

# VIVO for Data Analysts

- [Background](#)
- [Getting Rectangles of Data](#)
- [Getting Graphs of Data](#)
- [Repeatable Processes](#)
- [Distributed Queries](#)
- [References](#)

## Background

VIVO represents data as triples. All data is represented and stored in the form subject, predicate, object. All entities are identified by URI. The W3C has developed standards for RDF (Resource Description Framework) for such representations and for various serializations of RDF, including Turtle. If you are unfamiliar with this method for data representation, see the references. A typical VIVO for a large research institution could have well over 10 million triples. The triples are defined using an ontology. The ontology is described here: [Ontology Reference](#) Understanding which triples are needed for an analysis can be challenging. The VIVO community is here to help. Questions regarding data and data extraction using the techniques below can be posted to one of the [VIVO Google Groups](#). You may also wish to contact the VIVO providers at your institution who may be able to help with some of the technologies involved.

## Getting Rectangles of Data

To get rectangles of data, use SPARQL queries. SPARQL is a simple query language designed for use with triple stores. Use of SPARQL is described here: [SPARQL Queries](#)

## Getting Graphs of Data

The entire triple store can be unloaded for use in a local triple store, and for local query. This is recommended for sites wishing to make repeated analyst queries of the data. Community-editions of a triple stores are available with no cost. Stardog is a popular, stable, and free triple store that can be used for this purpose. See <http://stardog.com>

To unload the triple store to a set of triples, use jena3tools, available here: <https://github.com/vivo-project/jenatools>

## Repeatable Processes

To get data from VIVO on a regular basis, you may wish to work with your VIVO providers to create an API that you can use that will provide required data. The [Data Distribution API](#) is designed for this purpose and can be configured to return specified data, including parameterized data via configurable addresses in configurable data formats.

## Distributed Queries

Some applications involve getting data from multiple VIVOS. VIVOS running version 1.10 and above provide a [Triple Pattern Fragments](#) endpoint which can be used to rapidly get all triples from a VIVO matching a triple pattern.

## References

1. RDF 1.1 Primer <https://www.w3.org/TR/rdf11-primer/>
2. RDF 1.1 Turtle <https://www.w3.org/TR/turtle/>
3. Börner, Conlon, Corson-Rikert, and Ding (eds) VIVO: A Semantic Approach to Scholarly Networking and Discovery, Morgan-Claypool Publishers, 2012. 160 pages.
4. Allemang, and Hendler. Semantic Web for the Working Ontologist, second edition. Morgan-Kaufmann Publishers, 2011. 354 pages.
5. DuCharme. Learning SPARQL: Querying and Updating with SPARQL 1.1. O'Reilly, 2011. 235 pages.