Chapter 23

Flapjack: Data Management and Analysis for Genetic Circuit Characterization

Carolus Vitalis, Guillermo Yáñez Feliú, Gonzalo Vidal, Macarena Muñoz Silva, Tamara Matúte, Isaac Núñez, Fernán Federici, and Timothy J. Rudge

Abstract

Flapjack presents a valuable solution for addressing challenges in the Design, Build, Test, Learn (DBTL) cycle of engineering synthetic genetic circuits. This platform provides a comprehensive suite of features for managing, analyzing, and visualizing kinetic gene expression data and associated metadata. By utilizing the Flapjack platform, researchers can effectively integrate the test phase with the build and learn phases, facilitating the characterization and optimization of genetic circuits. With its user-friendly interface and compatibility with external software, the Flapjack platform offers a practical tool for advancing synthetic biology research.

This chapter provides an overview of the data model employed in Flapjack and its hierarchical structure, which aligns with the typical steps involved in conducting experiments and facilitating intuitive data management for users. Additionally, this chapter offers a detailed description of the user interface, guiding readers through accessing Flapjack, navigating its sections, performing essential tasks such as uploading data and creating plots, and accessing the platform through the pyFlapjack Python package.

Key words Data management, Genetic circuit characterization, SBOL, Web application, Visualization tools

1 Introduction

Flapjack [1] plays a crucial role in the Design, Build, Test, Learn (DBTL) cycle of engineering synthetic genetic circuits. By offering a comprehensive set of features for managing, analyzing, and visualizing kinetic gene expression data and associated metadata, Flapjack facilitates the seamless integration of the *Test* phase with the *Build* and *Learn* phases. With its user-friendly interface and interaction with external software tools such as LOICA [2] and SynBioHub [3], Flapjack is an invaluable resource for advancing research in synthetic biology.

Jeffrey Carl Braman (ed.), *Synthetic Biology: Methods and Protocols*, Methods in Molecular Biology, vol. 2760,
https://doi.org/10.1007/978-1-0716-3658-9_23,

Flapjack can handle large volumes of measurement data and associated metadata. Researchers can effectively collect, organize, and analyze data from numerous circuits, conditions, assays, and laboratories. This enables a comprehensive understanding of circuit behaviors in diverse contexts, leading to improved performance and functionality of genetic circuits.

Flapjack provides an intuitive interface for exploring and analyzing measurement data. Its interactive visualization tools empower researchers to dynamically examine and interpret results, uncovering patterns and correlations within the data. This capability aids in the identification of potential circuit optimizations and design improvements, ultimately enhancing the efficiency and effectiveness of the DBTL cycle.

Moreover, the Flapjack platform can interact with different tools and workflows via its REST API and Python package. This integration enables researchers to incorporate Flapjack into existing workflows, leveraging its capabilities with other tools. By facilitating interoperability, Flapjack enhances the efficiency and versatility of the DBTL cycle, allowing researchers to fully leverage the potential of synthetic genetic circuits in various applications.

2 Materials

Flapjack can be used entirely from a web browser using the front-end user interface. For more complex analysis, post-processing and custom plotting users may optionally install the pyFlapjack Python package. Installing pyFlapjack via pip will automatically install some required packages. These prerequisites include Python 3.7 or a later version. We recommend the use of an environment manager, such as Anaconda (https://www.anaconda.com/).

2.1 Installation

The pyFlapjack Python package is distributed using the Python Package Index (PyPI), which utilizes the Pip Installs Package (PIP) for installation and update management. pyFlapjack can be installed using the following commands:

```
pip install pyflapjack
```

To verify that the installation was successful, users should be able to run the following command with no errors:

```
import flapjack
```

3 Methods

3.1 Data Model

The data model used in Flapjack is analogous to typical laboratory workflows and data organization, making data management intuitive to the user. The data model can be seen in Table 1.

3.2 Accessing Flapjack

Access the Flapjack page at http://flapjack.rudge-lab.org/. The front page is shown in Fig. 1.

3.2.1 Creating an Account

Here, the user will find two buttons. Select "Ready to get started? Sign Up!" to create a new account.

On the next page, provide the Username, Email, and Password, and click "Register."

3.2.2 Logging in to an Existing Account

If the user has already registered in the platform, simply select the button "Already have an account? Log In!" Enter the Username and Password on the next page and then press "Login."

3.3 Home Page

Figure 2 shows Flapjack's home page after signing up. On the home page, three buttons are found, each one with its description:

- Upload: Kinetic data from a microplate reader and other sources can be uploaded along with associated metadata.
- Browse: Browse published studies, assays, and available DNA.
- Search and Analyse: Query public and private data to visualize, analyze, and model.

3.4 Preparing and Uploading Data in Flapjack

Flapjack is a robust platform that facilitates the efficient uploading and management of data obtained from plate readers. This chapter provides a detailed guide on preparing and uploading data files into Flapjack. It covers the necessary steps for file preparation, including adding relevant sheets with assay information. It outlines the process of uploading the file to Flapjack, creating new studies, and associating data with existing information in the platform's database. By following these instructions, researchers can seamlessly integrate their experimental data into Flapjack for further analysis and exploration.

3.4.1 Preparing the Data File

Data obtained directly from a plate reader can be uploaded to Flapjack as an Assay after minor adjustments.

First, once the Excel file is obtained from the plate reader, we will get something like Fig. 3. It is necessary to add some sheets with relevant metadata about the Assay to this file.

The first sheet with our data must be renamed to "Data," so Flapjack can identify the data it contains. This step is crucial; otherwise, Flapjack will not recognize the file, and the upload cannot be continued.

Table 1
Flapjack models and their attributes. Attributes in italics are optional when creating an object of that type

Model	Attributes	Description
Study	name description *doi* owner *shared_with* public	A project, for example, a paper or report, that corresponds to a particular question a researcher wants to address.
Assay	study name machine description temperature	Measurement experiments performed to explore different study aspects. This includes replicates and varying experimental conditions.
Media	owner name description	Composition of the substrate which drives the genetic circuit, media in the case of live cell assays, or extract for cell-free experiments.
Strain	owner name description	The chassis organism—if any—hosting the genetic circuit.
Chemical	owner name description *pubchemid*	Any supplementary chemicals that interact with components of the genetic circuit.
Supplement	owner name chemical concentration	
Dna	owner name *sboluri*	Describes the synthetic DNAs encoding a genetic circuit, including links to part composition and sequence via the corresponding SBOL URIs.
Vector	owner name dnas	
Sample	assay *media* *strain* *vector* supplements row col	Corresponds to the basic unit subject to measurement, for example, a colony or a well in a microplate.
Signal	owner name description color	
Measurement	sample signal value time	The raw measurement value recorded for a particular sample during an assay at a particular time.

