MEETINGS WITH JAKE

**06/03**

*To ask:*

-learned about ComponentDefinition object and ModuleDefinition object. As I understand it, CD is related to the actual definitions/annotations/sequences of a plasmid while the MD describes the interactions between two CD objects. The MD refers to the component definitions through a collection?

*Notes:*

-component = definition, module = action

-pw363 has 3 instances, so it will be a ComponentDefinition with each instance of it being a FunctionalComponent

-make the Homespace something like “bu.edu/dasha”, this will be replaced when things get uploaded to SynBioHub by the URL that will contain a standard prefix

-measurements are done as annotations – they need number, unit, what it represents

-each column will be its own MD with each plasmid as a FC

-make it in one Python file

**06/05**

*To ask:*

-created FC version of every plasmid, with 3 FC for the plasmid that has 3 values in the table

-I set the measurement value and the measurement unit as Annotations with a string value

-what should I do for the plasmid that do not have a value listed? Put an Annotation with an empty string (in the case that it could be added on later?) or leave it out altogether?

-can I send you my code to look at?

-Do I have to create new FunctionalComponents though? Are FunctionalComponents unique to the ModuleDefinition they exist in? Should I name them something else to differentiate? Can I do everything in a loop, replacing the name of the MD (aka the name of the column) each time, and then just putting in different data values for each FC in each new column?

*Notes:*

-MD = collections, CD = top level

FC is a child object of MD

-FC name is extension of MD name -- use URI’s

-MD as parent, find Python functions to find all the children of the parent (aka all the FC)

-don’t worry about input/outputs right now

-Maps-To lives in the module or FC:

-they are children of the module definition

-they hook the maps to onto the FC for the Plasmid (bigger piece), so it points to the smaller piece

-a third pointer to it from the module definition

(3 way relationship)

-make the Python code into SBOL XML

-load it into SynBioHub (make accounts), make a private collection and upload, send share link to Jake

-separate what you want to pull out of the sheet from the code that goes through the sheet and extracts information

-module.functionalComponent

For documentation: uploaded first XML file onto SynBioHub and was able to successfully translate first column of Experimental DNA sample tab into sbol format

**6/6/19**

*To ask:*

-what is the end goal regarding the excel import? Should I try to maximize user input? For example right now I added a loop that allows the user to enter the sheet name until they put one in that exists in the excel file.

-I am thinking that this way different pieces of info could be extracted from different locations

-What is the correct “stem” for the units and measurements and stuff?

*Notes:*

-put full URL into python (before the annotation)

-ok to have it in SynBioHub

-add conversion functions to turn illegal characters into % symbols and stuff

-for the object naming: experiment name – sample – 1:dnax

-add Types for the measurements

Figured out how to import excel file into python, finished organizing the rest of the columns into components and modules

**06/10/19**

*To ask:*

-currently the way I am searching for plasmids is by looking for anything that starts with a lowercase ‘p’ followed by a capital letter in the first column, which is an issue because some google spreadsheets have the plasmid without the lowercase p

-other way I was thinking about doing it would be to take anything under the header name ‘plasmid number’ until it reached an empty cell, because as I saw on another spreadsheet there were multiple groups of experiments under one another in the spreadsheet

*Notes:*

-DON’T look with lowercase, look at the structure of the Excel sheet instead (can’t assume there are naming conventions)

-add a thing that directly uploads everything to SynBioHub from the python code

-don’t force a choice—let the user decide if they want to log into SynBioHub personal or LCP collection

-you can also write to Excel—log error messages and write to Excel

-the end goal is to start connecting what I have made to an existing object in LCP or GenBank

06/12: figured out how to upload things directly to SynBioHub by having the user enter a password, reworked the code so it works on multiple test spreadsheets (including ones that don’t follow the specific plasmid naming convention—instead of searching for ‘p’ it searches in the first column, and also it deals with repeat plasmids and data tables positioned underneath one another)

**06/14**

-still a little confused as to how to implement my code in Excel—what spreadsheet will this go into? Is it something you can import into a spreadsheet with existing data, or is it like a “template” spreadsheet?

-how to extract the descriptions for Module Definitions? Right now they are listed as “optional” and are also not organized in the same way on both spreadsheets

-tried testing on JHT10 and there are two columns that are named the same (0.1F20S is twice and 0.01F20S is twice)