Subcellular Modeling of Individual Neurons in the Visual Thalamus

– from Dendritic Inputs to Somatic Response

Huimiao Chen

September 8, 2023

1. Background and Data Description

2. Basic Regression Modeling

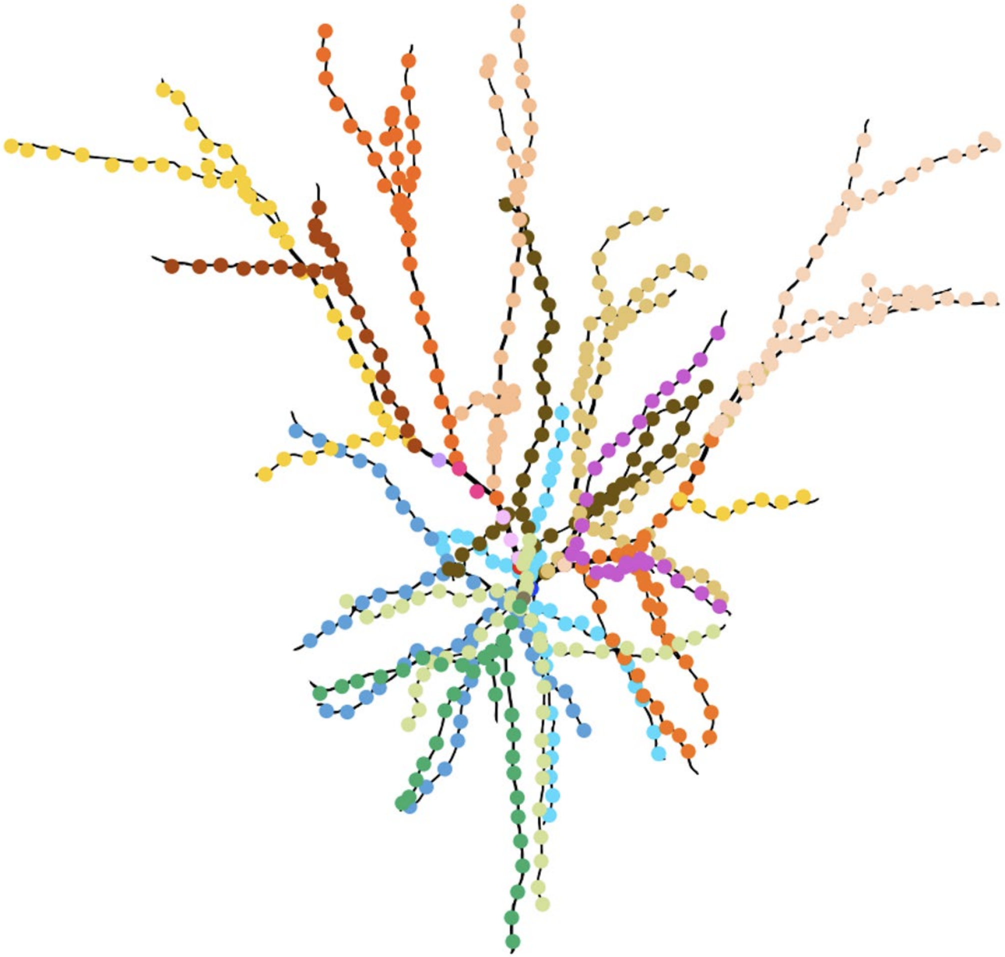
3. Reevaluating Data and Refining Data Restoration

4. Biologically Interpretable Modeling

5. Limitations and Future Work

1. Background and Data Description
2. Basic Regression Modeling
3. Reevaluating Data and Refining Data Restoration
4. Biologically Interpretable Modeling
5. Limitations and Future Work

Background and Data Description



Subcellular modeling for input-output transformation of neurons

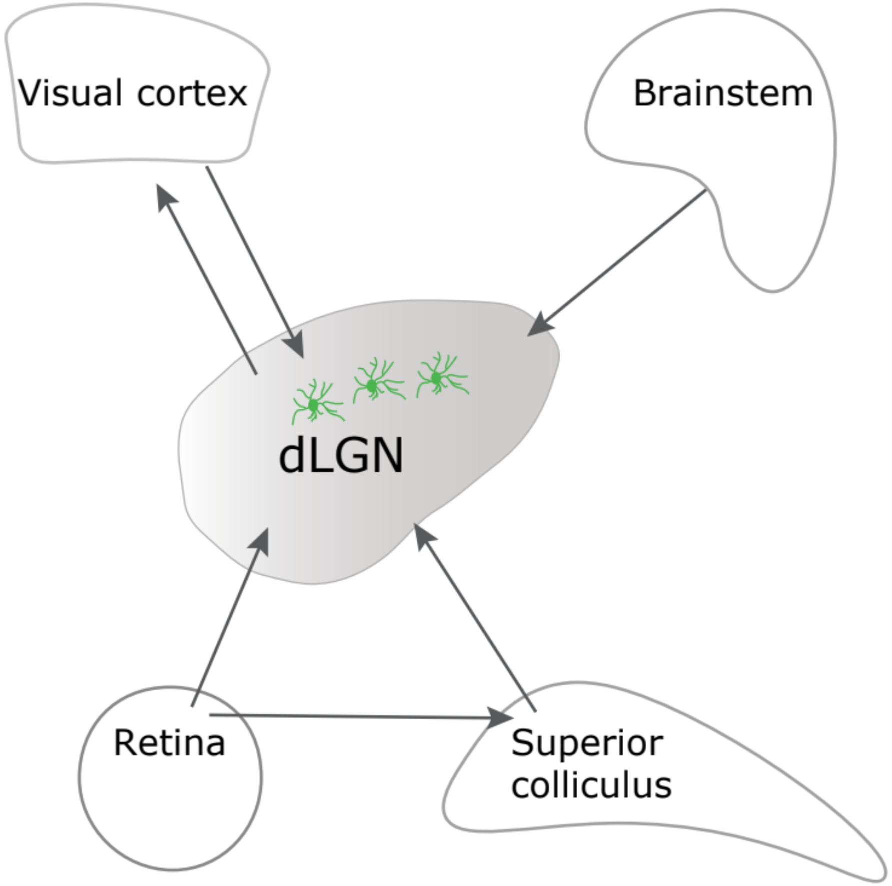
- Dendrites perform complex computations on the incoming signals, such as summation, multiplication, and nonlinearities.
- Dendritic integration is essential for neural coding, learning, and memory in the brain.

1. Spatial Integration: collecting signals from multiple synapses, and integrating them – specific input patterns.

2. Temporal Integration : integrating signals over time, and either amplifying or attenuating incoming signals – specific input patterns.

3. Dendritic Branching: complex branches, enabling intricate signal processing, allowing neurons to compute responses based on stimulus characteristics like orientation, frequency, or direction.

Background and Data Description



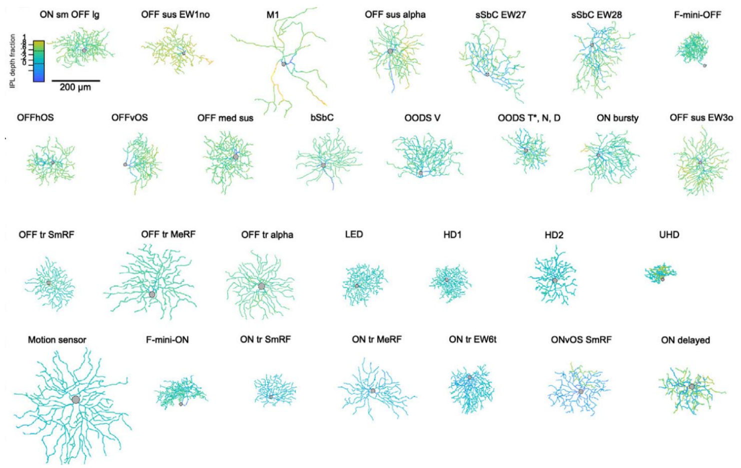
Mouse imaging-forming pathway

Now, consider the visual pathway...

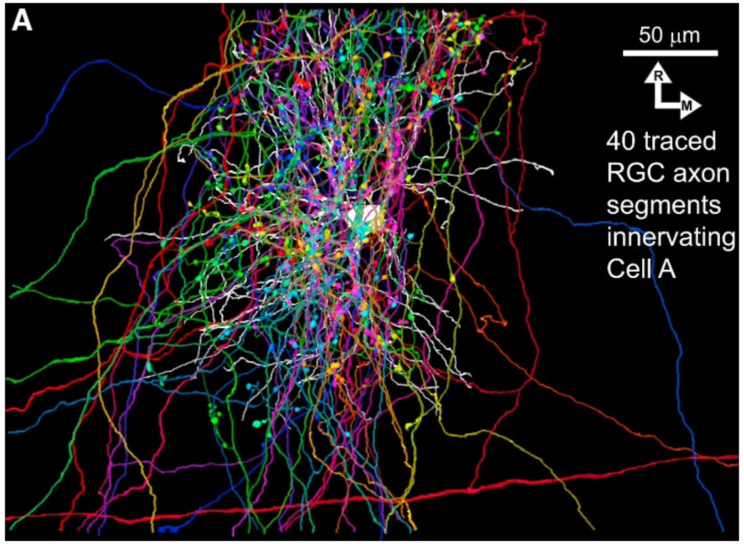
From physical perspective:
The dLGN location shows the significance of its neurons in cracking the long-range functional connectivity of the visual pathway.
Analyzing the computational properties of dLGN neurons helps us to gain insights into how visual information is processed across multiple brain regions.

From algorithmic perspective:
Signals input into the dLGN represent the earliest stages of visual data processing within the visual pathway.
Understanding of dLGN neurons is important for figuring out the fundamental mechanisms that underlie the whole intricate processing.

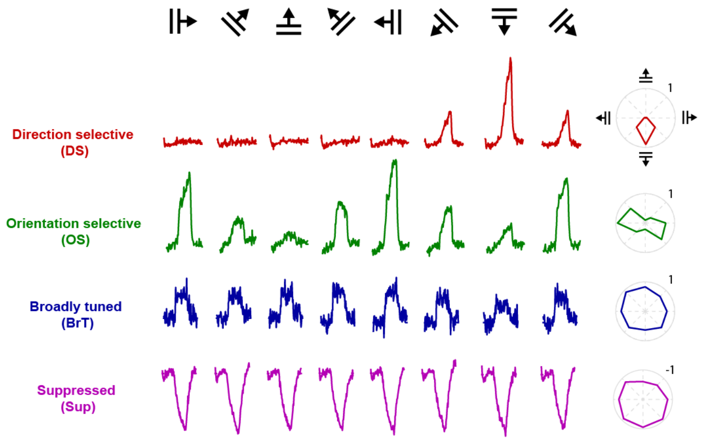
Background and Data Description



Over 40 types of retinal ganglion cells in mouse retina



Tens of RGCs can converge to a single dLGN neuron



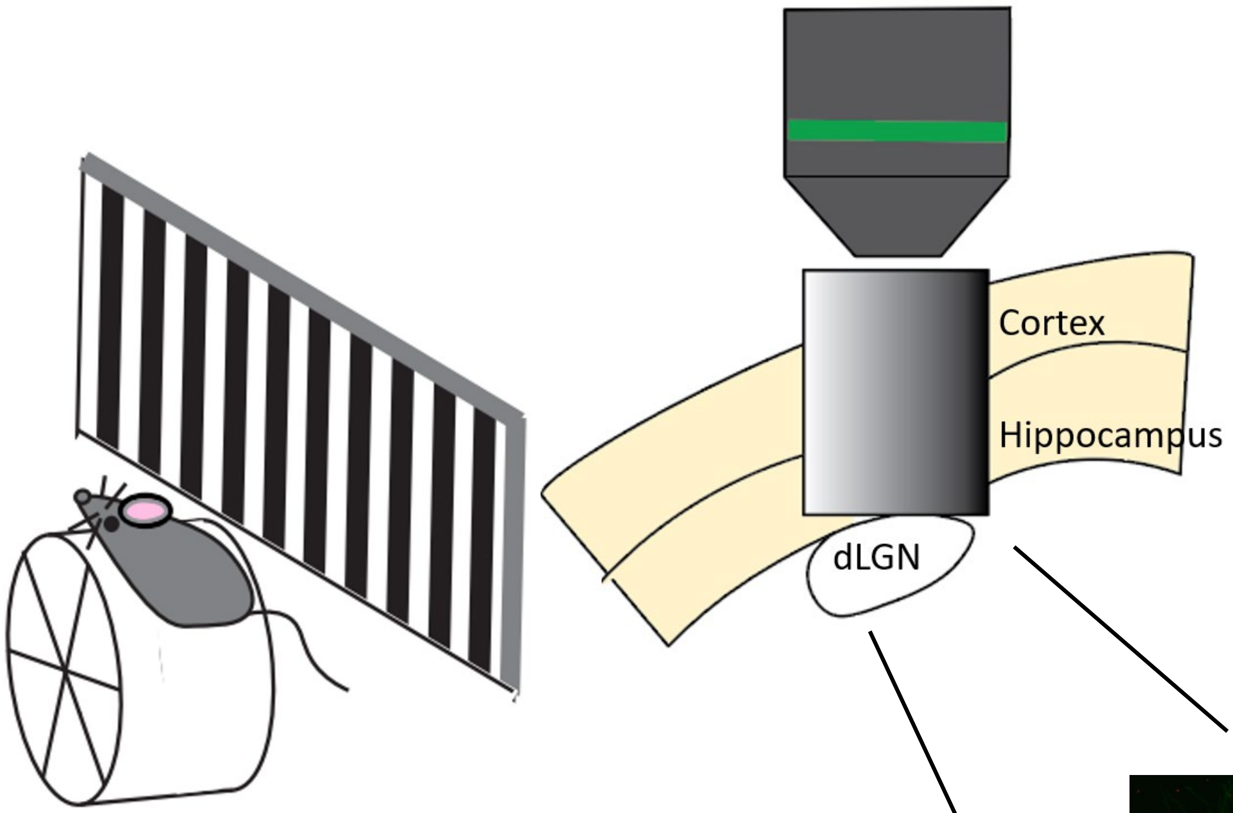
Functional categories of dLGN neurons

dLGN neurons receive very rich inputs.

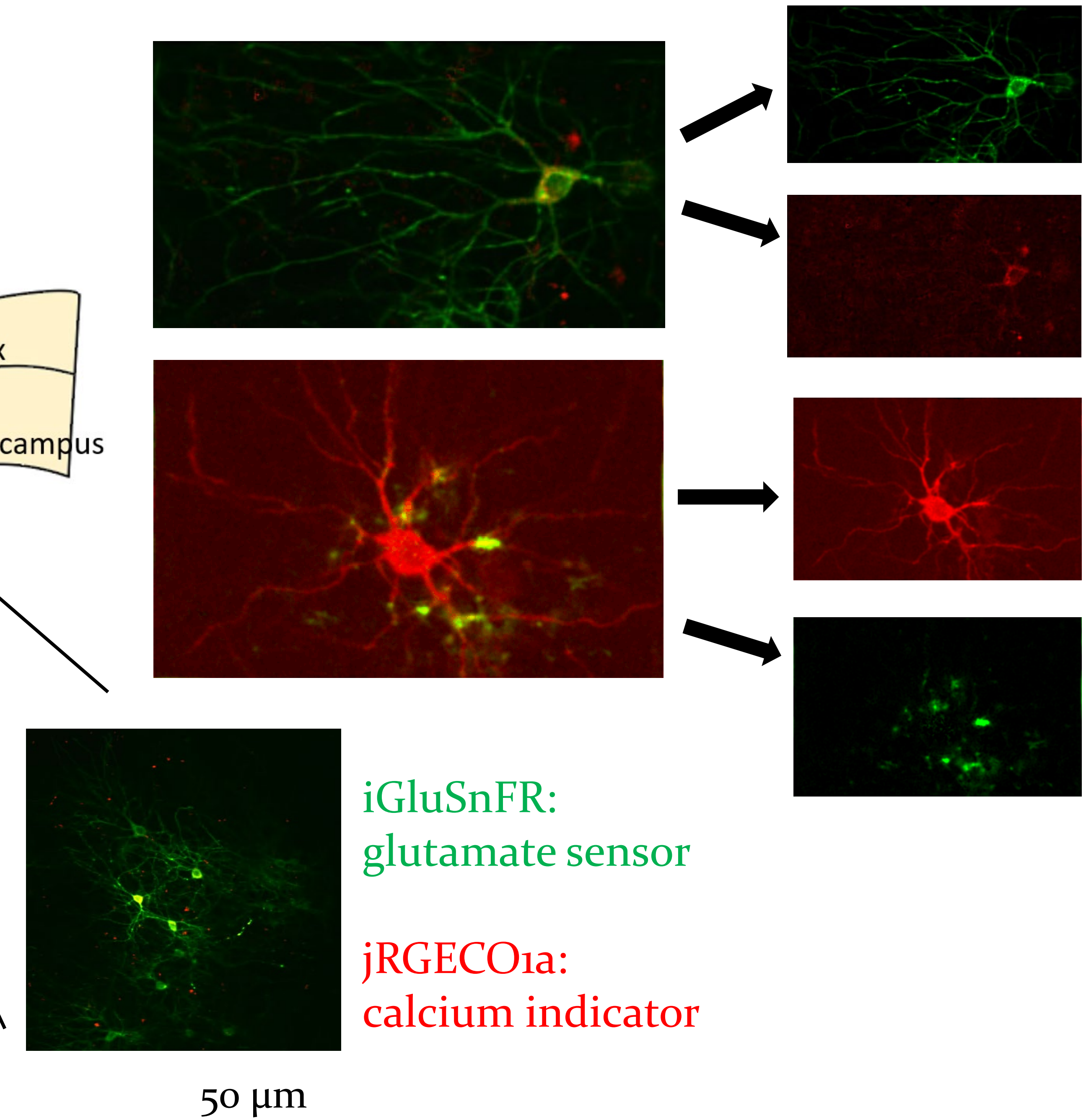
Such data diversity serves as a crucial foundation for building computational models and analyzing the complex properties.

It also underscores the necessity of a comprehensive understanding of these neurons

Background and Data Description



2-photon imaging of somatic firing and
excitatory inputs to dendrites in the dLGN



Background and Data Description

48 conditions of visual Stimuli + 3 blank trials (in total 51)

8 (orientation) × 2 (tf) × 3 (sf) = 48 (condition)

Orientation	0°						...	315°					
Temporal frequency (cycle per second, Hz)	1			4			...	1			4		
Spatial frequency (cycle per degree)	0.02	0.08	0.32	0.02	0.08	0.32	...	0.02	0.08	0.32	0.02	0.08	0.32

Each condition lasts 2 s and after a 2-s intertrial.

Each repeat includes and randomizes all 51 conditions.

51 × 4 × 10 / 60 = 34 min – adding 2 s at the end leads to in total 34 min + 2 s for 1 run, including 10 repeats.

2p system freq: 15.63 Hz; visual stimui screen freq: 60.5 Hz; NiDaq freq: 4000 Hz.

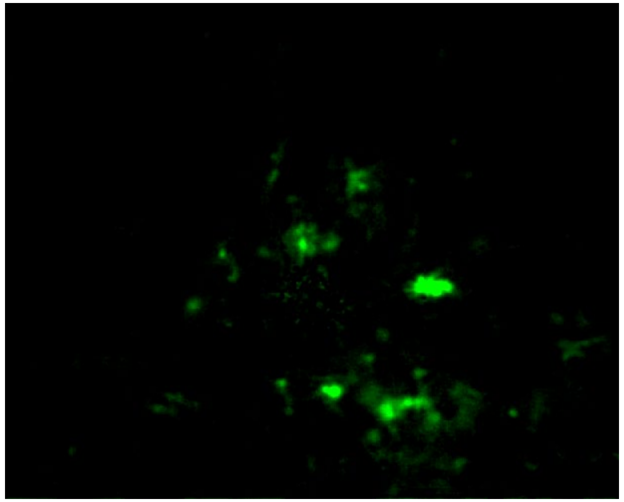
We have 32500 frames from 2p system, which is 32500 / 15.63 / 60 = 34.66 min > 34 min + 2 s.

How to synchronize and locate? NiDaq tells the exact onset time of visual stimuli within the 2p period.

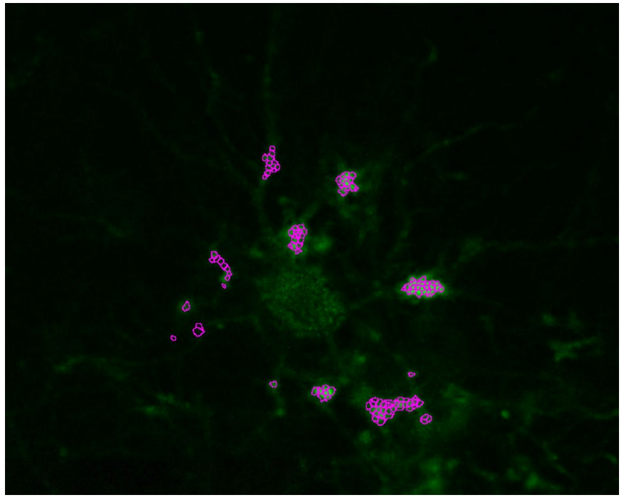
Background and Data Description

If for one cell, we use 3-run data, then we have: $3 \times 10 \times 48 \times 2$ s data during the visual stimuli → F values, dF/F values. Full data is 3×32500 frames' 2p images. We have same form pupil size data.

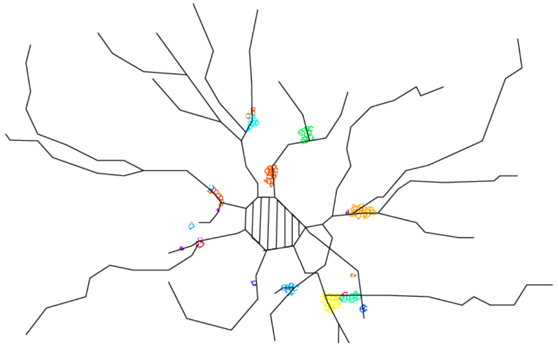
Data processing and extraction:
After preprocessing like motion alignment and PCA (Principal Component Analysis) denoising, Algorithms, like CNMF (Constrained Non-Negative Matrix Factorization), are used to extract the significant components for green fluorescence and grouped them; red fluorescence at the soma is extracted.



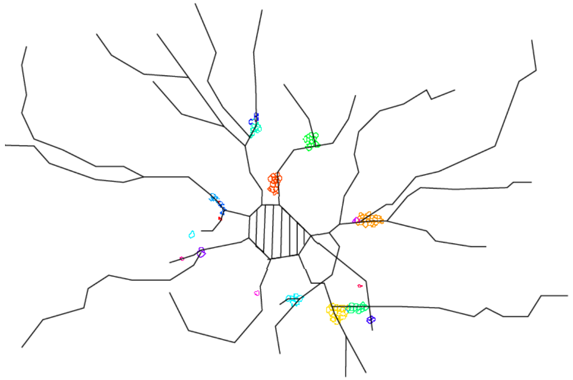
dF/F mean images of eight directions



Components extracted from the hot spots



Input components grouped by noise correlation

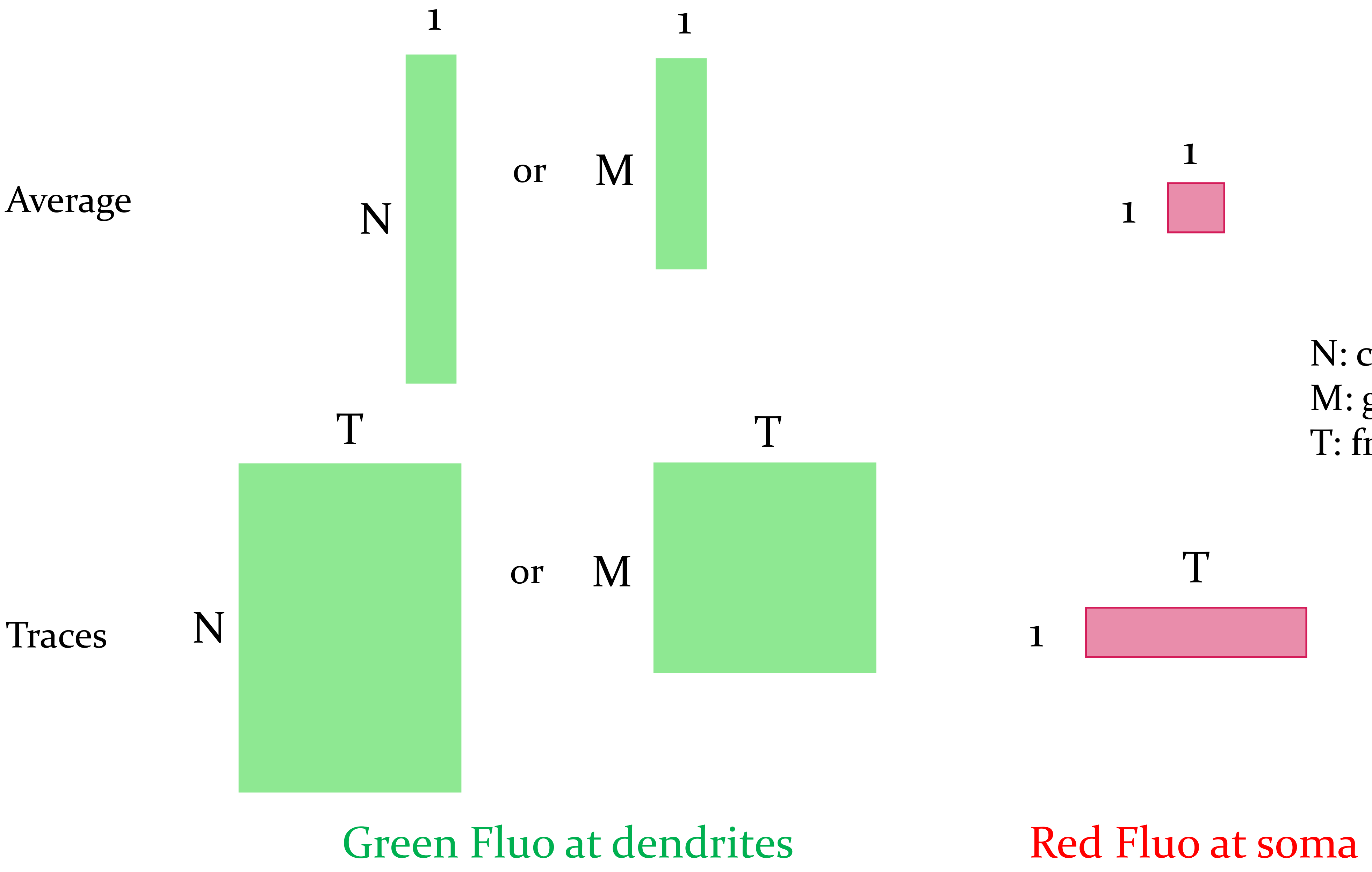


Input components grouped by spontaneous activity correlation

Background and Data Description

Then what data we have?

(Pupil size data not included here.)



N: component number
M: group number
T: frame number

1. Background and Data Description

2. Basic Regression Modeling

3. Reevaluating Data and Refining Data Restoration

4. Biologically Interpretable Modeling

5. Limitations and Future Work

Basic Regression Modeling

Linear Regression
- Ordinary Linear Regression
- Ridge Linear Regression
- Elasticnet Linear Regression

Power-Law Regression

Exponential Regression

Logistic Regression

This part we only use the average data of green and red fluorescence (dF/F) during the visual stimuli to fit models.

We in total have $48 \times 10 \times 3 = 1440$ pairs of green and red data; 95% of them are used for fitting and 5% for test.

E.g., cell CL090_230515 has 281 components (or 25 groups) for green data, then the shape of green data is (1440, 281) (or (1440, 25)); the shape of red data is (1440, 1).

Basic Regression Modeling

Linear Regression

Ordinary least squares Linear Regression.