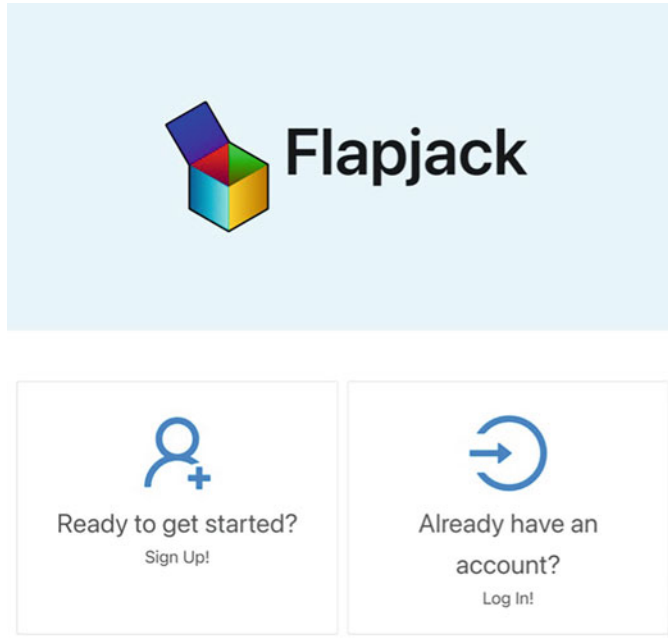


Fig. 1 Flapjack frontend front page, seen before logging in

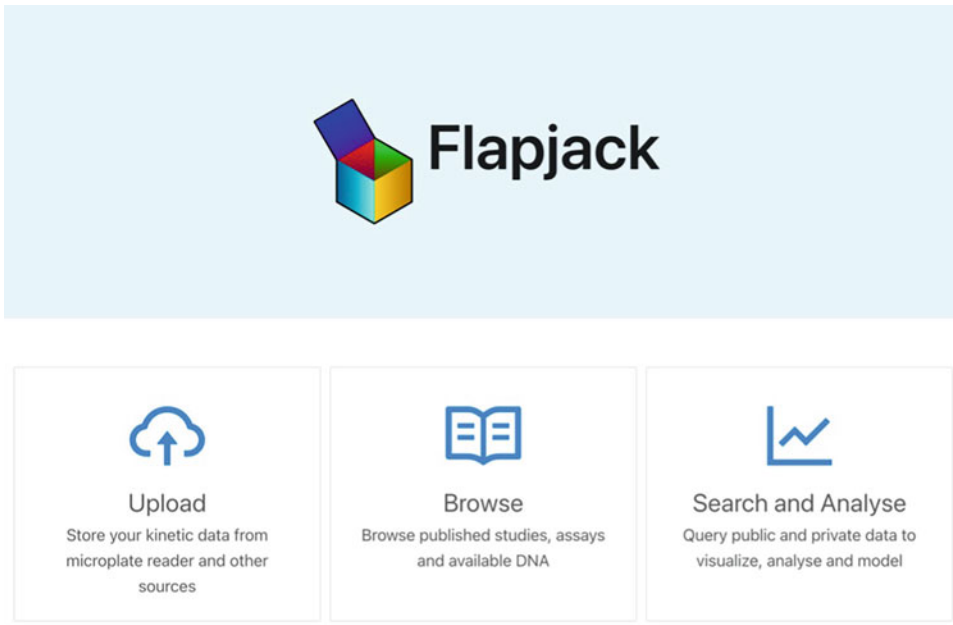
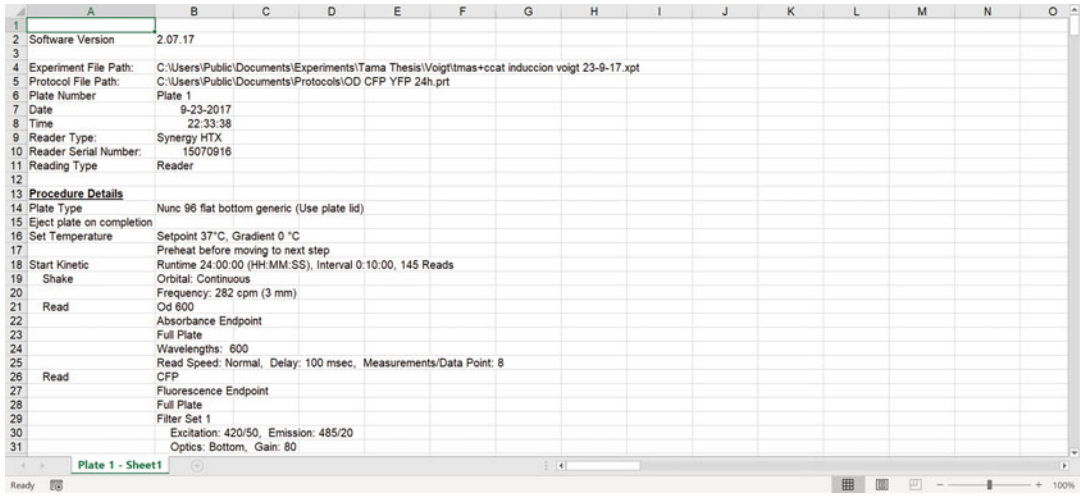


Fig. 2 Flapjack home page, seen after logging in



	A	B
1		
2	Software Version	2.07.17
3		
4	Experiment File Path:	C:\Users\Public\Documents\Experiments\Tama Thesis\Voigt\tmas+ccat induccion voigt 23-9-17.xpt
5	Protocol File Path:	C:\Users\Public\Documents\Protocols\OD CFP YFP 24h.prt
6	Plate Number	Plate 1
7	Date	9-23-2017
8	Time	22:33:38
9	Reader Type:	Synergy HTX
10	Reader Serial Number:	15070916
11	Reading Type	Reader
12		
13	**Procedure Details**	
14	Plate Type	Nunc 96 flat bottom generic (Use plate lid)
15	Eject plate on completion	
16	Set Temperature	Setpoint 37°C, Gradient 0 °C
17		Preheat before moving to next step
18	Start Kinetic	Runtime 24:00:00 (HH:MM:SS), Interval 0:10:00, 145 Reads
19	Shake	Orbital: Continuous
20		Frequency: 282 cpm (3 mm)
21	Read	Od 600
22		Absorbance Endpoint
23		Full Plate
24		Wavelengths: 600
25		Read Speed: Normal, Delay: 100 msec, Measurements/Data Point: 8
26	Read	CFP
27		Fluorescence Endpoint
28		Full Plate
29		Filter Set 1
30		Excitation: 420/50, Emission: 485/20
31		Optics: Bottom, Gain: 80

Plate 1 - Sheet1

Ready

Fig. 3 An example of an Excel data file produced by a plate reader

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Media	1	2	3	4	5	6	7	8	9	10	11	12
2	A	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
3	B	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
4	C	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
5	D	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
6	E	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
7	F	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
8	G	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol
9	H	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicero	M9-glicerol

Fig. 4 Media metadata sheet, which is added to the original Excel file from a plate reader to specify the media in which each Sample (well) was grown

The first sheet to be added is the "Media," shown in Fig. 4, which will indicate the media used in each well of the plate. As the plate reader follows an arrangement of rows from A to H and columns from 1 to 12, we will follow this arrangement, reserving the first cell A1 as the sheet's title. Thus, in cells A2 to A9, we will have the axis of the rows, while from B1 to M1, we will have the axis of the columns. Then, we will fill each cell with the media that was incorporated. If no media has been incorporated, we write "None" in the cell. In this case, the same media was incorporated in all the wells; therefore, we repeat the value in all the cells.

