The *DAVIDQuery* package in Bioconductor: Retrieving data from the DAVID Bioinformatics Resource

Roger S. Day†‡ , Alex Lisovich†

June 6, 2010

†Department of Biomedical Informatics, ‡Department of Biostatistics
University of Pittsburgh

1 Introduction

DAVID (Database for Annotation, Visualization and Integrated Discovery) is a bioinformatics resource developed by the National Institute of Allergy and Infectious Diseases at Frederick in conjunction with the Laboratory of Immunopathogenesis and Bioinformatics (LIB), SAIC Frederick. This resource is described as "a graph theory evidence-based method to agglomerate species-specific gene/protein identifiers the most popular resources including NCBI, PIR and Uniprot/SwissProt. It groups tens of millions of identifiers into 1.5 million unique protein/gene records." Further information can be found in published articles [1][2].

As of this time, maintenance of the DAVID resource is supervised by Dr. Richard Lempicki. The resource is accessed interactively at `http://david.abcc.ncifcrf.gov/`. The interactive interface provided there is suitable for many purposes, but for a bioinformatician using R an automated procedural solution is needed. The convention for executing queries via formation of URL attribute-value strings is provided at `http://david.abcc.ncifcrf.gov/content.jsp?file=DAVID_API.html`. Although this is described as an application program interface (API), the desired query result is not directly provided by the immediate return page, and two rounds of "screen-scraping" and URL formulation are required to retrieve the query results from a program.

In Spring 2010, the DAVID interface changed. This package has been modified to work with the new interface. In particular, the "Gene ID Conversion tool"

was excluded from the new DAVID API and required a separate implementation as outlined at the end of the next session.

2 Types of identifiers and reports

As of this version, there are three important attributes in the URL specification. The `"id"` attribute will hold the proband identifiers about which information is to be retrieved. The `id` values are combined in a single string joined by commas. The `"type"` attribute will hold a string indicating the type of the identifiers. The list of legitimate values for `type` has increased from 15 to 37 and includes the "Not sure" type which causes the DAVID system to infer the type based on the ID list content. The choices, described as "DAVID's recognized gene types", now are obtained directly from the page `http://david.abcc.ncifcrf.gov/tools.jsp`.

The legitimate values for `type` (excluding "Not Sure") are:

AFFYMETRIX_3PRIME_IVT_ID	AFFYMETRIX_EXON_GENE_ID	AFFYMETRIX_SNP_ID
AGILENT_CHIP_ID	AGILENT_ID	AGILENT_OLIGO_ID
ENSEMBL_GENE_ID	ENSEMBL_TRANSCRIPT_ID	ENTREZ_GENE_ID
FLYBASE_GENE_ID	FLYBASE_TRANSCRIPT_ID	GENBANK_ACCESSION
GENOMIC_GI_ACCESSION	GENPEPT_ACCESSION	ILLUMINA_ID
IPI_ID	MGI_ID	OFFICIAL_GENE_SYMBOL
PFAM_ID	PIR_ID	PROTEIN_GI_ACCESSION
REFSEQ_GENOMIC	REFSEQ_MRNA	REFSEQ_PROTEIN
REFSEQ_RNA	RGD_ID	SGD_ID
TAIR_ID	UCSC_GENE_ID	UNIGENE
UNIPROT_ACCESSION	UNIPROT_ID	UNIREF100_ID
WORMBASE_GENE_ID	WORMPEP_ID	ZFIN_ID

The third attribute is `"tool"`, which refers to the type of report to be generated. Values which return useful results are the strings `"gene2gene"`, `"list"`, `"geneReport"` (the latter two nearly equivalent), `"annotationReport"`, and `"geneReportFull"`. The other choices for `tool`, related to DAVID's Functional Annotation tools, generate much more complex output and cannot be handled by this package at this time.

A fourth attribute, the `"annot"` attribute, is relevant to the `"annotationReport"`, tool. It names the additional columns to appear in the annotation report. For other tools, `"annot"` does not appear to affect the returned results, and is generally set to `NULL`.

