Using the R Programming Language for VIVO Application Programming

Mike Conlon, PhD COO UF Clinical and Translational Science Institute, Gainesville, Florida and the VIVO Collaboration*

The R Programming Language

R is an open source, open development computing environment and language for statistical computing and graphics[1]. R is popular in biostatistics, bioinformatics, financial market analysis, social network analysis and geospatial modeling. As a programming language, R is expressive and compact with a large collection of powerful functions and tools and operators for data representation, analysis and display. On-line tutorials are available for learning both basic and advanced R programming[2].

Some simple examples:

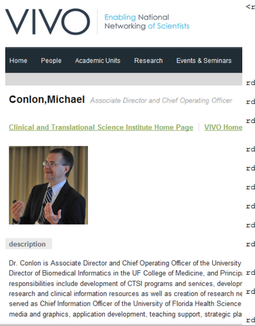
```
x<-5                # create an object x and assign
                    # it the integer value 5
myurl<-"http://vivo.ufl.edu/individual/mconlon"
                    # assign text string to url
v<-rnorm(1000)      # generate 1000 random normal
                    # variates and assign to v
hist(v)             # draw a histogram of v
```

VIVO Data is RDF

VIVO represents all its data using Resource Description Framework (RDF)[3]. RDF represents all data as "triples" of the form subject predicate object. Subjects, objects and predicates are represented in an ontology. See a standard text for descriptions of RDF and ontologies.[4] The VIVO ontology describes people participating in research activity, as well as elements common to these people and their activities -- grants, events, projects, publications and more.[5]

RDF as XML

RDF Schema[6] (RDFS) is a description language for RDF represented in Extensible Markup Language (XML). XML is readily processed by application programs. VIVO can present all its data as either Hypertext Markup Language (HTML), for human reading through a browser, or as XML for application programs and tools. The XML produced by VIVO conforms to the RDF Schema standard. For example, the url in the R sample above can be seen below as rendered HTML (left) or as XML/RDF Schema (right).




```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.0="http://vivo.ufl.edu/ontology/vivo-ufl/"
    xmlns:j.1="http://xmlns.com/foaf/0.1/"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:j.2="http://vivoweb.org/ontology/core#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description
rdf:about="http://vivo.ufl.edu/individual/n25562">
      <j.2:featuredIn
rdf:resource="http://vivo.ufl.edu/individual/n6868"/>
      <j.2:principalInvestigatorOn
rdf:resource="http://vivo.ufl.edu/individual/n4702"/>
      <j.2:workPhone>352 273 8872</j.2:workPhone>
      <j.2:featuredIn
rdf:resource="http://vivo.ufl.edu/individual/n3884"/>
      <j.2:hasProfessionalServiceActivity
rdf:resource="http://vivo.ufl.edu/individual/n3709"/>
      <j.2:educationalBackground
rdf:resource="http://vivo.ufl.edu/individual/n4821"/>
      <j.2:featuredIn
rdf:resource="http://vivo.ufl.edu/individual/n6584"/>
      <j.2:featuredIn
rdf:resource="http://vivo.ufl.edu/individual/n180"/>
      <j.2:featuredIn
rdf:resource="http://vivo.ufl.edu/individual/n1162"/>
      <j.2:workFax>352 273 8703</j.2:workFax>
      <rdf:type
rdf:resource="http://vivoweb.org/ontology/core#FacultyMember"/>
      <j.2:personInPosition
rdf:resource="http://vivo.ufl.edu/individual/n2762"/>
      <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>
      <j.2:hasResearchArea
rdf:resource="http://vivo.ufl.edu/individual/n3883"/>
```

VIVO Applications

VIVO applications are software systems using VIVO data. Existing systems such as Drupal or Sakai can be extended to use VIVO data. Here we show simple R programs which consume and display VIVO data. VIVO applications can be written in any computer language capable of accessing web pages and processing RDF. We use R because of its simplicity and display capabilities.

VIVO applications "read" VIVO data by fetching VIVO data via HTTP. There is no "application programming interface" (API) nor special VIVO software routines to learn. The format of the VIVO data is published via its ontology.[5] This makes VIVO data far easier to consume in applications and repurpose than systems requiring the use of proprietary APIs. VIVO data is open and accessible to all via simple web page fetch.