We continue with the next sheet, which we will call "Strains." Following the previous arrangement, we place the cell strain used in each well, and again if no strain has been used in a particular well, we will fill this cell with the value "None." In this case, our last row contains the value "None," indicating that these Samples are controls. Flapjack will automatically recognize these control Samples and use them to correct for background fluorescence and optical density.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	DNA 1	1	2	3	4	5	6	7	8	9	10	11	12
2	A	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd
3	B	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd	StdStd
4	C	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2
5	D	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2	TMA2
6	E	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5
7	F	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5	TMA5
8	G	None	None	None	None	None	None	None	None	None	None	None	None
9	H	None	None	None	None	None	None	None	None	None	None	None	None
10													
11	DNA 2	1	2	3	4	5	6	7	8	9	10	11	12
12	A	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT
13	B	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT
14	C	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT
15	D	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT
16	E	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT
17	F	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT	CcaT
18	G	None	None	None	None	None	None	None	None	None	None	None	None
19	H	None	None	None	None	None	None	None	None	None	None	None	None

Fig. 5 DNA metadata sheet added to the original Excel file from a plate reader, showing how multiple plasmids can be present in each Sample (well). "None" indicates that the well does not contain cells with synthetic DNA (negative controls)

Next, we generate the DNA sheet, and in it, we place—again following the previous arrangement—the plasmids we have used. If we have used more than one plasmid, we generate a new table below, leaving an empty row. In our example, cells were co-transformed with two plasmids; hence, we have two tables "DNA 1" and "DNA 2," as shown in Fig. 5.

Finally, we created the sheet "Chemicals," where we will add the Chemicals used, following the same arrangement as before. Chemicals are any substance that is added to the media as a Supplement at a given concentration, such as an inducer of gene expression. Be careful; the concentrations must be in molar; consider that when entering the values.

In the end, we should have the following sheets: Data, Media, Strains, DNA, and Chemicals.

3.4.2 Uploading the File

We are now ready to upload the file to Flapjack.

Once logged in (*see* Subheadings 3.2.1 and 3.2.2), place the cursor over your username, where the "Upload" option will be displayed, as shown in Fig. 6. After clicking, the page shown in Fig. 7 will be displayed. The fields to be filled are covered in the next sections.

Study

A Study represents a group of Assays that were performed for some research purpose, perhaps to assess the effect of growth conditions on gene expression. Select one of the studies available from the drop-down menu. Here, only studies owned or shared with the user will be displayed. If the study does not previously exist, select the "Create new study" button. In our example, we will create a new one named "Example."

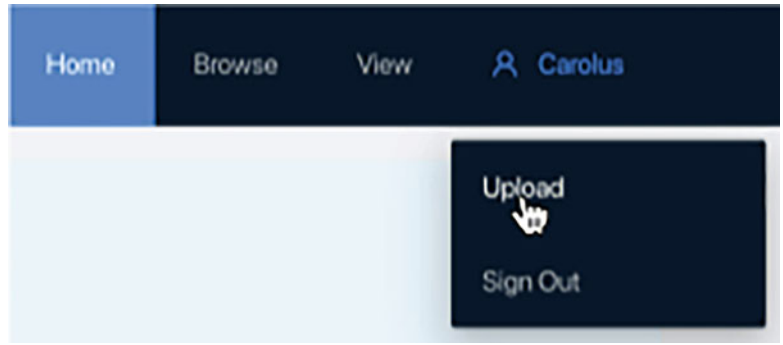


Fig. 6 Flapjack dropdown menu showing the Upload option

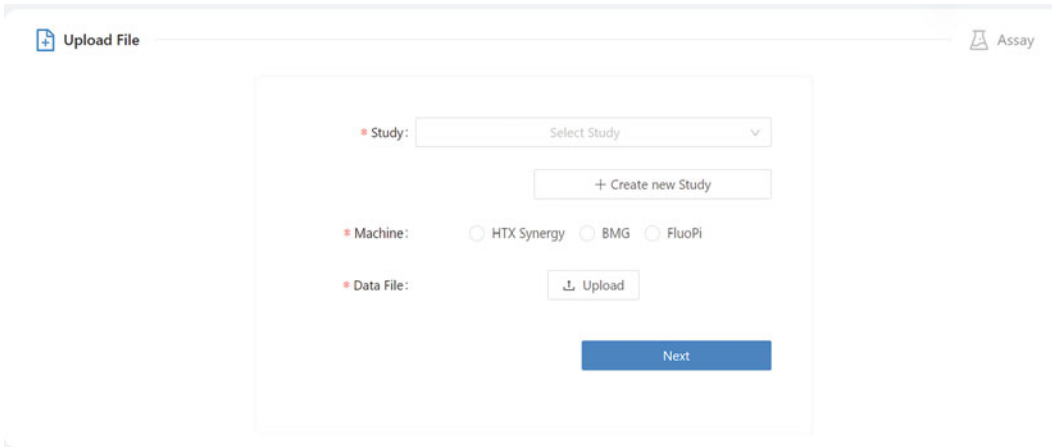


Fig. 7 Upload File dialog showing the first metadata entry form

Creating a New Study

For a new study, three fields are required: Name, Description, and DOI. The user will be asked to provide these fields, as shown in Fig. 8. The DOI is optional but good practice if available. There is also an option to make the study public. After entering the details, select "Create Study."

Machine

It indicates on which type of instrument the experiment was conducted. This is important because each machine delivers different files, and Flapjack treats them accordingly. In our example, it is an HTX Synergy plate reader.

Data File

The file to be uploaded is selected. The user can select the trash can icon if the file is to be removed and replaced with another one. Then select "Next."

Now, we will be asked for the details of the Assay. We must fill in the fields: Name, Description, and the temperature in degrees Celsius at which the experiment was conducted. All of them are mandatory.

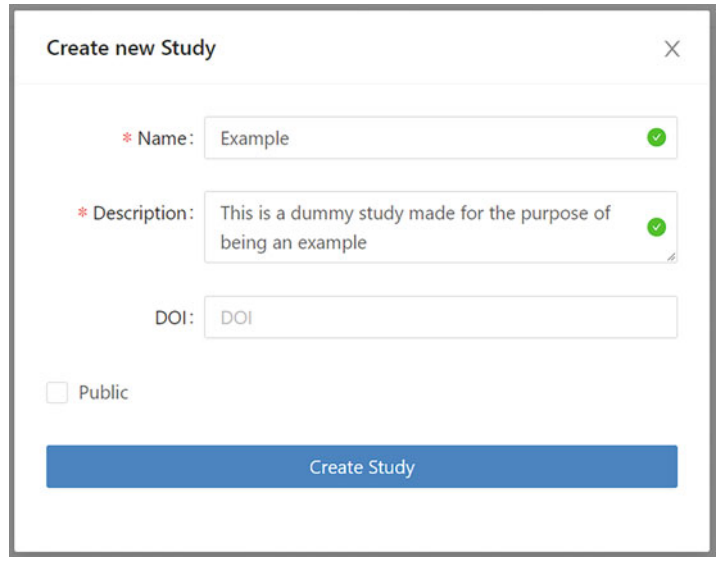


Fig. 8 Create new Study popup window showing metadata entry form for describing a new Study

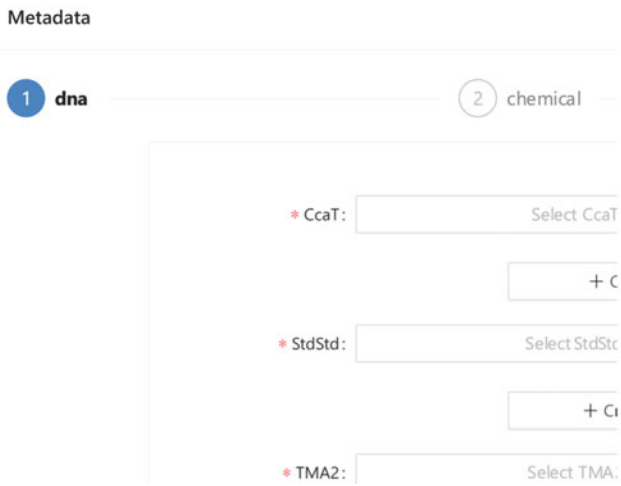


Fig. 9 Metadata entry form for describing the DNA used in the Assay to be uploaded

Remember that we can have several Assays per Study, so we must place a name accordingly. In our example, we will call it "Assay 1," but feel free to provide a more detailed name. Once this is done, press Submit.

