/\*! \page tutorial libSBOLj v.02 Tutorial

<center>

03/09/2011

Michal Galdzicki

</center>

\section A A: Write SBOL RDF or Json data from a GenBank File

*<i>The code in this example can be found in libSBOLjUseExample.Main</i>*

The basic SBOL use case is to represent a DNA level design of a DNA construct. Legacy designs of biological constructs are likely to be found in GenBank flatfile format, as DNA sequence editor software (eg ApE, VectorNTI, GD-ICE’s Vector Editor) used to create them has the ability to export this format. Additionally the GenBank format is the *de-facto* standard for exchange of annotated DNA sequence data between software. Therefore, for our first example we convert a GenBank file from the BIOFAB Pilot Project into SBOL RDF and Json serializations.

\subsection html\_out HTML output

## Use SBOLutil class

The SBOLutil class contains static methods, namely utilities for the GenBank to SBOL conversion use case. To use SBOLutil you first instantiate an object of this type.

## Read GenBank file RichSequence

The BioJava library utilities (BioJava-1.7.1) are used to parse a GenBank flat file formatted file and create a RichSequenceIterator. The RichSequence objects correspond to one GenBank record from a file. Since, GenBank files can contain multiple records, the collection of records needs to be iterated over. However, as is the case in our example, there is often one GenBank record in a file. The file BFa\_8.15.gb contains the sequence for a plasmid vector and annotations for a GFP expression cassette. The SBOLutil method fromGenBankFile is a convenience method to read GenBank files.

Iterator aRsIter = SBOLutil.fromGenBankFile("test\\test\_files\\BFa\_8.15.gb");

## Get a new Library object from RichSequence objects

The SBOLutil method fromRichSequenceIter steps through the RichSequenceiterator object and creates a Library SBOL object. The output contains the DnaCompnents for each RichSequence object (ie GenBank Record) and its annotations.

Library aBioFABlib = SBOLutil.fromRichSequenceIter(aRsIter);

More detail: fromRichSequenceIter loops over the iterator using the readRichSequence method to read each RichSequence object, and to map the information contained within RichSequence to an SBOL DnaComponent. Finally, it adds each DnaComponent to a Library which is returned. The readRichSequence method can be used independently if only one RichSequence is expected.

## Serialize the Library object to RDF or Json

Finally, to serialize the SBOL data, found in the Library object, there are two methods that allow you to serialize the Library object either to RDF or to Json.

String jsonString = SBOLutil.toJson(aBioFABlib);

String rdfString = SBOLutil.toRDF(aBioFABlib);

# Read SBOL RDF data into a Library object

Now that the

SBOLservice aS = SBOLutil.fromRDF(rdfString);

Library lib = aS.getLibrary("BioFabLib\_1");

System.out.println("lib Contains: "+lib.getComponents().iterator().next().getName());

# Create a New SBOL Library object

# Access members of Library