If the query contains `tool=list` or `tool=geneReport`, then the result (after formatting) is a three-column character data frame. If the query contains `tool=geneReportFull`, then the result (after formatting) is a list with each element corresponding to an identifier in the ID list. If the query contains `tool=gene2gene`, then the result (after formatting) is a list with each element

corresponding to a functional group selected by a DAVID algorithm. The formats are documented in detail in the manual documents for the function `formatDAVIDResult`.

As was mentioned before, the Gene ID Conversion Tool is not included into the latest version of API and can be accessed only through the online query system. To overcome this limitation, we introduced the new tool value, `"geneIdConversion"`, and implemented the conversion by programmatically reproducing the Gene ID Conversion Tool workflow as follows. First, the list of IDs to be converted from the given ID type is submitted to the DAVID online "tools.jsp" service using the HTTP message post. Second, the DAVID check 'at least 80 percent of samples should be mapped' turned off by accessing the hidden URL "submitAnyway.jsp". This ensures that the input ID list can contain any percentage of correct IDs and still be mapped properly. Third, the request for ID conversion is sent by posting the HTTP message to the DAVID conversion service. The resulting page is scrapped, the URL of the conversion result file is obtained and the file is retrieved. As the conversion results file is a well formatted table represented by a tab delimited .txt file,no further formatting of the DAVIDQueryResult is needed. The `"annot"` attribute values in this case are the same as for `"type"`, with addition of an extra item, `DAVID` (the DAVID unique gene identifier), and define the type of gene ID conversion to be performed.

3 Motivating setting

Our group received results of a proteomic mass spectrometry experiment that generated over 12,000 protein UNIPROT identifiers, and needed to compare these results to a microarray experiment that utilized the Affymetrix U133 Plus 2 chip. Therefore the 12,000 identifiers needed to be mapped as well as possible to Affymetrix probe-sets which could confidently be assigned to protein-coding genes. There are numerous strategies for accomplishing this mapping, such as utilizing the Affymetrix NetAffx resource or NCBI Entrez, but each approach is known to generate an occasional incorrect answer. Utilizing DAVID appears to be at minimum competitive with the others, and possibly the best approach.

An early version of `DAVIDQueryLoop` was used to retrieve matching probe-sets. These results, together with comparisons to alternative mapping methods, are to be reported in a manuscript in preparation. This work was initially performed by Kevin McDade at the University of Pittsburgh, later automated by us; he is continuing with some related innovative sequence-based analysis.

It should be noted that, as of last look, the retrieval of Affymetrix probe-set IDs via the DAVID API did not allow for restricting the result to a specified chip. Lists of probe-sets by chip name are available at DAVID. The function `getAffyProbesetList` is provided in this package to retrieve the list for the chip of interest, for intersection with lists of probe-sets retrieved from DAVID

via DAVIDQueryLoop. (We caution that there is no guarantee that these probeset lists match comparable lists obtained elsewhere.)

4 Launching a single query

A single query is accomplished with the function DAVIDQuery. The mechanics involve formulating a query URI, launching it and retrieving identifiers from the returned HTML, formulating and launching a new query, retrieving a result file name from the returned HTML, and finally retrieving the file itself. Formatting of the final result is the default option. (The result file remains on the server for 24 hours.)

4.1 Structured and unstructured

A raw HTML character stream is transmitted by DAVID. By default, an attempt to structure the results will be made. A structuring function is defined for each tool. There is no guarantee that the structuring functions will continue to work if or when the formats of the pages returned by DAVID change. Also, not all combinations of the query arguments have been tested, and there may be combinations of ids, type, annot, tool for which the tool's structuring function does not work correctly. When a look at the raw stream is desired, for example if the structuring fails or the result is unexpected, then the call can be made with the argument assignment: DAVIDQuery(formatIt=FALSE). This allows the user to receive the raw character table actually returned.