After the file has been uploaded to the platform, the window shown in Fig. 9 will appear with data to be completed according to our particular file. In this window, the user must match the data—DNA names, for instance—with the data available on the server using the drop-down menu or the search function. If the value is

Fig. 10 Popup window to create a new DNA

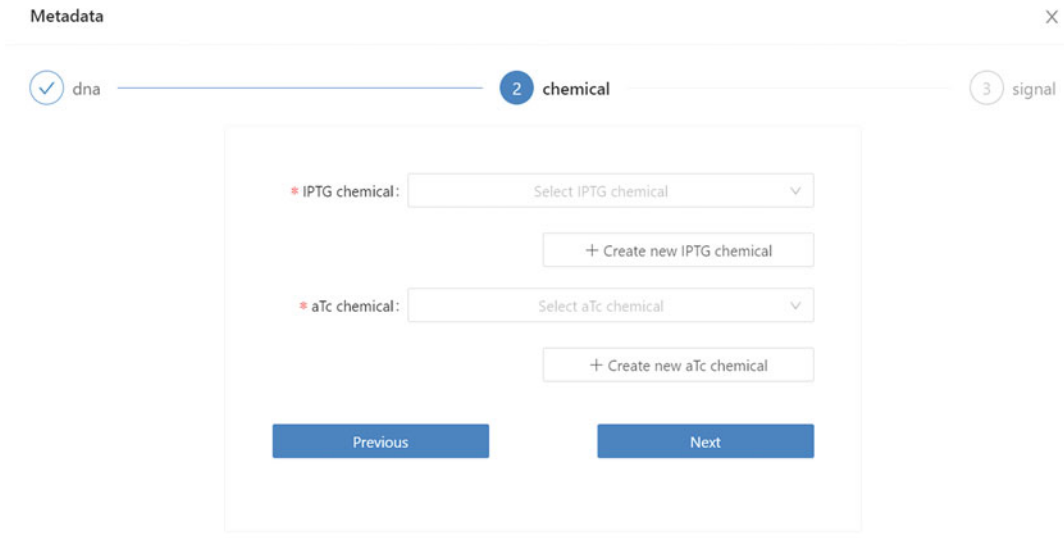


Fig. 11 Metadata entry form to describe chemicals (e.g., inducers) added to Samples in the Assay

not found, it can be added with the button "Create new plasmid," where the name entered in the Excel file for that plasmid is meant to be entered here.

The following pop-up windows will appear when a plasmid value is created, shown in Fig. 10. The name to be assigned to it (required) and an SBOL Uri (optional) must be entered here, then "Create DNA" must be pressed.

Then, we must enter the Chemical details, following the same steps as above, as indicated in Fig. 11.

If the chemical is not found, select the "Create new chemical" button, where a pop-up window will appear, as depicted in Fig. 12, with fields to fill in: Name, Description, and PubChem ID (optional).

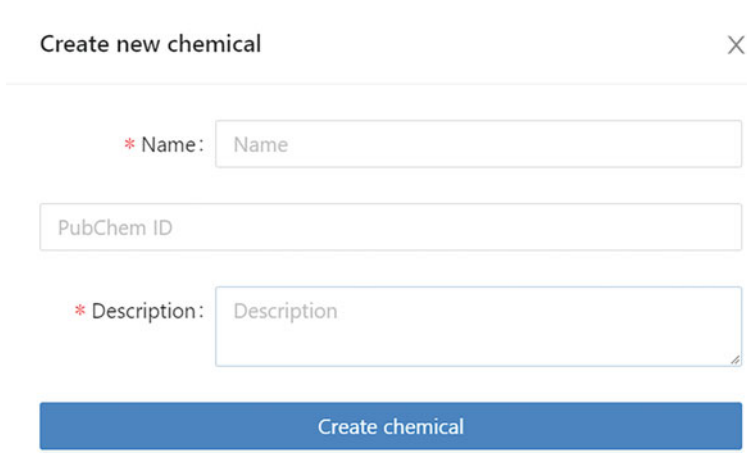


Fig. 12 Popup window to create a new chemical (e.g., inducer)

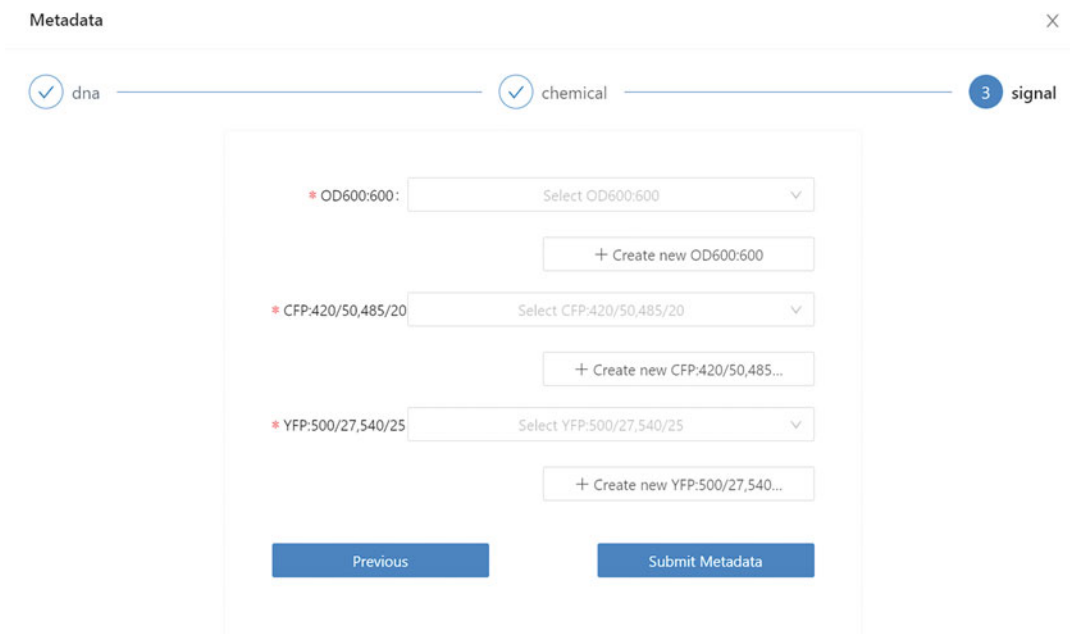


Fig. 13 Metadata entry form to describe the Signals measured in the Assay to be uploaded

Once this is completed, press "Next" to go to the next section.

