**Supplemental Information 2 (User Protocol).**

**Integrated knowledge mining, genome-scale modeling, and machine learning for predicting *Yarrowia lipolytica* bioproduction**

**Authors.** Jeffrey J. Czajka1\*, Tolutola Oyetunde1,2, Yinjie J Tang1†

1Department of Energy, Environmental and Chemical Engineering, Washington University, St. Louis, MO, 63130, USA.

2Takeda Pharmaceutical Company Ltd, 300 Shire Way, Lexington, MA, 02421, USA.

**Corresponding**

**\*Email: jjczajka@wustl.edu**

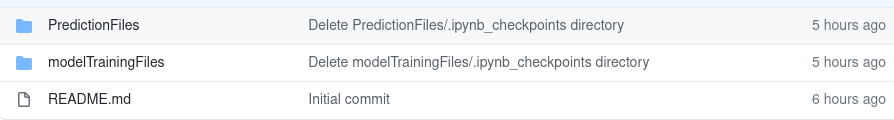
†Email: yinjie.tang@seas.wustl.edu

**Contents: Computational Strain Design Protocol and Tutorial.**

**Computational Strain Design Usage and Tutorial.**

A hybrid genome-scale model and machine learning (GSM-ML) model was developed. The final model, M21iYL (described in the main text) can be leveraged to screen and predict engineered strain design outputs. To facilitate this, a template was developed and deposited at the GitHub repository: <https://github.com/jjczajka/YlipCSD>. This file aims to provide an example on how to use the code for strain design predictions. An important note for using this platform. M21iYL was trained on strain designs that all resulted in some product production. Thus, the model has the implicit assumption that the submitted designs are capable of producing product (i.e., heterologous enzymes are functional).

Follow the GitHub link (<https://github.com/jjczajka/YlipCSD>) to find instructions for installing the environmental dependencies within the README file. The prediction code is in the depository *PredictionFiles* (**Fig. 1**).

**Figure 1. View of the GitHub repository.**

Within the *PredictionFiles* directory are several files (**Fig. 2)**:

computationalStrainPrediction.ipynb – code for generating the predictions from a strain design.

Supplemental Excel File 6- CSD Template.xlsx – worksheet where strain information and designs to be screened are input.

The rest of the files are not necessary for the user to access (but are dependencies for the rest of the program):

