A tutorial on how to use the

Frequency-Independent Biological Signal Identification (FIBSI) Program

By: Max A. Odem (max.odem@uth.tmc.edu; max.neuro.odem@gmail.com) and Ryan M. Cassidy (rmcassidy@pm.me)

This is the program described in the publication: Cassidy RM, Bavencoffe AG, Lopez ER, Cheruvu SS, Walters ET, Uribe RA, Krachler AM, Odem MA. TITLE. *Journal* 2020 XX(XX): XXXX-XXXX. doi:XXXX

The FIBSI source code, readme file, and tutorial data are provided online at https://github.com/rmcassidy/FIBSI_program

Table of Contents

Getting started

Prerequisites

1. **FIBSI.py** – The core analysis program
2. **README.md** – A description of usage, flags, explanations, and background related to FIBSI
3. **NeuronTutorial_RAandNA.csv** – Raw membrane potential data collected from 3 types of rodent sensory neurons
4. **Batch for FIBSI_v1_0_1 Tutorial – RA and NA Neurons.bat** – Windows batch file with input commands
5. **ZfishTutorial_AB_Fed#10.csv** – Raw fluorescence data collected from a zebrafish larva fed with paramecia
6. **Batch for FIBSI_v1_0_1 Tutorial – Zfish AB_Fed#10.bat** – Windows batch file with input commands
7. Python and non-core Python v3 packages
    a. **Anaconda v2019.7.0.0 or higher** distributed by Anaconda, Inc., Austin, TX
    OR
    b. **Python v3.5.2 or higher** distributed by Python Software Foundation, Beaverton
    c. Python v3 packages **matplotlib** and **numpy**

Items 1-6 can be downloaded online at https://github.com/rmcassidy/FIBSI_program.

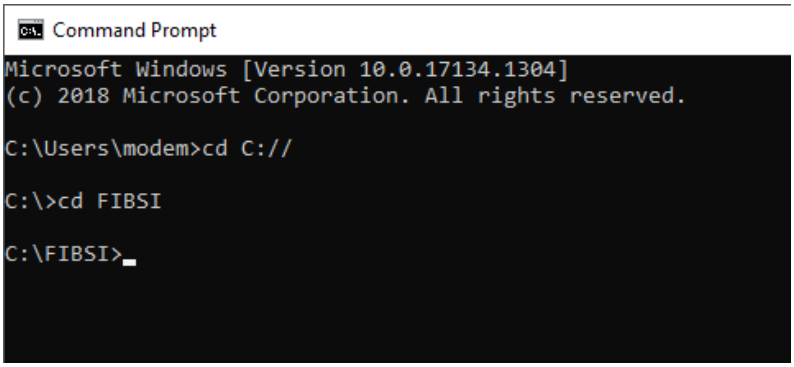
Choose whether to use item 7a or items 7b-c. Visit https://www.anaconda.com/ and https://www.python.org/ to download and install Anaconda and Python versions appropriate for your operating system. **Note:** If prompted during installation, include Python and/or Anaconda to your PATH variable.

Note: This tutorial was written and tested using a Windows 10 computer.

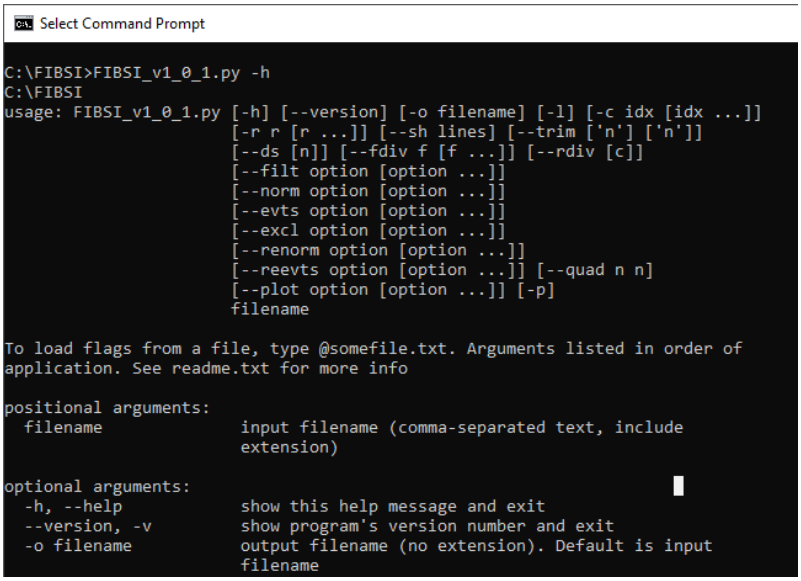
Installation

1. Create a new folder on a drive of your choosing (e.g., FIBSI folder on the C: drive)
2. Drop FIBSI.py, README.md, the tutorial .csv, and the tutorial .bat files into the new folder

3. Using the Command Prompt, change the working directory to the new folder



4. Test the FIBSI.py help message to ensure it is installed and invoked properly by typing "FIBSI_vX_X_X.py -h" and pressing enter, a usage description with available arguments (i.e., flags) should appear



Note: For Windows 10 users, PowerShell also works, but you may need to invoke Python by inserting "python" or "py" before "FIBSI_vX_X_X.py –h" in the command line.

Using FIBSI to analyze data collected from a time series

Explicit descriptions of the underlying processing and functions used by FIBSI will not be discussed in this tutorial. Only those functions necessary to replicate the analyses performed in our original publication (Cassidy et al., *JOURNAL*, YEAR) will be described. For examples of the processing and flags used to call various functions, background, and conceptual design, please carefully read through the README.md provided on the GitHub page.

Key terms and abbreviations

There are many key terms and abbreviations used in FIBSI, namely the flags and functions. It is recommended you refer to README.md for a full description. For quick reference, the few used in this tutorial include:

- fity – the reference Y series fit via a user-specific normalization
- rmd – running median, one of multiple normalizations that can be selected (e.g., running mean, least squares regression)
- dfy – the residual Y series ($Y - fity = dfy$)
- evts – detected events
- AUC – area under the curve
- DSF – depolarizing spontaneous fluctuation
- AP – action potential
- ROI – region of interest

Formatting raw data for input

FIBSI.py reads X-Y data series in a comma-separated format. The extension must be included with the filename (e.g., FILENAME.csv or FILENAME.txt) when inputting data in FIBSI. No spaces are permitted within the filename or output filename (to be designated by the user) unless the entire filename or output name are surrounded with quotation marks (e.g., “spaces are inconvenient.csv”).