4.2 Examples

```
> library("DAVIDQuery")

This is DAVIDQuery Version  1.20.0   2010-06-10

> result = DAVIDQuery(type="UNIPROT_ACCESSION", annot=NULL, tool="geneReportFull")
> names(result)

 [1] "ids"               "firstURL"          "firstStageResult"
 [4] "DAVIDaction"       "secondURL"         "secondStageResult"
 [7] "hasSessionEnded"   "downloadFileName"  "downloadURL"
[10] "DAVIDQueryResult"

>
```

The result has been structured into a list of lists. Printing is suppressed due to the size of the output. The code `DAVIDQuery(testMe=TRUE)` is the equivalent of the DAVIDQuery call above.

The result of the simpler query using `tool="geneReport"` is a matrix:

```
> Sys.sleep(10)  ### Assure that queries are not too close in time.
> result = DAVIDQuery(type="UNIPROT_ACCESSION", annot=NULL, tool="geneReport")
> result$firstURL

[1] "http://david.abcc.ncifcrf.gov/api.jsp?type=UNIPROT_ACCESSION&ids=O00161,O75396&tool=gen

> result$secondURL

[1] "http://david.abcc.ncifcrf.gov/geneReport.jsp?rowids=814170,803794&annot=814170,803794"

> result$downloadURL

[1] "http://david.abcc.ncifcrf.gov/data/download/gr_E31D02532DA31365650707898.txt"

> result$DAVIDQueryResult

      ID                                                    Gene.Name
2 O75396 SEC22 vesicle trafficking protein homolog B (S. cerevisiae)
3 O00161                      synaptosomal-associated protein, 23kDa
       Species
2 Homo sapiens
3 Homo sapiens

>
```

The Gene Functional Classification query is obtained by the query clause `tool="gene2gene"`. The returned value has a complex structure which we attempt to translate into a corresponding R object respecting the structure, using the function `format-Gene2Gene`.

```
> Sys.sleep(10)  ### Assure that queries are not too close in time.
> result = testGene2Gene(details=FALSE)
> length(result)

[1] 4

> names(result[[1]])
```

```
[1] "type"    "value"   "details"
```

Convenience functions are provided to assist with integrating genomic and proteomic data:

```
> Sys.sleep(10)  ### Assure that queries are not too close in time.
> affyToUniprot(details=FALSE)
```

```
      From     To      Species                                Gene.Name
1 88736_AT O00161 Homo sapiens synaptosomal-associated protein, 23kDa
2 88736_AT Q13602 Homo sapiens synaptosomal-associated protein, 23kDa
3 88736_AT O00162 Homo sapiens synaptosomal-associated protein, 23kDa
4 88736_AT Q6IAE3 Homo sapiens synaptosomal-associated protein, 23kDa
5 88736_AT A8K287 Homo sapiens synaptosomal-associated protein, 23kDa
```

```
> Sys.sleep(10)  ### Assure that queries are not too close in time.
> uniprotToAffy(details=FALSE)
```

```
     From                    To      Species
1  O00161             209130_AT Homo sapiens
2  O00161           214544_S_AT Homo sapiens
3  O00161      G1374812_3P_A_AT Homo sapiens
4  O00161          229773_PM_AT Homo sapiens
5  O00161              42622_AT Homo sapiens
6  O00161             229773_AT Homo sapiens
7  O00161             U55936_AT Homo sapiens
8  O00161          209130_PM_AT Homo sapiens
9  O00161            32179_S_AT Homo sapiens
10 O00161          RC_N25249_AT Homo sapiens
11 O00161            46749_I_AT Homo sapiens
12 O00161            32178_R_AT Homo sapiens
13 O00161      RC_AA342059_S_AT Homo sapiens
14 O00161        209131_PM_S_AT Homo sapiens
15 O00161              88736_AT Homo sapiens
16 O00161           209131_S_AT Homo sapiens
17 O00161 MMUGDNA.23322.1.S1_AT Homo sapiens
18 O00161  HS.122505.0.A1_3P_AT Homo sapiens
19 O00161          RC_H02552_AT Homo sapiens
20 O00161        214544_PM_S_AT Homo sapiens
21 O00161       G13277555_3P_AT Homo sapiens
                                Gene.Name
1  synaptosomal-associated protein, 23kDa
2  synaptosomal-associated protein, 23kDa
3  synaptosomal-associated protein, 23kDa
```

```
4  synaptosomal-associated protein, 23kDa
5  synaptosomal-associated protein, 23kDa
6  synaptosomal-associated protein, 23kDa
7  synaptosomal-associated protein, 23kDa
8  synaptosomal-associated protein, 23kDa
9  synaptosomal-associated protein, 23kDa
10 synaptosomal-associated protein, 23kDa
11 synaptosomal-associated protein, 23kDa
12 synaptosomal-associated protein, 23kDa
13 synaptosomal-associated protein, 23kDa
14 synaptosomal-associated protein, 23kDa
15 synaptosomal-associated protein, 23kDa
16 synaptosomal-associated protein, 23kDa
17 synaptosomal-associated protein, 23kDa
18 synaptosomal-associated protein, 23kDa
19 synaptosomal-associated protein, 23kDa
20 synaptosomal-associated protein, 23kDa
21 synaptosomal-associated protein, 23kDa
```