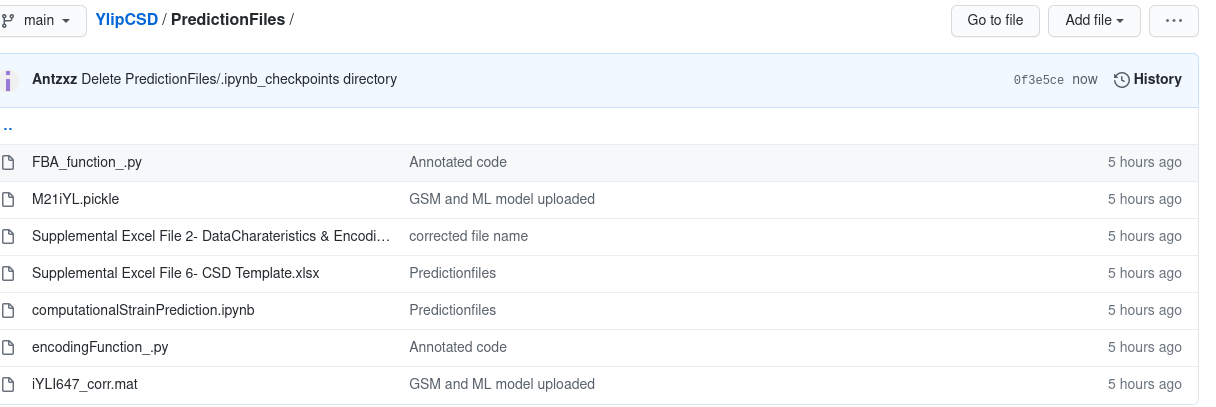
M21iYL.pickle – GSM-ML trained file

FBA\_function\_.py – function for generating the required FBA features

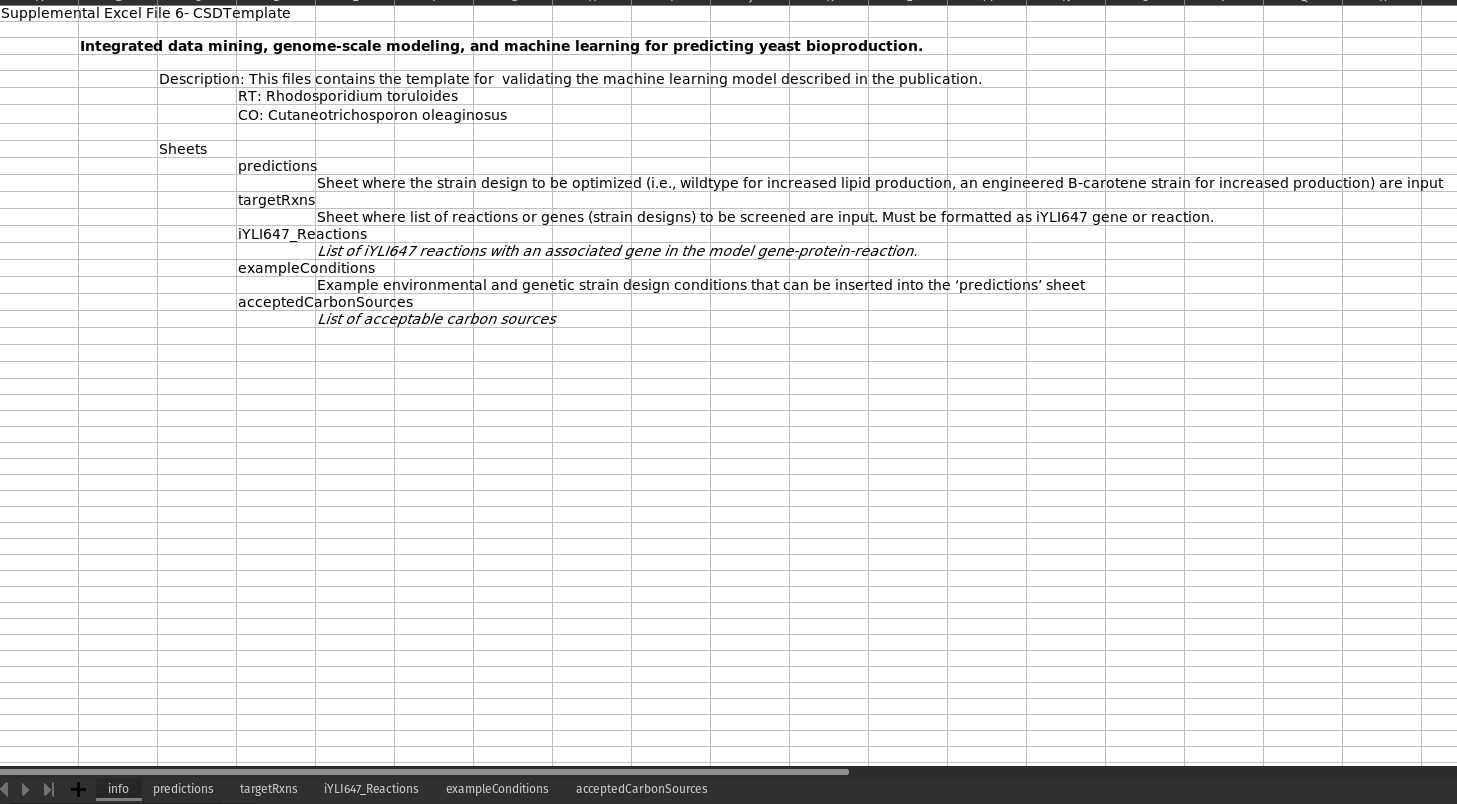
encodingFunction\_.py – function for encoding the information into features for ML training

