The ENVISIONQuery package in BioConductor: Retrieving data through the Enfin-Encore annotation portal.

Alex Lisovich[1], Roger Day[1,2]

April 19, 2011

[1]Department of Biomedical Informatics, [2]Department of Biostatistics
University of Pittsburgh

1 Overview

The Enfin-Encore (EnCore) is the integration platform for the ENFIN European Network of Excellence [1], which provides a portal to various database resources with a special focus on systems biology (http://code.google.com/p/enfin-core/). EnCore is appealing to developers of bioinformatics applications, because of the scope and variety of EnCore's annotation resources, and because its collection of web services* utilizes a common standard format (EnXML: http://code.google.com/p/enfin-core/wiki/wp1_encore_enxml). Web services can communicate client applications using a variety of programming languages. Many bioinformaticians work in R, so an R-based solution is desirable.

The ENVISIONQuery package provides programmatic access to the EnCore web services in R. EnCore's capabilities evolve rapidly, so the architecture of ENVISIONQuery enables rapid integration of the new services when they appear.

*__IMPORTANT NOTE: The Java Web Services utilized by the package require Java 1.6 or higher to be installed, otherwise the package installation will fail giving the warning abot the wrong Java version.__

2 Design considerations

2.1 Motivating setting

Our group studied differential expression between endometrial cancer tissues with normal tissues. We received an Orbitrap proteomic mass spectrometry experiment that generated over 12,000 protein UNIPROT identifiers, as well as an Affymetrix U133 Plus 2 microarray experiment on the same samples. The analysis required mapping UNIPROT identifiers to Affymetrix probe-sets, intending to produce an integrated view of expression at protein and transcript levels tied to the same gene. We examined several strategies for accomplishing this mapping. We found evidence that utilizing Enfin-Encore services is at minimum competitive, and possibly superior to other approaches [2].

Initially, the motivation ENVISIONQuery was speifically to automate the retrieval of ID mapping information. Later, it became evident that a broad approach would make a wide range of Enfin-Encore services available to R users. Moreover, the Enfin-Encore architecture allows one to combine queries into the pipelines, using the given query results as an input for the next query utilizing the unified EnXML data format. Consequently, the scope of the ENVISIONQuery package expanded to cover additional services provided by the Enfin-Encore portal, the final goal being to cover the whole range of services which the Enfin-Encore online front ends, EnVision and EnVision2 provide.

The ENVISIONQuery is a second (the first being the DAVIDQuery) in a series of packages devoted to ID mapping related information retrieval and performance analysis. At the moment our group is

preparing two more packages for submission to BioConductor: the IdMappingAnalysis package for ID mapping performance analysis and the IdMappingRetrieval package providing a unified framework for ID mapping related information retrieval from various online services for further analysis by the IdMappingAnalysis package. DAVIDQuery and ENVISIONQuery are utilized by the IdMappingRetrieval package.

2.2 Comparison to biomaRt

The biomaRt package available from BioConductor is an extremely powerful tool covering a wide range of data types and sources. On the other hand, the Enfin Encore (and ENVISIONQuery), though at an early stage of development, provides access to some types of data that not covered by biomaRt, the IntAct and PICR being the examples (see section 5). Thus, we believe ENVISIONQuery would nicely complement biomaRt.

When we began the integration study described above, biomaRt did not actively support UNIPROT. For this reason, we explored other solutions, leading us to use the Enfin Encore online interactive front end (EnVision). Recently, we returned to biomaRt, comparing its results with Enfin Encore in mapping Affymetrix probeset IDs to UNIPROT Accessions. Despite the fact that both systems access presumably the same Biomart (www.biomart.org) data source through the Ensembl (www.ensembl.org), the results were somewhat different. For the HG U133 Plus 2 micro array, either biomaRt or ENVISIONQuery returned UNIPROT Accession matches for 32438 probesets. Of those probesets, 70% of the returned UNIPROT lists were identical, for 28% the lists overlapped but were not identical, and for 2% only one of the two methods returned a UNIPROT accession. The Enfin Encore team informed us that there is more than one instance of the Biomart data source underlying database, and that Ensembl and Enfin Encore ID mapping service access different ones. We are not certain which service is more accurate.