Ordinary Linear Regression

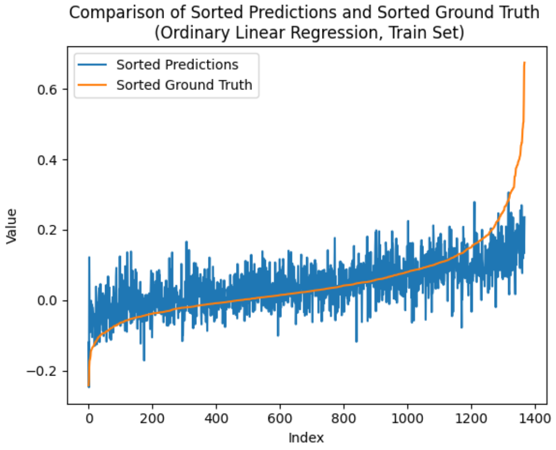
Linear Regression fits a linear model with coefficients to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

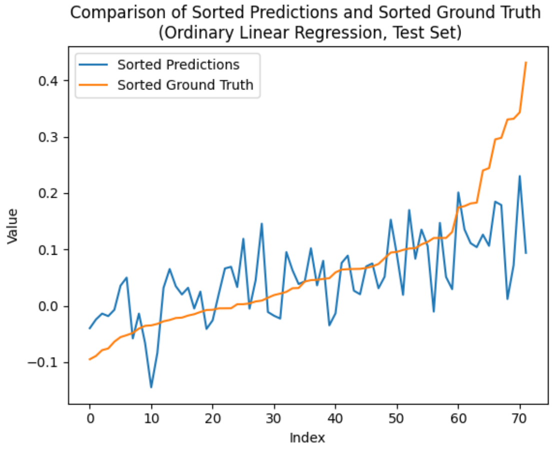
Sample: Cell CL090_230515

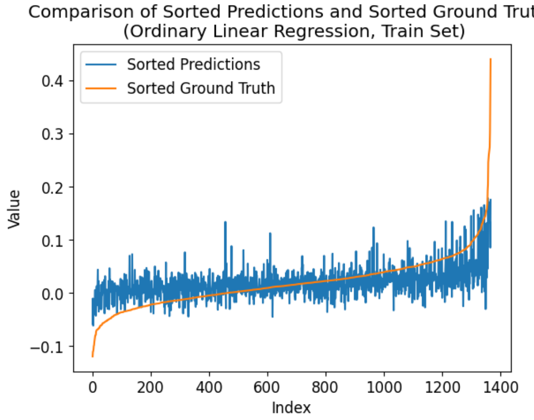
	Train data	Test data
R2 score	0.43182030844123154	0.3610848833977214
Mean squared error	0.006159697059993445	0.007927335708322457
Correlation coefficient	0.6571303587882936	0.6140976886111208

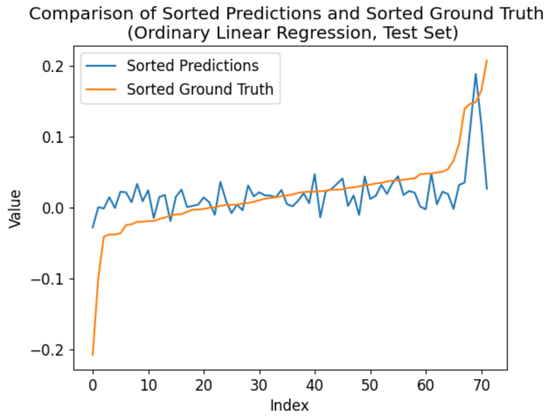
Sample: Cell CL075_230303

	Train data	Test data
R2 score	0.36242152070799216	0.35909155398992
Mean squared error	0.0014309492673555517	0.0019353809880106538
Correlation coefficient	0.602014551907171	0.6012948658925962









Basic Regression Modeling

Linear Regression

Ridge Linear Regression

Linear least squares with l2 regularization.

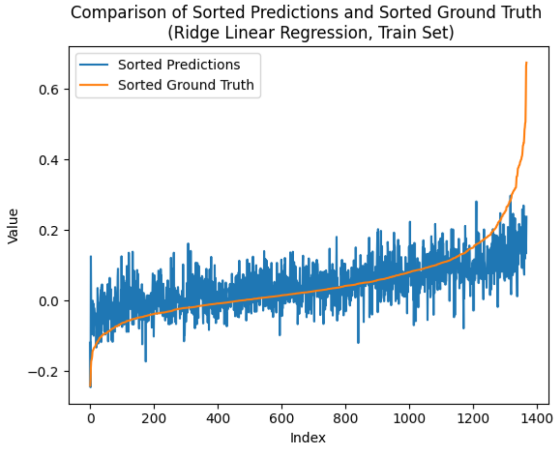
Minimizes the objective function:

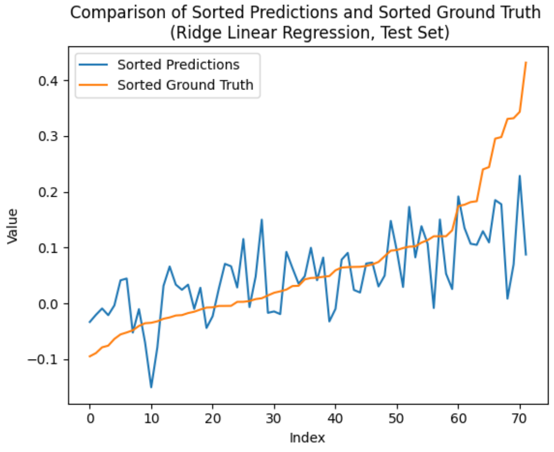
$$||y - Xw||_2^2 + \alpha||w||_2^2$$

This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm.

Sample: Cell CL090_230515

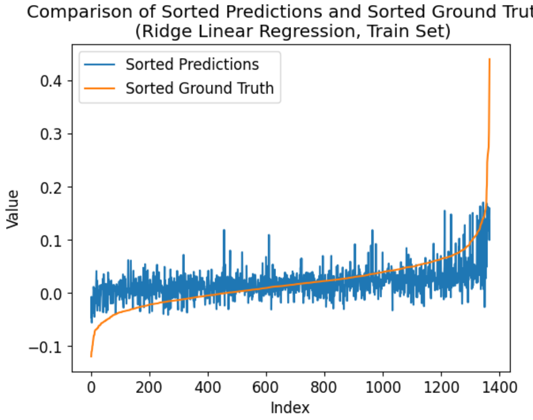
	Train data	Test data
R2 score	0.43062317675322503	0.35050150845424777
Mean squared error	0.006172675293203 73	0.008058648873285792
Correlation coefficient	0.6562194390940245	0.6047541953182045

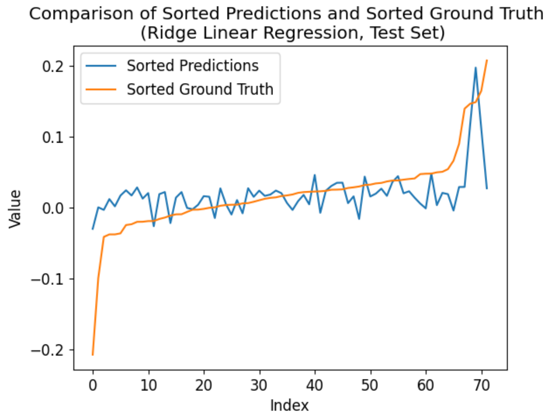




Sample: Cell CL075_230303

	Train data	Test data
R2 score	0.35094054377913975	0.3512424255181108
Mean squared error	0.0014567165980582566	0.0019590833656456547
Correlation coefficient	0.5925651096246126	0.5932188323094388





Basic Regression Modeling

Linear Regression

Elasticnet Linear Regression

Linear regression with combined L1 and L2 priors as regularizer.

Minimizes the objective function:

$$1/(2 * n_{samples}) * ||y - Xw||_2^2 + \alpha * l1_{ratio} * ||w||_1 + 0.5 * \alpha * (1 - l1_{ratio}) * ||w||_2^2$$

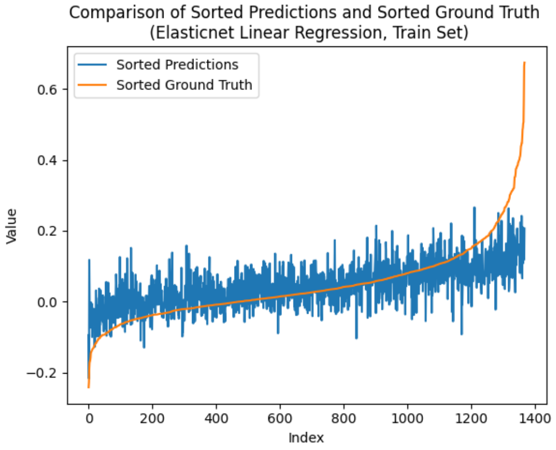
If controlling the L1 and L2 penalty separately, that this is equivalent to:

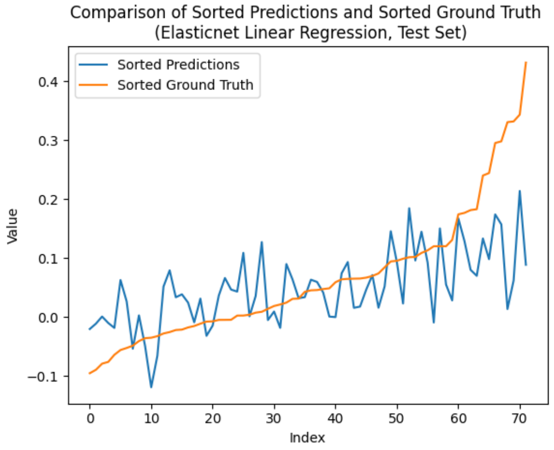
$$a * ||w||_1 + 0.5 * b * ||w||_2^2$$

where: $\alpha = a + b$ and $l1_{ratio} = a/(a + b)$.

Sample: Cell CL090_230515

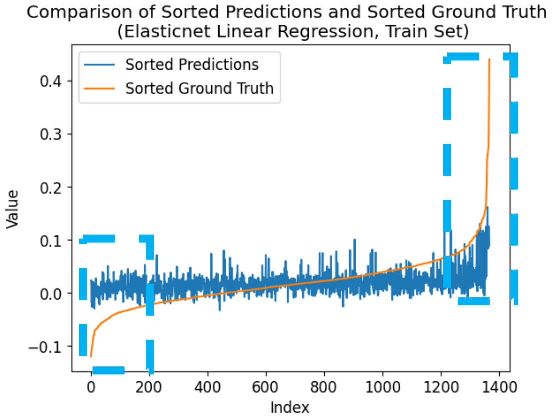
	Train data	Test data
R2 score	0.4110127314829676	0.33535179119658987
Mean squared error	0.006385274236585692	0.008246618904776503
Correlation coefficient	0.6428087118878939	0.593481482809575

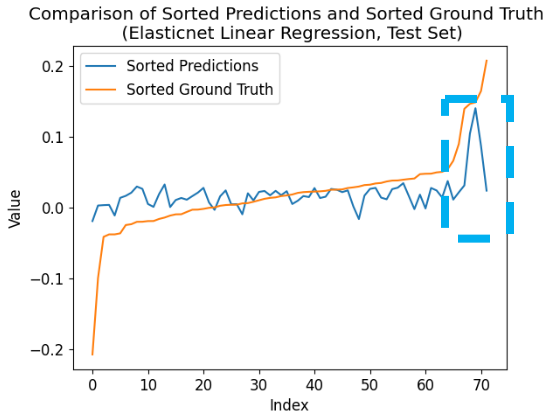




Sample: Cell CL075_230303

	Train data	Test data
R2 score	0.24334455576550063	0.3370103668241261
Mean squared error	0.0016981965499813	0.002002060573994312
Correlation coefficient	0.4970718988854777	0.6136636898586912





Basic Regression Modeling

Power-Law Regression

Mathematically, a power-law relationship can be expressed as:

$$y = AX^C$$

Here, I modify it, shown as:

$$y = A(X + B)^C + D$$

where, $X = \beta_1 x_1 + \beta_2 x_2 \cdots + \beta_N x_N$. $X + D$ is a linear regression part. A, B, C, D, β_1, β_2, ...,β_N are parameters to be determined.

Note: We need to fix the exponent to fit the other parameters. Because when the base is negative the exponent should be a fraction whose denominator is even. That is, we can only predetermine the exponent before fitting.

Algorithms are needed to help looking for numbers for exponent if trying different exponents.

Basic Regression Modeling

Power-Law Regression

Sample: Cell CL090_230515

- **Example: Exponent = 5**

$C = 5.$

Fitted Parameters A, B, D, β_1, β_2, ...,β_N

	Train data	Test data
R2 score	0.44977053881398854	0.3711914289705994
Mean squared error	0.005965096684632098	0.007801938801087381
Correlation coefficient	0.6706493415945699	0.6219617870348478

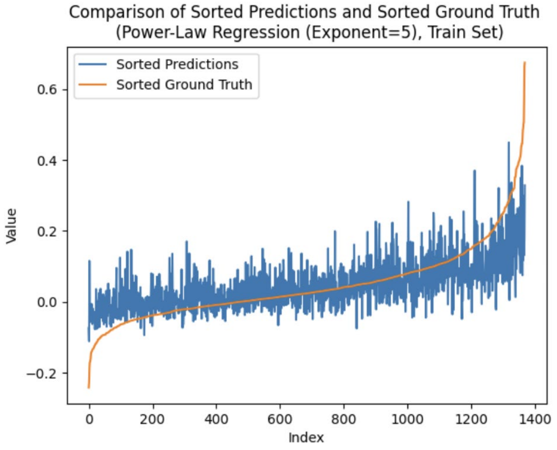
- **Example: Exponent = 5 (only fit A and D)**

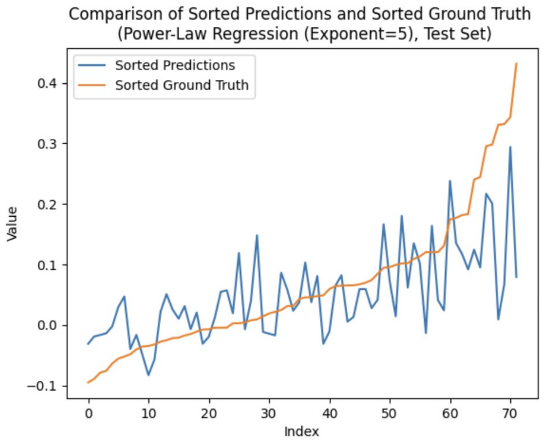
$C = 5.$

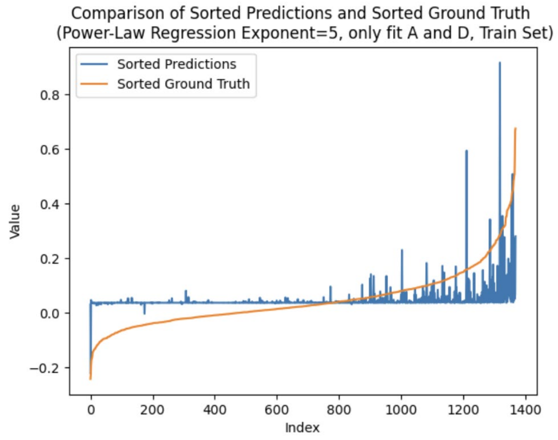
β_1, β_2, ...,β_N given by Ordinary Linear Regression.

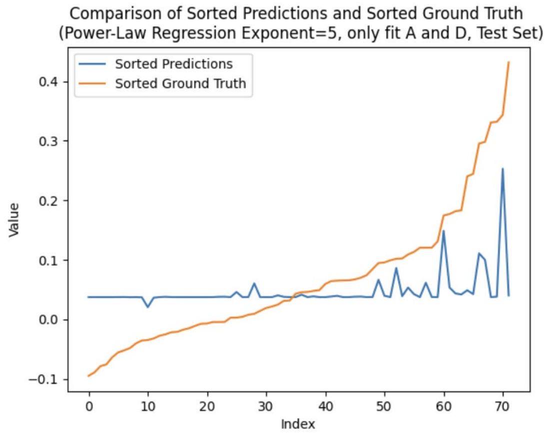
Fitted Parameters A, D

	Train data	Test data
R2 score	0.16867571360029543	0.16149921366803333
Mean squared error	0.00901247587500699	0.010403693780630378
Correlation coefficient*	0.41070148964947206	0.4740522878239656









Simply adding an exponent to the linear regression doesn't help!

Note: actually if we are not using linear regression correlation coefficient is not a suitable metric.

Basic Regression Modeling

Sample: Cell CL090_230515

Power-Law Regression

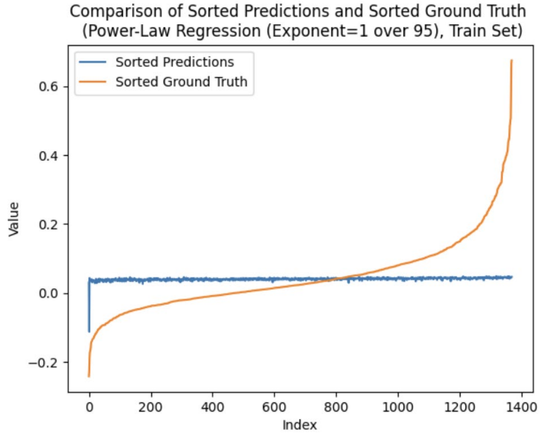
Try different exponents

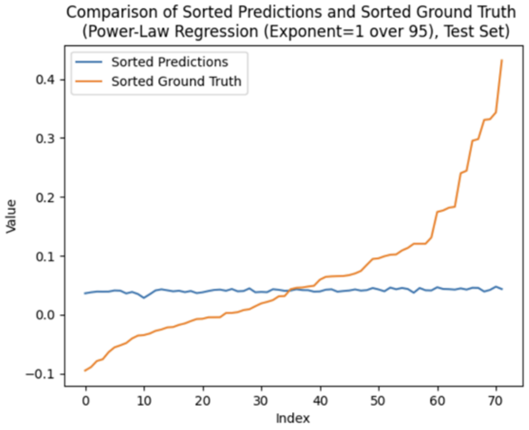
Exponent: 1/95

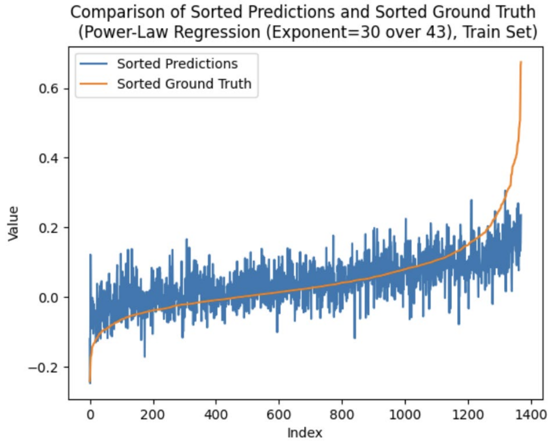
	Train data	Test data
R2 score	0.035784263044422133	-0.01178350132389605
Mean squared error	0.01045316636333401	0.012553698090272793
Correlation coefficient	0.4156817350556336	0.5492586362659945

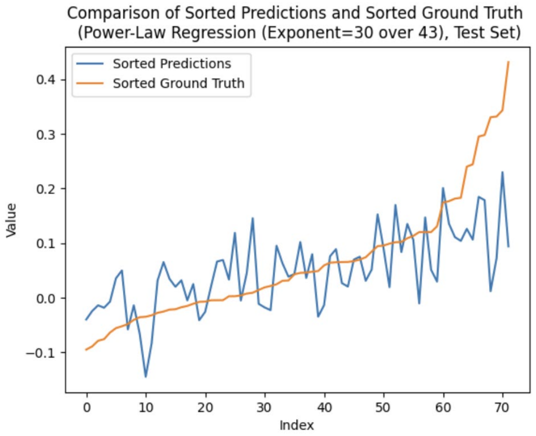
Exponent: 30/43

	Train data	Test data
R2 score	0.43176358007773163	0.36105993515151313
Mean squared error	0.006160312058280804	0.007927645253546677
Correlation coefficient	0.6570871953391387	0.6140764914279518









Basic Regression Modeling

Sample: Cell CL090_230515

Power-Law Regression

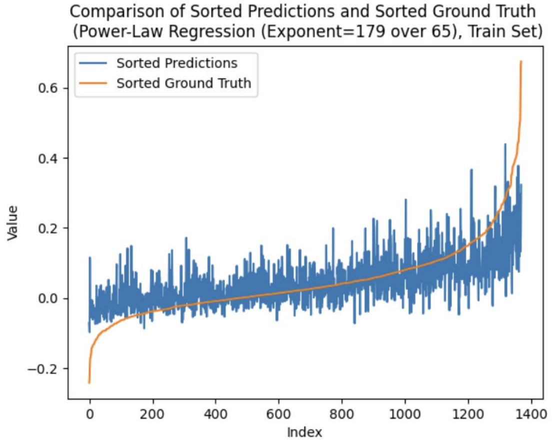
Try different exponents

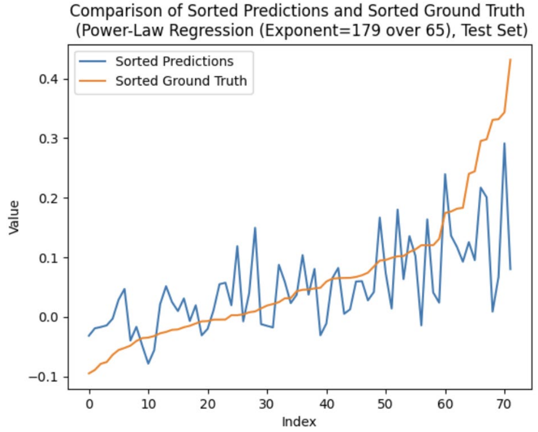
Exponent: 179/65

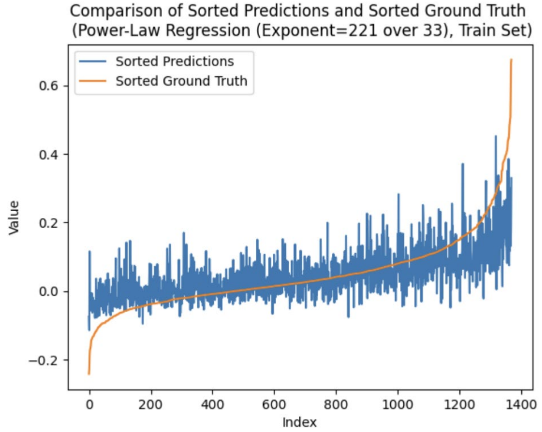
	Train data	Test data
R2 score	0.4496429961042646	0.3719683808539297
Mean squared error	0.0059664793888465975	0.0077922987749738555
Correlation coefficient	0.6705542454661818	0.6224739461963064

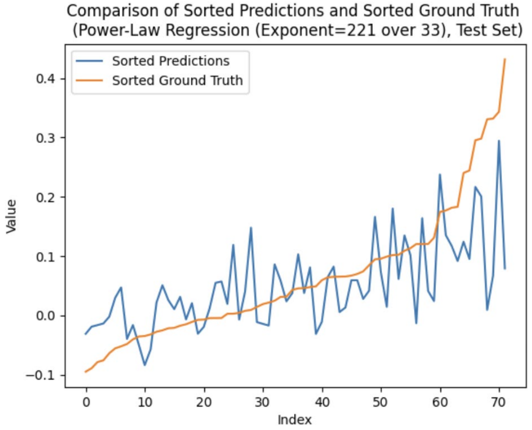
Exponent: 221/33

	Train data	Test data
R2 score	0.44976921796448766	0.37100964253069046
Mean squared error	0.0059651110041034	0.007804194315316004
Correlation coefficient	0.6706483564323005	0.6218374974348415









Basic Regression Modeling

Sample: Cell CL090_230515

Power-Law Regression

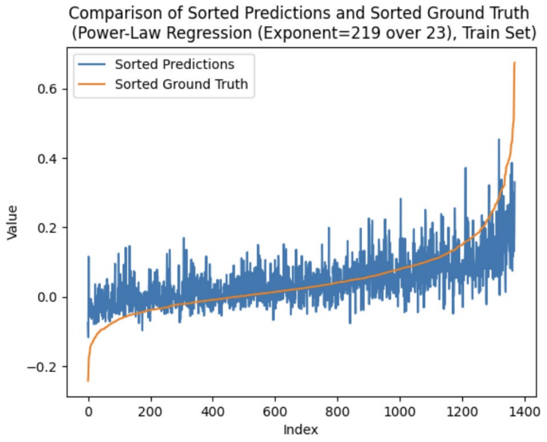
Try different exponents

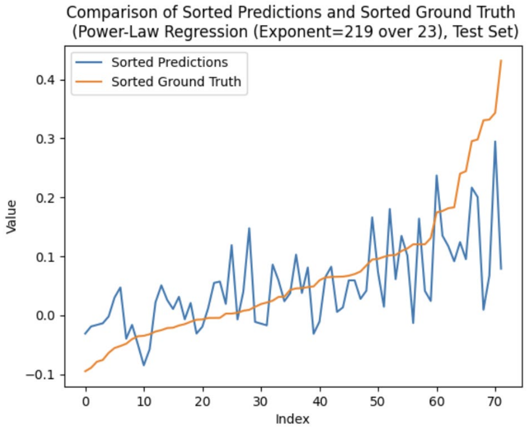
Exponent: 219/23

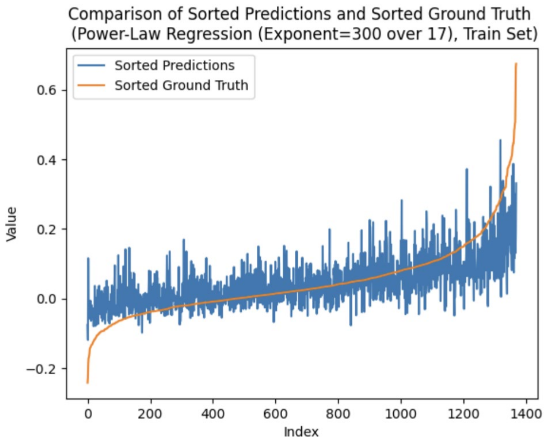
	Train data	Test data
R2 score	0.44976054121933307	0.37087312756007484
Mean squared error	0.005965205069629524	0.007805888124044452
Correlation coefficient	0.6706418874831295	0.6217442394863175

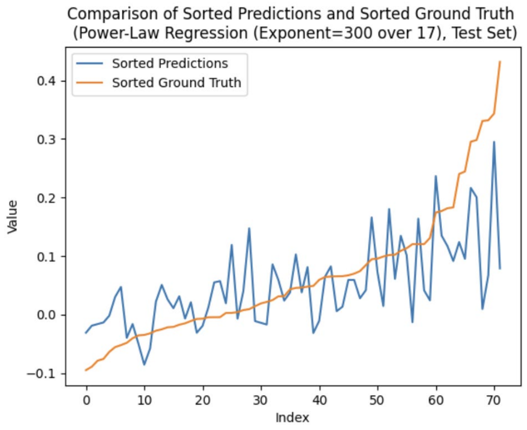
Exponent: 300/17

	Train data	Test data
R2 score	0.449745532921901	0.37070570241504563
Mean squared error	0.005965367776193391	0.007807965450587863
Correlation coefficient	0.6706306978729792	0.6216287490698504







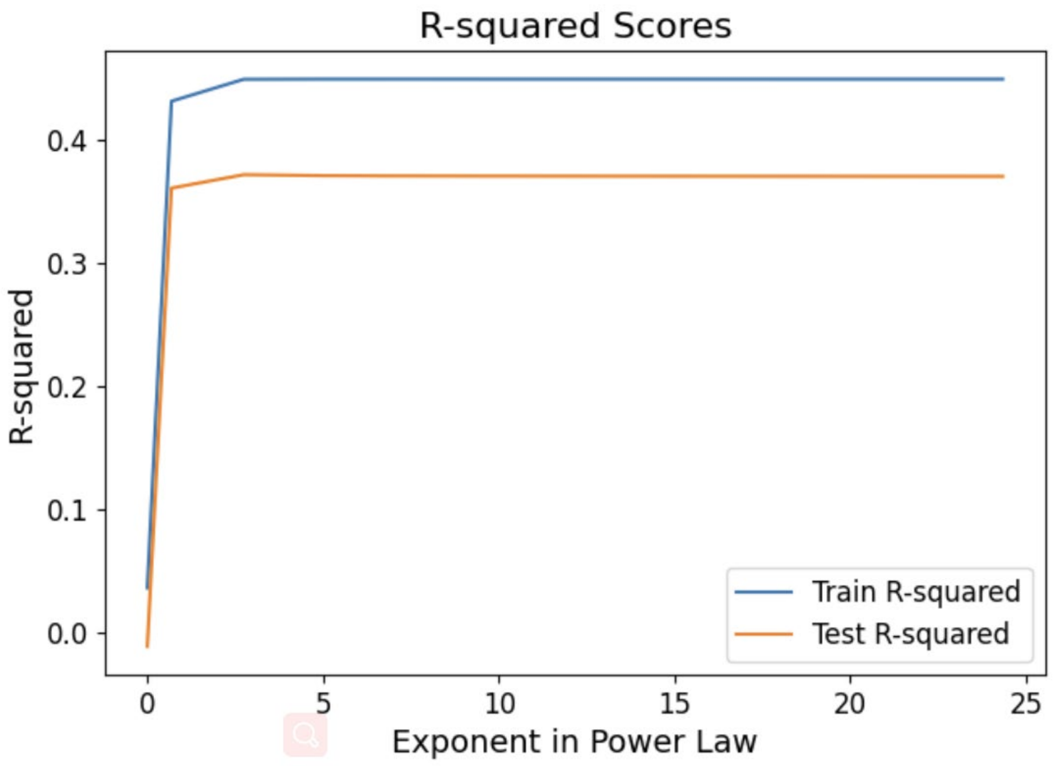


Basic Regression Modeling

Sample: Cell CL090_230515

Power-Law Regression

Comparison of different exponents



As the exponent increases, we can see the R2 score on longer increases more (saturation).