iYLI647\_corr.mat – GSM structure

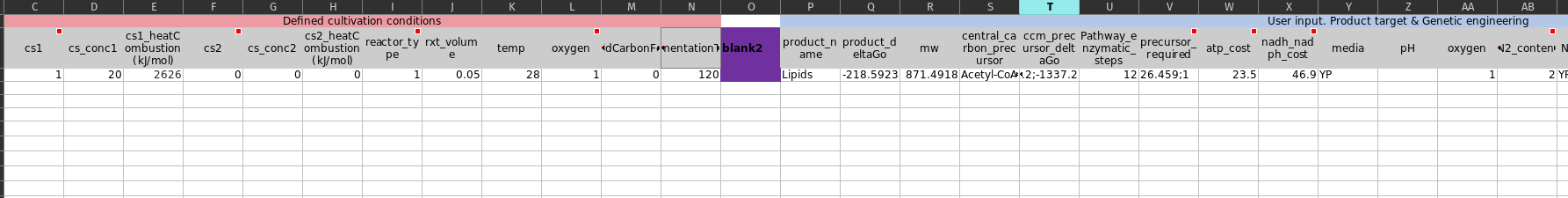
Supplemental Excel File 2- DataCharateristics & Encoding.xlsx – worksheet needed for generating encoded features.

**Figure 2. View of the GitHub prediction file directory.**

The *Supplemental Excel File 6- CSD Templat*e spreadsheet is the data input file. The starting strain (Wildtype or previously engineered strain) is input to the “**predictions**” sheet within the spreadsheet. The “**targetRxns**” sheet is where the knock-out reactions are input. Multiple strain designs can be input for screening. Opening the file will show the following screen (**Fig. 3**)

**Figure 3. View of the initial info sheet contained within the Supplemental Excel File 6- CSD Template.**

Navigating to the “**predictions**” sheet will allow the user to input the initial strain design (**Fig. 4**). Computational knockouts will be conducted in this initial design. There are two sections within the spreadsheet, the “Defined cultivation conditions” and the “User input. Product target & Genetic engineering” sections. The Defined cultivation section is where users can set the environmental conditions. The default is set to 20 g/L glucose in 50 mL of YPD medium and a shaking flask vessel.

**Figure 4. View of predictions spreadsheet where the initial strain is input.**

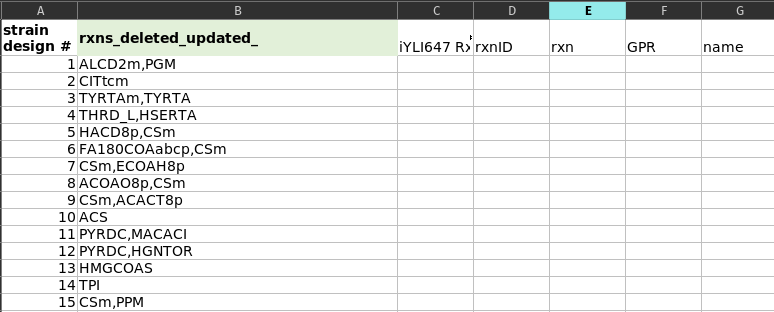
The “User input. Product target & Genetic engineering” section is where the product and strain design construct are input. The default target is “Lipid” production from a wild-type strain with no previous genetic engineering performed. The “**exampleConditions**” sheet provides more cultivation conditions that can be copied directly to the ‘predictions’ sheet as well as an example of a B-carotene engineered strain (**Fig. 5**).

