001" ;
        bibo:suffixName "PhD" .

<http://vivo.med.cornell.edu/individual/cwid-sea2003> foaf:firstName "Sharon" ;
        foaf:lastName "Abramovitz" ;
        core:email "[redacted]" ;
        wcmc:cwid "sea2003" ;
        bibo:suffixName "MD" .

<http://vivo.med.cornell.edu/individual/cwid-rabrams> foaf:firstName "Robert" ;
```

Installation

Prerequisites

1. Make sure the following are installed on the desired machine:
2. Java (SE) 1.6 or higher, http://java.sun.com
3. Apache Ant 1.7 or higher, http://ant.apache.org
4. Download Google Refine version 2.5 from https://github.com/OpenRefine/OpenRefine/wiki/Downloads. If you have Subversion installed and access to a command line, you can use the following command to checkout the 2.5 release:
5. svn checkout http://google-refine.googlecode.com/svn/tags/2.5 grefine-googlecode-2.5
6. Download the VIVO plugin and Google Refine extension from http://sourceforge.net/projects/vivo/files/Utilities, or click on one of the following links:
   grefine-vivo-1.5.tar.gz or grefine-vivo-1.5.zip
7. Unpack and you will find the following files and folder: i) grefine-vivo-extension; ii) README
8. Download and install VIVO release version 1.5 from http://sourceforge.net/projects/vivo/files/, if you have not already done so.

Google Refine+VIVO Extension Installation

1. Copy `grefine-vivo-extension` to Google Refine's extension folder so that it becomes `/extension/grefine-vivo-extension`
2. Rename from `/extension/grefine-vivo-extension` to `/extension/vivo` (required)
3. Modify `/extensions/build.xml` to add vivo to build and clean:

```
<target name="build">
<echo message="Building extensions" />
<ant dir="sample/" target="build" />
<ant dir="jython/" target="build" />
<ant dir="freebase/" target="build" />
<ant dir="gdata/" target="build" />
<ant dir="vivo/" target="build" />
</target>
<target name="clean">
<echo message="cleaning extensions" />
<ant dir="sample/" target="clean" />
<ant dir="jython/" target="clean" />
<ant dir="freebase/" target="clean" />
<ant dir="gdata/" target="clean" />
<ant dir="vivo/" target="clean" />
</target>
```

4. Run `./refine clean`
5. Run `./refine build`
6. To start Google Refine, run `./refine`

Contributing to the project

We welcome anyone with a deep interest in Google Refine and/or VIVO to get involved.

The following mailing lists are highly recommended for support and discussions:

1. google-refine@googlegroups.com
2. vivo-dev-all@lists.sourceforge.net

References

1. The Google Refine VIVO extension was adapted and modified from the RDF extension project (http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension/).

Useful tips

1. Google Refine expression to get the resource URI from reconciled cells

e.g. cells.fullname.recon.best.id