Fetch VIVO Data via HTTP

To use R for VIVO application programming, you will want to get and install the XML Library.[7] This library provides all the tools you will need to fetch pages and extract values for processing and display in R. To fetch a VIVO page from a URL, execute the one line below:

```
my.rdf<-readLines(url(myurl))
```

The variable named `my.rdf` is created and contains the RDF Schema text from the remote VIVO page as shown previously.

Create an XML Parse Tree

The resulting RDF can be parsed into a tree for further processing. Many objects in VIVO have parent-child relationships.

```
my.tree<-xmlParse(myurl)
```

The variable `my.tree` is created by fetching the remote page and parsing the XML found there.

Use XPath to Extract Data Values

A tree can be searched for values satisfying an XPath[8] query.

```
my.nodes<-getNodeSet(my.tree,"//j.2:workPhone")
```

The matching node(s) are then stripped to get values

```
my.workphone<-xmlValue(my.nodes[[1]])
```

The variable `my.workphone` now contains the value "352 273 8872"

Single and Multiple Values

VIVO RDF contains single valued elements and multi-valued elements. The R code shown above is for a single valued response. getNodeSet will return multiple values in an R list structure for further processing.

Crawling RDF

In some cases, the objects returned by VIVO are RDF URIs for other objects. This is the basis of the semantic web – interlinked references to objects expressed as RDF. Resolving such references can be called "crawling" or "dereferencing." Consider the organizational structure of a university. Each "org" may have subOrganizations, which are each "orgs". A uri for the University of Florida in VIVO returns its subOrganizations. Each is an RDF URI for the subOrganization – a college, institute or department. Using R, we can access each organization and recursively process its subOrganizations to generate a complete tree structure for the university as a whole. The code does just that. `processOrg` returns the entire organizational structure of the university (or any other university with a VIVO URI). `getURI` is a helper function for creating URIs from RDF XML attributes.

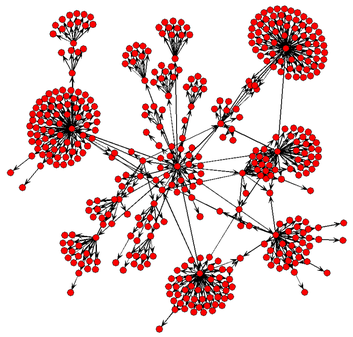
```
processOrg<-function(uri){
  x<-xmlParse(uri)
  u<-NULL
  name<-xmlValue(getNodeSet(x,"//rdfs:label")[[1]])
  subs<-getNodeSet(x,"//j.1:hasSubOrganization")
  if(length(subs)==0) list(name=name,subs=NULL)
  else {
    for(i in 1:length(subs)){
      sub.uri<-getURI(xmlAttrs(subs[[i]])["resource"])
      u<-c(u,processOrg(sub.uri))
    }
    list(name=name,subs=u)
  }
}
```

Displaying Results Using statnet[9]

statnet is an open source suite of packages for R used for network. The organizational structure of the University of Florida is displayed as a directed graph below. The root node is in the center. Directed vertices point to subOrganizations.

Large clusters represent the College of Medicine, The Institute for Food and Agricultural Sciences, the extension offices, and the College of Liberal Arts and Sciences.

The figure was produced using the code above, followed by transformation to a statnet edgelist, then a network object named `uf.g`. The network object was plotted with the single R function `plot(uf.g)`



Next Steps

If you are new to programming you will find R a bit difficult. Experienced programmers will find R to be relaxing and powerful. Writing R functions involves a bit of research to find the best functions for the task at hand. The compactness of R makes it easy to read for the experienced R programmer. If you are not an experienced programmer, you may wish to team with someone who is.

R is particularly well suited for extracting data, tabulating, reporting and displaying data. The statnet community is adding social network analysis tools. R is less well-suited for interactive applications. Such applications might be written with Web 2.0 front-end tools, while using R for back-end data extraction, processing and graphics generation.

The R programming language augmented by the XML tools for data extraction and the statnet tools for social network display and analysis provide a powerful and ready made toolbox for VIVO application programming.