Input file formatting notes:

1. The X and Y data must be found within distinct columns

2. Sampling interval
    a. Must be consistent for a given X-Y series
    b. Can differ between series if the series are input using separate commands (using either the same source file or separate source files)
    c. Must be consistent if inputting two or more series from the same source file using a single command
3. If multiple series are combined into a single source file, then the rows corresponding to series 1 must be labeled separately from the rows for series 2, and so on

	A	B	C	D
1	**Label**	**Y variable (e.g., fluoresence units)**	**X variable (time)**	
2	Cell1	4500.00	1.00	
3	Cell1	3500.00	2.00	
4	Cell1	3000.00	3.00	
5	Cell1	2900.00	4.00	
6	Cell1	2800.00	5.00	
7	Cell1	...	...	
8	Cell1	4600.00	1000.00	
9	Cell2	4900.00	1.00	
10	Cell2	4800.00	2.00	
11	Cell2	4600.00	3.00	
12	Cell2	3700.00	4.00	
13	Cell2	2000.00	5.00	
14	Cell2	...	...	
15	Cell2	3600.00	1000.00	
16	Cell3	2100.00	1.00	
17	Cell3	2500.00	2.00	
18	Cell3	2700.00	3.00	
19	Cell3	3000.00	4.00	
20	Cell3	3600.00	5.00	
21	Cell3	...	...	
22	Cell3	4900.00	1000.00	
23				

Formatting a command for input

Commands can be written using Command Prompt or run in batches using .bat files (can be edited like text files), or .sh files for Unix-like systems. Be mindful of proper syntax and spelling, as most errors will be due to inputting a command incorrectly and/or misspellings and mislabeling. **Note:** FIBSI uses a 0-index system, so column A in an Excel spreadsheet is column 0, B is column 1, and so on.

For this tutorial, commands will include the following components:

- Invoking FIBSI (required)
- Source filename for input (required)
- Filename for output (-o flag, required)

- Assigning the X and Y data columns (-c flag, required since default is -c 0 1)
- Assigning the series ID-containing column and series names to be analyzed (-r flag, required for this tutorial since the example files have multiple rows of named series per each value X)
- Assigning a normalization factor (--rdiv flag, optional)
- Assigning the fitting method (--norm flag, required for event analysis)
- Assigning the event detection method and cutoffs (--evts flag, required for event analysis)
- Assigning a post-hoc fitting method, event detection method, and cutoffs (--renorm and --reevts flags, all optional)
- Assigning how processed data are visualized using the interactive pyplot viewer (-p and --plot flags, all optional)

Example command input

FIBSI_v1_0_0.py filename.csv -o filename_results -c 15 1 -r 0 Cell1 Cell2 Cell3 Cell4 --rdiv Cell4 --norm rmd,50 --evts dfy xc,1 yc,0,0 --renorm above --reevts dfy xc,1 yc,0,0 -p --plot raw fity dfy evts save_csv save_png,200

The example command will do the following (flags written in order of processing and color coded):

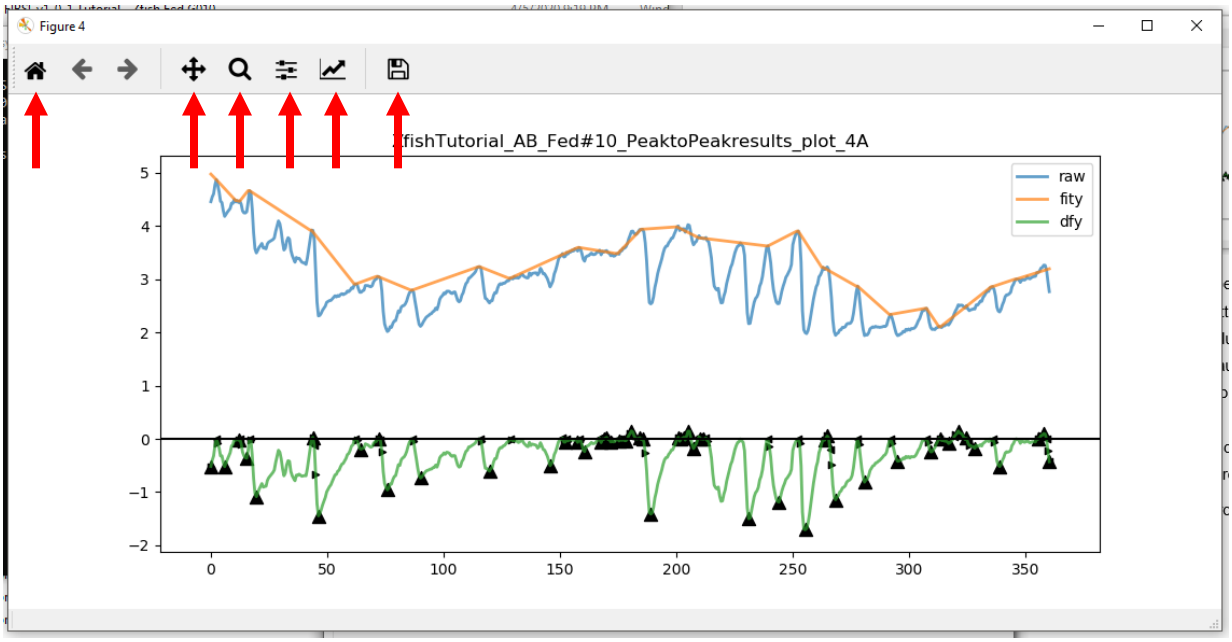
1. Invoke FIBSI_v1_0_0.py to analyze filename.csv and all results will be dumped into the root folder with the filename_results moniker
2. The X data are located in column 15 and Y data are in column 1 – give X, then Y
3. There are 4 series in column 0 with row IDs Cell1, Cell2, Cell3, and Cell4 – give column, then row IDs
4. Cells 1-3 will be divided by the Cell4 series – if dividing by a series, then all series must have the same sampling interval
5. The series for Cells 1-3 will be individually fitted and normalized using a running median with a window of 50 X units – this generates the fity line; window uses whatever units X data are collected in
6. Event detection is performed on the residual Y data with an X-cutoff of at least 1 unit and no Y-cutoffs in the negative or positive amplitude – this generates the dfy line; X and Y cutoffs use whatever units data are collected in

7. The dfy line is used to identify peaks (or troughs if “below” is invoked with --renorm flag) and the raw Y series is refitted using a peak-to-peak “above” renormalization – this generates new fity and dfy lines
8. Event detection is performed on the new dfy line with X- and Y-cutoffs
9. pyplot is invoked to plot the raw Y series, fity, and dfy with events labeled; All series data shown in pyplot will be saved as a .csv file in the order input and a .png image of the pyplot viewer will be saved at 200 dpi resolution

Note: In pyplot, the fity, dfy, and events will reflect the peak-to-peak renormalization because the --renorm flag was invoked. If --renorm was not invoked, then fity, dfy, and events would reflect the first normalization method (in this example, the running median).