Finally, we fill in the fields for signal, as shown in Fig. 13. Again, we can create a new signal if not found, as illustrated in Fig. 14. To do this, select the "Create new signal" button where the signal is equivalent to the value. Here, we must complete the fields: Name, Description, and Color. All are required.

The "Submit Metadata" button should be selected after the data entry is verified to be correct. A loading bar will appear indicating the progress of the upload, and that is it; the data has been successfully uploaded to Flapjack.

Fig. 14 Popup window to create a new Signal

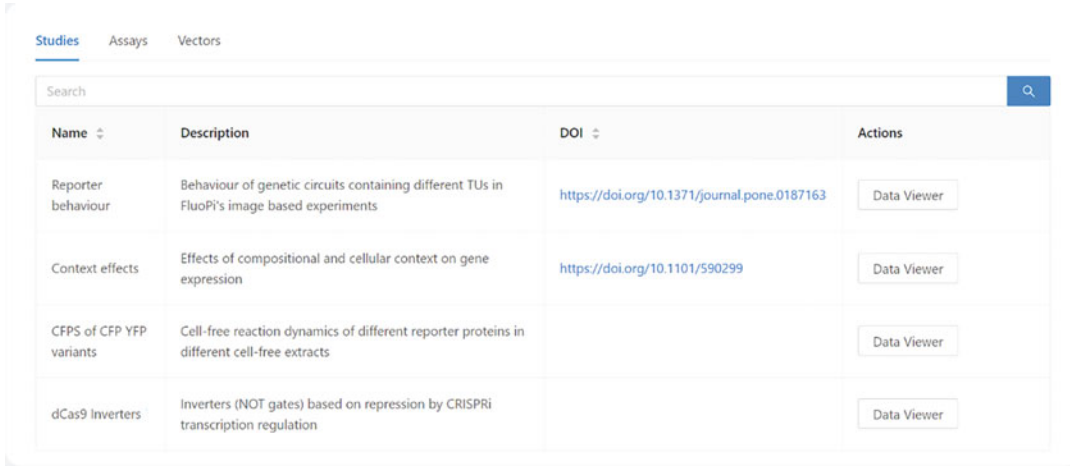


Fig. 15 Flapjack Browse Page, which lists Studies, Assays and Vectors available to the user. The Studies tab is selected. The other two tabs—Assays and Vectors—are shown

3.5 Browse Page

All the Studies, Assays, and Vectors available for review are displayed in the Browse page as seen in Fig. 15, either because they are public or because they have been shared with this account. The Search box can be used in both the Studies and Assays sections. To remove the search, simply select the x button.

3.5.1 Studies

The list of studies can be accessed in the first tab, corresponding to Studies. This tab has the following parameters: Name, Description, and DOI, which allow us to identify the different studies.

Actions

Next to these parameters, we find the Subheading *Actions* which allows us to see the different actions we can perform in the study. Depending on whether the Study's owner is the user or not, the

available Actions will vary. The image below shows that an owner can manage the data. For instance, the owner can access the *Manage* button, which displays three options: *Share*, *Make public*, or *Delete*.

Share

When sharing a study, a pop-up window will appear. The user email of the person with whom the study will be shared is entered here. This window also shows a list of all the people who were given access to this study.

The message "Study successfully shared" will be displayed on the screen after pressing the Share button. Access to the study can be removed by selecting the trash can icon next to the user's email if the decision to share it with someone is changed at any point. The message "Study unshared successfully" will appear on the screen after the trash can icon has been pressed.

Make Public

The study can be made public by selecting this button so that anyone can browse its content without individual sharing. This function is independent of the Share function, so access to the study will be maintained by those previously indicated if the study is made private again after being public.

Delete

The study will be removed from the database by this button. No confirmation is required, and the action cannot be undone. All associated Assays and their measurement data will be removed. Caution is advised.

3.5.2 Assays

An Assay is a procedure which measures Signals produced by a set of Samples, such as a kinetic plate-reader experiment. In this tab, we can see the name. ID, description, study to which the assay is associated, temperature, machine, and button to view the data.

3.5.3 Vectors

Once in the Vectors tab, we can see the Vectors, which are the synthetic DNA transformed into the cell Strain. A SynBioHub URI (link) is associated with each Vector, which allows quick access to the corresponding SBOL representation for more detailed metadata. SynBioHub is shown in Fig. 16. One Transcriptional Unit is reviewed in this example.

3.6 View Page

The Data Viewer button under the Actions tab can be selected to view the data associated with a Study, Assay, or Vector. The Analysis page is then reached. It can also be accessed via the View tab on the top bar.

A new tab, named *Analysis* #, is generated when the View Page is opened, where # is the number of the tab according to when it was generated.

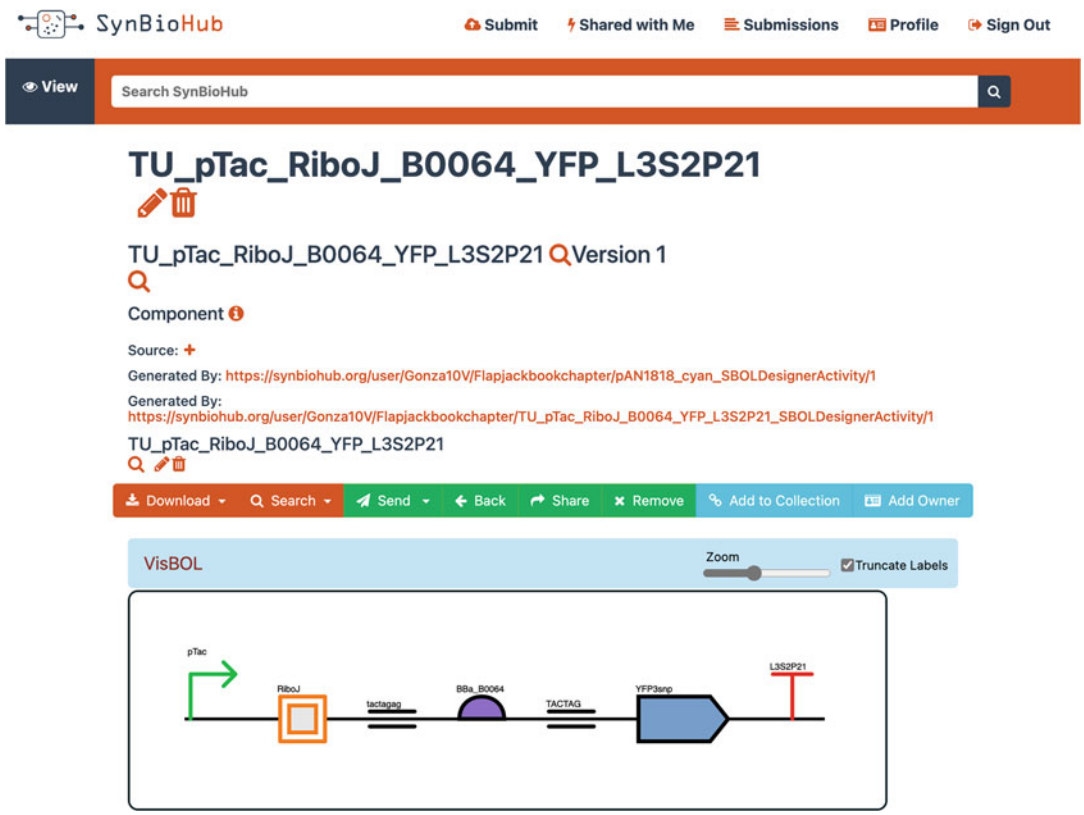


Fig. 16 SynBioHub entry linked from Flapjack, which can be accessed via the Browse page for Vector

The tab can be renamed to any name by pressing the pencil icon (Fig. 17).