It should be noted that EnXML file format imposes a significant performance penalty on the query system due to high percent of redundant information contained within the xml text file. In our experience, the EnXML file is 10 to 20 times larger than the comparable csv-format file which biomaRt returns, with comparable ratio for time expended. Therefore, large queries should probably use biomaRt, if the discrepancies described above are of little concern. Otherwise, our recommendation is to use the ENVISIONQuery when the query size is small enough (up to a few hundred input IDs), or when the service desired is not supported by biomaRt.

2.3 ENVISIONQuery and Java Web Services

As mentioned above, the Encore platform is implemented in Java as a collection of Web Services, and therefore, the most robust way to integrate it into the custom application would be to use the language supporting the web service paradigm. In fact, the EnCore team recommends using Java in client applications and provides extensive support to facilitate this style of development. From the other hand, R does not have such capabilities. To overcome this limitation, the package is subdivided in two distinctive subsystems: the Java library providing the set of classes serving as wrappers for EnCore web services and the R counterpart communicating with Java using the rJava package available from CRAN and providing the package high level functionality including user interaction, preparing data for online submission, formatting the raw EnXML data, filtering data on a set of constraints etc. Each web service implemented within Java is registered in R during the package initialization in a special data structure which contains the information on how to access the Java methods, what services are available, what tools and options can be used for a particular service, as well as additional information allowing to interactively define the processing details as necessary. As a result, adding an additional EnCore service is a straightforward process involving adding a Web Service into the Java library (any major Java IDE like Eclipse or NetBeans has a built-in support for this), implementing a wrapper suitable for call from R through rJava and updating the package initialization function to include the newly developed service. Provided the conversion of the EnXML file into the R data frame is straightforward (being the case for majority of services), the effort of adding the new functionality is reduced and the whole process the package updating and maintenance gets simplified.

3 Types of identifiers, services and reports

As of this version, the ENVISIONQuery there are following important attributes in the package query syntax. The `"ids"` attribute determines the set of identifiers about which information is to be retrieved. It could be either a character vector of IDs (UNIPROT or Affymetrix probeset IDs as an example) or EnXML compliant text in the form of either character string or file. The `"typeName"`determines the type of input identifier set. The `"serviceName"`defines the type of service from which information should be retrieved, and `"toolName"`defines the tool to be used within the particular service. The `"options"` attribute defines any additional query parameters (maximum number of pathways for Reactome service being an example) while the `"filter"` attribute defines the additional filtering to be applied to the formatted results (organism species, micro array platform etc.). A summary of ENVISIONQuery currently supported functionality is given in a Table 1.

Table 1: Services Summary

Service	**Tools**	**Description**	**Source**
ID Conversion	Affy2Uniprot, Uniprot2Affy[1]	Convert Affy probeset to UNIPROT Convert UNIPROT to Affy probeset	Ensembl www.ensembl.org
Intact	FindPartners	Find protein-protein interaction (UNIPROT)	EMBL-EBI www.ebi.ac.uk/intact
Picr	mapProteinsAdv	Map protein identifiers between various DBs[2]	EMBL-EBI www.ebi.ac.uk/Tools/picr
Reactome	FindPathAdv	Finds pathways for specified proteins	Reactome www.reactome.org

[1]As of the end of January 2011 this service is temporarily unavailable. The development team has assured that it will be operational by the end of February 2011.

[2]The list of supported databases and their identifiers can be found at the bottom of the following page: http://www.ebi.ac.uk/Tools/picr/WSDLDocumentation.do.

For detailed description of services as well as options supported for a particular service please use the links provided in Table 2.

Table 2: Service references

Service	Documentation URL
ID Conversion	http://code.google.com/p/enfin-core/wiki/wp1_encore_webservices_affy2uniprot
Intact	http://www.enfin.org/encore/wsdl/enfin-intact.wsdl [1]
Picr	http://code.google.com/p/enfin-core/wiki/wp1_encore_webservices_picr
Reactome	http://code.google.com/p/enfin-core/wiki/wp1_encore_webservices_reactome

[1]At the moment, only formal service description is available. For informal description, please refer to www.ebi.ac.uk/intact.