5 Launching large queries

To control performance of the DAVID website, and to assure that queries launched by the website can be successfully processed, policy limits are implemented. When a user needs to retrieve answers which would exceed these limits if a single query is attempted, the function `DAVIDQueryLoop` can be used. It attempts to slow successive calls and to reduce the query size, sufficiently to meet the website policies with a little to spare.

6 Limitations

This package cannot use semantic interoperability, due to the nature of DAVID API. This entails risk that future modifications to DAVID will cause functions in this package to fail. In fact, this did occur in the Spring of 2010, entailing a major refactoring of this package.

7 Future improvements and adaptations

We would like to create a package targeted more generally to data analysis combining protein expression data with mRNA expression data. The main focus, initially at least, will be to provide support for mapping between protein identifiers, for example those returned by Sequest from mass spectrometry experimental results, and probe-set identifiers for microarray chips. Multiple mapping

methods will be implemented and compared, extending ongoing research in our group.

Ideally, the information in DAVID would be directly available via a grid service. Neither the DAVID team nor we have current plans to implement that, but note that Martin Morgan's team working with caBIG has developed extensive tools for bridging between R and the caBIG's caGRID, using the package *RWebServices* from Bioconductor.

8 Session information

This version of DAVIDQuery has been developed with R 2.11.0.

R session information:

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

- R version 3.0.0 (2013-04-03), `x86_64-apple-darwin10.8.0`
- Locale: `C`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: DAVIDQuery 1.20.0, RCurl 1.95-4.1, bitops 1.0-5
- Loaded via a namespace (and not attached): tools 3.0.0

9 Acknowledgements

Brad Sherman and Da Wei Huang of the DAVID project kindly reviewed this package and documentation. Their corrections and encouragement were invaluable.

Thanks are due to Drs. Larry Maxwell and Thomas Conrads for provision of the data and scientific collaborations that motivated this work, Kevin McDade and Uma Chandran for discussions on the identifier-mapping problem, and Richard Boyce for careful review of the package and documentation. Grant support includes funding from the Gynecologic Diseases Program, a collaboration whose bioinformatics components include Walter Reed Army Medical Center, University of Pittsburgh, and Windber Research Institute. Additional support came from the Telemedicine and Advanced Technology Research Center (TATRC).

10 References

[1] Huang D.W., Sherman B.T., Tan Q., Kir J., Liu D., Bryant D., Guo Y., Stephens R., Baseler M.W., Lane H.C. et al. (2007) DAVID Bioinformatics Resources: expanded annotation database and novel algorithms to better extract biology from large gene lists. Nucleic Acids Res., 35, W169-W175.

[2] Huang D.W., Sherman B.T. and Lempicki R.A. (2008) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat. Protoc., doi: 10.1038/nprot.2008.211.