The View page has three options: Query, Analysis, and Plot.

3.6.1 View Page Filters and Options

Query

The Query section is subdivided into six filters to select a specific set of measurement data, which are:

- Studies
- Assays
- Vector
- Strain
- Media
- Signal

In this example, we will select *Reporter behavior* in the Study filter, as seen in Fig. 17.

All related Assays (and their components, such as Vector, Strain, Media, and Signal) will be automatically selected by selecting this Study. Filters can be removed by checking each field as desired.

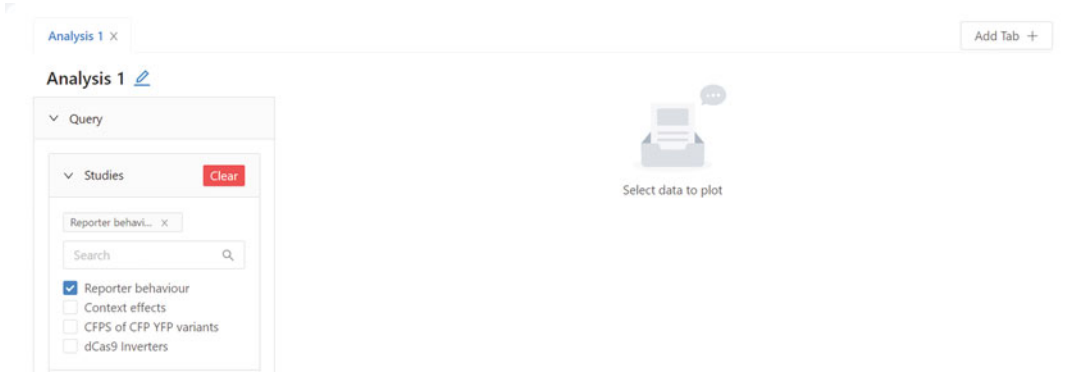


Fig. 17 Flapjack View page. Example of a Query section filter. Here, the *Reporter behaviour* study has been selected

Analysis

The data can be analyzed by selecting the desired option from here. The available analysis types are listed below. More details can be obtained in references [1] and [4].

- Expression Rate (Indirect)
- Expression Rate (Direct)
- Expression Rate (Inverse)
- Rho
- Alpha
- Induction Curve
- Heatmap
- Kymograph

Plot Options

- Normalize: The default option is no normalization (None), but you also have options for Temporal Mean, Mean/std, and Min/Max.
- Subplots: Data can be grouped into different subplots according to the user's plot needs. The Signal is the default option, but Study, Assay, Vector, Media, Strain, and Supplement are also available.
- Lines/Markers: Different line and marker colors can be plotted according to the user's needs. The default option is to group by Vector, but other available options are Study, Assay, Media, Strain, Supplement, and Signal.
- Plot: This determines whether to show all Samples or to compute their mean and/or standard deviation. Mean +/− std is the default option chosen, but Mean and All data points are also available.

Each filter or option has a Clear button to clear the settings that have been selected.

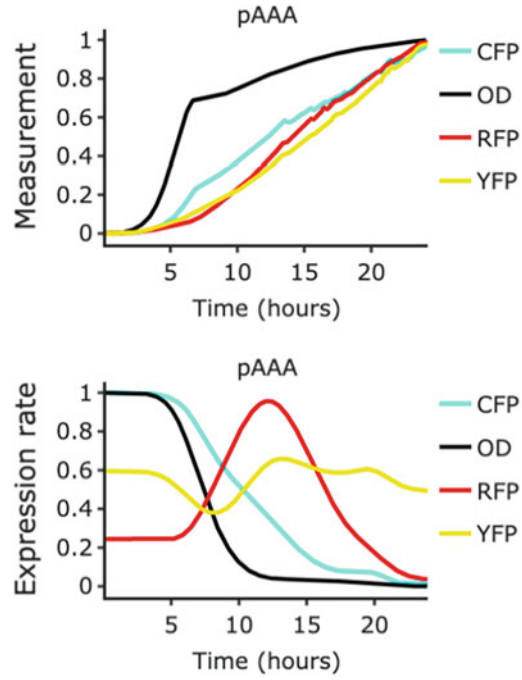


Fig. 18 Plots downloaded from Flapjack's front end, showing raw measurement data (top) and computed gene expression rate (CFP, RFP, and YFP) and growth rate (OD)

3.7 Creating a Plot

Now that we have familiarized ourselves with the interface, we will recreate the plot shown in Fig. 18 step by step using the Flapjack frontend.

Once we have previously logged into our account (*see* Subheadings 3.2.1 and 3.2.2), we must go to the View tab and, under the option "Studies," select the study "Context effects."

Now we can click on the Plot button. Immediately a Plot will be generated, and we will see the loading percentage on the screen. However, when plotting with the default options, we get a plot that does not look much like our target plot. This is because we have not defined the best parameters for the display of this particular case.

We can see that the default plot shows us 14 vectors (pAAA, pGEA, pEAA, pECA, pEDA, pEFA, pGAA, pGDA, pGCA, pGFA, pBAA, pBCA, pBDA, and pBFA), which is correct because this Study was made with all those vectors, but this time we only want to visualize the vector pAAA. Hence, the first thing we will do is to deselect the other vectors. To perform this task, we open the option Vector and remove all vectors except pAAA, then proceed to click on Plot again, obtaining Fig. 19.

We now have only the data that interest us; however, these are separated into four plots because the default option groups subplots according to Signal. To have them all together in one plot, we

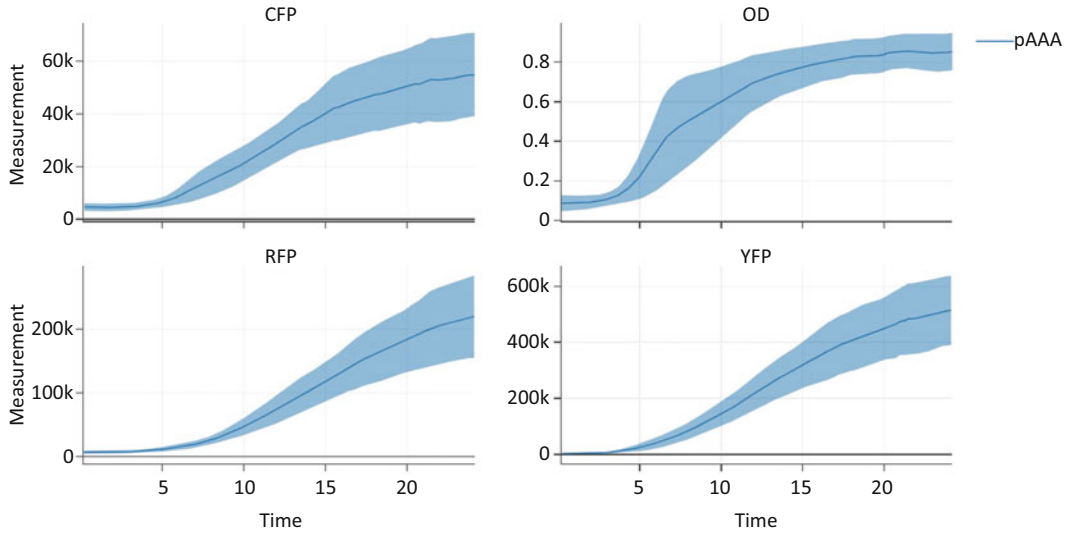


Fig. 19 Plots from the Context effects study showing mean and standard deviation of measurements of fluorescence (CFP, RFP, and YFP) and optical density (OD)