Using the pyplot interactive viewer

FIBSI incorporates the pyplot interactive viewer from matplotlib for visualizing how the raw data are processed and for comparing the results of the event detection to the raw data. Basic functions (in order left to right, red arrows) include reset original view, pan axes, zoom to specified area, change subplot settings, edit axes and line parameters, and save to file. Events are displayed and bounded by isosceles triangles (black) on the dfy line. Rightward point triangles are event start points, leftward are event end points, and upward/downward are peaks/troughs.



Workflow examples using real biological data

Here we will demonstrate the workflow to analyze two different types of biological data. Tutorial data and commands for rodent sensory neurons and zebrafish are provided online on the GitHub page. We recommend walking through these examples to familiarize yourself with the workflow so you may begin analyzing your own data.

Measuring fluctuations in membrane potential in rodent sensory neurons

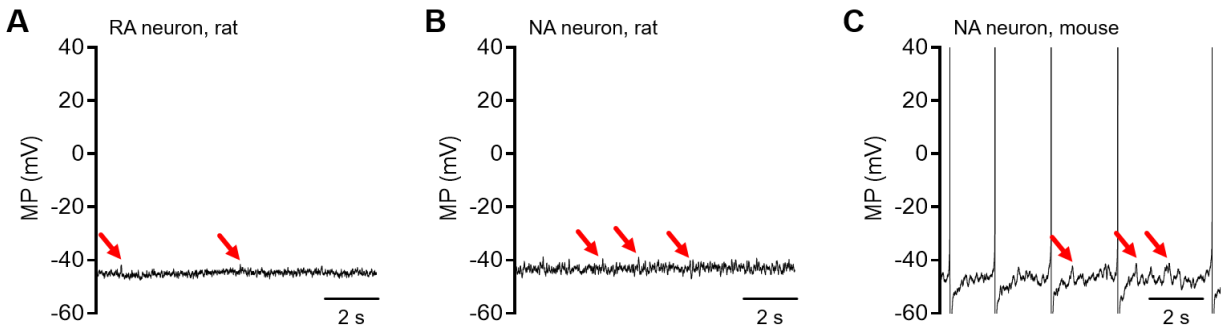
Background and Methodology

In Odem et al., *Pain*, 2018, we used a prototype version of the event-detection algorithm in FIBSI to analyze voltage fluctuations in small-diameter sensory neurons (putative nociceptors) dissociated from rat dorsal root ganglia. These depolarizing spontaneous fluctuations (DSFs) bridge the gap between resting membrane potential and the threshold for generating an action potential (AP). Under injury and inflammatory-like conditions when nociceptors are depolarized, DSFs increase in amplitude and frequency to drive increased firing activity and presumably pain (Yang et al., *J Neurosci*, 2014).

In Cassidy et al., *JOURNAL*, YEAR, we used a subset of sensory neurons dissociated from naïve rats and mice. All neurons were depolarized to a holding potential of -45 mV for 30 s to model pain-like conditions when nociceptors are more likely to be active and generate DSFs. The voltage recordings were extracted as X-Y data series for DSF analysis.

Whole-cell current-clamp recordings have a low signal-to-noise ratio, and irregular, low-amplitude deviations from a neuron's resting membrane potential can be difficult to quantify and distinguish from noise. To overcome this challenge, FIBSI uses a running median to model slow changes in the baseline membrane potential. Then, the event-detection algorithm identifies voltage fluctuations above and below the median that meet user-defined duration and amplitude thresholds. Under our testing conditions (e.g., equipment, extra- and intracellular solutions), the measured noise is ~0.5 mV when no cell is in circuit. An amplitude cutoff of 1.5 mV ensures measured fluctuations are not attributable to non-biological noise. Pilot analyses indicated a duration cutoff of 20 ms eliminated most other high-frequency noise in our recordings. These cutoffs may vary depending on your lab environment, electrophysiological equipment, and solutions used. For this tutorial, the same method will be used (previously reported in Odem et al., Pain, 2018) to process 3 example neurons (panels **A-C** in figure below, example DSFs marked

with red arrows) in **NeuronTutorial_RAandNA.csv** (commands written in the **Batch for FIBSI_v1_0_1 Tutorial – RA and NA Neurons.bat** file):

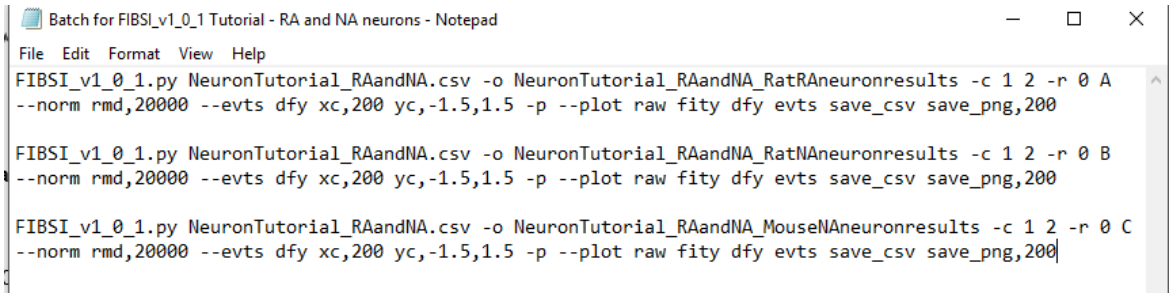


Processing the tutorial data

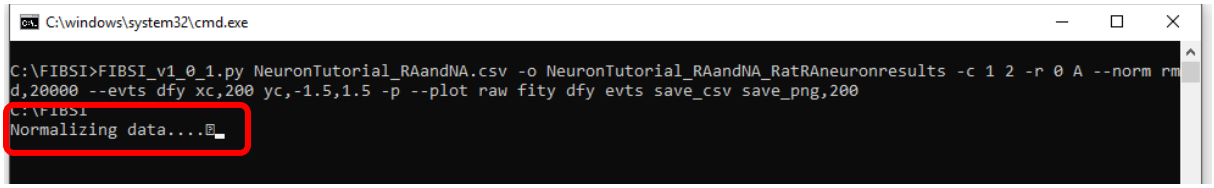
1. Drop the **NeuronTutorial_RAandNA.csv** and **Batch for FIBSI_v1_0_1 Tutorial – RA and NA Neurons.bat** files into your root folder
2. Inspect each file
    a. The .csv should have 4 columns with 30,000 data points for neurons A-C seen above