Obtaining R, Packages and Code Examples

Download installers for R for Windows, Mac or Linux from the R Home Page[1]. The installer does the rest. To install the XML and statnet packages, execute the R commands :

```
install.packages("XML", repos =
     "http://www.stats.ox.ac.uk/pub/RWin")
library(XML)
install.packages("statnet")
library(statnet)
```

All code displayed and used on this poster is available at vivo.sourceforge.net

References

[1]R Project Home Page www.r-project.org
[2]Resources to help you learn and use R www.ats.ucla.edu/stat/R
[3]Resource Description Framework (RDF) www.w3c.org/RDF
[4]Dean Allemang and Jim Hendler (2008) *Semantic Web for the Working Ontologist*, Morgan Kaufmann, 352 pp.
[5]VIVO Ontology http://sourceforge.net/projects/vivo/files/Ontology/vivo-core-1.1.owl
[6]RDF Vocabulary Description Language 1.0: RDF Schema http://www.w3.org/TR/rdf-schema/
[7]Lang, Duncan Temple *Tools for parsing and generating XML in R,* http://www.omegahat.org/RSXML/
[8]XML Path Language (Version 1.0) http://www.w3.org/TR/xpath/
[9]Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris (2003) Software Tools for the Statistical Modeling of Network Data. Version 2.1-1. Project home page at http://statnet.org, URL http://CRAN.R-project.org/package=statnet

*VIVO Collaboration: Cornell University: Dean Krafft (Cornell PI), Manolo Bevia, Jim Blake, Nick Cappadona, Brian Caruso, Jon Corson-Rikert, Elly Cramer, Medha Devare, Elizabeth Hines, Huda Khan, Brian Lowe, Joseph McEnerney, Holly Mistlebauer, Stella Mitchell, Anup Sawant, Christopher Westling, Tim Worrall, Rebecca Younes. University of Florida: Mike Conlon (VIVO and UF PI), Chris Barnes, Cecilia Botero, Kerry Britt, Erin Brooks, Amy Buhler, Ellie Bushhousen, Linda Butson, Chris Case, Christine Cogar, Valrie Davis, Mary Edwards, Nita Ferree, George Hack, Chris Haines, Sara Henning, Rae Jesano, Margeaux Johnson, Meghan Latorre, Yang Li, Paula Markes, Hannah Norton, Narayan Raum, Alexander Rockwell, Sara Russell Gonzalez, Nancy Schaefer, Dale Scheppler, Nicholas Skaggs, Matthew Tedder, Michele R. Tennant, Alicia Turner, Stephen Williams. Indiana University: Katy Borner (IU PI), Kavitha Chandrasekar, Bin Chen, Shanshan Chen, Jeni Coffey, Suresh Deivasigamani, Ying Ding, Russell Duhon, Jon Dunn, Poornima Gopinath, Julie Hardesty, Brian Keese, Namrata Lele, Micah Linnemeier, Nianli Ma, Robert H. McDonald, Asik Pradhan Gongaju, Mark Price, Yuyin Sun, Chintan Tank, Alan Walsh, Brian Wheeler, Feng Wu, Angela Zoss. Ponce School of Medicine: Richard J. Noel, Jr. (Ponce PI), Ricardo Espada Colon, Damaris Torres Cruz, Michael Vega Negrón. The Scripps Research Institute: Gerald Joyce (Scripps PI), Catherine Dunn, Brant Kelley, Paula King, Angela Murrell, Barbara Noble, Cary Thomas, Michaeleen Trimarchi. Washington University School of Medicine in St. Louis: Rakesh Nagarajan (WUSTL PI), Kristi L. Holmes, Caerie Houchins, George Joseph, Sunita B. Koul, Leslie D. McIntosh. Weill Cornell Medical College: Curtis Cole (Weill PI), Paul Albert, Victor Brodsky, Mark Bronnimann, Adam Cheriff, Oscar Cruz, Dan Dickinson, Richard Hu, Chris Huang, Itay Klaz, Kenneth Lee, Peter Michelini, Grace Migliorisi, John Ruffing, Jason Specland, Tru Tran, Vinay Varughese, Virgil Wong. This project is funded by the National Institutes of Health, U24 RR029822, "VIVO: Enabling National Networking of Scientists".