**Table

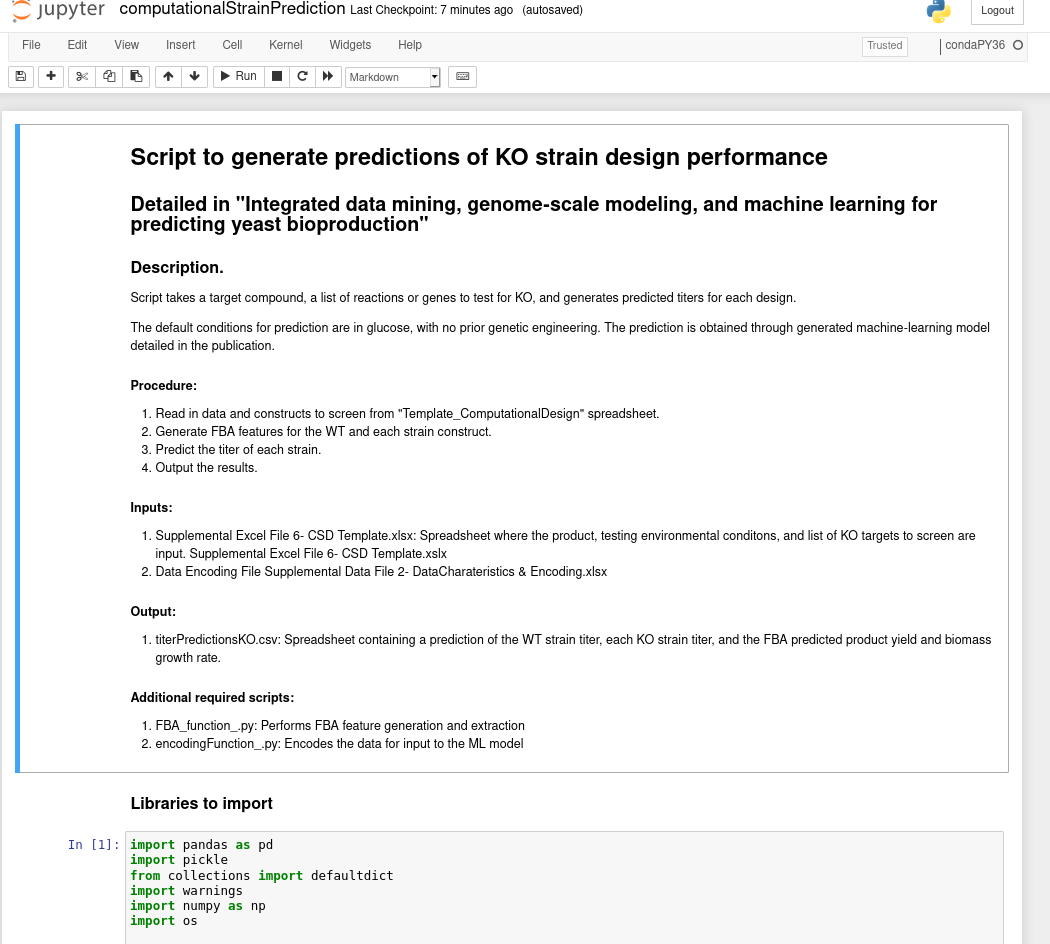
Description automatically generated with medium confidence**

**Figure 5. View of ‘exampleConditions’ sheet which contains example environmental, cultivation, and genetic conditions to input as the initial strain.**

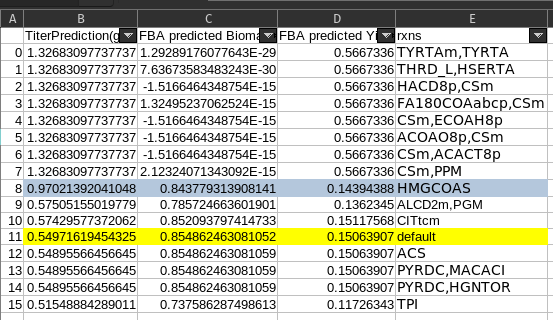
Navigating to the “**targetRxns**” sheet (**Fig. 6**) allows the user to input target knock out strain designs to test. This section can be left blank to only evaluate the strain input to the “**predictions**” sheet. The “rxns\_deleted\_updated\_” column is where users can insert the strain designs. For example, shown in the **Fig. 6** is 15 single and double knock-out strains (algorithm is not limited to only single or double knock-outs) to be screened. The program will also predict the initial strain production for 16 total predictions.

**Figure 6. View of targetRxns where the knock-outs to be screened are input.**

Running the “computitionalStrainPrediction.ipynb” Jupyter Notebook will predict the lipid output of the resulting 16 strain designs (default + 15 knock-out strains, **Fig. 7**).

**Figure 7. Screenshot of the computationalStrainPrediction.ipynb jupyter notebook.**

Lastly, the algorithm will output the strain design predicted titer (g/L), the FBA predicted biomass (1/h), and the FBA predicted product yield (mmol/(gDCW·h)) (**Fig. 8**). The 15 reaction set examined indicated that 10 of the designs will produce more lipid than the wildtype strain. However, seven of the 10 predicted strains show essentially no biomass growth. The FBA predicted biomass was not a feature in the trained model, so the FBA biomass growth does not affect the prediction. Thus, the engineer should further evaluate whether the knockouts would allow for physiological growth or if a knock-down strategy would be more appropriate. Three of the reactions have similar biomass growth rates and increased titer. The best performing strain involves knocking out the HMGCOAS reaction.



**Figure 8. Output of the CSD algorithm.** Titers (g/L), FBA predicted biomass (mmol/(gDCW·h)), FBA predicted yield, reaction list (rxns). The titers were sorted from the highest to lowest prediction.