	A	B	C	D	E
1	Label	Time (s)	Potential	Potential (V)	
2	A	0	-42.56	-0.04256	
3	A	0.0001	-42.59	-0.04259	
4	A	0.0002	-42.53	-0.04253	
5	A	0.0003	-42.69	-0.04269	
6	A	0.0004	-42.69	-0.04269	
7	A	0.0005	-42.62	-0.04262	
8	A	0.0006	-42.66	-0.04266	
9	A	0.0007	-42.44	-0.04244	
10	A	0.0008	-42.72	-0.04272	
11	A	0.0009	-42.47	-0.04247	
899995	C	29.9993	-48.19	-0.04819	
899996	C	29.9994	-48.23	-0.04823	
899997	C	29.9995	-48.29	-0.04829	
899998	C	29.9996	-48.29	-0.04829	
899999	C	29.9997	-48.23	-0.04823	
900000	C	29.9998	-48.23	-0.04823	
900001	C	29.9999	-48.19	-0.04819	

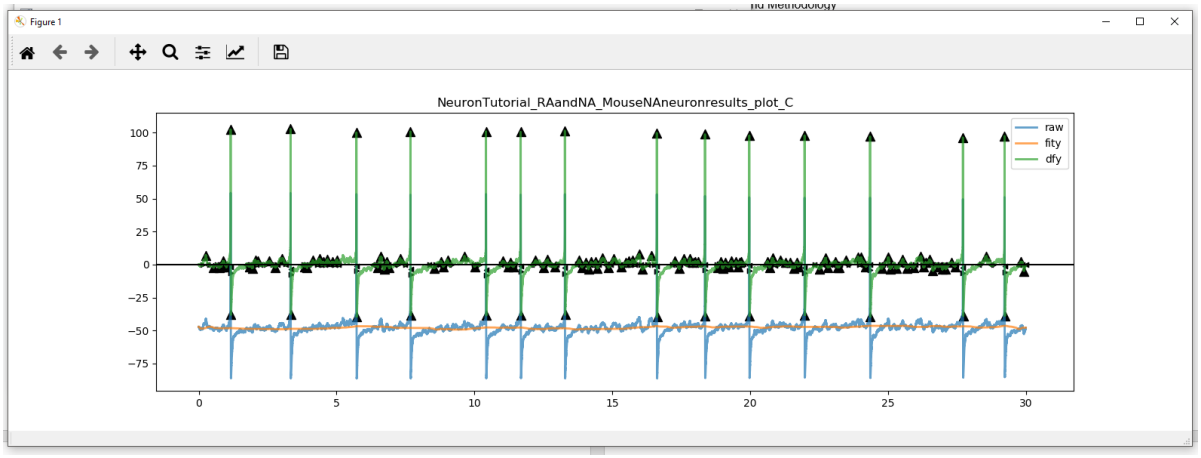
    b. The .bat file should have 3 commands, 1 for each neuron



3. Run FIBSI
    a. If running FIBSI using Command Prompt, change your directory to your root folder then type each command and hit enter; If running FIBSI using the .bat file, double click the .bat file in the root folder
    b. If run correctly, a real-time status will update as each command is processed



    c. The pyplot interactive viewer will open for each neuron in the order of commands processed; **Note:** in order to proceed, exit out of each pyplot viewer



4. Confirm output files were properly generated and dumped in your root folder
    a. 3 .log files, 1 for each neuron – tracks the settings used for analysis
    b. 3 .csv files, 1 for each neuron – contains the Y values for the raw, fity, and dfy lines
    c. 3 .png files, 1 for each neuron – captures the default view for each pyplot viewer
    d. 3 .txt files ending in _evts, 1 for each neuron – contains descriptive information for all detected events

e. 3 .txt files ending in _evts_formatted, 1 for each command – combines descriptive information for all detected events per command (defaults to events in 1 neuron if only processing 1 neuron per command)

Note: We recommend saving all files for your records, but only the .txt files in (e) will be used for post-processing analyses.

Analyzing the processed tutorial data

The evts_formatted.txt files contain all of the descriptive information necessary for analysis. Most of the columns are self-explanatory (e.g., event start time, end time, duration, amplitude). Column A is called “ROI” by default, but it corresponds to the label IDs used during command input (neurons A-C in this example). Column B “Quadrant” can be ignored when processing sensory neurons, it will be addressed in the next example using zebrafish. Column C “Direction” refers to whether the event on the dfy line points above “a” zero or below “b” zero. Therefore, “a” labels the DSFs and “b” labels the hyperpolarizing fluctuations (not the focus of this tutorial, but for more information please refer to Odem et al., Pain, 2018).

	A	B	C	D	E	F	G	H	I	J	K	L
1											NeuronTutorial_RA	
2	ROI	Quadrant	Direction	Start time	End time (	Peak time	Duration (	Amplitud	Midpoint	Half Durat	AUC	
3	C	4	a	0.1491	0.3161	0.2596	0.167	7.01	0.2326	0.0835	5668.12	
4	C	2	b	0.4015	0.6285	0.5387	0.227	-2.44	0.515	0.1135	-2251.76	
5	C	2	b	0.6808	0.7787	0.747	0.0979	-2.47	0.7297	0.04895	-1330.01	
6	C	2	b	0.7788	0.836	0.822	0.0572	-1.5	0.8074	0.0286	-389.915	
7	C	4	a	0.8445	0.9498	0.8887	0.1053	2.76	0.8971	0.05265	1620.025	
8	C	2	b	0.9797	1.0284	1.0053	0.0487	-2	1.004	0.02435	-437.04	
9	C	4	a	1.0309	1.1627	1.1585	0.1318	102.49	1.0968	0.0659	8919.26	
10	C	2	b	1.1628	1.7093	1.1679	0.5465	-38.03	1.436	0.27325	-32491.9	
11	C	2	b	1.7931	1.8613	1.8437	0.0682	-1.66	1.8272	0.0341	-581.045	
12	C	2	b	1.9091	1.9724	1.9492	0.0633	-2.75	1.9407	0.03165	-1080.45	
13	C	4	a	1.9729	2.0923	2.0492	0.1194	3.48	2.0326	0.0597	2405.72	
14	C	4	a	2.1039	2.23	2.1532	0.1261	2.76	2.1669	0.06305	1645.995	
15	C	4	a	2.4438	2.6363	2.5607	0.1925	2.85	2.54	0.09625	2667.37	

NeuronTutorial_RAandNA_MouseNAn

Analysis of the sensory neuron fluctuation data is straight forward in comparison to analyzing gut motility in zebrafish (next example in this tutorial). Users can extract the DSF properties of interest (e.g., amplitudes, durations, AUCs) and/or calculate additional information (e.g., frequency of DSFs ≥3 mV), and then use these values to make comparisons between neurons and experimental conditions.