Basic Regression Modeling

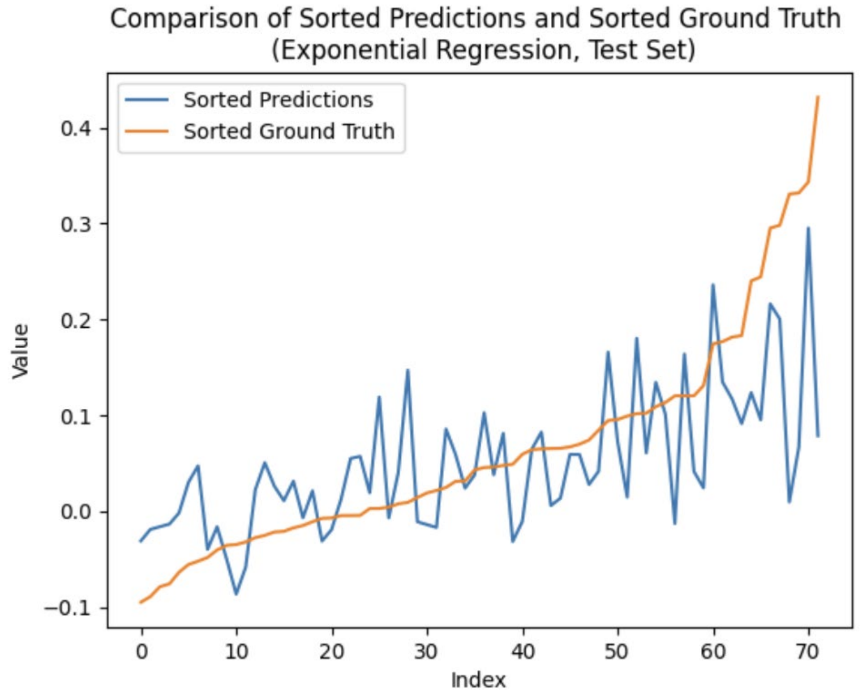
Sample: Cell CL090_230515

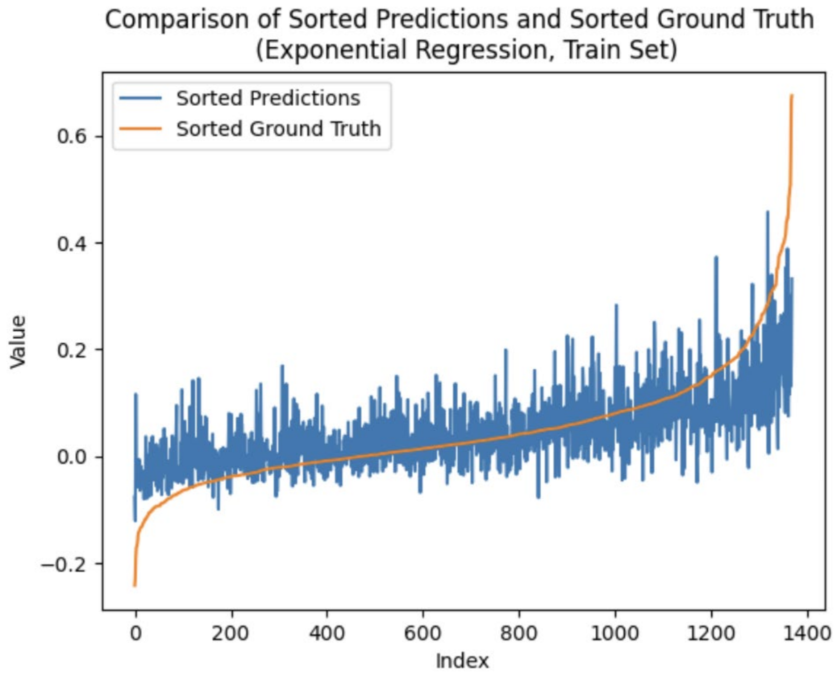
Exponential Regression

Let $B = (b_1, b_2, \ldots, b_N)$.

$$y = A \cdot e^{(b_1 \cdot x_1 + \ldots + b_N \cdot x_N)} + C$$

	Train data	Test data
R2 score	0.44972294262878365	0.3705444716933213
Mean squared error	0.005965612679987845	0.007809965919858963
Correlation coefficient	0.6706138550826338	0.6215183575736783





Basic Regression Modeling

Logistic Regression

Two types: OvR and Multinomial

OvR

In logistic regression, the "One-vs-Rest" (OvR) approach is a common strategy **for handling multi-class classification problems**. In this approach, multiple binary logistic regression models are trained, each one considering one class as the positive class and the rest of the classes as the negative class.

1. Calculate the sigmoid probabilities for each class using the logistic regression model.

   The key point of logistic regression is that **we model the log of the odds ($O = \frac{p}{1-p}$) as linear**. This is called logistic regression.

$$\eta = \operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

$$p = \operatorname{expit}(\eta) = \frac{e^\eta}{e^\eta + 1} = \frac{1}{1+e^{-\eta}}$$

   where logit and expit are inverse functions of each other. The logit function maps probabilities (values between 0 and 1) to real numbers (values between negative infinity and positive infinity). The expit function is a sigmoid function that maps any real-valued number to the range of 0 to 1.

$$\eta = \operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

$$p = \operatorname{expit}(\eta) = \frac{e^\eta}{e^\eta + 1} = \frac{1}{1+e^{-\eta}}$$

   For class i and data $\boldsymbol{X}$, η is considered as a linear combination (if we set intercept is True, then we have $\beta_{i,0}$) as follows.

$$\begin{aligned}\eta_i(\boldsymbol{X}) &= \operatorname{logit}\{P(Y=y_i \mid \boldsymbol{X})\} \\ &= \operatorname{logit}\{P(Y=y_i \mid X_1, X_2, \ldots, X_N)\} \\ &= \beta_{i,0} + \beta_{i,1}X_1 + \beta_{i,2}X_2 + \cdots + \beta_{i,N}X_N\end{aligned}$$

   All $\beta_{i,n}$ forms an I (number of classes) by N (input length, i.e., the number of features) matrix, resulting in a 2D matrix of probabilities for each class and each piece of input data.

2. Normalize the probabilities by dividing each probability by the sum of all probabilities (for each piece of data $\boldsymbol{X}$):

   normalized_probs = probabilities / sum(probabilities)

   This step ensures that the sum of the probabilities for all classes is equal to 1.

Basic Regression Modeling

Logistic Regression

Two types: OvR and Multinomial

Multinomial

Multinomial Logistic Regression, also known as Softmax Regression, is an extension of logistic regression that is used for multi-class classification problems. It allows for the prediction of probabilities across multiple mutually exclusive classes.

In multinomial logistic regression, the goal is to model the relationship between the predictor variables and the probabilities of each class. Instead of modeling binary outcomes, it deals with multiple classes. The model estimates the probabilities of each class and assigns the observation to the class with the highest probability.

The mathematical formulation of multinomial logistic regression involves the use of the softmax function. Given a set of predictor variables and their corresponding weights, the softmax function calculates the probabilities for each class.

Mathematically, the softmax function is defined as:

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

where $p_i = P(Y = y_i \mid \boldsymbol{X}) = P(Y = y_i \mid X_1, X_2, \ldots, X_N)$ is the probability of class i. In OvR, we model η as linear; here we model z as linear.
$z_i = \beta_{i,0} + \beta_{i,1}X_1 + \beta_{i,2}X_2 + \cdots + \beta_{i,N}X_N$ is is the linear combination of the predictor variables (if we set intercept is True, then we have $\beta_{i,0}$) and their corresponding weights for class i, and the sum is taken over all classes j.

To fit a multinomial logistic regression model, the weights or coefficients are estimated using maximum likelihood estimation or other optimization algorithms. The objective is to find the values of the weights that maximize the likelihood of observing the training data given the model.

Note that softmax function naturally guarantees that the sum of probabilities of all classes is 1, so no normalization is needed. But OvR needs normalization for 3 or more classes (see the following for special case: binary classification).

Special case: binary classification
Note: 'multinomial', for multi-class, achieves better performance than 'ovr'.

We use Multinomial.

Basic Regression Modeling

Logistic Regression

We digitize the output into J intervals (J classes for multi-class logistic regression).

$$p_i = \frac{e^{\beta_{i,0}+\beta_{i,1}X_1+\beta_{i,2}X_2+\cdots+\beta_{i,N}X_N}}{\sum_{j=1}^{J} e^{\beta_{j,0}+\beta_{j,1}X_1+\beta_{j,2}X_2+\cdots+\beta_{j,N}X_N}}, i = 1, \ldots, J$$

Fitted Parameters include a J by N matrix for coefficients and a J length vector for intercepts.

Considering that logistic regression is a classification, I apply two classifications to enhance the performance: one for fitting (training) and the other for evaluation.

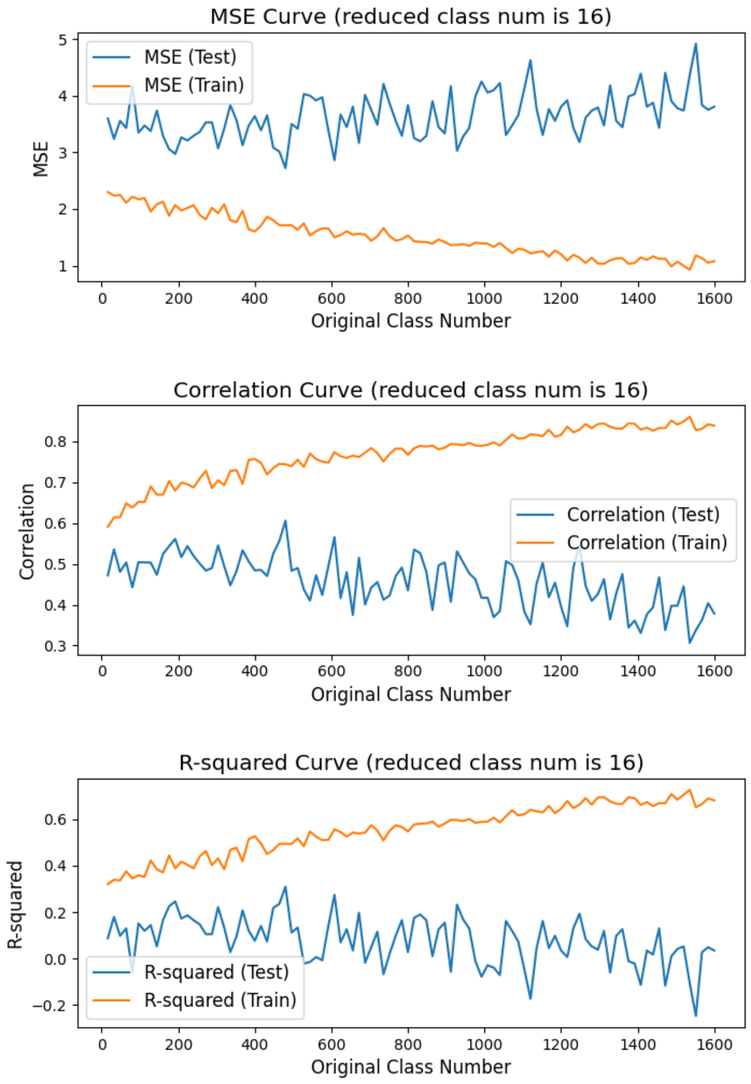
For example, in the training stage, I classify data into class_num (e.g., class_num = 160) intervals (hitogram); in test/evaluation stage, I evaluate the results with a same number of classes (e.g., reduced_class_num = 16). That is, for the example, class_num = 160 and reduced_class_num = 16, the in test/evaluation stage, classes 0, 1, …, 15 become one class, i.e., 0; …; classes 144, 145, …, 159 become one class, i.e., 15.

Fixing the reduced class number, I enumerate the original class number to see what a original class number is better.

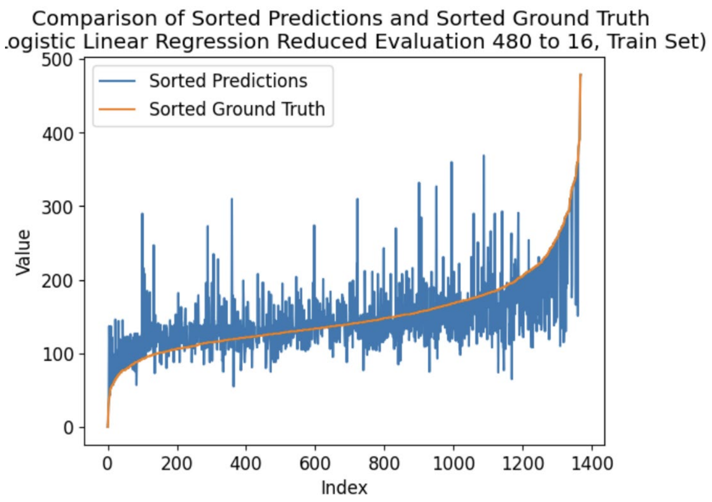
Basic Regression Modeling

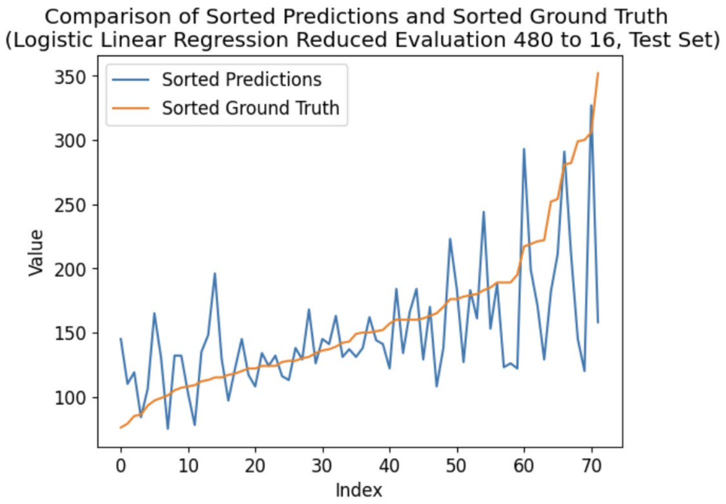
Sample: Cell CL090_230515

Logistic Regression



	Train data	Test data
R2 score	0.49425421012452153	0.3095552619991193
Mean squared error	1.7105263157894737	2.7222222222222223
Correlation coefficient	0.7442081479296433	0.6060794440849583





Best Original Class Number (Best R2 score on test set) is 480.
(some big spikes.)

Basic Regression Modeling

Comparison of Different Regressions with Same Metric

Sample: Cell CL090_230515

Classify all the predictions into 16 intervals/classes for the evaluation.

- **Ordinary Linear Regression**

	Train data	Test data
R2 score	0.4011018872884826	0.36944077498899164
Mean squared error	2.0255847953216373	2.486111111111111
Correlation coefficient	0.6340286499988343	0.6292717941149766

Ridge Linear Regression

	Train data	Test data
R2 score	0.3974276657381052	0.36944077498899164
Mean squared error	2.038011695906433	2.486111111111111
Correlation coefficient	0.6311655529544047	0.6290805964856068

- **Elasticnet Linear Regression**

	Train data	Test data
R2 score	0.37754364323018 04	0.2954645530603258
Mean squared error	2.1052631578947367	2.7777777777777777
Correlation coefficient	0.6144590454550366	0.566099913342907

- **Power-Law Regression**

	Train data	Test data
R2 score	0.42660530746169045	0.3377366798767063
Mean squared error	1.939327485380117	2.611111111111111
Correlation coefficient	0.653816054551323	0.6064419746929176

Exponential Regression

	Train data	Test data
R2 score	0.4291988756148981	0.32716864817261115
Mean squared error	1.9305555555555556	2.6527777777777777
Correlation coefficient	0.6558085630583136	0.5985681611539189

- **Logistic Regression**

	Train data	Test data
R2 score	0.49425421012452153	0.3095552619991193
Mean squared error	1.7105263157894737	2.7222222222222223
Correlation coefficient	0.7442081479296433	0.6060794440849583

Note: logistic regression does not guarantee a better numerical performance than other regression methods but with also taking into account the graphical results, logistic regression could be the best one.

Basic Regression Modeling

Deleting small groups improves?
Delete groups (axons) with less than 3 components.

Sample: Cell CL090_230515

Original

Exponential Regression

	Train data	Test data
R2 score	0.4291988756148981	0.32716864817261115
Mean squared error	1.9305555555555556	2.6527777777777777
Correlation coefficient	0.6558085630583136	0.5985681611539189

Logistic Regression

	Train data	Test data
R2 score	0.49425421012452153	0.3095552619991193
Mean squared error	1.7105263157894737	2.7222222222222223
Correlation coefficient	0.7442081479296433	0.6060794440849583

Deleting small groups

Exponential Regression

	Train data	Test data
R2 score	0.42530852338508673	0.36239542051959495
Mean squared error	1.9437134502923976	2.513888888888889
Correlation coefficient	0.6527532920188056	0.6251978930658115

Logistic Regression

	Train data	Test data
R2 score	0.4918767726507479	0.3095552619991193
Mean squared error	1.7185672514619883	2.7222222222222223
Correlation coefficient	0.7430224939475898	0.6060794440849583

Basic Regression Modeling

Does decay influence dF/F? Yes

For cell `CL090_230515` red data:

- mean(run1) = 0.0664
- mean(run2) = 0.0498
- mean(run3) = 0.0252

For cell `CL090_230515` green data:

- mean(run1) = 0.0378
- mean(run2) = 0.0456
- mean(run3) = 0.0377

For cell `CL075_230303` red data:

- mean(run1) = 0.0278
- mean(run2) = 0.0160
- mean(run3) = 0.0139

For cell `CL075_230303` green data:

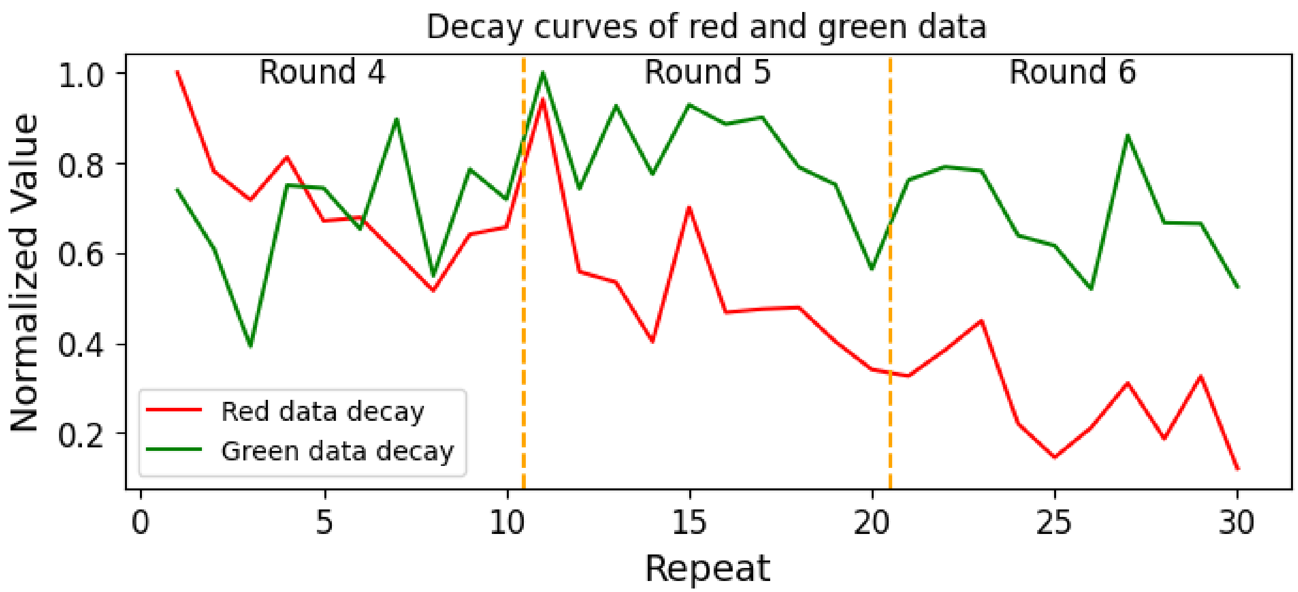
- mean(run1) = -0.0066
- mean(run2) = -0.0019
- mean(run3) = -0.0081

We can see obvious decay of the red dF/F data.

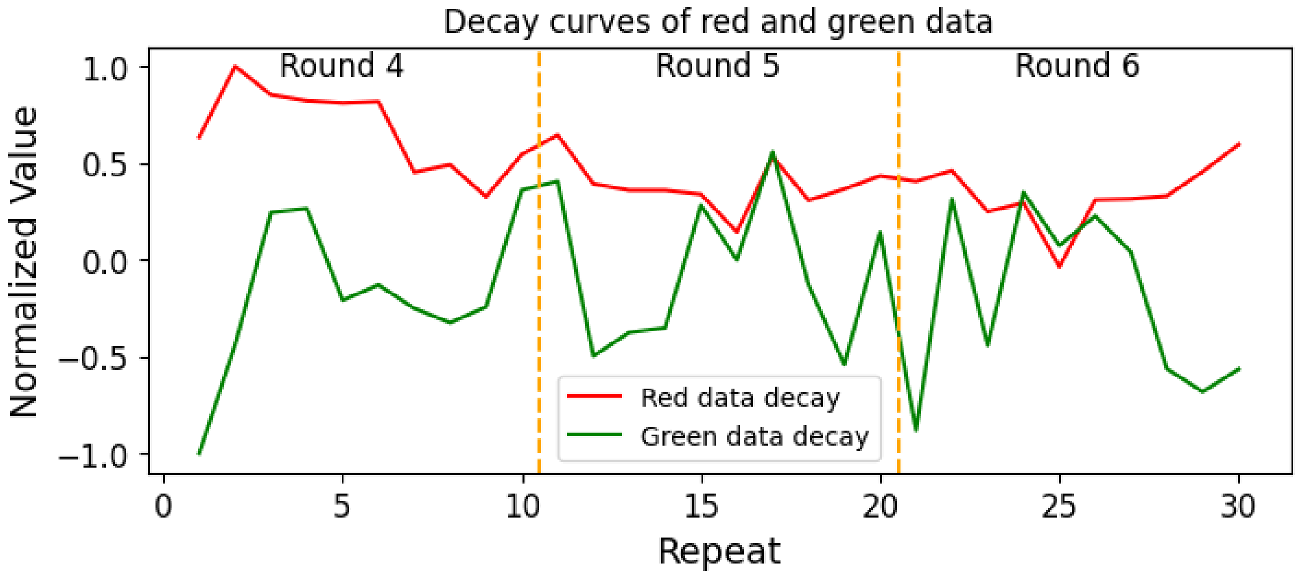
Basic Regression Modeling

Does decay influence dF/F? Yes

Sample: Cell CL090_230515



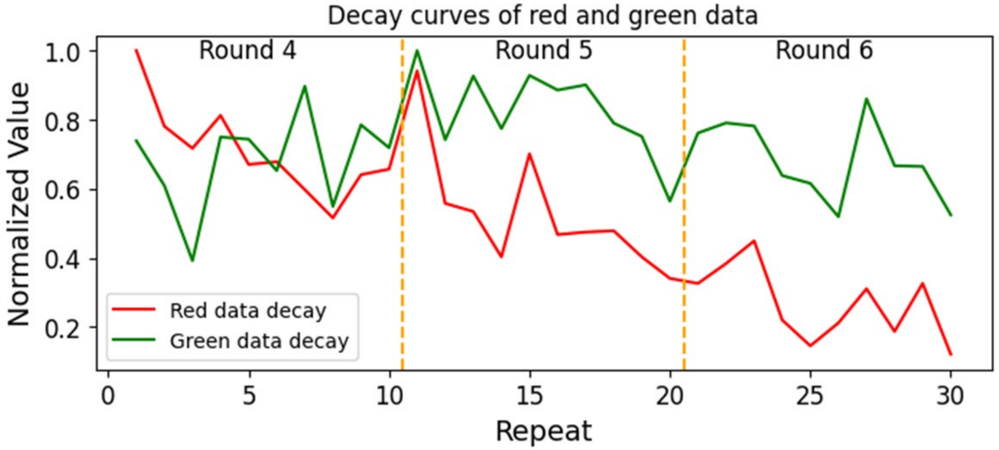
Sample: Cell CL075_230303

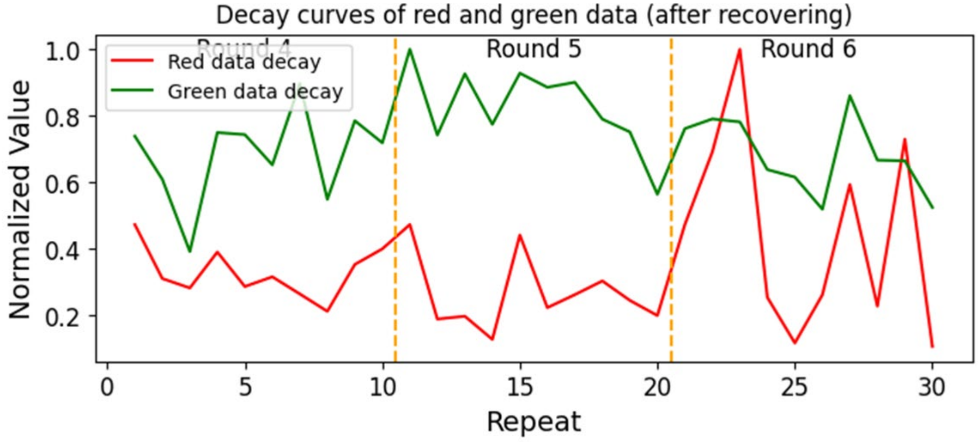


Basic Regression Modeling

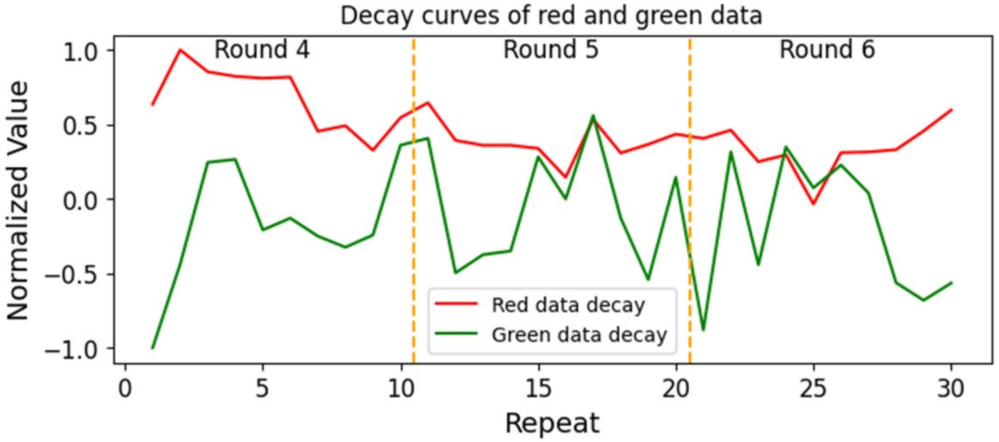
We use the exponential function to recover the decay because exponential function is memory less. After fitting we align them between runs.