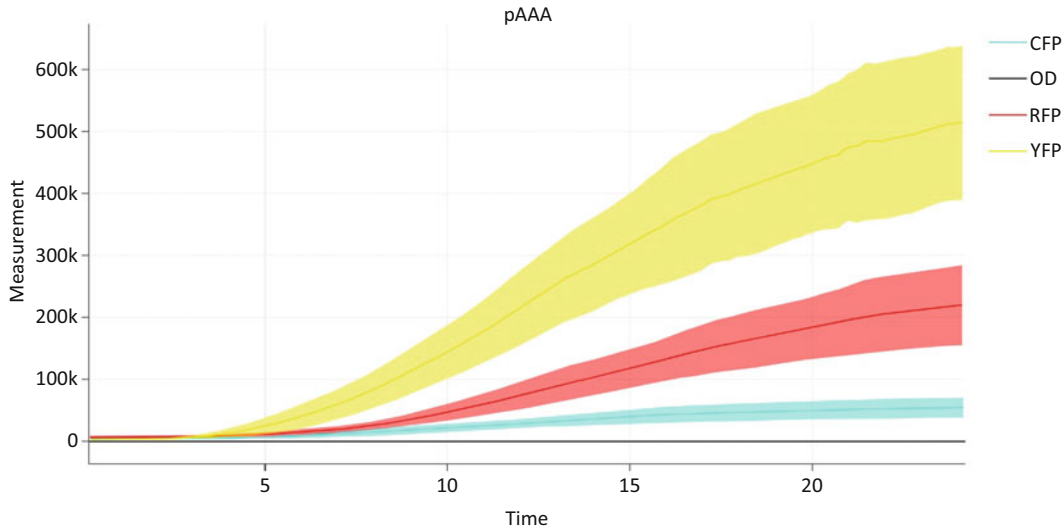


Fig. 20 Plot of mean and standard deviation of all the Signals measured for Vector pAAA in the Context effects Study

must group them by Vector. To do this, we go to the Plot Options section and, under Subplots, select "Vector," Also, under Lines/Markers, we will select "Signal," to show each measurement channel in a different color. We proceed to select Plot again, and we should have something like Fig. 20.

It can be observed that we are approaching the original plot. However, we see the raw measurements here, so we must still normalize the data for better visualization. For this, we go to the

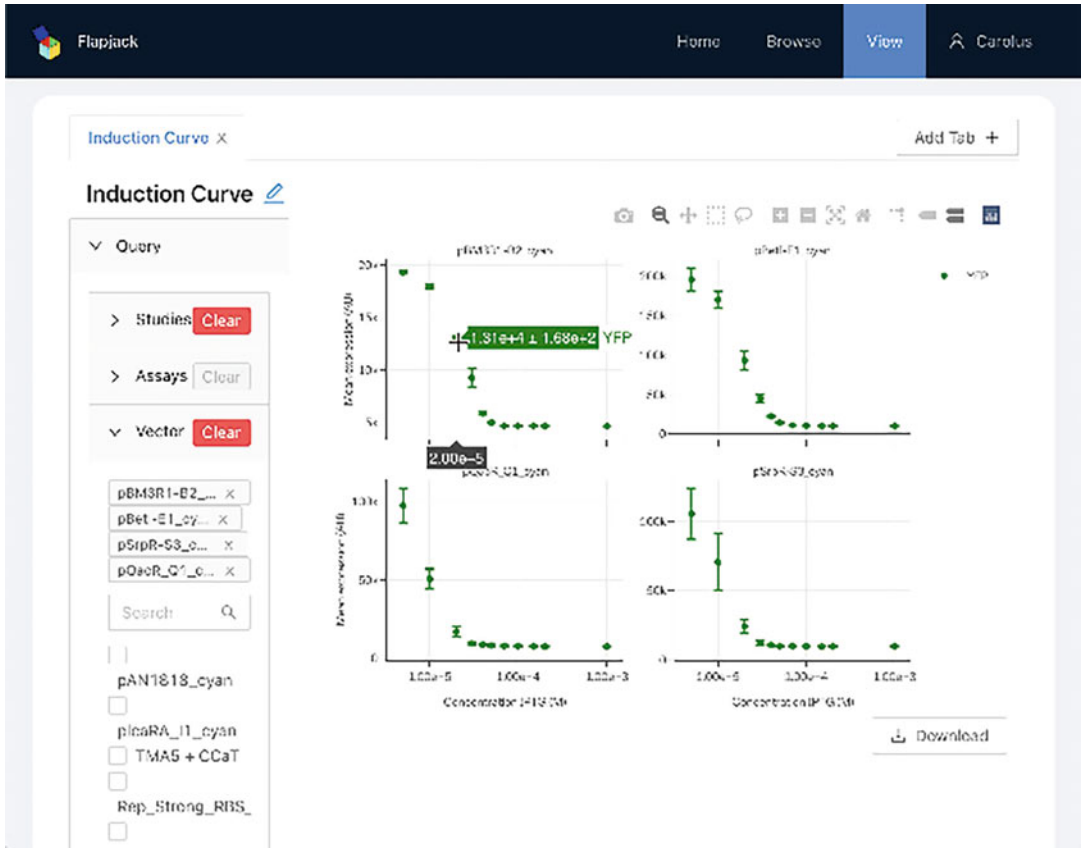


Fig. 21 Visualization and analysis of the Induction Curve study. Here four signal inverters are characterized by plotting their mean fluorescence as a function of added IPTG inducer

Normalize option and select "Min/Max," to eliminate distractions, we will remove the standard deviation, selecting only "Mean" under Plot. We proceed to press Plot, and finally, we have correct the plot.

3.7.1 Induction Curve

In Fig. 21, we consider a Study characterizing signal inverter circuits that use repressor proteins to produce low gene expression when the input signal (IPTG in this case) is at high concentration, and vice versa. To characterize the behavior of these circuits we use the Induction curve analysis type, choosing the appropriate signal Chemical. In this plot we have grouped subplots by Vector to show four different inverter circuits, and lines/markers are colored according to Signal (YFP). We can see that the circuits are functional.

3.8 pyFlapjack

This section covers the basics of how to use the pyFlapjack Python Package. It should be noted that studies can only be accessed by their owners, by those who have been granted access, or by anyone if they are public.