Notes regarding suprathreshold DSFs: The list of DSFs contains both sub- and suprathreshold DSFs (i.e., those triggering an AP, amplitudes ~90-100 mV). Once AP threshold is reached, the

AP waveform occludes the underlying trigger DSF, and so the information in columns E-K cannot be used to describe the DSF itself. Instead, E-K describes the total DSF+AP waveform to the running median reference line at y = 0. We recommend users manually differentiate between sub- and suprathreshold DSFs during analysis, and if users want to conservatively estimate the amplitude of a given suprathreshold DSF then substitution is necessary (see methods in Odem et al., Pain, 2018). Briefly, calculate the difference in mV between the neuron's AP threshold and the membrane potential corresponding to the start time of the DSF+AP waveform. This can be done by cross referencing the event start time with the membrane potential in the .csv file ending in _plot_ID. For example, the first DSF+AP waveform to occur in neuron C in this tutorial begins at 1.0309 s ("a" event with amplitude of 102.49 mV). In the .csv file ending in _plot_C, time point 1.0309 is located at row #10311 and it corresponds to a membrane potential of -48.16 mV. The AP threshold for neuron C is -39.11 mV, so the substituted DSF amplitude is:

$$(-39.11\ mV) - (-48.16\ mV) = \ 9.05\ mV$$

Not all DSF+AP waveforms are substituted, we do not want the analysis to be tautological (i.e., recounting APs under a new name). Some neurons fire APs in rapid bursts and occlude underlying DSFs. It is unclear in these circumstances whether a single DSF triggers a burst of APs and/or multiple triggering DSFs occur in series but cannot be seen. It is also unclear how the afterhyperpolariztion potential component of the AP and residual ionic conductances influence DSFs. To account for these uncertainties, we apply substitutions according to the following conservative criteria:

Following the first AP in a burst, does ≥1 clearly distinct DSFs and/or hyperpolarizing fluctuations occur before the next AP? (This would suggest the next DSF+AP waveform is unique)

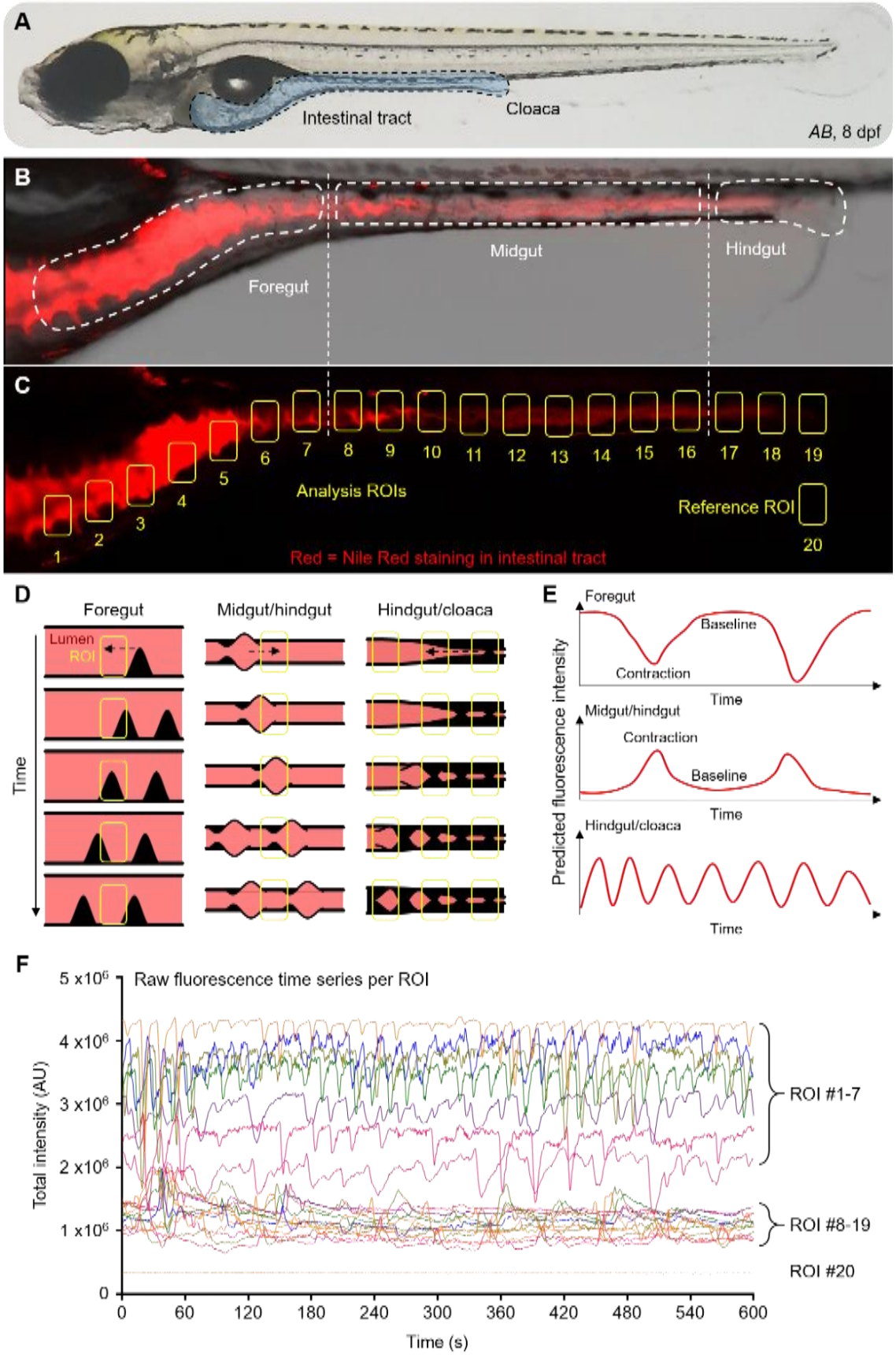
- If yes, calculate the substitution amplitude for the next DSF+AP waveform
- If no, has 200 ms elapsed between the end of the preceding afterhyperpolarization potential and the next DSF+AP waveform? (A duration of 200 ms is expected to be a sufficient amount of time for the residual ionic conductances of the afterhyperpolarization to end)
    - If yes, calculate the substitution
    - If no, do not calculate the substitution

