Retrieving data from the DAVID Bioinformatics Resource

University of Pittsburgh

January 10, 2009

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 me thod 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.

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 legitimate values for `type` are:

AFFY_ID	ENTREZ_GENE_ID	GENBANK_ACCESSION
GI_ACCESSION	PIR_ID	PIR_NREF_ID
REFSEQ_MRNA	REFSEQ_PROTEIN	REFSEQ_RNA
UNIPROT_ACCESSION	UNIPROT_ID	UNIREF100_ID
GENPEPT_ACCESSION	REFSEQ_GENOMIC	UNIGENE

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`.

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. The bulk of the work being performed by Kevin McDade at the University of Pittsburgh.

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

`AffyProbesetList` 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 probe-set 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(structureIt=FALSE)`. This allows the user to receive the raw character table actually returned.

4.2 Examples

```
> library("DAVIDQuery")
> 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 DAVDQuery call above.

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

```
> Sys.sleep(10)
> 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=ge
```

```
> result$secondURL
```

```
[1] "http://david.abcc.ncifcrf.gov//geneReport.jsp?rowids=,3027979,2878470&annot=null"
```

```
> result$downloadURL
```

```
[1] "http://david.abcc.ncifcrf.gov//UserDownload/GR_8CC054DCC54E.txt"
```

```
> result$DAVIDQueryResult
```

```
  UNIPROT_ACCESSION                                        DAVID Gene Name
3            O00161                    SYNAPTOSOMAL-ASSOCIATED PROTEIN, 23KDA
4            O75396 SEC22 VESICLE TRAFFICKING PROTEIN-LIKE 1 (S. CEREVISIAE)
                Species
3 9606:9606:Homo sapiens
4 9606:9606: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 `formatGene2Gene`.

```
> Sys.sleep(10)
> result = testGene2Gene(details = FALSE)
> length(result)
```

```
[1] 5
```

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

```
NULL
```

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

```
> Sys.sleep(10)
> affyToUniprot(details = FALSE)

[1] "AFFY_ID"
[1] "88736_AT"
[1] "Gene Name"
[1] "SYNAPTOSOMAL-ASSOCIATED PROTEIN, 23KDA"
[1] "Species"
[1] "9606:Homo sapiens"
[1] "UNIPROT_ACCESSION"
[1] "O00161, O00162, Q13602, Q6IAE3, "
$`88736_AT`
$`88736_AT`$AFFY_ID
[1] "88736_AT"

$`88736_AT`$`Gene Name`
[1] "SYNAPTOSOMAL-ASSOCIATED PROTEIN, 23KDA"

$`88736_AT`$Species
[1] "9606:Homo sapiens"

$`88736_AT`$UNIPROT_ACCESSION
[1] "O00161" "O00162" "Q13602" "Q6IAE3"


attr(,"annot")
[1] "UNIPROT_ACCESSION"
attr(,"ids")
[1] "88736_AT"
attr(,"tool")
[1] "annotationReport"
attr(,"type")
[1] "AFFY_ID"

> Sys.sleep(10)
> uniprotToAffy(details = FALSE)

[1] "UNIPROT_ACCESSION"
[1] "O00161"
[1] "Gene Name"
[1] "SYNAPTOSOMAL-ASSOCIATED PROTEIN, 23KDA"
[1] "Species"
[1] "9606:Homo sapiens"
[1] "AFFY_ID"
[1] "209130_AT, 209131_S_AT, 214544_S_AT, 229773_AT, 32178_R_AT, 32179_S_AT, 42622_AT, 46749
```

```
$ids
[1] "O00161"

$firstURL
[1] "http://david.abcc.ncifcrf.gov//api.jsp?type=UNIPROT_ACCESSION&ids=O00161&tool=annotatio

$firstStageResult
[1] "\n\n\n\n\n\n \n \n<html>\n<head></head>\n<body>\n<form name=\"apiForm\" method=\"POST\"

$DAVIDaction
[1] "annotationReport.jsp"

$secondURL
[1] "http://david.abcc.ncifcrf.gov//annotationReport.jsp?rowids=,3027979&annot=AFFY_ID"

$secondStageResult
[1] "\n\n\n\n<SCRIPT LANGUAGE=\"JavaScript\">\nvar page='summary';\n</SCRIPT>\n\n  \n  \n\n\
```

```
[15] "MMUGDNA.23322.1.S1_AT"   "RC_AA342059_S_AT"
[17] "RC_H02552_AT"            "RC_N25249_AT"
[19] "SNP_A-1642283"           "SNP_A-1670379"
[21] "SNP_A-1838249"           "SNP_A-1917769"
[23] "SNP_A-2134743"           "SNP_A-2151716"
[25] "SNP_A-2252591"           "SNP_A-2271650"
[27] "SNP_A-2274689"           "SNP_A-2274693"
[29] "SNP_A-2295472"           "U55936_AT"
```

```
attr(,"annot")
[1] "AFFY_ID"
attr(,"ids")
[1] "O00161"
attr(,"tool")
[1] "annotationReport"
attr(,"type")
[1] "UNIPROT_ACCESSION"
```

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 communication with the DAVID team, it appears that improvements in the DAVID API itself are desired but unlikely to reach the top of the work queue in the foreseeable future. This *DAVIDQuery* package has withstood the transition from DAVID 2007 to DAVID 2008. Therefore we anticipate that maintenance in the face of changes will not be as brittle as one might fear. If or when the API is modified, this package may be adapted accordingly.

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 2.8.0 GUI 1.26 (5256).

R session information:

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

- R version 2.9.0 (2009-04-17), `i386-pc-mingw32`
- Locale: `LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MON`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: DAVIDQuery 1.2.0
- Loaded via a namespace (and not attached): RCurl 0.94-1

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.