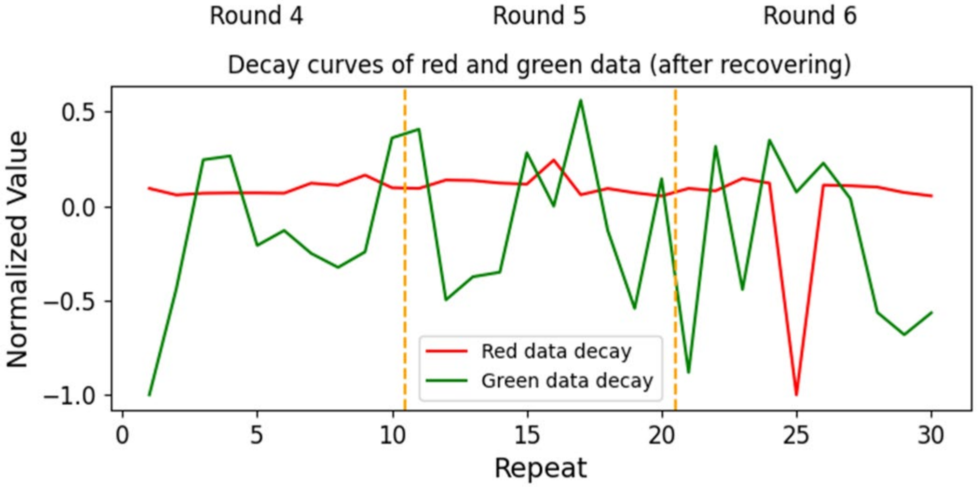
Sample: Cell CL090_230515





Sample: Cell CL075_230303

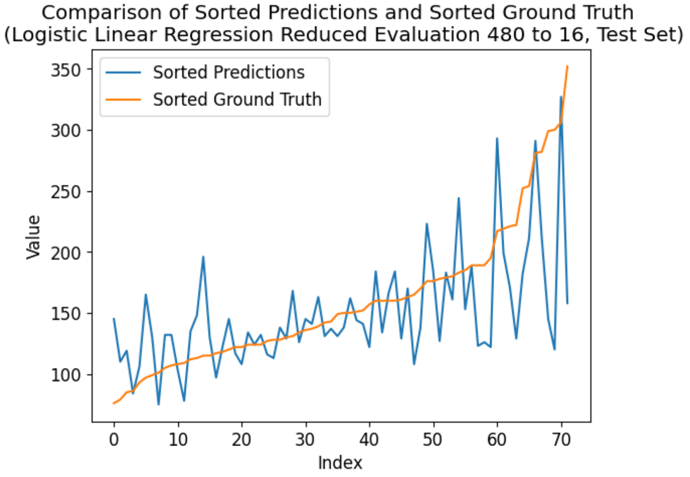


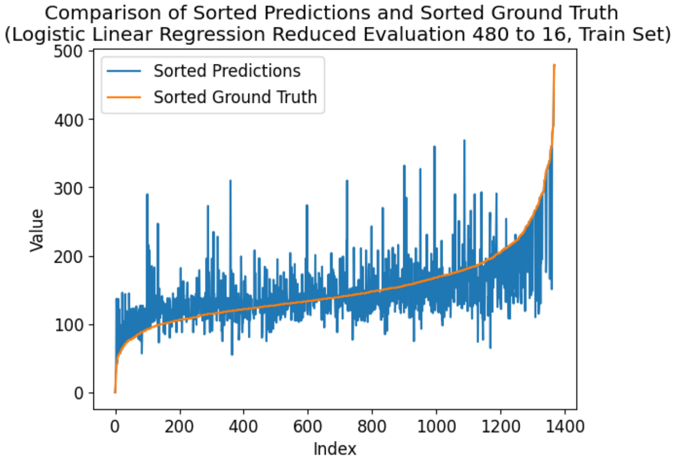


Basic Regression Modeling

Sample: Cell CL090_230515

We use recovered data to refit the logistic Model.





Before red data recovering

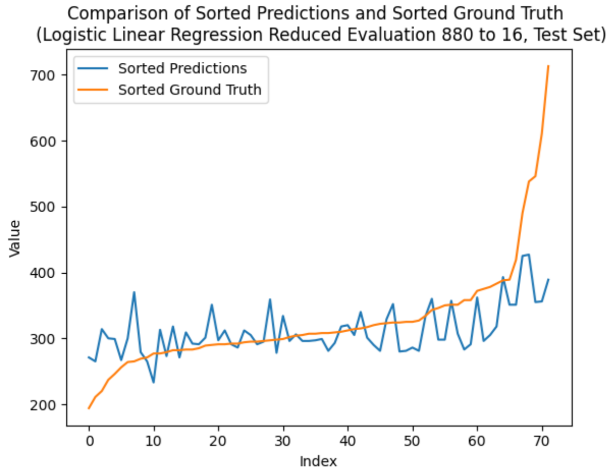
test eval:
Mean squared error: 2.7222
Correlation coefficient: 0.6060
R2 score: 0.3095

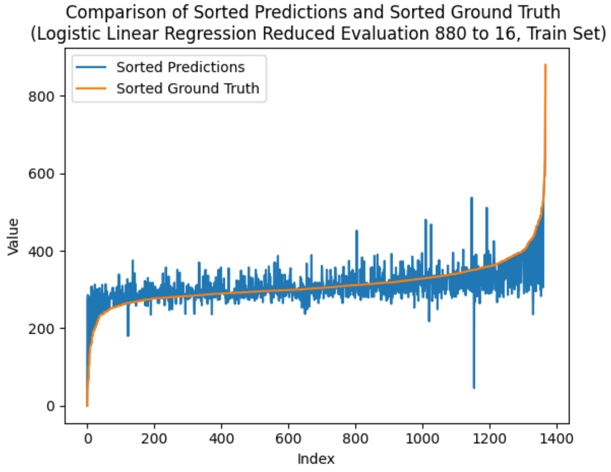
train eval:
Mean squared error: 1.7185
Correlation coefficient: 0.7430
R2 score: 0.4918

After red data recovering

test eval:
Mean squared error: 1.2916
Correlation coefficient: 0.6760
R2 score: 0.3941

train eval:
Mean squared error: 0.6038
Correlation coefficient: 0.6907
R2 score: 0.4257

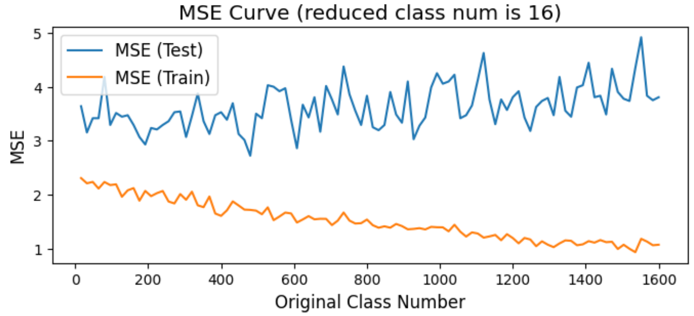


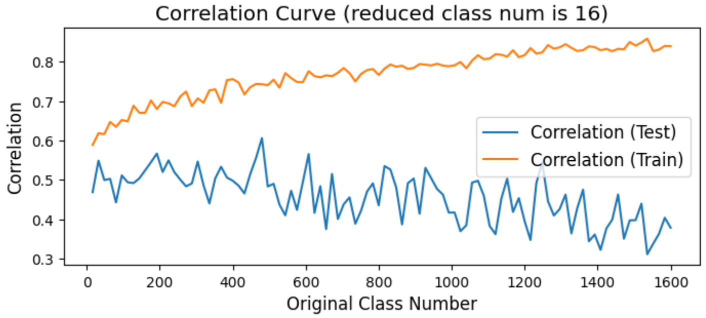


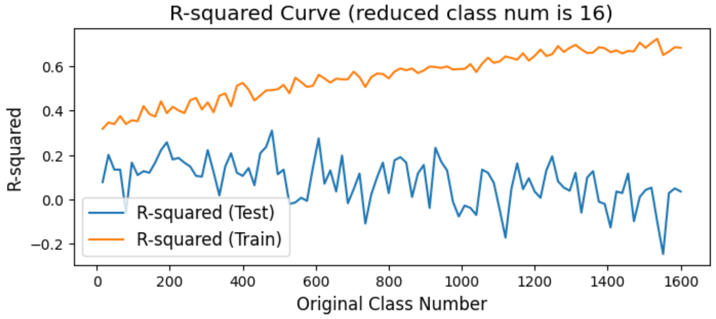
Basic Regression Modeling

Sample: Cell CL090_230515

We use recovered data to refit the logistic Model.

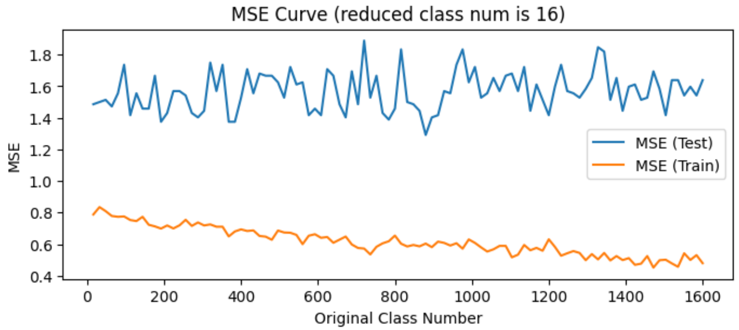


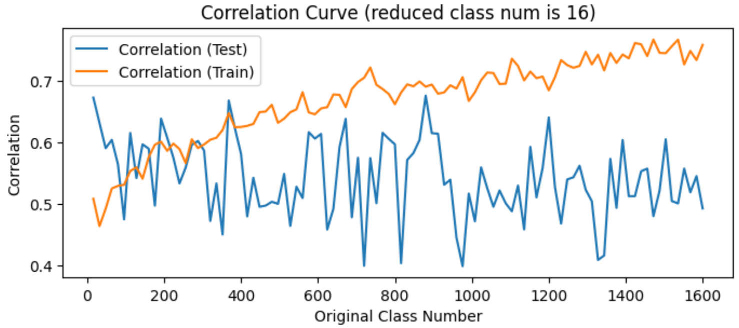


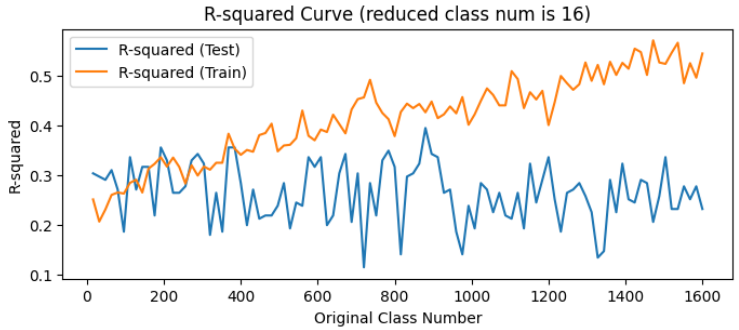


Before red data recovering

After red data recovering







Basic Regression Modeling

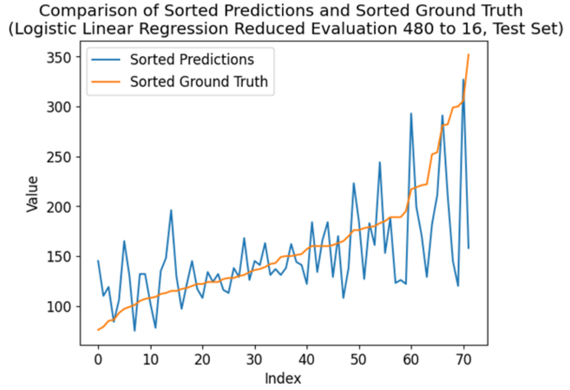
Sample: Cell CL090_230515

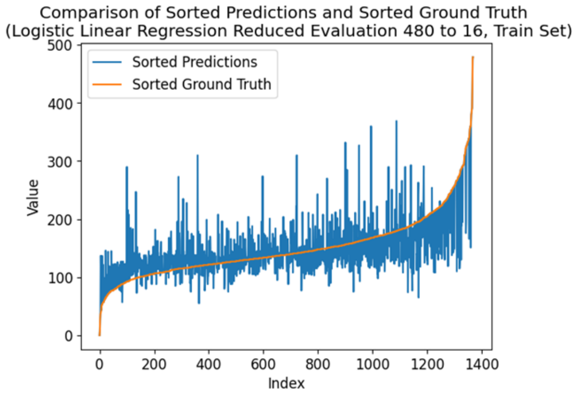
Does the pupil size improve the fitting? Not sure.

These two fittings have similar trends, which are slightly better than the other two, though the “fine” one doesn’t have obviously better numerical performance (probably because of deviation of few points).

w/o decay restoration
w/o pupil size

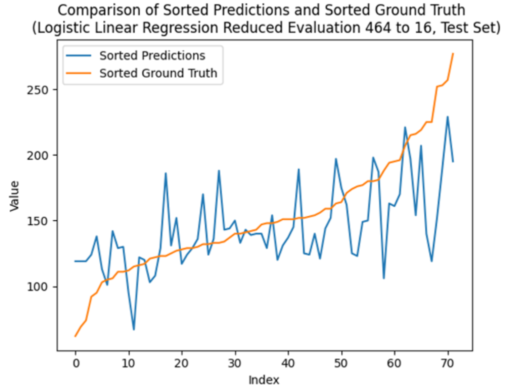
mse test: 2.72
r2 score test: 0.31
mse train: 1.72
r2 score train: 0.49

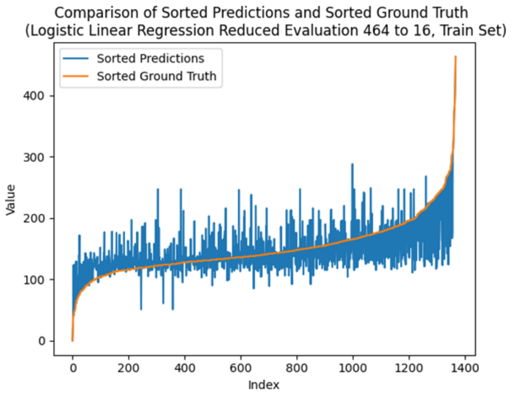




w/ decay restoration
w/o pupil size

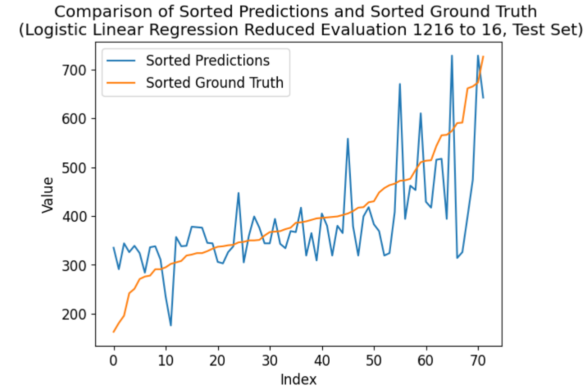
mse test: 1.56
r2 score test: 0.30
mse train: 1.26
r2 score train: 0.49

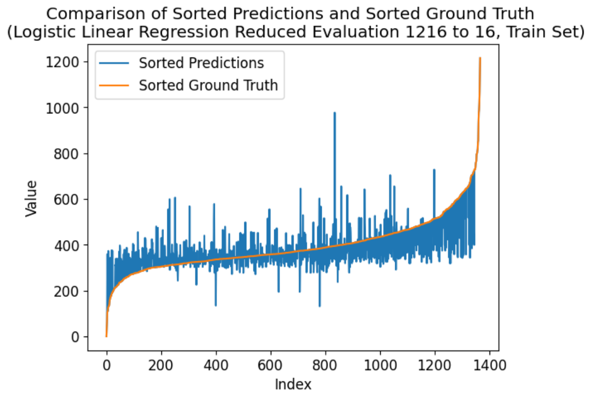




w/ decay restoration
w/ pupil size (mean)

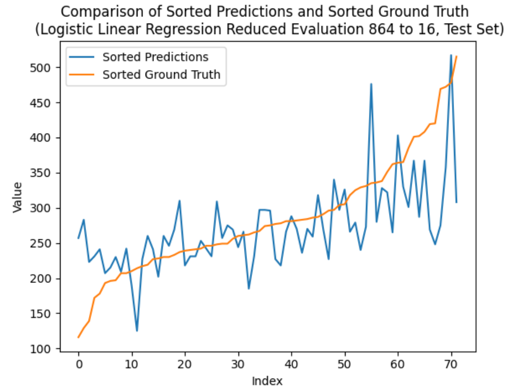
mse test: 1.35
r2 score test: 0.39
mse train: 0.96
r2 score train: 0.61

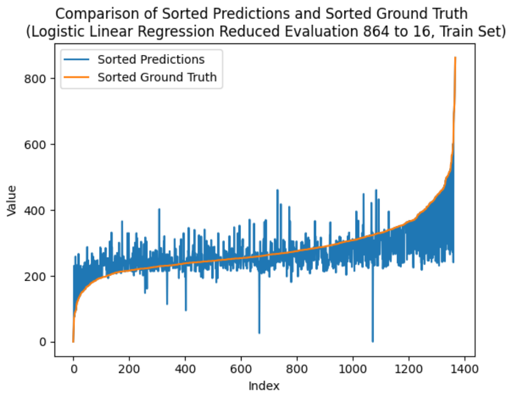




w/ decay restoration
w/ pupil size (fine)

mse test: 1.57
r2 score test: 0.29
mse train: 1.19
r2 score train: 0.52





Basic Regression Modeling

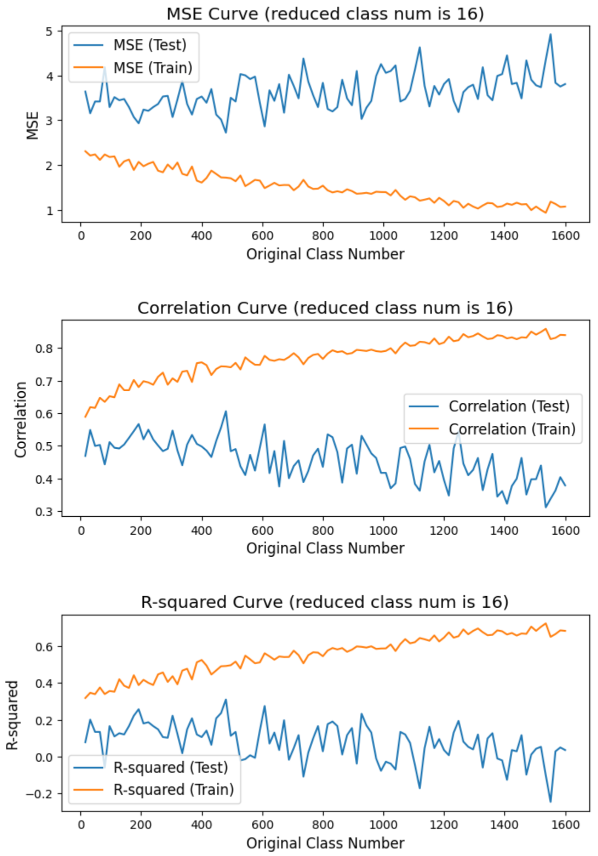
Sample: Cell CL090_230515

Does the pupil size improve the fitting? Not sure.

Average performance

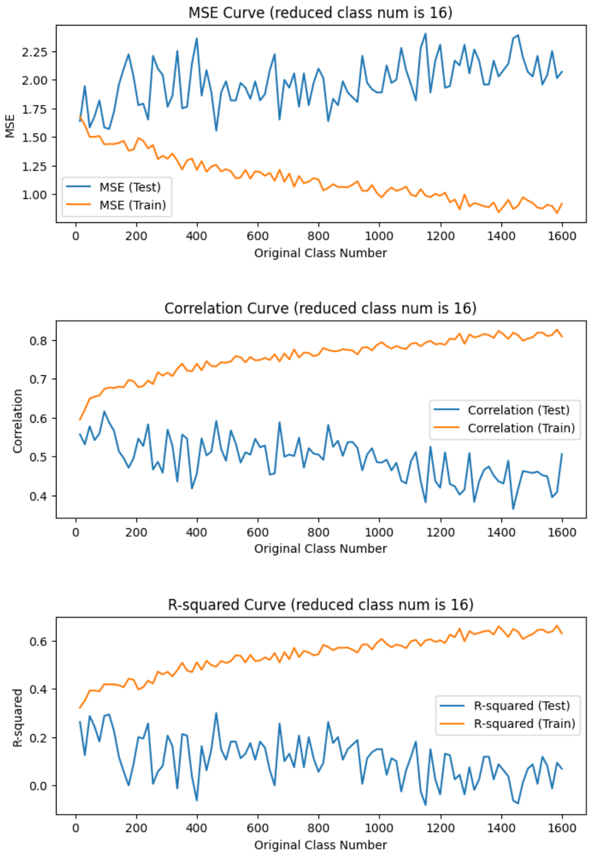
**w/o decay restoration
w/o pupil size**

mean of mse test: 3.63
mean of r2 score test: 0.08
mean of mse train: 1.51
mean of r2 score train: 0.55



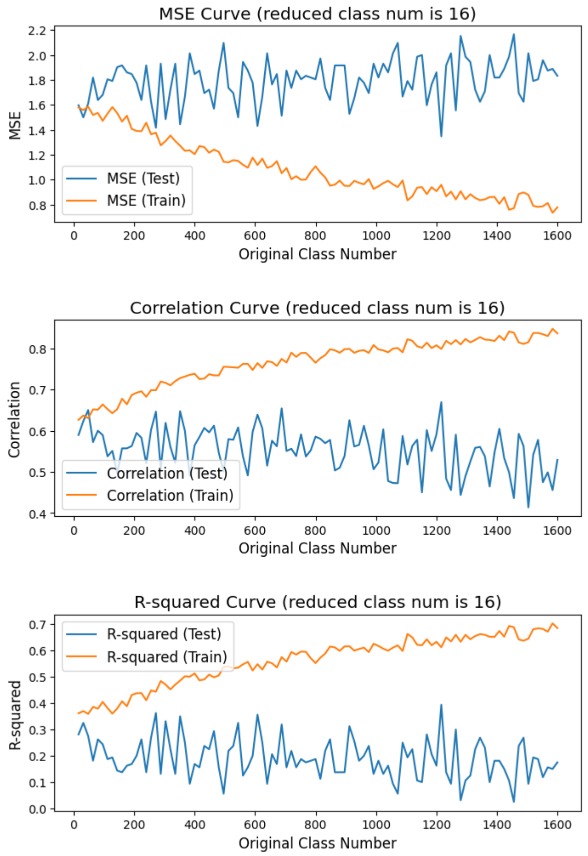
**w/ decay restoration
w/o pupil size**

mean of mse test: 1.98
mean of r2 score test: 0.11
mean of mse train: 1.13
mean of r2 score train: 0.54



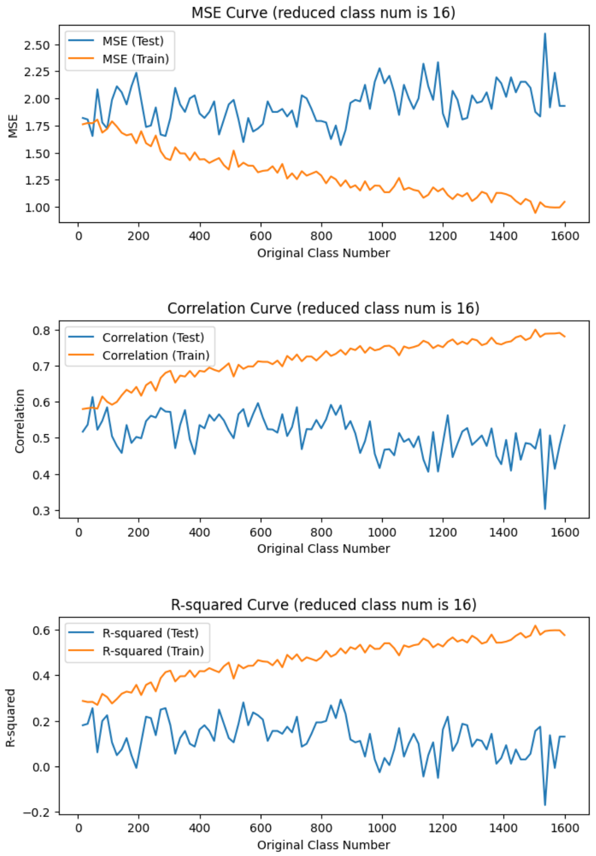
**w/ decay restoration
w/ pupil size (mean)**

mean of mse test: 1.79
mean of r2 score test: 0.19
mean of mse train: 1.08
mean of r2 score train: 0.56



**w/ decay restoration
w/ pupil size (fine)**

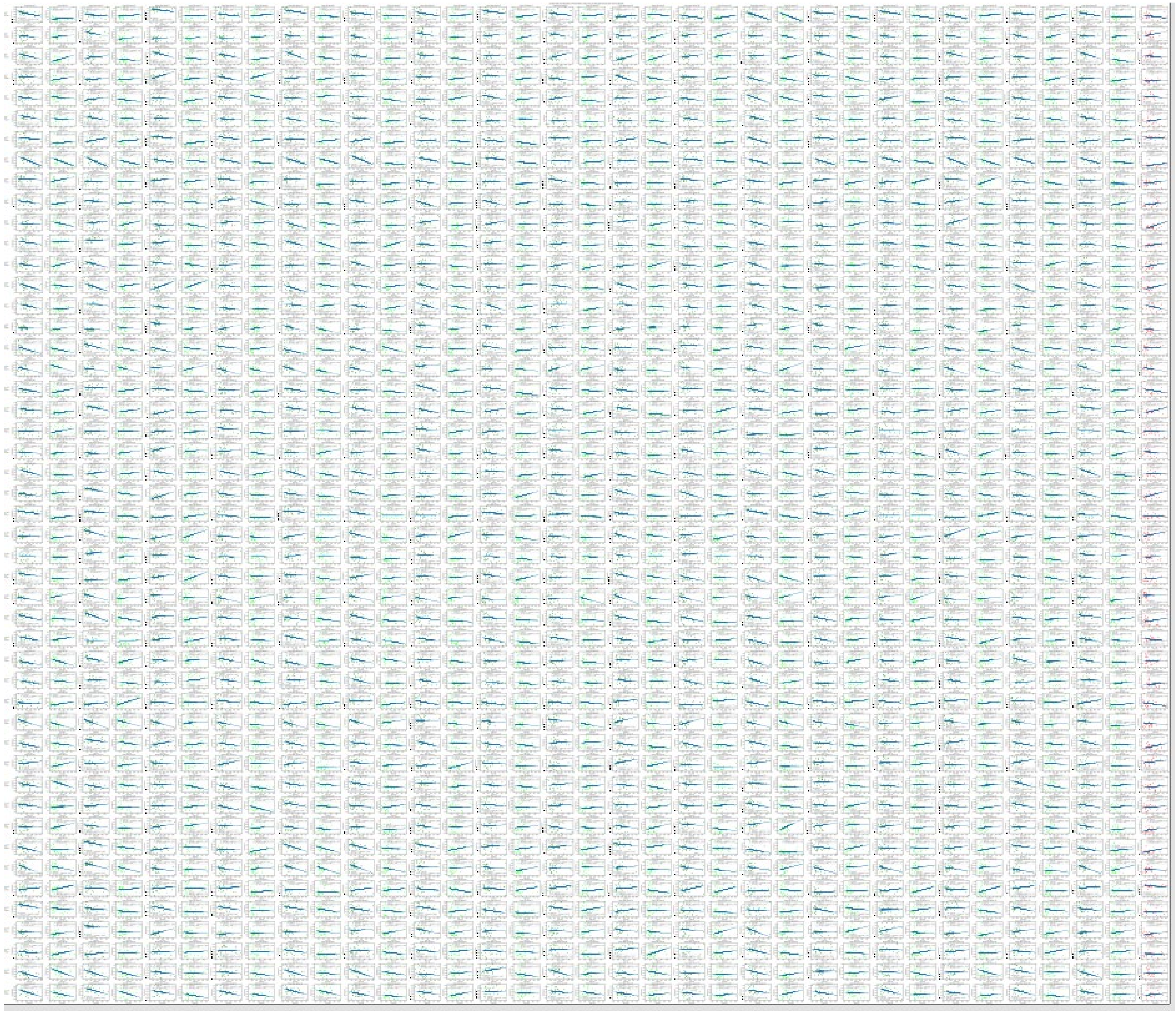
mean of mse test: 1.94
mean of r2 score test: 0.13
mean of mse train: 1.31
mean of r2 score train: 0.47



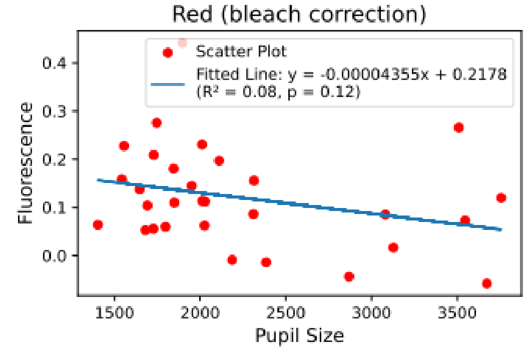
Basic Regression Modeling

Samples: Cell CL090_230515
Cell CL075_230228
Cell CL079_230324

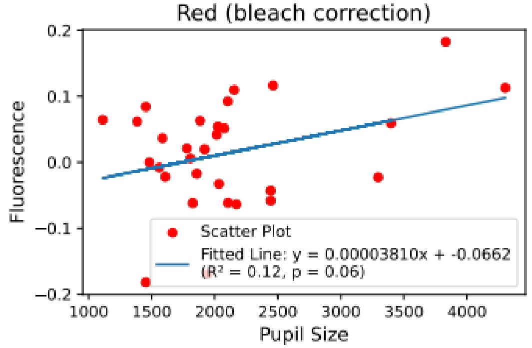
Does the pool size improve the fitting? Not very sure.