4 Getting started

4.1 Exploring package capabilities

The code snippet below shows how to check what services as well as tools and options for a given service are supported by `"ENVISIONQuery"` package.

```
> library(ENVISIONQuery);

This is  ENVISIONQuery   1.24.0   2013-08-20

> #check available services
> getServiceNames();

[1] "Reactome"

> #check available tools for a given service
> service<-getService("ID Conversion");
> getToolNames(service);

NULL

> #check available input type for a given tool
> service<-getService("Reactome");
> tool<-getTool(service,"FindPathAdv");
> getInputTypes(tool);

[1] "Uniprot ID"      "Enfin XML"       "Enfine XML file"

> #getAvailable options for a given service
> getServiceOptions("Picr");

NULL
```

4.2 Using basic ENVISIONQuery functionality

The set of queries below retrieves the formatted information from all 4 services supported in current version of the package using the default service options and providing the the set of request specific IDs as well as service, tool and input type attributes.

```
> #convert the Affy probeset IDs to Uniprot IDs
> res<-ENVISIONQuery(ids=c("1553619_a_at","1552276_a_at","202795_x_at"),
+ serviceName="ID Conversion",toolName="Affy2Uniprot",typeName="Affymetrix ID");
> print(res);

NULL

> #retrieve the pathways for given Uniprot ID(s)
> res<-ENVISIONQuery(ids="P38398",serviceName="Reactome",typeName="Uniprot ID");

 Error in if (!(toolName %in% toolNames)) {: argument is of length zero

ENVISIONQuery: No results found

> print(res[1:5,]);

NULL

> #retrieve protein-protein interactions
> res<-ENVISIONQuery(ids="P38398",serviceName="Intact",typeName="Uniprot ID");
> print(res[1:5,]);

NULL

> #convert EnSembl IDs to Uniprot IDs
> res<-ENVISIONQuery(ids=c("ENSP00000397145","ENSP00000269554"),
+ serviceName="Picr",typeName="Protein ID");
> print(res);

NULL
```

4.3 Using Options and Filters

In order to alter the default service behavior, one needs to supply the options and or filter set in the form of the list where item names and values define to options/filters types and values. The code snippet below illustrates how to:

1. Retrieve the data from PICR service specifying the databases which the Uniprot IDs should be mapped to.
2. Retrieve data from Reactome service adding coverage and total protein count for a given pathway to the output.
3. Convert Affymetrix probeset IDs to Uniprot IDs filtering output on micro array type and organism species.

```
> #match Uniprot IDs to EnSembl and TrEMBL
> options<-list("enfin-picr-search-database"=c("ENSEMBL_HUMAN","TREMBL"));
> res<-ENVISIONQuery(ids="P38398",options=options,
+ serviceName="Picr",typeName="Protein ID");
> print(res);

NULL

> #retrieve the pathways for given Uniprot ID(s)sorting them by coverage
> #and calculating the total protein count
> options<-list("enfin-reactome-add-coverage"="true",
+ "enfin-reactome-sort-by-coverage"="true");
> res<-ENVISIONQuery(ids="P38398",options=options,
+ serviceName="Reactome",typeName="Uniprot ID");

 Error in if (!(toolName %in% toolNames)) {: argument is of length zero

ENVISIONQuery: No results found

> print(res[1:5,]);

NULL

> #convert the Affy probeset IDs to UniProt IDs restricting
> #output by micro array type and organism species
> filter<-list(Microarray.Platform="affy_hg_u133_plus_2",
+ organism.species="Homo sapiens");
> res<-ENVISIONQuery(ids=c("1553619_a_at","1552276_a_at","202795_x_at"),
+ filter=filter, serviceName="ID Conversion",
+ toolName="Affy2Uniprot",typeName="Affymetrix ID");
> print(res);

NULL
```