3.8.1 Importing pyFlapjack

To import pyFlapjack, use the following command:

```
import flapjack
```

3.8.2 Connecting to Flapjack

A class called Flapjack is defined by the pyFlapjack package, which connects to the Flapjack web application via its API. The Flapjack instance –either online or local– to which it connects can be specified. The main instance of Flapjack is connected to this example.

```
fj = flapjack.Flapjack(flapjack.rudge-lab.
org:8000)
```

The following command is used to log in after the creation of the Flapjack object:

```
fj.log_in(username=user, password=passwd)
```

3.8.3 Functions

The Flapjack package provides several functions to access the Flapjack data model. These functions make HTTPS requests to the Flapjack web application to retrieve the required data and metadata.

get Function

This function retrieves information about objects from a particular table in the database.

```
variable_name = fj.get('model', attribute=value)
```

Where `variable_name` is the assigned name for easy later access. `fj.get` states that the get function is being called. Table 1 lists the values accepted for `model` and `attribute`, and `value` is the value is the value that is sought.

Here is an example of using the get function to find a study called Context effects:

```
study = fj.get('study', name='Context effects')
```

The database may be queried against any of the attributes of the model in question, and the keyword argument "search" specified to query all attributes.

Possible errors:

If the user specifies a model not present in Flapjack, like typing `studies` instead of `study`, an error stating "model studies does not exist" will pop up.

An empty data frame is returned if no objects are found matching the query.

Create Function

This function creates a new object. All the information needed to create the object must be specified; otherwise, an error is obtained. Refer to Table 1.

```
variable_name = fj.create('model', attribute_1=value_1, attribute_n=value_n)
```

Where `variable_name` is the assigned name for easy later access. `fj.create` states that the create function is being called. The values accepted for `model` and `attribute` are listed in Table 1, and `value` is the value the user wants to assign to the corresponding attribute.

Important: It is necessary to specify every required attribute for the model to be created; otherwise, an error will be encountered.

Here is an example of using the create function to create a study called "Test study" and with the description "This is a test":

```
study = fj.create('study', name='Test study', description='This is a test')
```

Consider that we only provided values for `name` and `description` in this example because these values are required, as shown in Table 1.

Delete Function

Allows the user to delete the object. The user has to specify the model id.

```
variable_name = fj.delete('model', id_number)
```

Where `variable_name` is the assigned name for easy later access. `fj.delete` states that the delete function is being called, and `id_number` is the model's id at which the attributes are to be modified, which may be obtained by calling the get function to query for existing objects. The values accepted for `model` and `attribute` are listed in Table 1.

Here is an example of deleting an assay with id = 0:

```
deleting_assay = fj.delete('assay', 0)
```

Analyzing Data

To analyze data in Flapjack, we first use the get function to identity the metadata. For example,

```
study = fj.get('study', name='voigt inverters RVs')
vector = fj.get('vector', name='pAN1818_cyan')
vector2 = fj.get('vector', name='pSrpR-S3_cyan')
media = fj.get('media', name='M9 Glicerol')
```

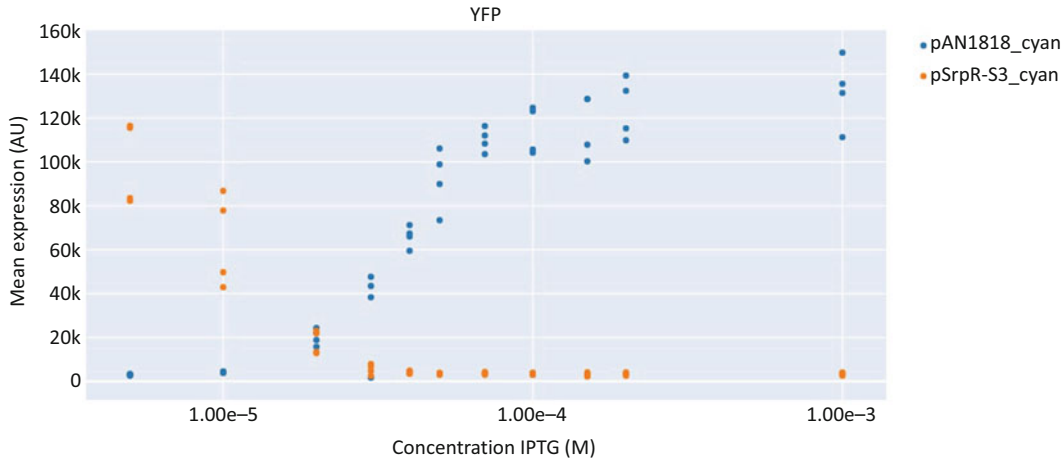


Fig. 22 Induction curves plot generated using pyFlapjack. Induction curves of the signal receiver pAN1818 and the signal inverter pSrpR-S3 showing mean fluorescence at different concentrations of inducer IPTG

```
strain = fj.get('strain', name='Top10')
iptg_chem = fj.get('chemical', name='IPTG')
yfp = fj.get('signal', name='YFP')
biomass_signal = fj.get('signal', name='OD')
```

Then, we can use the plot function. The example shows how to query by study, vector, and signal. For the analysis, select as type Induction curve with function mean expression (the value to plot for each inducer concentration). Keyword arguments for the analysis include analyte (the inducer Chemical) and biomass_signal (the Signal representing culture growth). The grouping of data can be specified as parameters corresponding to the options in the front end. The final result can be seen in Fig. 22.

```
fig = fj.plot(study=study.id,
 vector=[vector.id, vector2.id],
 signal=yfp.id,
 type='Induction Curve',
 analyte=iptg_chem.id[0],
 function='Mean Expression',
 biomass_signal=biomass_signal.id,
 normalize='None',
 subplots='Signal',
 markers='Vector',
 plot='All data points'
 )
fig
```

4 Notes

4.1 Patch Function

This function allows the user to modify something that already exists. The user has to specify the model id.

```
variable_name = fj.patch('model', id_number, attribute=value)
```

Where `variable_name` is the assigned name for easy later access. `fj.patch` states that the patch function is being called, and `id_number` is the id of the study at which the attributes are to be modified. The values accepted for `model` and `attribute` are listed in Table 1, and `value` is the new value the user wants to assign.

Here is an example where we modify the name of a study with ID 0 to "Changed name":

```
study = fj.patch('study', 0, name='Changed name')
```

References

1. Yáñez Feliú G, Earle Gómez B, Codoceo Berrocal V, Muñoz Silva M, Nuñez IN, Matute TF, Arce Medina A, Vidal G, Vitalis C, Dahlin J, Federici F, Rudge TJ (2021) Flapjack: data management and analysis for genetic circuit characterization. ACS Synth Biol 10:183–191. https://doi.org/10.1021/acssynbio.0c00554
2. Vidal G, Vitalis C, Rudge TJ (2021) LOICA: integrating models with data for genetic network design automation. ACS Synth Biol 2021.09.21.460548-2021.09.21.460548. https://doi.org/10.1021/acssynbio.1c00603
3. McLaughlin JA, Myers CJ, Zundel Z, Mlslrll G, Zhang M, Ofiteru ID, Goñi-Moreno A, Wipat A (2018) SynBioHub: a standards-enabled design repository for synthetic biology. ACS Synth Biol 7:682–688. https://doi.org/10.1021/ACSSYNBIO.7B00403
4. Vidal G, Vitalis C, Muñoz Silva M, Castillo-Passi C, Yáñez Feliú G, Federici F, Rudge TJ (2022) Accurate characterization of dynamic microbial gene expression and growth rate profiles. Synth Biol 7. https://doi.org/10.1093/SYNBIO/YSAC020