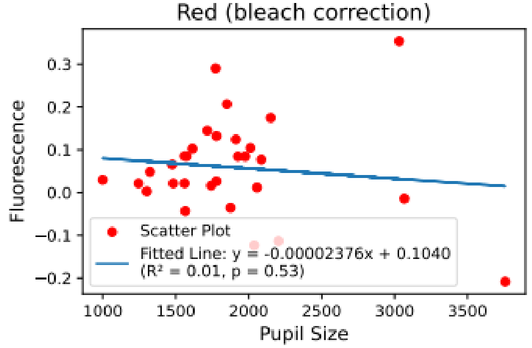
Ori: 0°
TF: 1
SF: 0.02



Ori: 0°
TF: 1
SF: 0.08



Ori: 0°
TF: 1
SF: 0.32



Conclusions:

1. Numerical metrics are difficult to distinguish the performance of linear and non linear regression methods.

2. Logistic regression is seemingly the best one when we considering both the numerical metrics and the graphic results.

3. Decay restoration is very important for the red data and for the fitting.

4. It’s still not sure whether the pupil size can affect the fitting.

1. Background and Data Description

2. Basic Regression Modeling

3. Reevaluating Data and Refining Data Restoration

4. Biologically Interpretable Modeling

5. Limitations and Future Work

Reevaluating Data and Refining Data Restoration

How to further improve the performance?

Rethink what data we want and use it for what model.

- If using the average data of green and red fluorescence, we can only have one 1440 pairs of green and red data – it is a small number of data pieces.

- If using the signal traces of green and red fluorescence we can cut a period from the traces anywhere and we can have much more data.

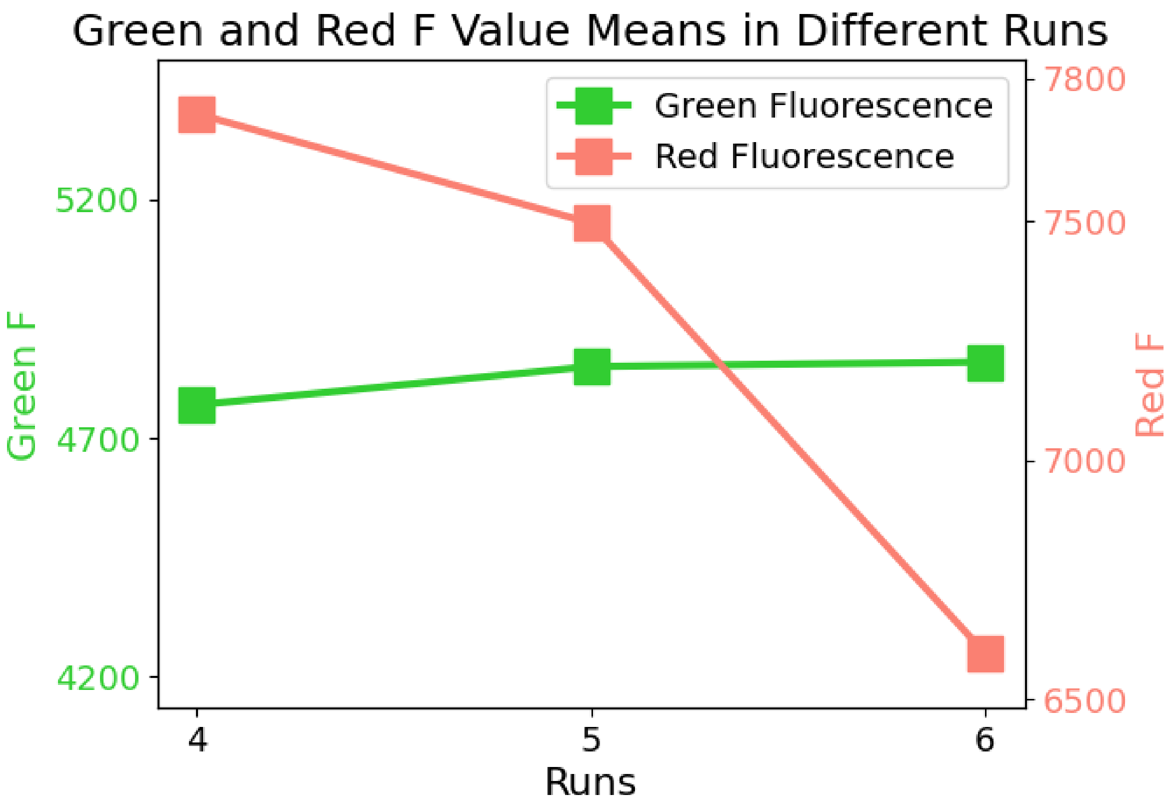
- For modeling, we don’t need to consider whether the data is during the visual stimuli; we only need to care the data is the input and output of neurons.

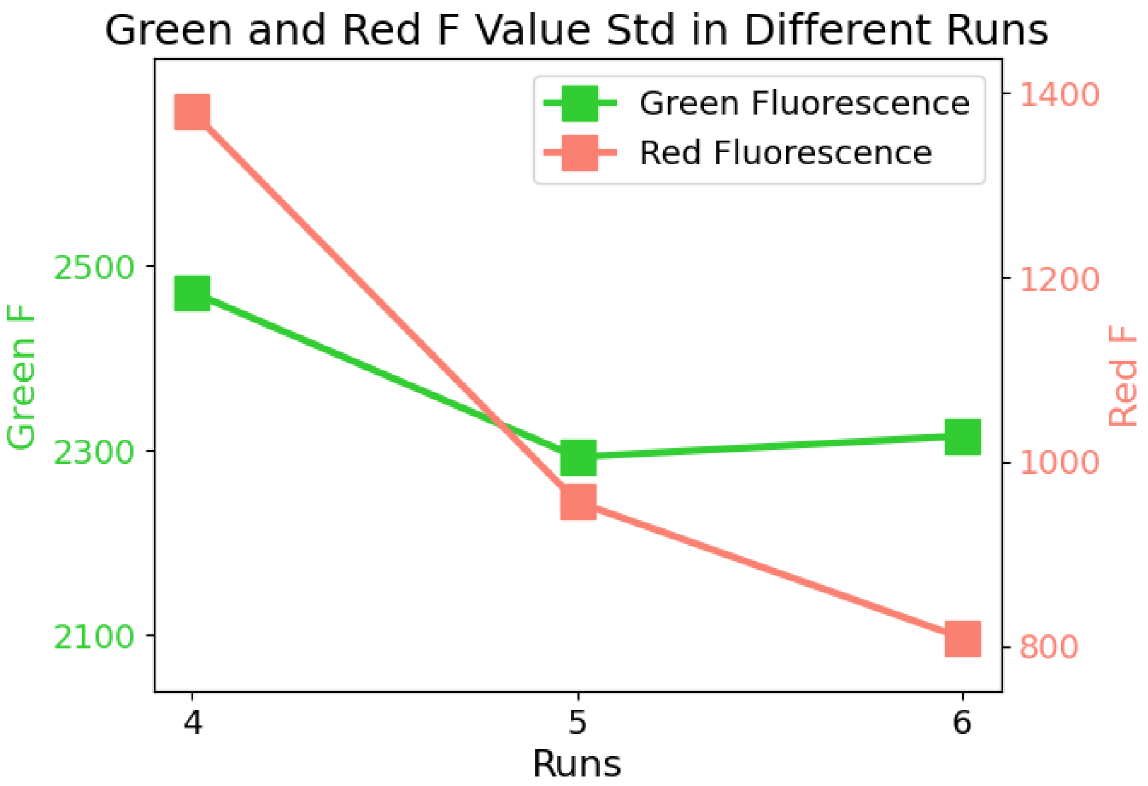
- Visual stimuli conditions can be used for further analysis.

- We may need to restore the all the trace data from both the mean and standard deviation aspects.

Reevaluating Data and Refining Data Restoration

Sample: Cell CL090_230515

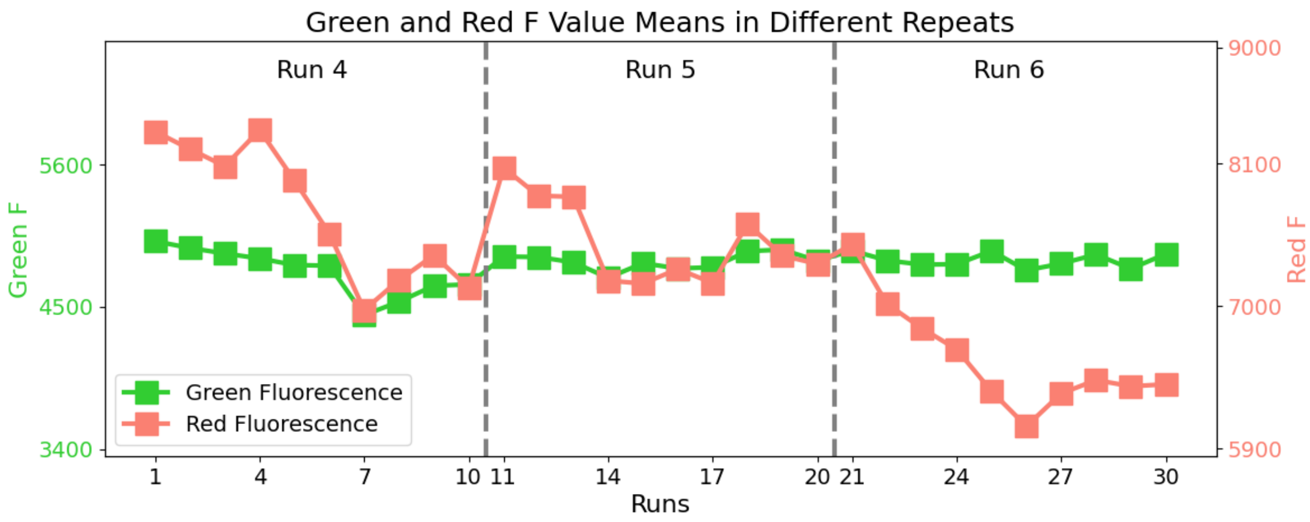


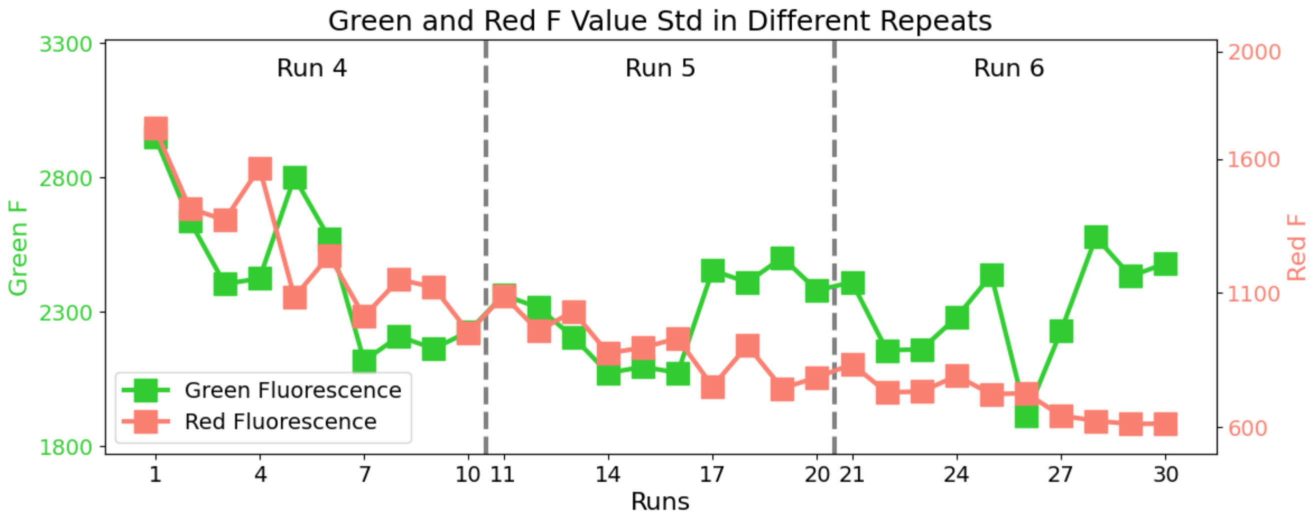


It can be seen that both the means and standard deviations (related to dF/F) decay as the time goes.

Reevaluating Data and Refining Data Restoration

Sample: Cell CL090_230515





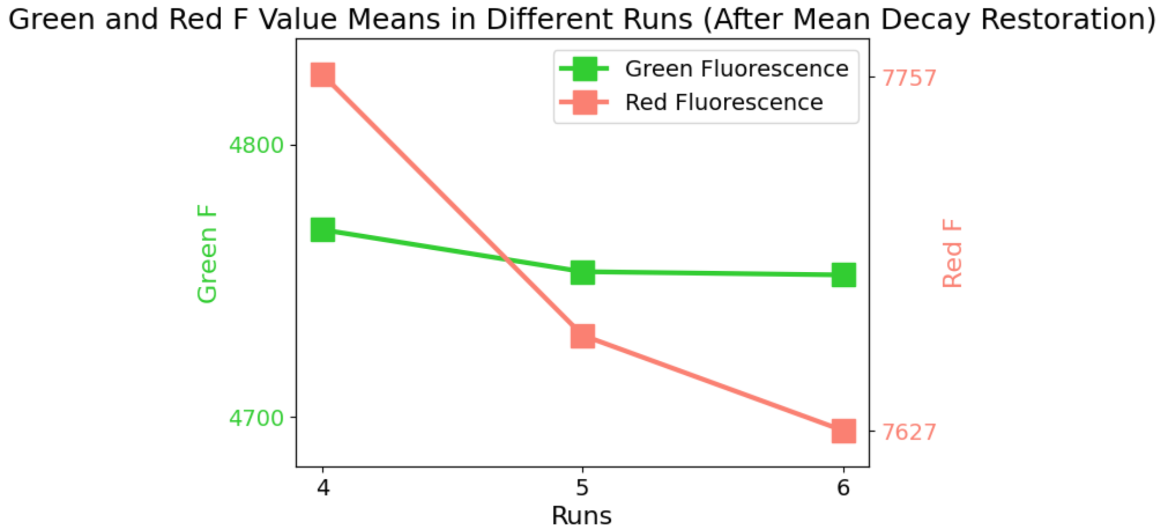
More clear in repeats.

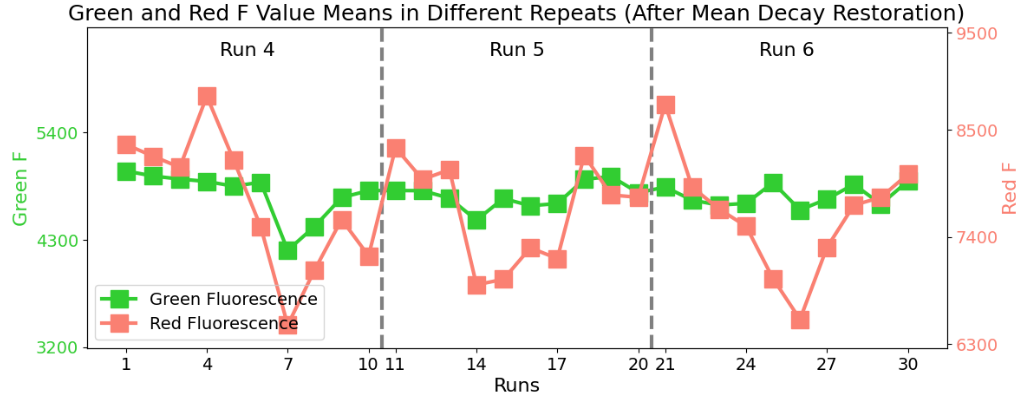
Run 6 seems to be with bad quality?

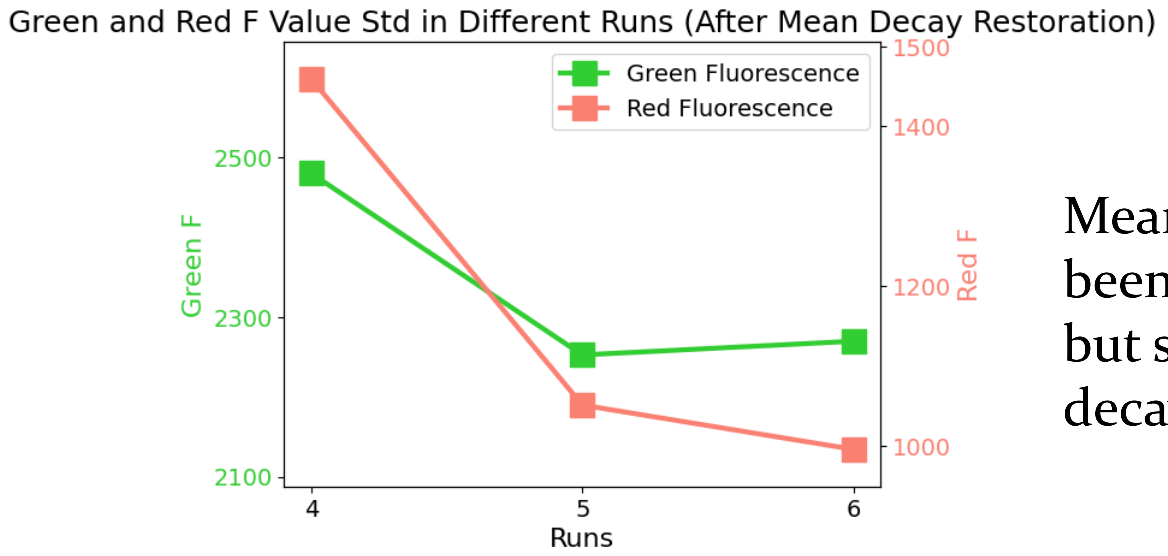
Reevaluating Data and Refining Data Restoration

Sample: Cell CL090_230515

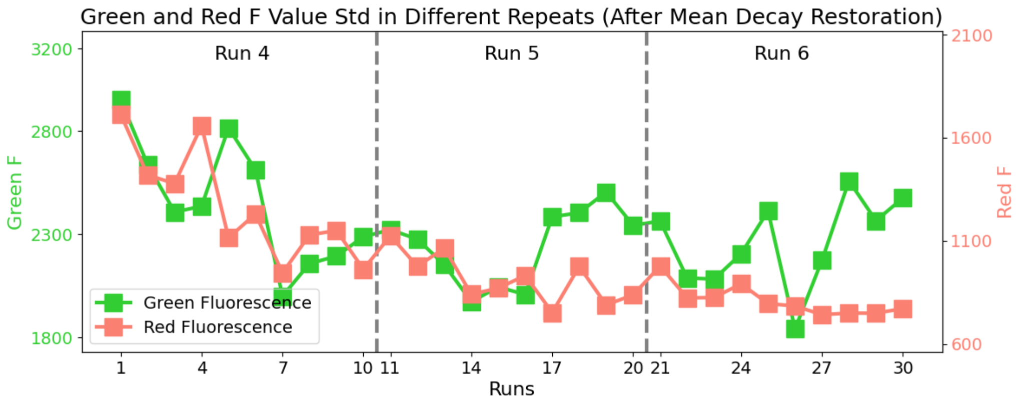
Still using exponential function to restore.
First retore the mean and then restore the standard deviation.







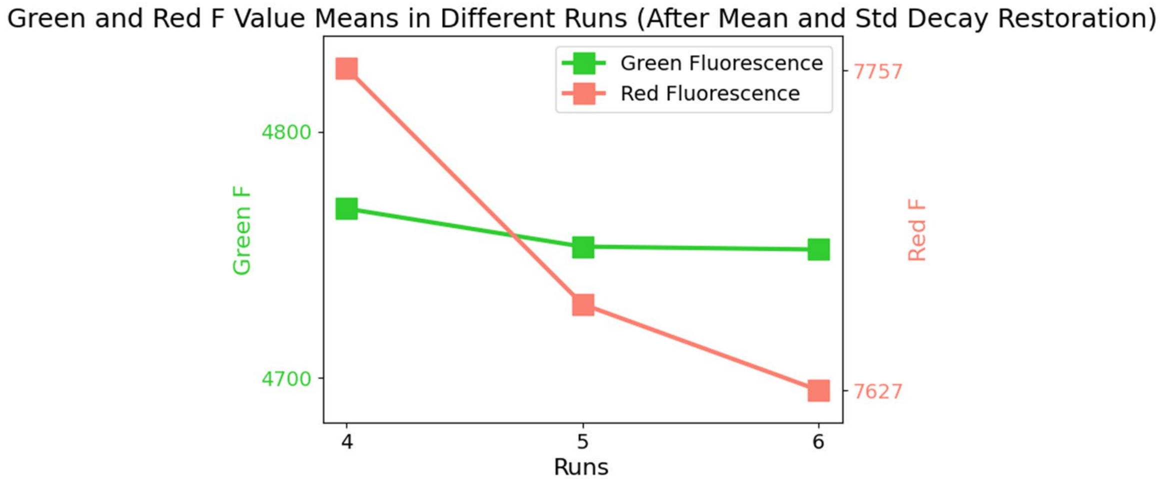
Means have been stable but std still decay.

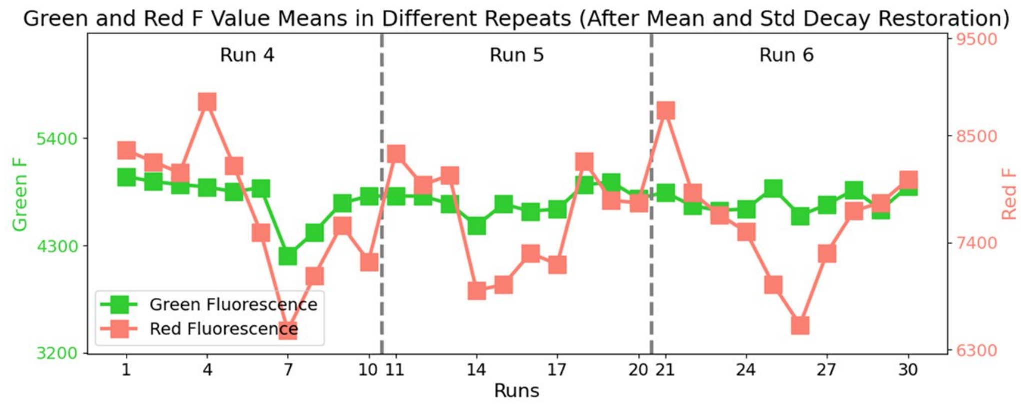


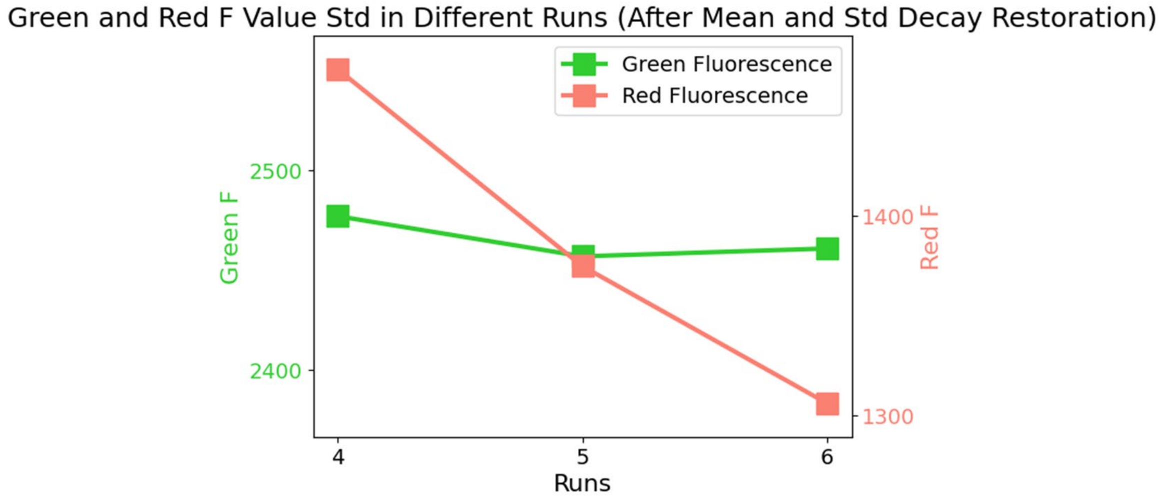
Reevaluating Data and Refining Data Restoration

Sample: Cell CL090_230515

Still using exponential function to restore.
First retore the mean and then restore the standard deviation.





