Getting VIVO Attributes from RDF Using R

VIVO presents its data via RDF. The RDF data comes in chunks. You issue the URI of the data you are interested in, and VIVO returns a chunk of RDF.

Getting RDF into R

For example, my "person" page can be found at http://vivo.ufl.edu/individual/n25562/n25562.rdf. If you issue this URI to an appropriate function, a chunk of RDF is returned. In R, this is done using the code below:

```
conlon.uri <- "http://vivo.ufl.edu/individual/n25562/n25562.rdf"
conlon.rdf <- readLines(url(conlon.uri))
```

Executing these two lines returns the RDF as a series of text lines in the array conlon.rdf.

What comes next would be parsing and searching the RDF result for values of interest. But there is a much simpler way to do this. RDF, as presented by VIVO, is delivered as RDF Schema, an XML variant. That is, all RDF supplied by VIVO comes as XML pages. R has a significant capability for processing XML documents.

Using XML in R

The R XML package provides a significant collection of XML processing tools. You'll want this package for working with VIVO in R. See R Toolkit to get and set up the library.

Perhaps the most important is xmlParse, a function that parses an XML page and returns an R structure containing the XML parse tree representation. XML parse trees are data structures that can be easily searched and manipulated. In R processing of VIVO pages, we are predominately interested in search.

To use xmlParse, simply give it the url of a page. It returns a parse tree:

```
conlon.tree <- xmlParse(conlon.uri)
```

conlon.tree now contains a parse tree of the RDF page at conlon.uri.

The getNodeSet function provides a full XPath implementation for searching a parse tree and returning nodes specified by the XPath query. For example:

```
conlon.labels<-getNodeSet(conlon.tree,"//rdfs:label")
```

returns all the nodes in the tree at any depth that are rdfs:label nodes. These nodes can then be post processed to retrieve attributes and values.

Getting Simple VIVO Attributes

The get.vivo.attribute function shown below can be used to retrieve attributes from a parse tree. For example

```
conlon.email <- get.vivo.attribute(conlon.tree,"email")
```

Finds and returns my email address. "email" is an attribute in the VIVO ontology for a person. If we attempted to retrieve an attribute that was not in the ontology, or just not present on the page, the value returned is NA.

You can add a namespace to find attributes on the page from other ontologies. This is often used when the VIVO ontology has been extended locally. So, for example,

```
conlon.eracommonsid <-get.vivo.attribute(conlon.tree,"eraCommonsID",
namespace="http://vivo.ufl.edu/ontology/vivo-ufl")
```

Finds and returns my eraCommonsID. This attribute was added by UF and is defined in the vivo-ufl ontology.

get.vivo.attribute

The get.vivo.attribute function (see below) shows a simple way to get attributes from VIVO pages. It uses getNodeSet using the desired ontology (defaulting to the VIVO ontology). When it finds values, it selects the first one and returns it as a string value. If no values are found, NA is returned.

```
get.vivo.attribute <- function(x,attr,namespace="http://vivoweb.org/ontology/core#")\{
#
#   Given the xml parse tree of a VIVO RDF page in x, find the first attribute attr in
#   the given namespace.  Return NA if no such attribute
#
    xns<-getNodeSet(x,paste("//d:",attr,sep="",collapse=""),c(d=namespace))
    ifelse(length(xns)==0,NA,xmlValue(xns[1]))
\}
```