Measuring gut motility in zebrafish using fluorescence contrast imaging

Background and Methodology

We adapted FIBSI to analyze fluorescence contrast-based intensity measures of gut motility in zebrafish to characterize peristalsis contractions. In Cassidy et al., *JOURNAL*, YEAR, all zebrafish larvae were imaged using 19 ROIs positioned along the gut and 1 reference ROI (panel **A-C** in figure below). The total fluorescence intensity was extracted for each ROI as an X-Y series. The total fluorescent intensity of the Nile Red dye either increases or decreases depending on how the gut moves during muscle contractions (panel **D-F** in figure below). In Cassidy et al. we described 3 different kinds of contraction events:

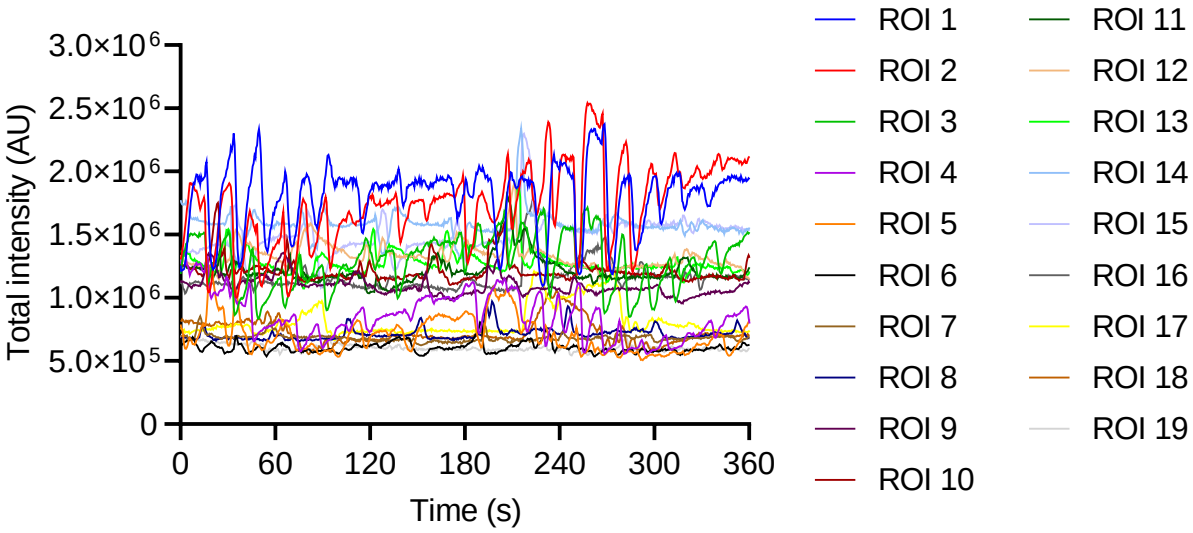
1. Narrowing of the gut
    a. Decrease in fluorescence intensity
    b. Predominantly observed in the foregut ROIs (ROIs #1-7 in the below example)
2. Widening of the gut
    a. Increase in fluorescence intensity
    b. Predominantly observed in the midgut/hindgut ROIs, sometimes observed in distal foregut ROIs (ROIs #8-19)
3. Undulations in the cloaca
    a. Generates an oscillation in the fluorescence intensity
    b. Observed in the hindgut ROIs (ROIs #17-19)


A
Intestinal tract
Cloaca
AB, 8 dpf
B
Foregut
Midgut
Hindgut
C
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Analysis ROIs
Reference ROI
Red = Nile Red staining in intestinal tract
D
Foregut
Midgut/hindgut
Hindgut/cloaca
Lumen ROI
Time
E
Foregut
Baseline
Contraction
Time
Midgut/hindgut
Contraction
Baseline
Time
Hindgut/cloaca
Time
Predicted fluorescence intensity
F
Raw fluorescence time series per ROI
5 x10⁶
4 x10⁶
3 x10⁶
2 x10⁶
1 x10⁶
0
Total intensity (AU)
0
60
120
180
240
300
360
420
480
540
600
Time (s)
ROI #1-7
ROI #8-19
ROI #20

Therefore, the type of movement influences whether the contraction event of interest is detected on the peak or trough of the change in fluorescence intensity. In order to detect the appropriate event start, peak, and end times, the renormalization method used should oppose the directional change in fluorescence intensity (i.e., use the --renorm above method for contractions that decrease fluorescence, and vice versa). Furthermore, the running median window size influences peak and trough detection, and thereby influences event detection accuracy. For optimal accuracy, longer window sizes are better for maintaining a stable baseline during large amplitude contractions (ROIs #1-7 in the above example) while smaller window sizes are better for a more fluid baseline in the midgut/hindgut (ROIs #8-19 in the above example). Choosing the best-fit method and optimal window size requires careful observation of videos synced with the fluorescence time series data as well some trial and error to determine an optimal fit for the data.

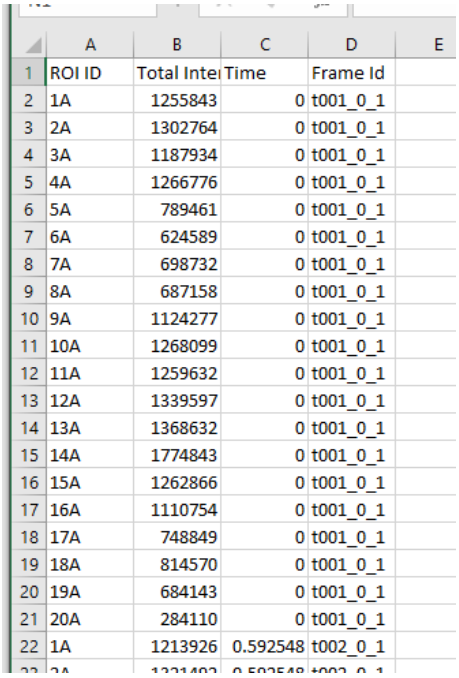
For this tutorial, the following methods will be used to analyze **ZfishTutorial_AB_Fed#10.csv** (commands written in the **Batch for FIBSI_v1_0_1 Tutorial – Zfish AB_Fed#10.bat** file, raw data shown below):