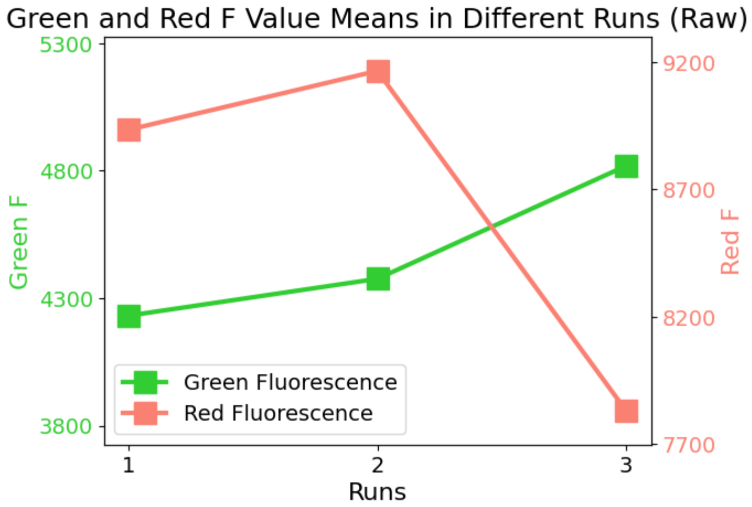


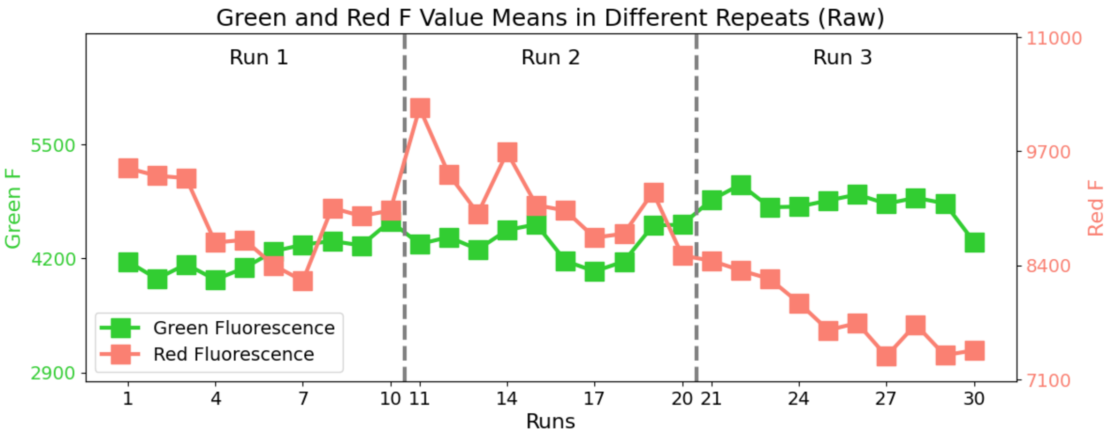
Reevaluating Data and Refining Data Restoration

Sample: Cell CL075_230228

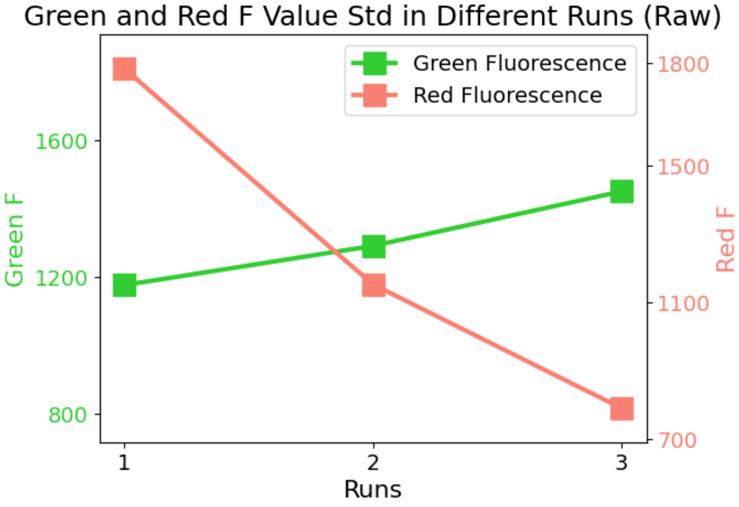
Raw signal traces

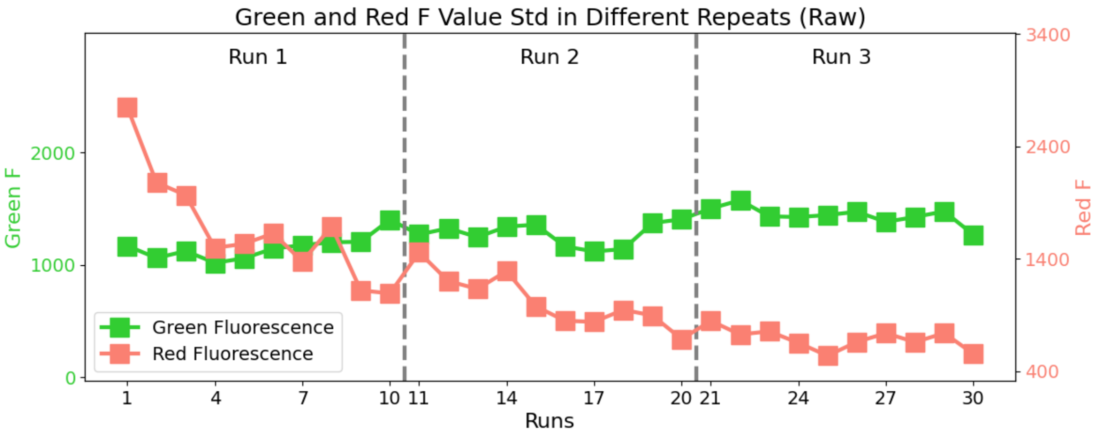
mean of each run of green:
run 1-- mean: 4231.62
run 2-- mean: 4376.03
run 3-- mean: 4819.07
mean of each run of red:
run 1-- mean: 8934.01
run 2-- mean: 9164.82
run 3-- mean: 7832.18





std of each run of green:
run 0 -- std: 1177.21
run 1 -- std: 1291.94
run 2 -- std: 1451.44
std of each run of red:
run 0 -- std: 1784.19
run 1 -- std: 1152.54
run 2 -- std: 790.13





Reevaluating Data and Refining Data Restoration

Fitting and Recovering for Mean Restoration

Green Restoration
coefficient in the exponent: 0.006480
coefficient in the exponent: 0.000619
coefficient in the exponent: -0.00336
Red Restoration
coefficient in the exponent: -0.012168
coefficient in the exponent: -0.021185
coefficient in the exponent: -0.016837

For this cell, in the restoration for green data, there are two coefficients in the exponent of the exponential function are positive, that is, the curve is not decay but increasing.

Previously, in the fitting of the cell CL090_230515, all the coefficients are negative.

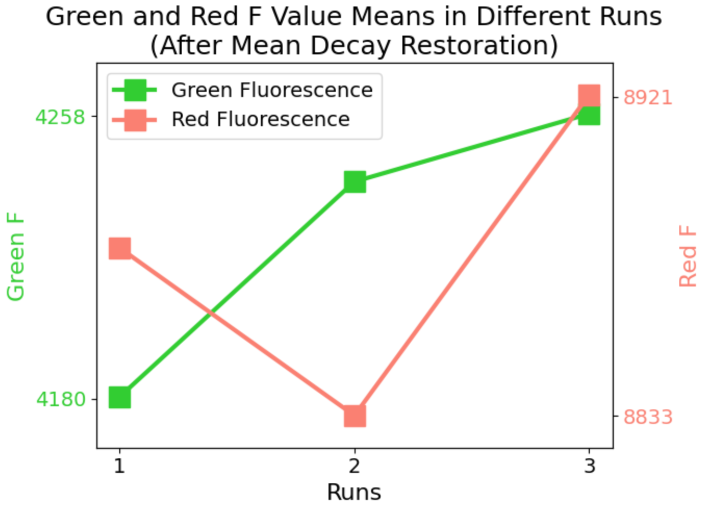
So, the restoration is no longer only for decay recovering but also has the data normalization purpose.

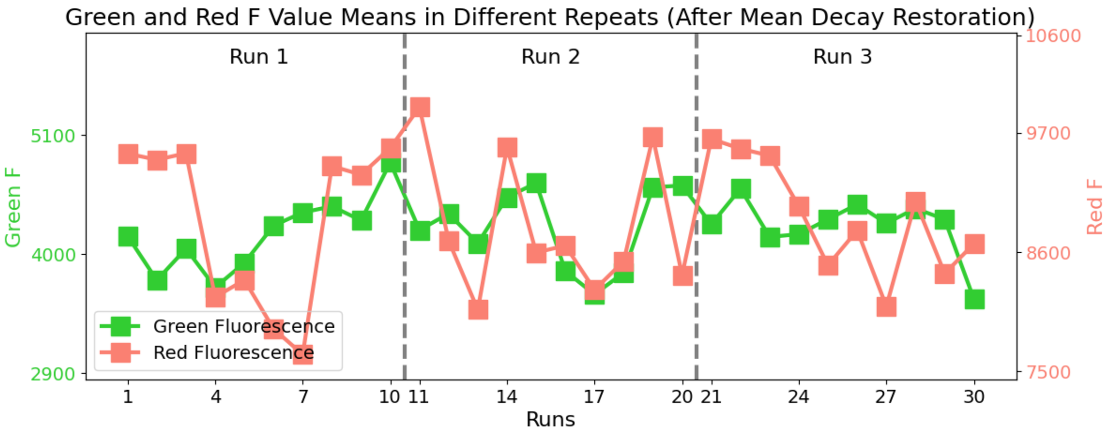
Reevaluating Data and Refining Data Restoration

Sample: Cell CL075_230228

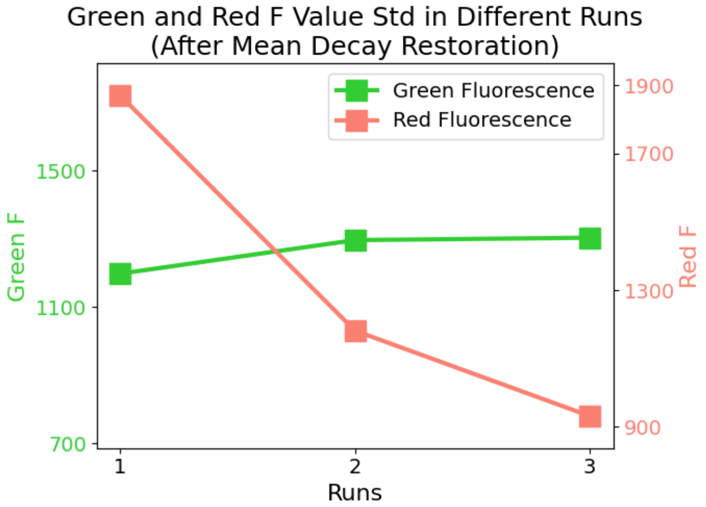
Mean restored signal traces

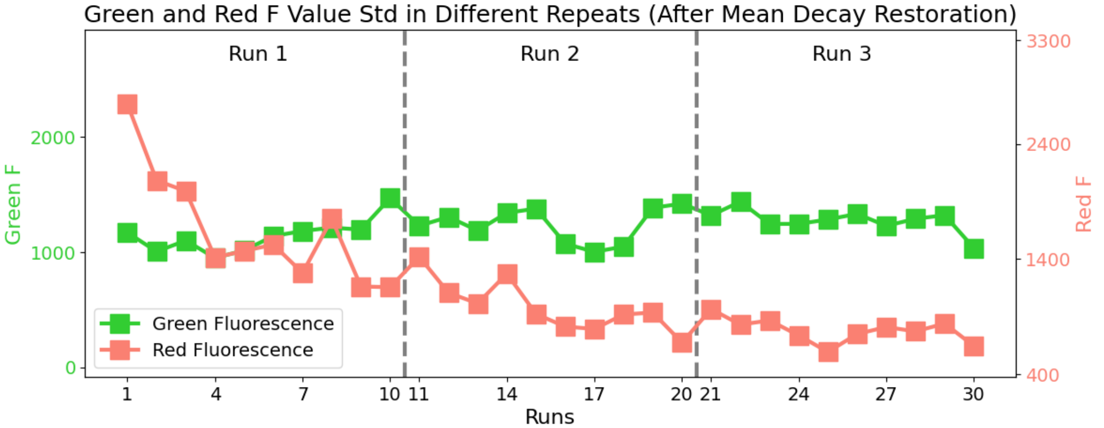
mean of each run of green:
run 1 -- mean: 4180.54
run 2 -- mean: 4239.89
run 3 -- mean: 4258.65
mean of each run of red:
run 1 -- mean: 8879.32
run 2 -- mean: 8833.06
run 3 -- mean: 8921.59





std of each run of green:
run 1 -- std: 1197.70
run 2 -- std: 1295.90
run 3 -- std: 1302.90
std of each run of red:
run 1 -- std: 1870.70
run 2 -- std: 1181.94
run 3 -- std: 932.01





Reevaluating Data and Refining Data Restoration

Sample: Cell CL075_230228

Fitting and Recovering for Std Restoration

Green Restoration
coefficient in the exponent: 0.004730
coefficient in the exponent: 0.001408
coefficient in the exponent: -0.009869
Red Restoration
coefficient in the exponent: -0.117278
coefficient in the exponent: -0.080808
coefficient in the exponent: -0.042174

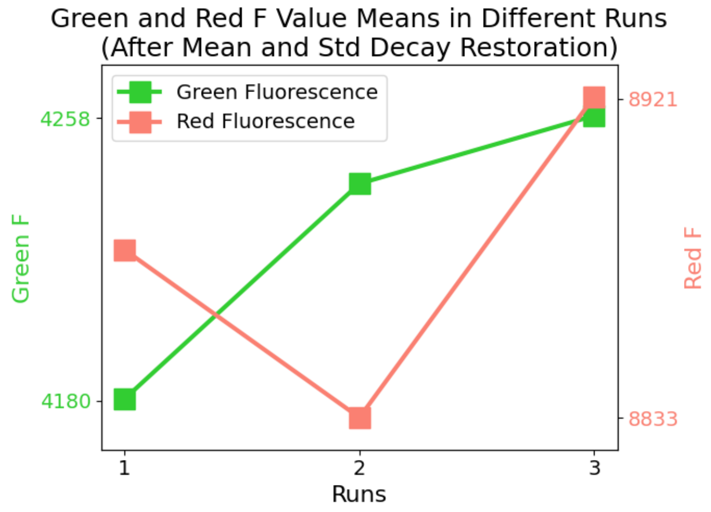
Also, there are two coefficients in the exponent of the exponential function are positive.

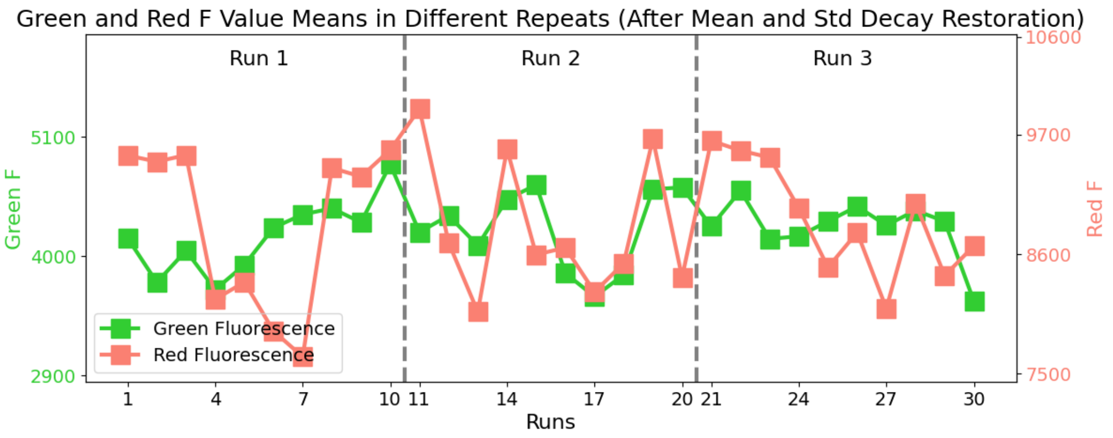
Reevaluating Data and Refining Data Restoration

Mean and Std restored signal traces

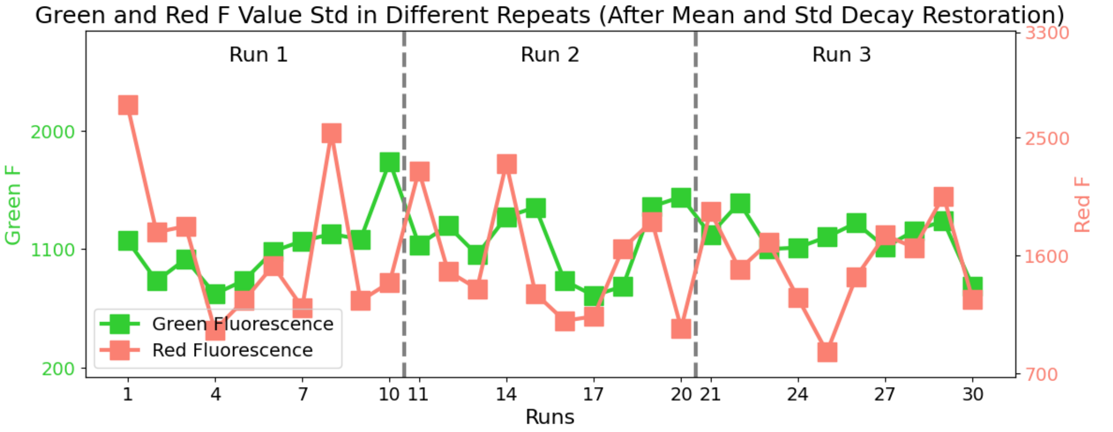
Sample: Cell CL075_230228

mean of each run of green:
run 1 -- mean: 4180.54
run 2 -- mean: 4239.89
run 3 -- mean: 4258.65
mean of each run of red:
run 1 -- mean: 8879.32
run 2 -- mean: 8833.06
run 3 -- mean: 8921.59





std of each run of green:
run 1 -- std: 1186.16
run 2 -- std: 1241.24
run 3 -- std: 1224.70
std of each run of red:
run 1 -- std: 1888.78
run 2 -- std: 1721.93
run 3 -- std: 1656.93



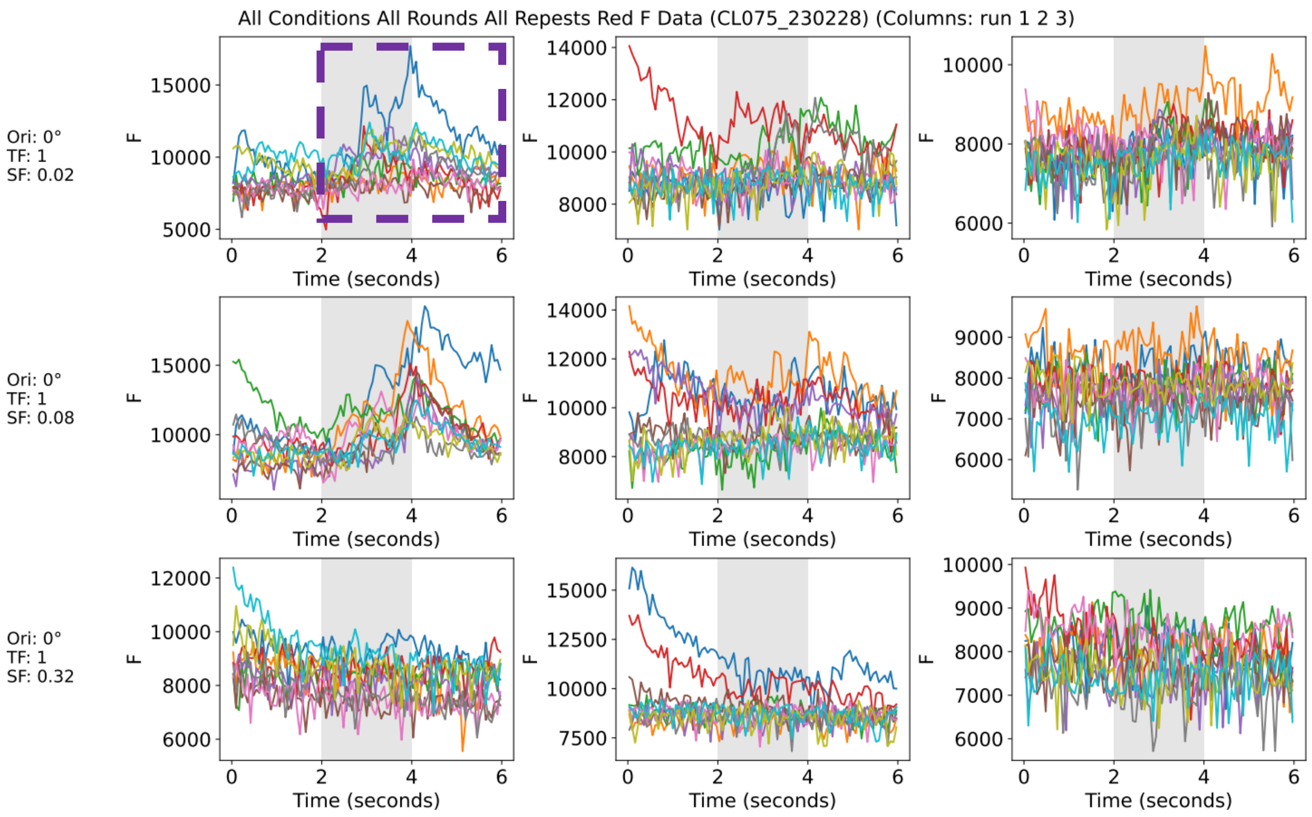
Reevaluating Data and Refining Data Restoration

Sample: Cell CL075_230228

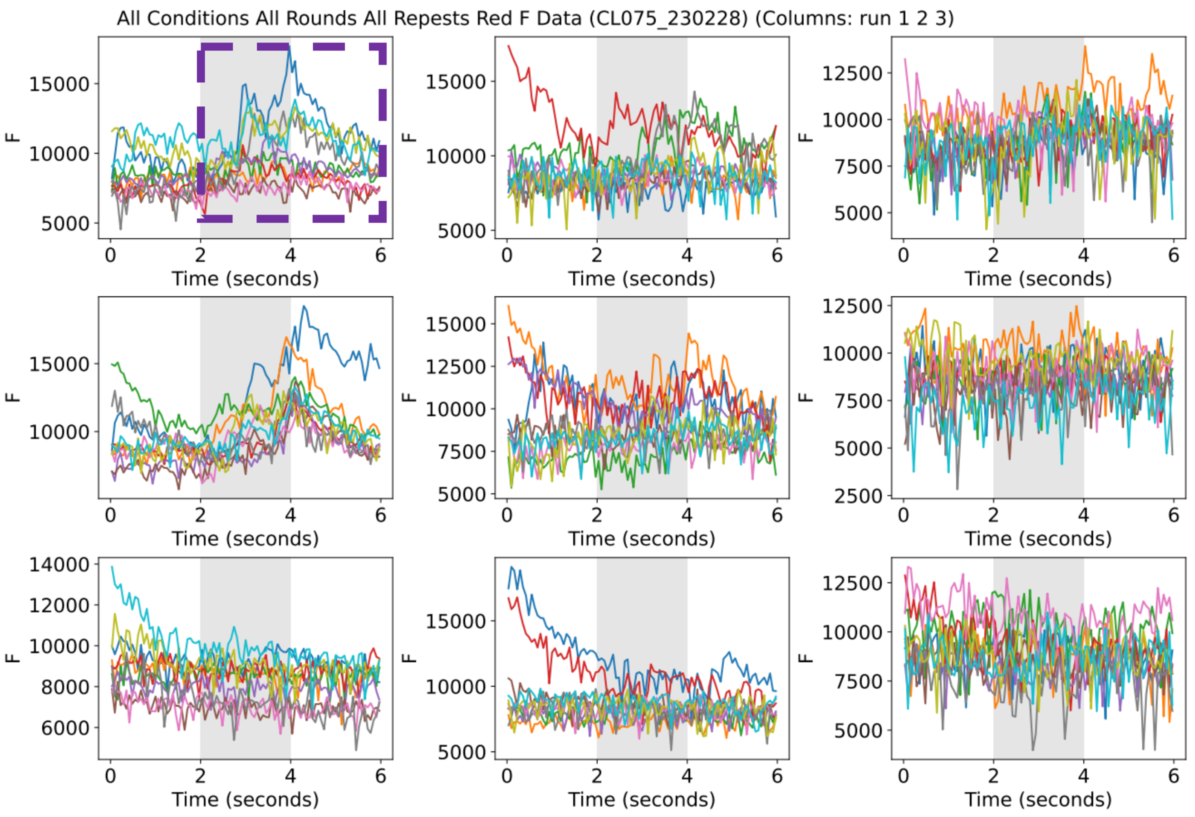
Check tuning curves to see whether the restoration influences tuning properties.

Red Data

Raw Data



After Mean and Std Restoration



Values become higher, especially Runs 2 and 3 .

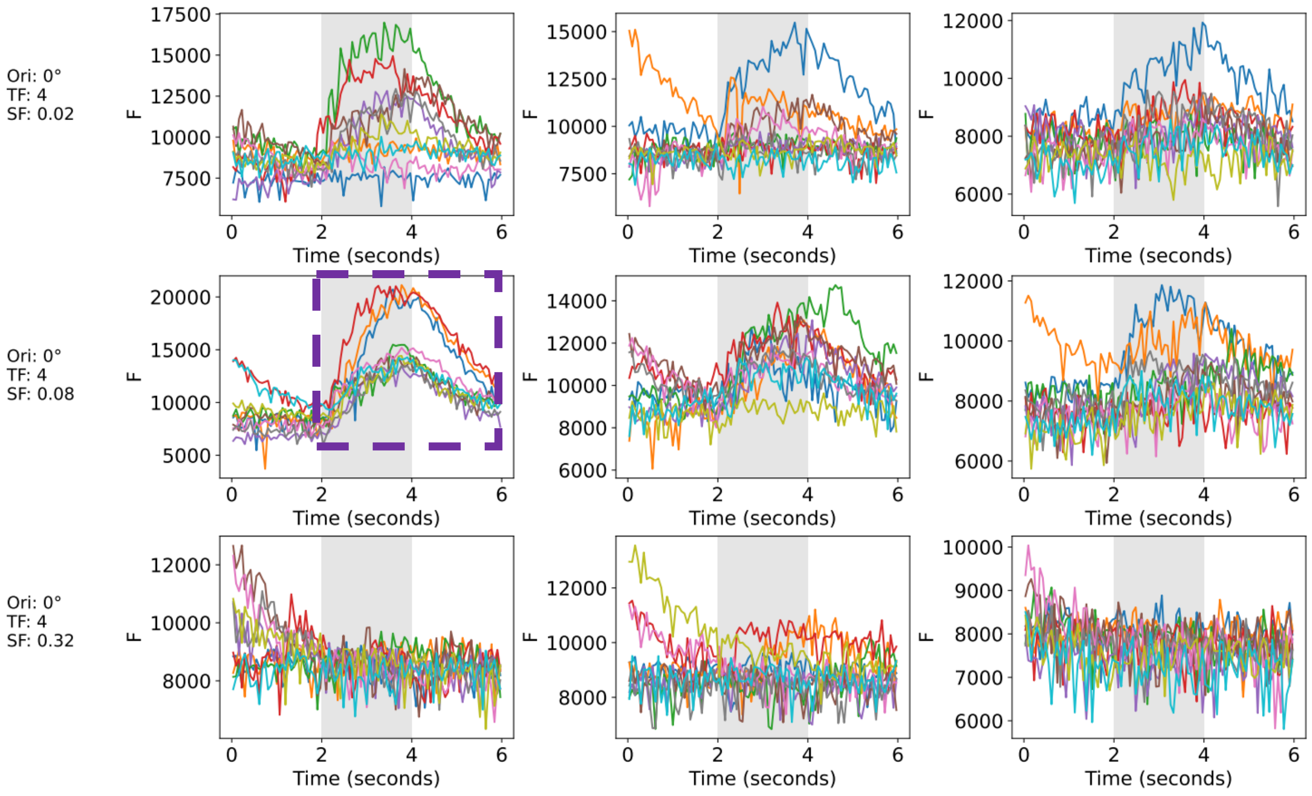
Reevaluating Data and Refining Data Restoration

Sample: Cell CL075_230228

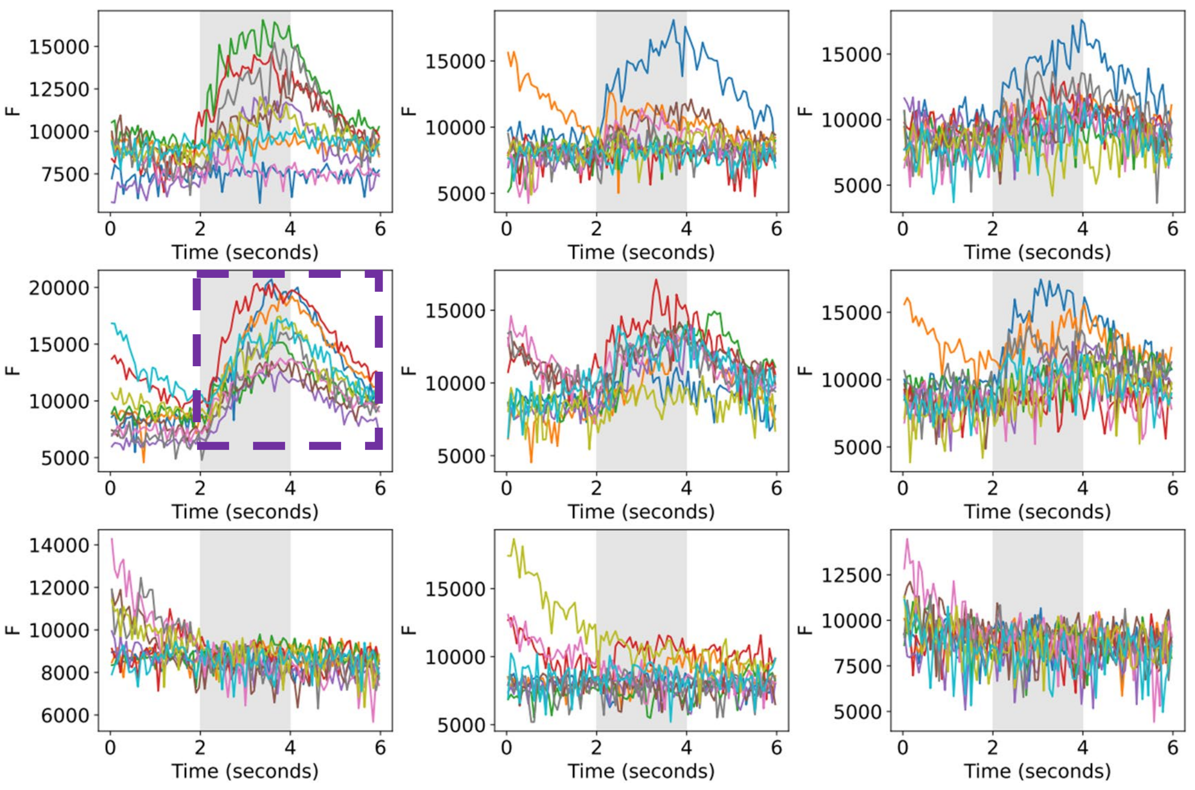
Check tuning curves to see whether the restoration influences tuning properties.