4.4 Cascading query requests.

The Enfin Encore architecture allows to use the EnXML output file as an input for another service, the resulting EnXML file being superposition of all intermediate results. The code snippet below shows how the ENVISIONQuery can be used to combine Intact and Reactome service requests into a single pipeline. Note, that as of the recent `"ENVISIONQuery"` does not support the formatted output for a cascaded call but we plan to provide it in the nearest future. In the meantime, the `"XML"` can be used to explore the resulting EnXML file.

```
> #Intact-Reactome cascaded request for UniProt ID(s).
> IntactReactomeXML<-ENVISIONQuery(ids="P38398",
+ serviceName=c("Intact","Reactome"),typeName="Uniprot ID",verbose=TRUE);
> #convert xml text into XMLDocument
> #using XML package for further exploring
> if(!is.null(IntactReactomeXML)){
+ xmlDoc<-xmlTreeParse(IntactReactomeXML,useInternalNodes = TRUE, asText=TRUE);
+ class(xmlDoc);
+ }
```

4.5 Defining ENVISIONQuery request attributes interactively

Sometimes it's desirable to specify the service, tool and/or input format interactively at the run time. This can be accomplished by specifying the "menu" (or omitting the parameter) instead of providing a concrete service, tool or input type name as illustrated below.

```
> filter<-list(Microarray.Platform="affy_hg_u133_plus_2",
+ organism.species="Homo sapiens");
> res<-ENVISIONQuery(ids=c("1553619_a_at","1552276_a_at","202795_x_at"),
+ filter=filter, serviceName="menu",toolName="menu",typeName="menu");
> print(res);
```

5 Running large queries

The current version of Enfin Encore services imposes a limitation on a number of input identifiers in the query request. To overcome this limitation, the ENVISIONQuery call processes data in chunks (using ENVISIONQuery.loop and ENVISIONQuery.chunks internally) returning the list of EnXML documents in case of unformatted output and merging the partial data frames into a single one in case the formatted output is requested. Unfortunately, the Enfin Encore documentation does not specify the limit on a number of input identifiers and if the limit is exceeded, just silently returns the input EnXML document or partial (and varying) resulting set. For Affy probeset to Uniprot conversion (Affy2Uniprot service) we have determined that the fail-safe chunk size is 1000 (default). For other services, the size could be different. In case the instability is observed, the chunk size can be altered by user.

6 Session information

This version of ENVISIONQuery has been developed with R 2.11.0.

R session information:

```
> toLatex(sessionInfo())
```

- R version 3.4.0 (2017-04-21), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=en_US.UTF-8`, `LC_ADDRESS=en_US.UTF-8`, `LC_TELEPHONE=en_US.UTF-8`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=en_US.UTF-8`
- Running under: `Ubuntu 16.04.2 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.5-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.5-bioc/R/lib/libRlapack.so`

- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: ENVISIONQuery 1.24.0, XML 3.98-1.6, rJava 0.9-8
- Loaded via a namespace (and not attached): compiler 3.4.0, tools 3.4.0

7 Acknowledgments

We would like to thank Rafael Jimenez from the Enfin EnCore team for an enjoyable discussion and the great effort he has put into expanding the ID mapping related EnCore web services to better suite our needs. We also would like to thank Himanshu Grover, a graduate student in Biomedical Informatics at the University of Pittsburgh, for the great effort and useful suggestions he has made during the package preparation and testing.

8 References

[1] Florian Reisinger, Manuel Corpas1, John Hancock, Henning Hermjakob, Ewan Birney, and Pascal Kahlem1. ENFIN - An Integrative Structure for Systems Biology. Data Integration in the Life Sciences Lecture Notes in Computer Science, 2008, Volume 5109/2008, 132-143, DOI: 10.1007/978-3-540-69828-9_13

[2] Identifier mapping performance for integrating transcriptomics and proteomics experimental results Roger S. Day1, Kevin K. McDade1, Uma Chandran1, Alex Lisovich, Thomas Conrads, Brian Hood, V.S.Kumar Kolli, David Kirchner, Traci Litzi, G. Larry Maxwell, in submission to BMC Bioinformatics