1. Peak-to-peak “above” analysis in ROIs #1-7; window size = 50 X units
2. Trough-to-trough “below” analysis in ROIs #8-16; window size = 10 X units
3. Peak-to-peak “above” analysis for the fish cloaca in ROIs #17-19; window size = 10 X units



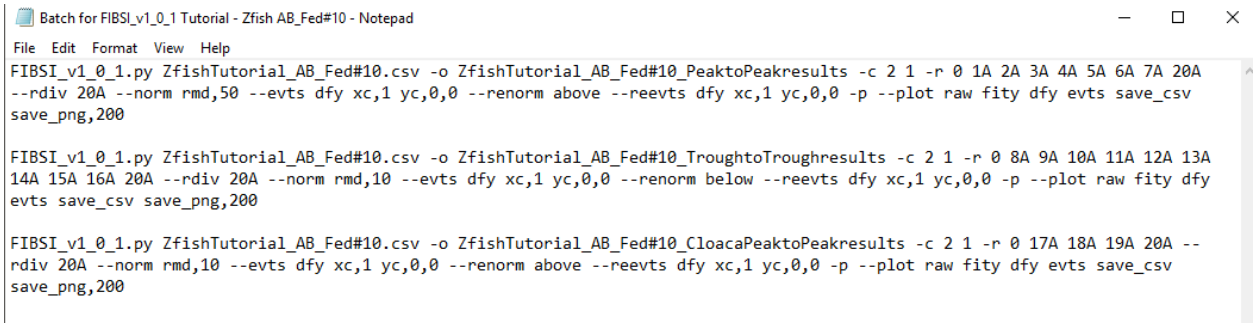
Processing the tutorial data

1. Drop the **ZfishTutorial_AB_Fed#10.csv** and **Batch for FIBSI_v1_0_1 Tutorial – Zfish AB_Fed#10.bat** files into your root folder
2. Inspect each file
    a. The .csv should have 4 columns, with 609 frames for ROIs 1A-20A



	A	B	C	D	E
1	ROI ID	Total Inte	Time	Frame Id	
2	1A	1255843	0	t001_0_1	
3	2A	1302764	0	t001_0_1	
4	3A	1187934	0	t001_0_1	
5	4A	1266776	0	t001_0_1	
6	5A	789461	0	t001_0_1	
7	6A	624589	0	t001_0_1	
8	7A	698732	0	t001_0_1	
9	8A	687158	0	t001_0_1	
10	9A	1124277	0	t001_0_1	
11	10A	1268099	0	t001_0_1	
12	11A	1259632	0	t001_0_1	
13	12A	1339597	0	t001_0_1	
14	13A	1368632	0	t001_0_1	
15	14A	1774843	0	t001_0_1	
16	15A	1262866	0	t001_0_1	
17	16A	1110754	0	t001_0_1	
18	17A	748849	0	t001_0_1	
19	18A	814570	0	t001_0_1	
20	19A	684143	0	t001_0_1	
21	20A	284110	0	t001_0_1	
22	1A	1213926	0.592548	t002_0_1	

    b. The .bat file should have 3 commands

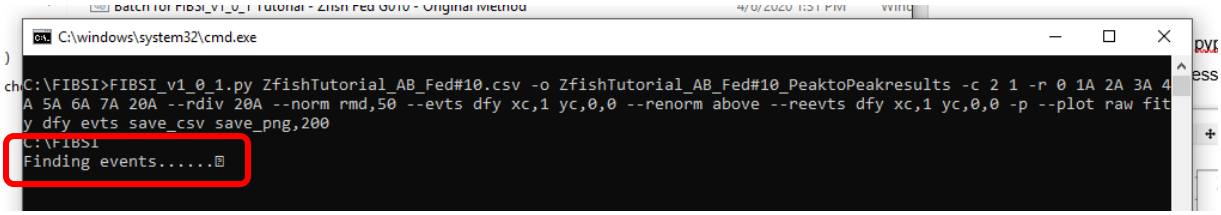


Batch for FIBSI_v1_0_1 Tutorial - Zfish AB_Fed#10 - Notepad

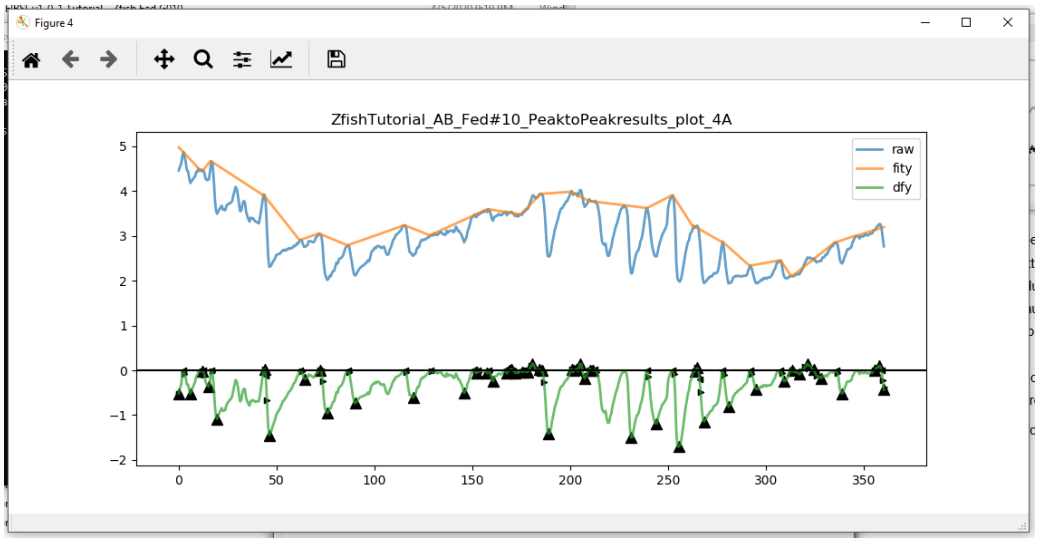
File Edit Format View Help

```
FIBSI_v1_0_1.py ZfishTutorial_AB_Fed#10.csv -o ZfishTutorial_AB_Fed#10_PeaktoPeakresults -c 2 1 -r 0 1A 2A 3A 4A 5A 6A 7A 20A
--rdiv 20A --norm rmd,50 --evts dfy xc,1 yc,0,0 --renorm above --reevts dfy xc,1 yc,0,0 -p --plot raw fity dfy evts save_csv
save_png,200

FIBSI_v1_0_1.py ZfishTutorial_AB_Fed#10.csv -o ZfishTutorial_AB_Fed#10_TroughtoTroughresults -c 2 1 -r 0 8A 9A 10A 11A 12A 13A
14A 15A 16A 20A --rdiv 20A --norm rmd,10 --evts dfy xc,1 yc,0,0 --renorm below --reevts dfy xc,1 yc,0,0 -p --plot raw fity dfy
evts save_csv save_png,200

FIBSI_v1_0_1.py ZfishTutorial_AB_Fed#10.csv -o ZfishTutorial_AB_Fed#10_CloacaPeaktoPeakresults -c 2 1 -r 0 17A 18A 19A 20A --
rdiv 20A --norm rmd,10 --evts dfy xc,1 yc,0,0 --renorm above --reevts dfy xc,1 yc,0,0 -p --plot raw fity dfy evts save_csv
save_png,200
```