Red Data

Raw Data



After Mean and Std Restoration



Values become higher, especially Runs 2 and 3 .

Reevaluating Data and Refining Data Restoration

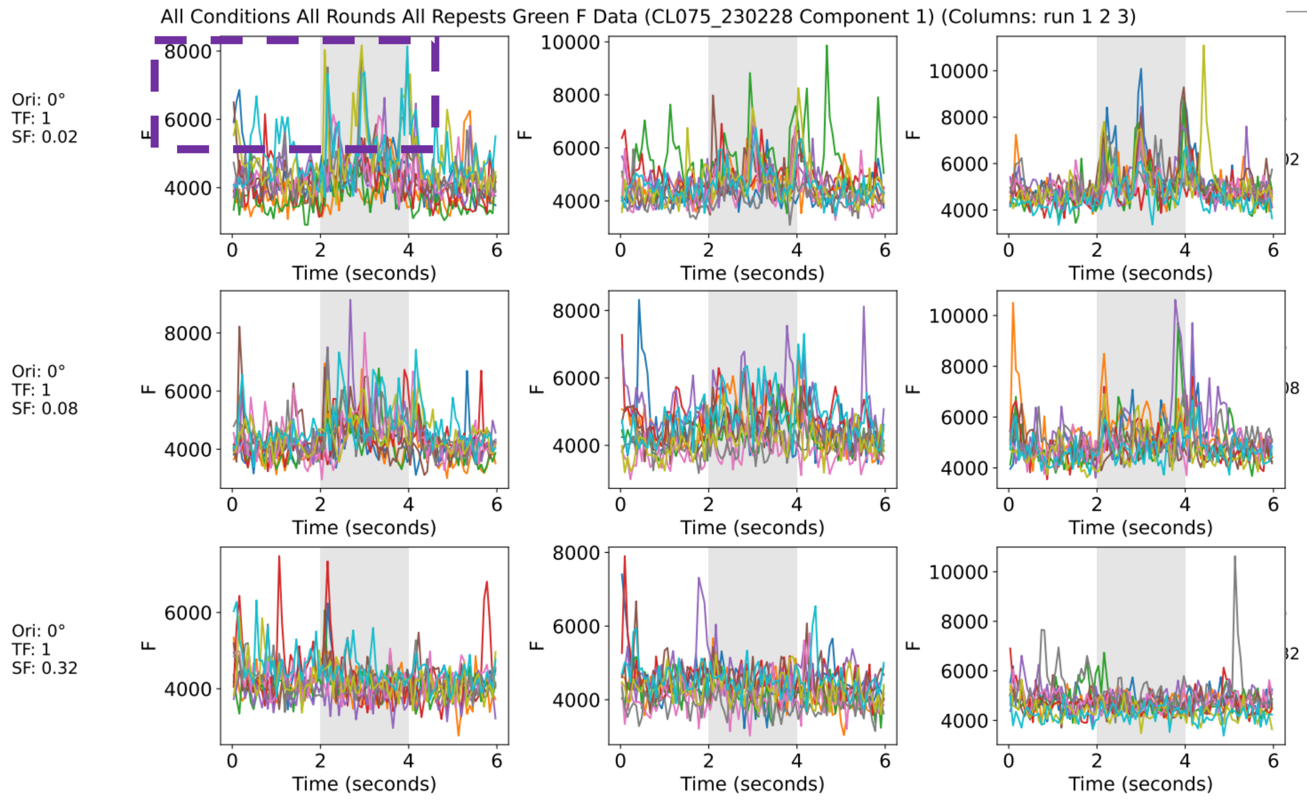
Sample: Cell CL075_230228

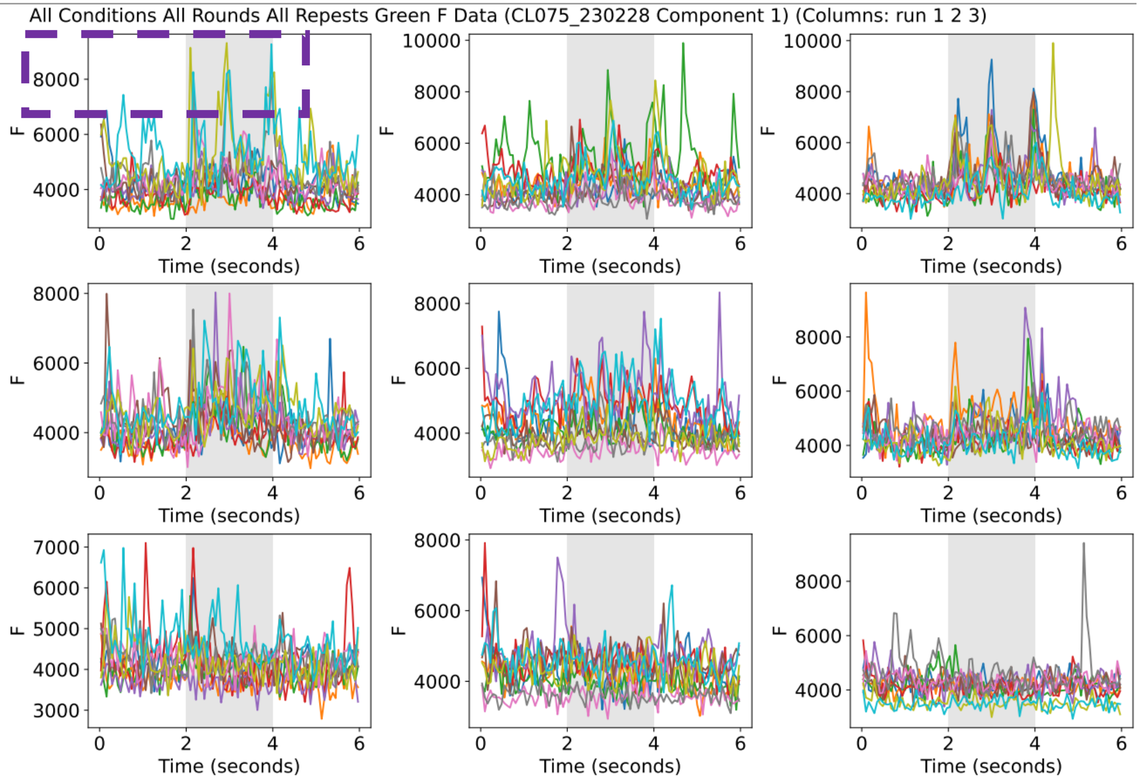
Check tuning curves to see whether the restoration influences tuning properties.

Green Data

Raw Data

After Mean and Std Restoration





More similar than red, because less restored; but a few responses become stronger.

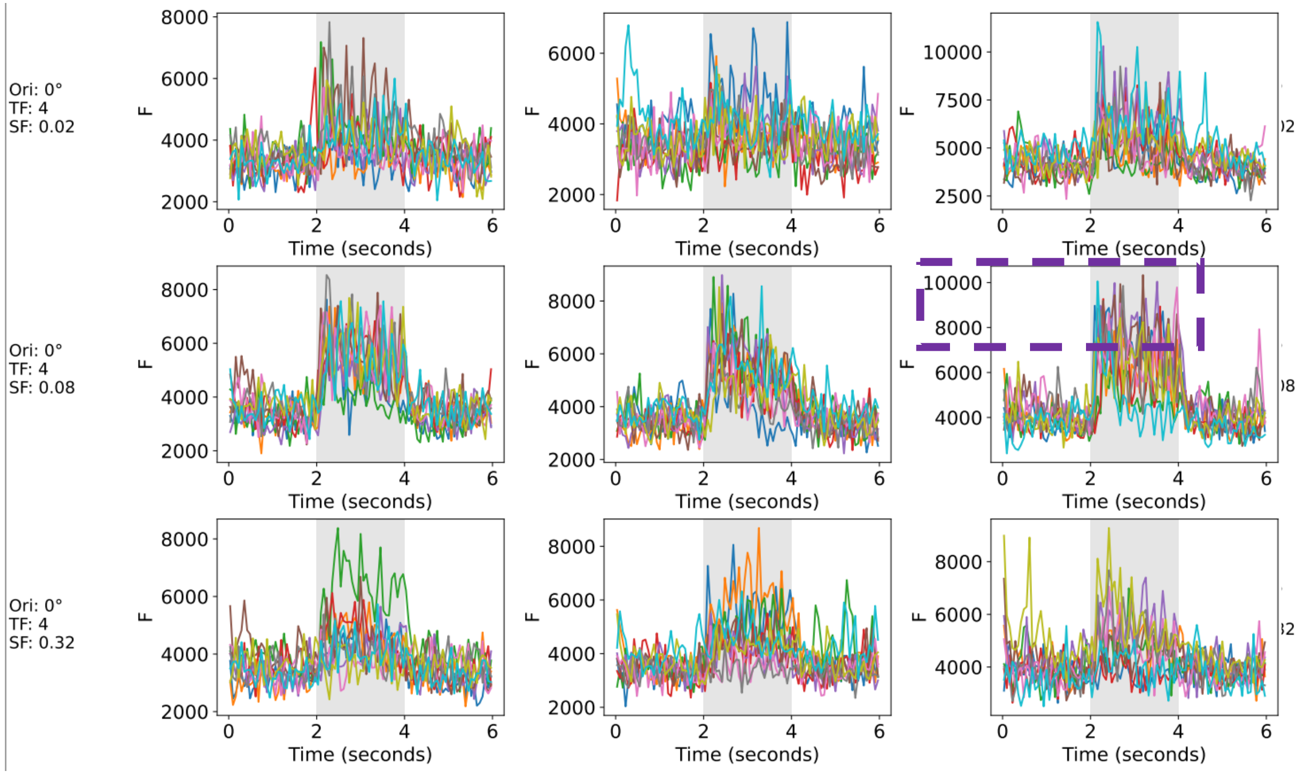
Reevaluating Data and Refining Data Restoration

Sample: Cell CL075_230228

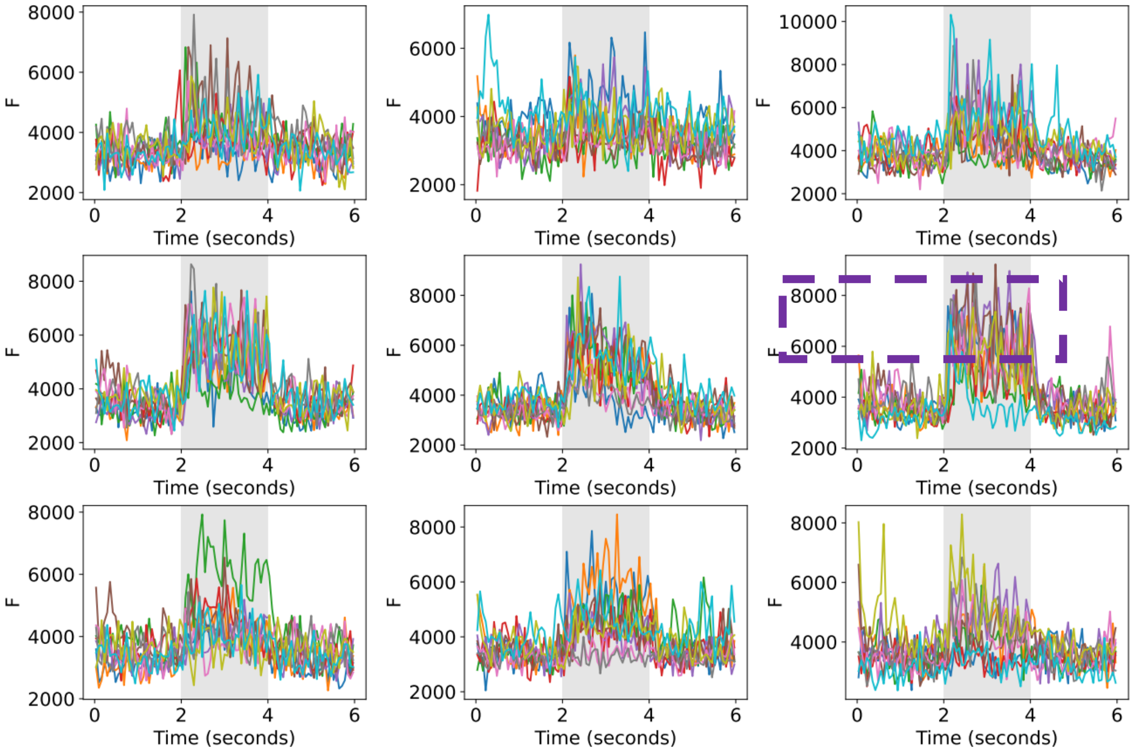
Check tuning curves to see whether the restoration influences tuning properties.

Green Data

Raw Data



After Mean and Std Restoration



Also, a few responses become weaker (e.g., for run 3), because this cell green std increases in later repeats, unlike cell090_230515.

Reevaluating Data and Refining Data Restoration

Conclusions:

1. Restoration of the whole signal trace is more accurate and more helpful, compared to the basic restoration used before.

2. Restoration of both the main and standard deviation makes the data better.

3. The restoration doesn’t destroy the tuning properties.

1. Background and Data Description

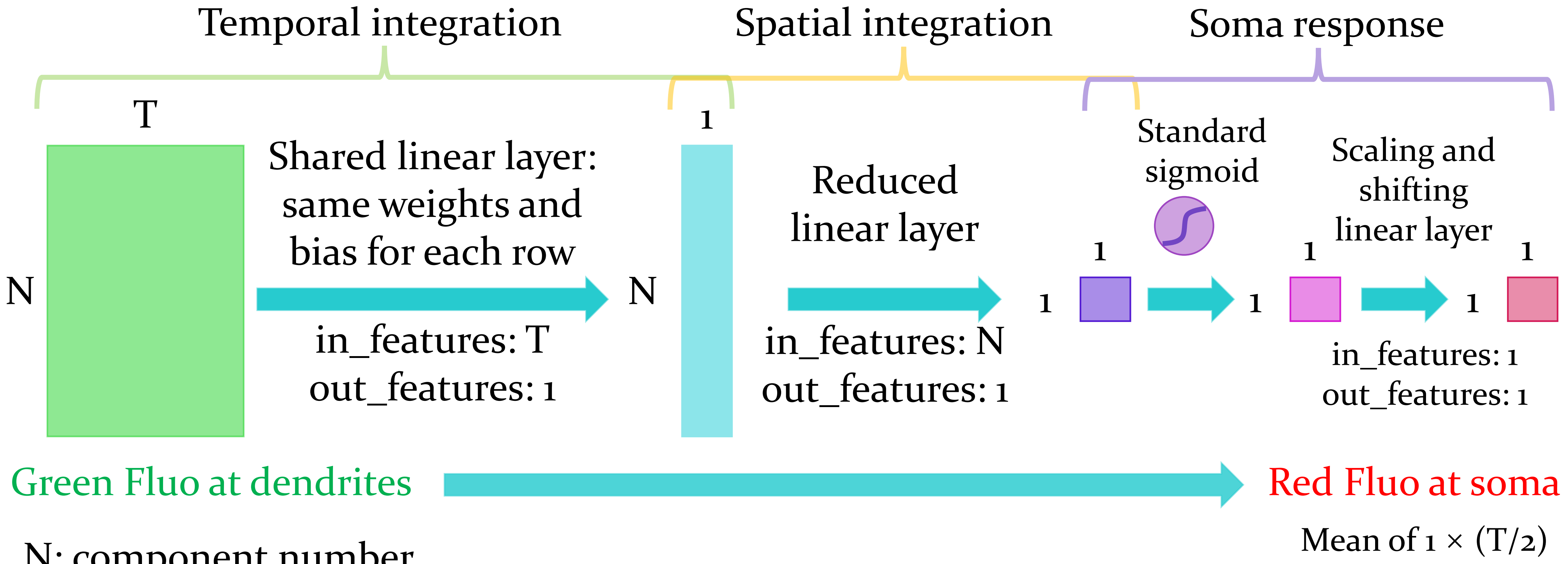
2. Basic Regression Modeling

3. Reevaluating Data and Refining Data Restoration

4. Biologically Interpretable Modeling

5. Limitations and Future Work

Biologically Interpretable Modeling



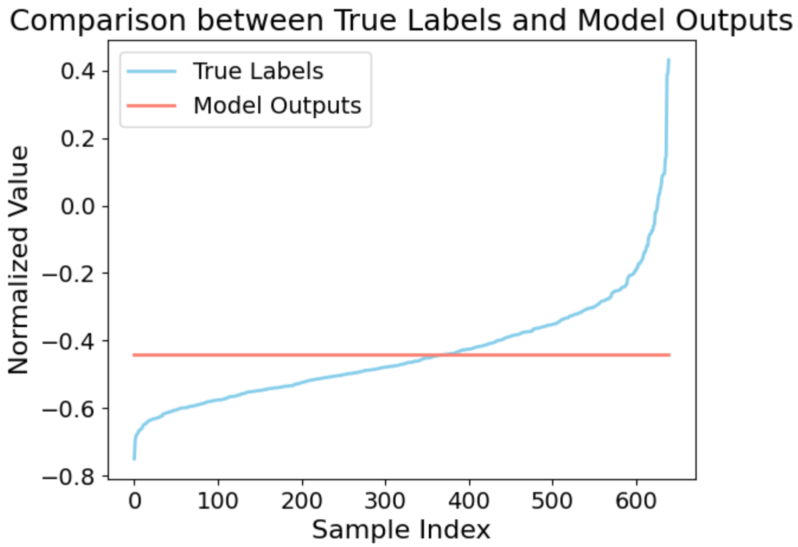
N: component number

T: frame number

Biologically Interpretable Modeling

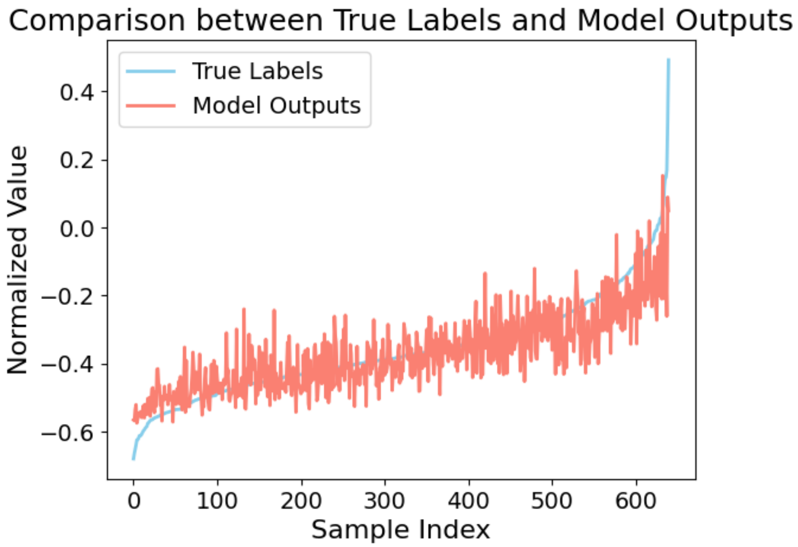
Sample: Cell CL090_230515

Raw



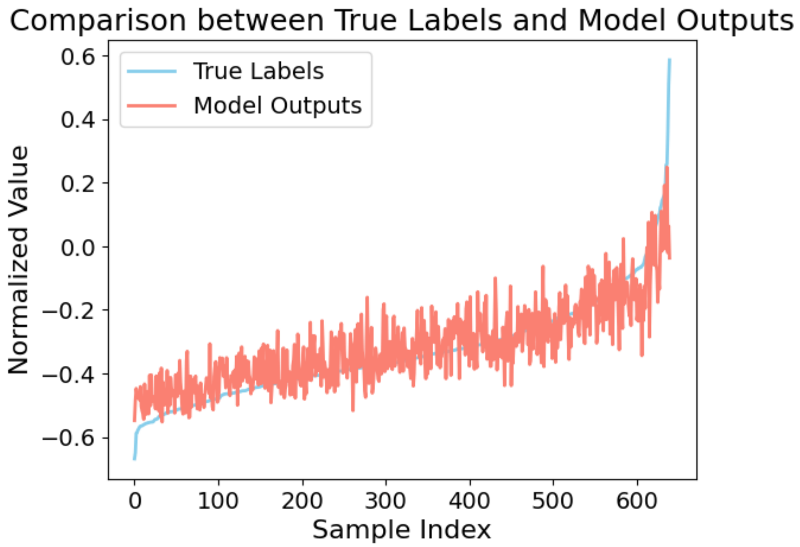
Loss = 0.0240

Mean restored



Loss = 0.0076

Mean and std restored



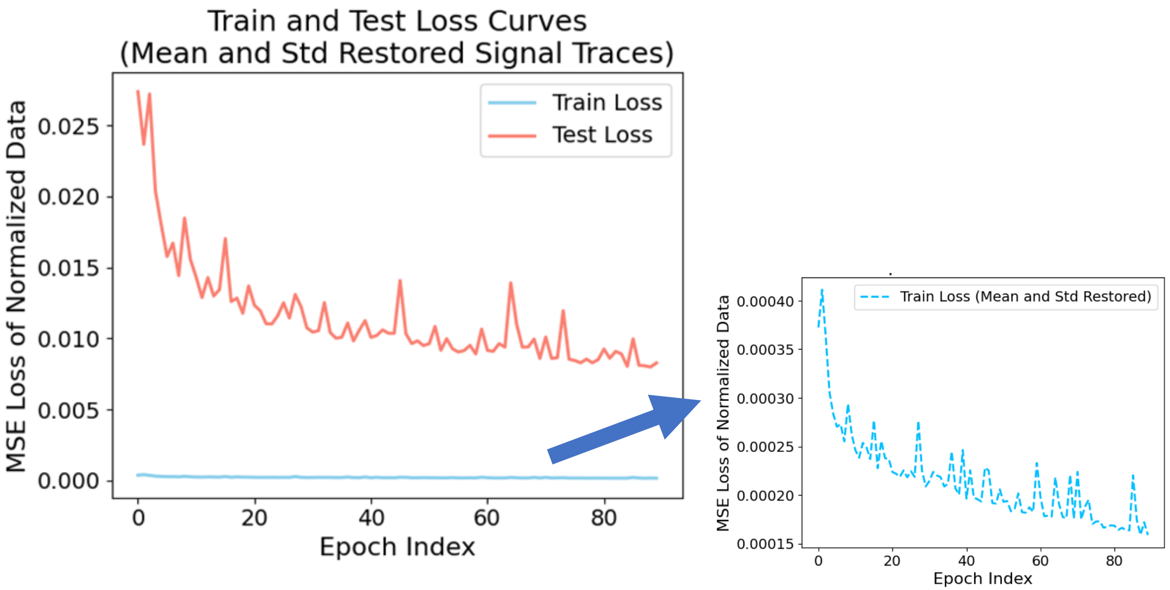
Loss = 0.0074

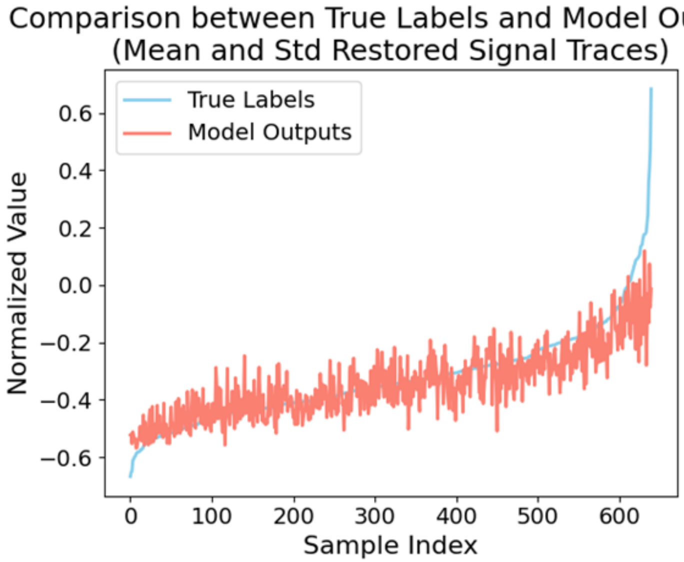
Last two similar, because we use the average of red as output. If we use the curve of red as output, I think this data would be better.

Biologically Interpretable Modeling

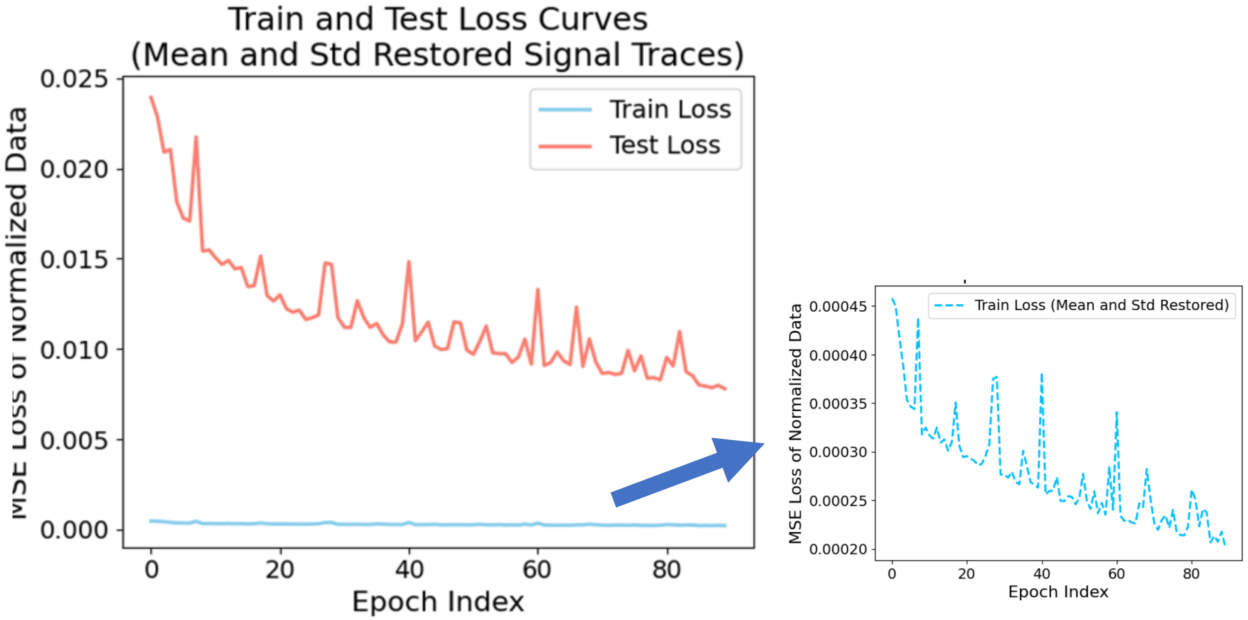
Sample: Cell CL090_230515

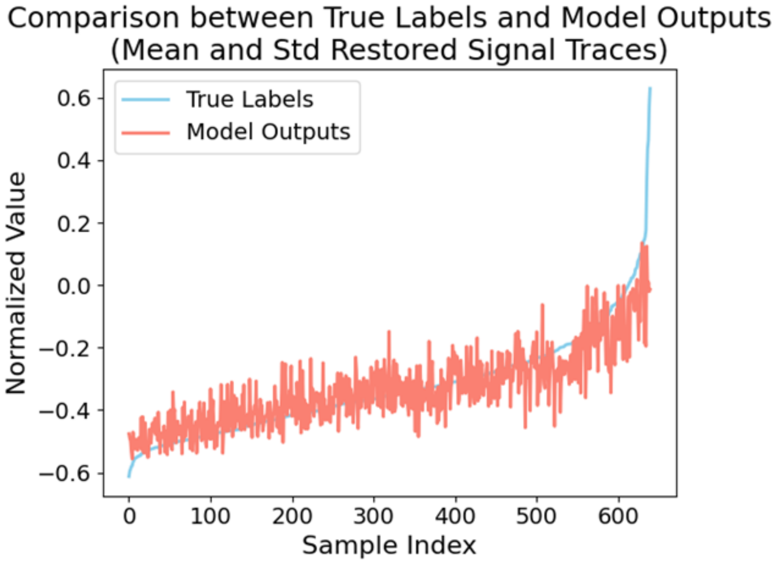
Use Runs 4, 5, 6





Use Runs 4, 5 (without Run 6)

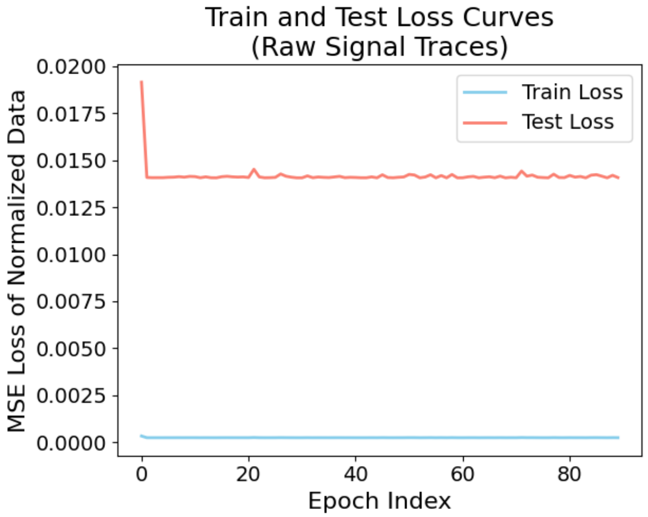


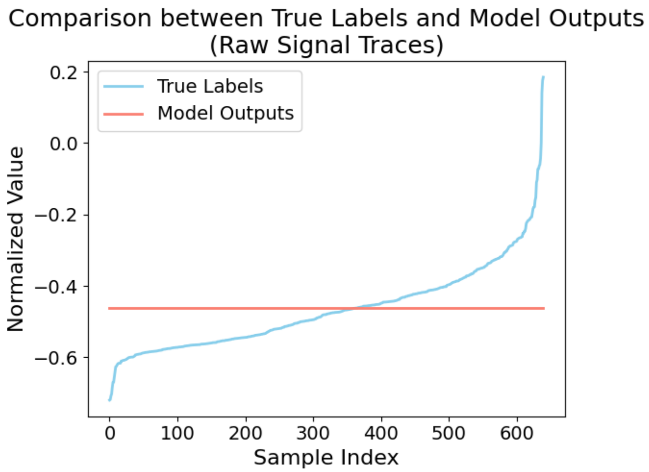


Biologically Interpretable Modeling

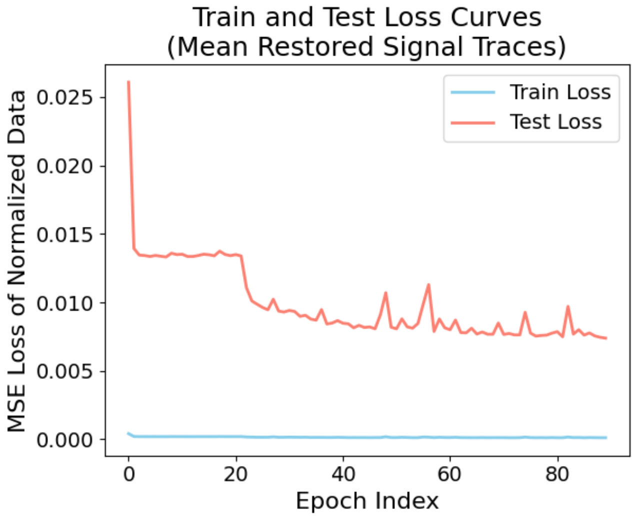
Sample: Cell CL075_230228

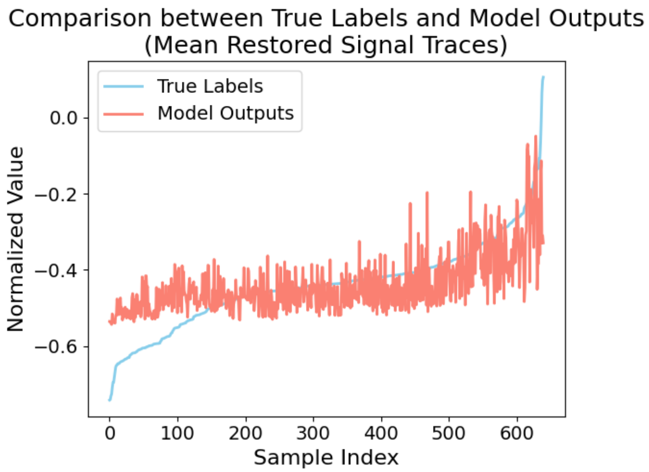
Raw



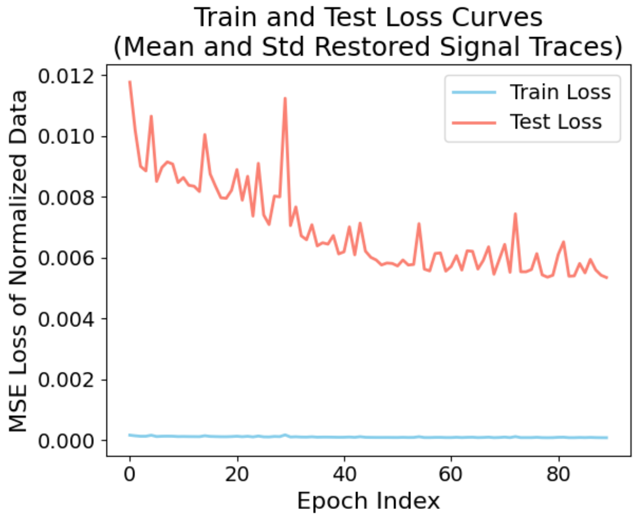


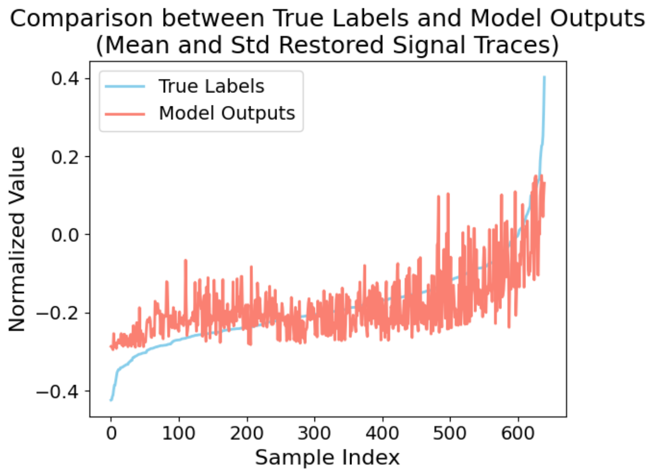
Mean restored





Mean and std restored



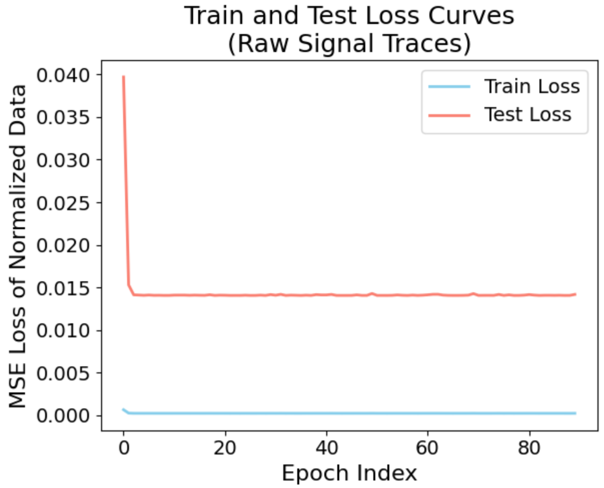


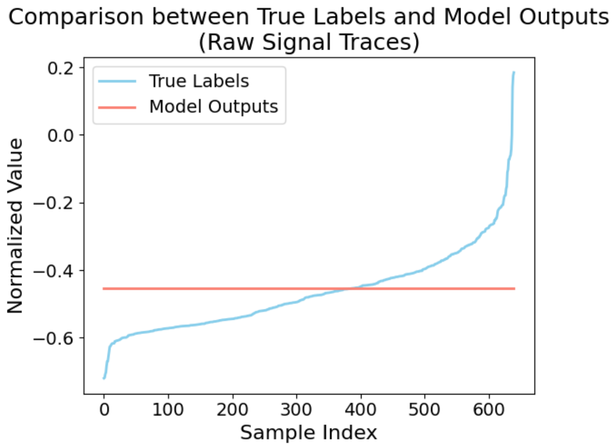
Biologically Interpretable Modeling

Sample: Cell CL075_230228

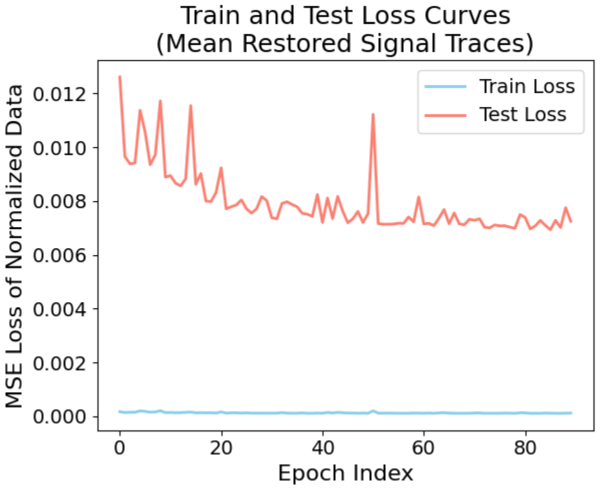
Try again to see whether it is robust

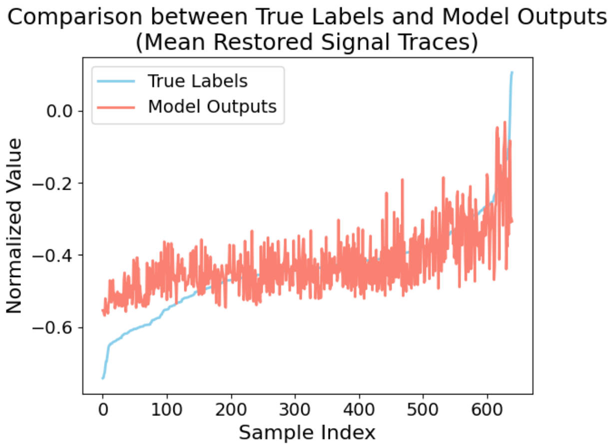
Raw



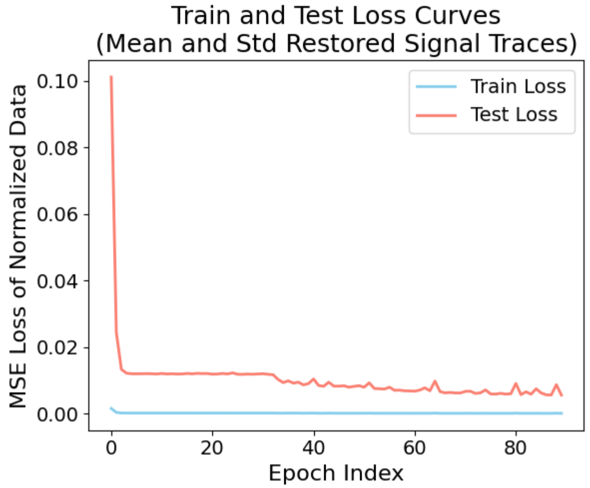


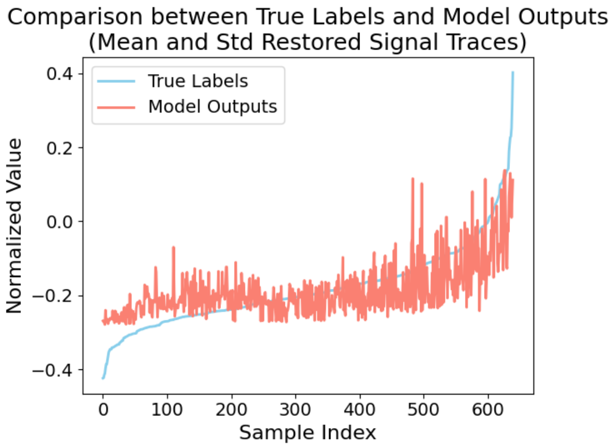
Mean restored





Mean and std restored





Biologically Interpretable Modeling

Sample: Cell CL075_230228

Raw

Epoch [1/90] - Train Loss: 0.000657 | Test Loss: 0.039699
Epoch [2/90] - Train Loss: 0.000257 | Test Loss: 0.015306
Epoch [3/90] - Train Loss: 0.000235 | Test Loss: 0.014138
Epoch [4/90] - Train Loss: 0.000234 | Test Loss: 0.014111
Epoch [5/90] - Train Loss: 0.000233 | Test Loss: 0.014081
Epoch [10/90] - Train Loss: 0.000233 | Test Loss: 0.014073
Epoch [20/90] - Train Loss: 0.000234 | Test Loss: 0.014098
Epoch [30/90] - Train Loss: 0.000236 | Test Loss: 0.014167
Epoch [40/90] - Train Loss: 0.000235 | Test Loss: 0.014126
Epoch [50/90] - Train Loss: 0.000238 | Test Loss: 0.014293
Epoch [60/90] - Train Loss: 0.000234 | Test Loss: 0.014096
Epoch [70/90] - Train Loss: 0.000238 | Test Loss: 0.014280
Epoch [80/90] - Train Loss: 0.000234 | Test Loss: 0.014099
Epoch [90/90] - Train Loss: 0.000236 | Test Loss: 0.014186

Mean restored

Epoch [1/90] - Train Loss: 0.000172 | Test Loss: 0.012611
Epoch [2/90] - Train Loss: 0.000142 | Test Loss: 0.009657
Epoch [3/90] - Train Loss: 0.000149 | Test Loss: 0.009384
Epoch [4/90] - Train Loss: 0.000152 | Test Loss: 0.009415
Epoch [5/90] - Train Loss: 0.000202 | Test Loss: 0.011374
Epoch [10/90] - Train Loss: 0.000137 | Test Loss: 0.008891
Epoch [20/90] - Train Loss: 0.000123 | Test Loss: 0.008319
Epoch [30/90] - Train Loss: 0.000117 | Test Loss: 0.008016
Epoch [40/90] - Train Loss: 0.000120 | Test Loss: 0.008241
Epoch [50/90] - Train Loss: 0.000113 | Test Loss: 0.007532
Epoch [60/90] - Train Loss: 0.000118 | Test Loss: 0.008152
Epoch [70/90] - Train Loss: 0.000114 | Test Loss: 0.007325
Epoch [80/90] - Train Loss: 0.000130 | Test Loss: 0.007498
Epoch [90/90] - Train Loss: 0.000122 | Test Loss: 0.007252

Mean and std restored

Epoch [1/90] - Train Loss: 0.001554 | Test Loss: 0.101179
Epoch [2/90] - Train Loss: 0.000364 | Test Loss: 0.024445
Epoch [3/90] - Train Loss: 0.000193 | Test Loss: 0.013309
Epoch [4/90] - Train Loss: 0.000176 | Test Loss: 0.012165
Epoch [5/90] - Train Loss: 0.000173 | Test Loss: 0.011983
Epoch [10/90] - Train Loss: 0.000172 | Test Loss: 0.011925
Epoch [20/90] - Train Loss: 0.000174 | Test Loss: 0.012054
Epoch [30/90] - Train Loss: 0.000172 | Test Loss: 0.011927
Epoch [40/90] - Train Loss: 0.000139 | Test Loss: 0.009035
Epoch [50/90] - Train Loss: 0.000129 | Test Loss: 0.008400
Epoch [60/90] - Train Loss: 0.000110 | Test Loss: 0.006820
Epoch [70/90] - Train Loss: 0.000101 | Test Loss: 0.006199
Epoch [80/90] - Train Loss: 0.000096 | Test Loss: 0.005987
Epoch [90/90] - Train Loss: 0.000088 | Test Loss: 0.005577

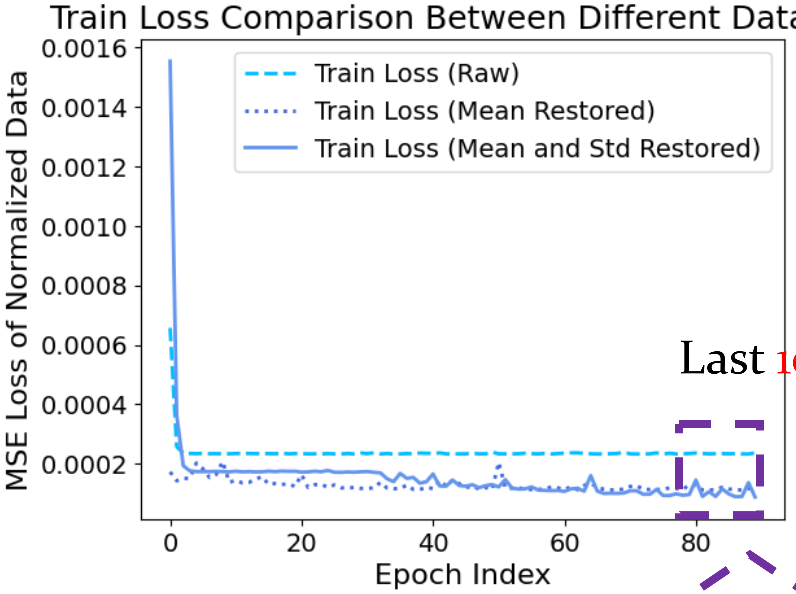
From left to right, train loss decreases – better fitting – higher data quality.

From left to right, test loss decreases – better fitting – higher data consistency.

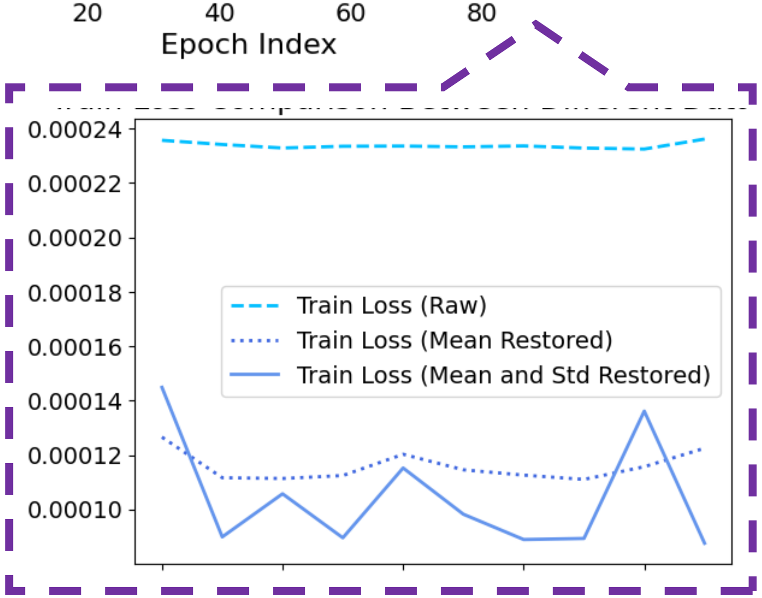
Biologically Interpretable Modeling

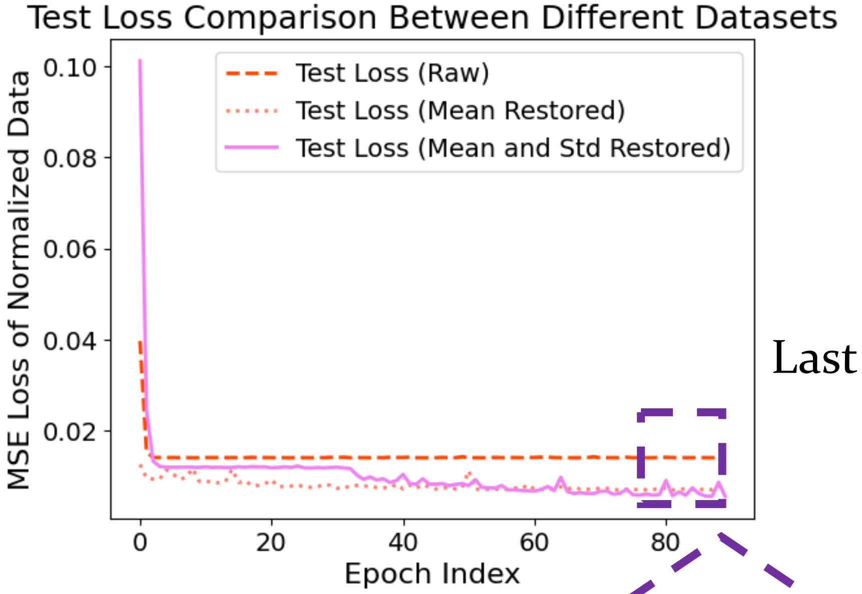
Sample: Cell CL075_230228

Compare train losses and compare test loss between different types of signal traces

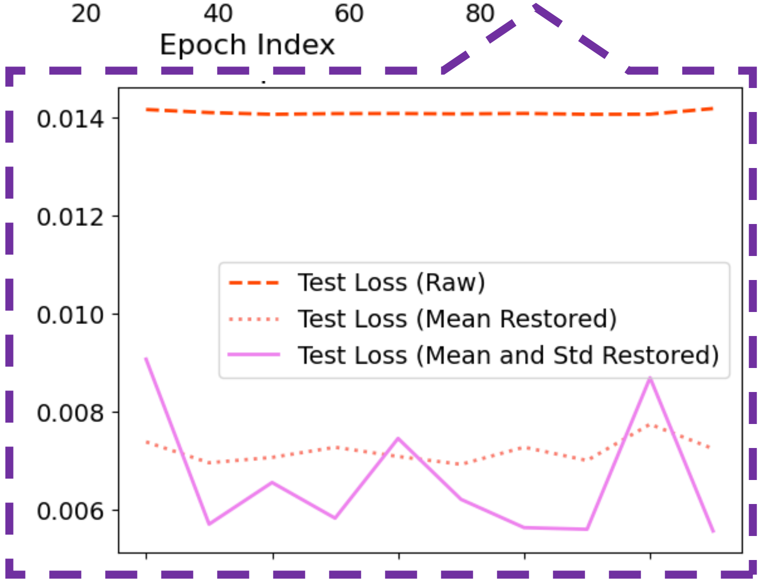


Last 10 epochs





Last 10 epochs

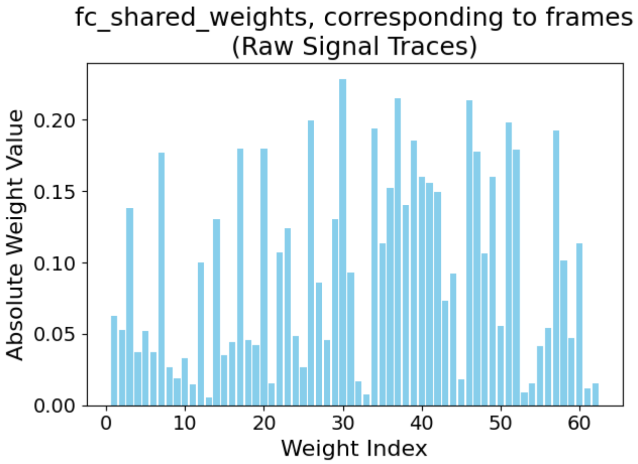


Biologically Interpretable Modeling

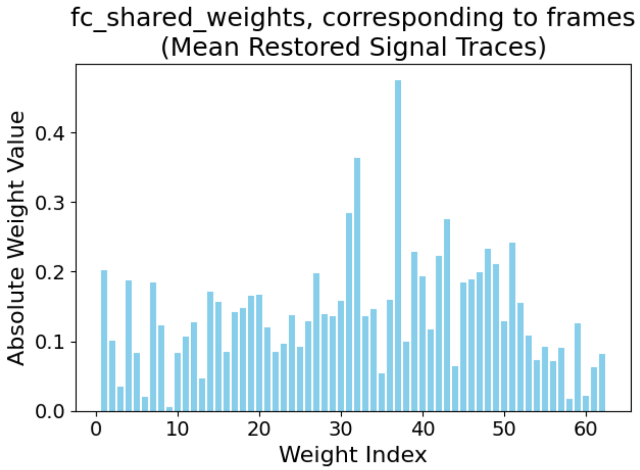
Sample: Cell CL075_230228

Comparing weights distribution, we can see the near time green data plays a more important role.

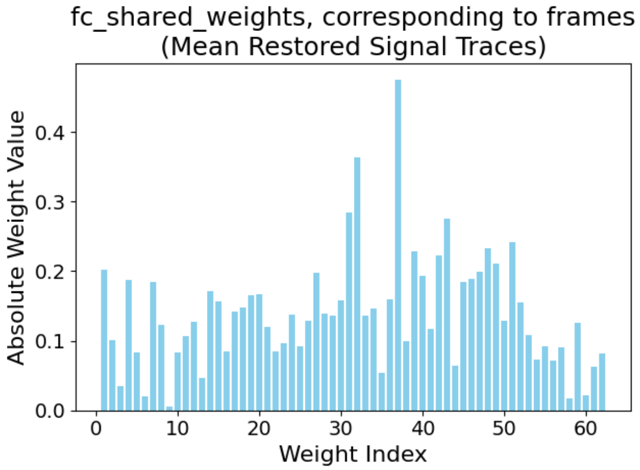
Raw



Mean restored



Mean and std restored



mean the first 31 weight absolute values of fc_shared_weights	0.081	0.126	0.103
mean the last 31 weight absolute values of fc_shared_weights	0.108 (+33.33%)	0.156 (+23.81%)	0.128 (+24.27%)

Biologically Interpretable Modeling

Basic Regression Models w/o Data Restoration

↓

Basic Regression Models w/ Basic Data Restoration

↓

Refining Decay Restoration w/ Full Raw Data

↓

Advanced Regression Models w/ Refined Decay Restoration

Conclusions:

1. Restoration provides us better quality data and using traces provides us more data, both contribute hugely to performance.

2. The model is with good performance and is to some extent biologically interpretable.

1. Background and Data Description

2. Basic Regression Modeling

3. Reevaluating Data and Refining Data Restoration

4. Biologically Interpretable Modeling

5. Limitations and Future Work

Limitations and Future Work

Limitations:

1. We only have excitatory inputs, don't know how other input affects the results.

2. We haven't done deconvolution of the fluorescence curves, probably resulting in some inaccuracy.

3. Whether most useful information has been extracted from 2p videos.

Future work:

1. Deeper analysis of the weights of the model.

2. Developing hierarchical model and checking whether structure information is important.

Thanks!