3. Run FIBSI
    a. If running FIBSI using Command Prompt, change your directory to your root folder then type each command and hit enter; If running FIBSI using the .bat file, double click the .bat file in the root folder
    b. If run correctly, a real-time status will update as each command is processed



c. The pyplot interactive viewer will open for each ROI in the order of commands processed; **Note:** in order to proceed, exit out of each pyplot viewer



4. Confirm output files were properly generated and dumped in your root folder
    a. 3 .log files, 1 for each command – tracks the settings used for analysis
    b. 19 .csv files, 1 for each ROI – contains the Y values for the raw, fity, and dfy lines
    c. 19 .png files, 1 for each ROI – captures the default view for each pyplot viewer
    d. 19 .txt files ending in _evts, 1 for each ROI – contains descriptive information for all detected events
    e. 3 .txt files ending in _evts_formatted, 1 for each command – combines descriptive information for all detected events in the ROIs processed per command

    Note: We recommend saving all files for your records, but only the .txt files in (e) will be used for post-processing analyses.

Analyzing the processed tutorial data

Column B “Quadrant” will be used in this example and will be addressed shortly. Column K “AUC” is provided by default, but we recommend to not use it in combination with our fluorescence

contrast assay; controls for the amount of Nile Red ingested by each zebrafish, and therefore amount of fluorescence present in the gut, are limited.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1											ZfishTutorial_AB_Fed#10_P		
2	ROI	Quadrant	Direction	Start time	End time (	Peak time	Duration (	Amplitud	Midpoint	Half Durat	AUC		
3	1A	0	a	0	0.592548	0	0.592548	0	0	0.296274	0		
4	1A	2	b	1.185096	8.295672	3.555288	7.110576	-0.81834	4.740384	3.555288	-5.78214		
5	1A	2	b	8.88822	9.480768	8.88822	0.592548	-0.05216	8.88822	0.296274	0		
6	1A	4	a	10.07332	11.25841	10.07332	1.185096	0.133388	10.66586	0.592548	0.125051		
7	1A	2	b	11.85096	15.9988	12.44351	4.147836	-0.29043	13.6286	2.073918	-1.25891		
8	1A	4	a	16.59134	17.18389	16.59134	0.592548	0.101864	16.59134	0.296274	0		
9	1A	2	b	17.77644	33.77524	21.33173	15.9988	-3.15347	25.47956	7.999398	-43.4915		
10	1A	2	b	34.36778	49.77403	37.33052	15.40625	-3.7753	42.07091	7.703124	-51.2029		
11	1A	2	b	50.36658	66.36538	55.10696	15.9988	-3.04412	58.0697	7.999398	-43.732		
12	1A	2	b	66.95792	80.58653	70.51321	13.6286	-2.06011	73.47595	6.814302	-22.7541		
13	1A	2	b	81.17908	93.03004	86.51201	11.85096	-1.7036	87.10456	5.92548	-19.4433		
14	1A	4	a	93.62258	94.80768	94.21513	1.185096	0.048224	94.21513	0.592548	0.03808		
15	1A	2	b	95.40023	102.5108	97.77042	7.110576	-0.57129	98.95552	3.555288	-3.00515		

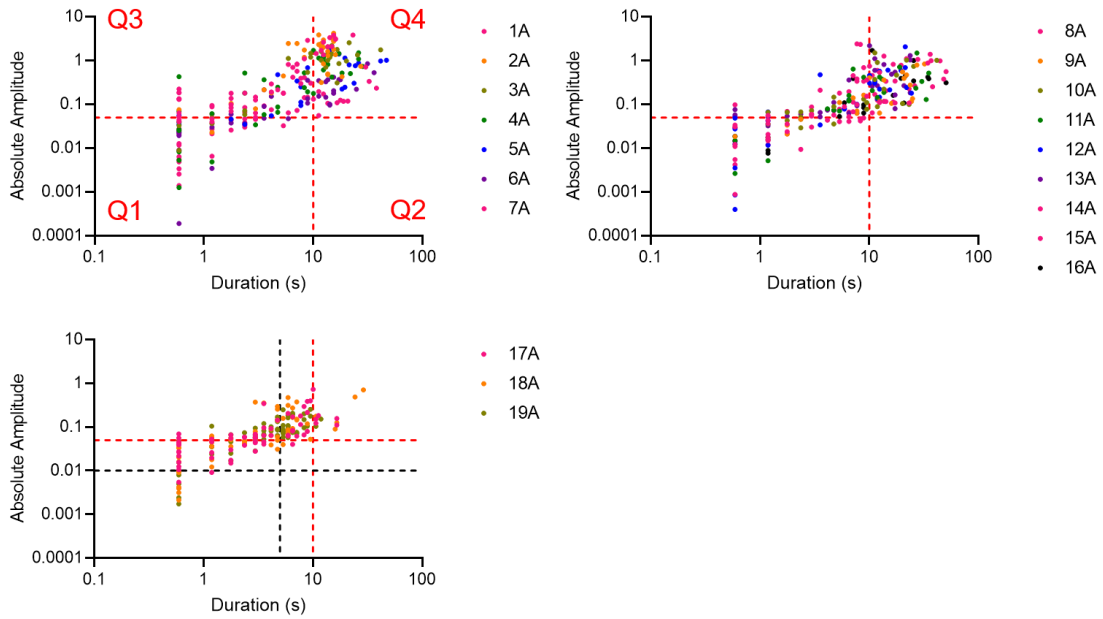
When analyzing gut motility data in zebrafish, the goal is to generate a raster plot to visualize contraction rhythmicity, assess FIBSI's accuracy, and to calculate the contraction frequency in each ROI. The frequencies can then be used to calculate the mean frequencies within the foregut, midgut, and hindgut regions, the whole gut, and for comparisons between zebrafish and experimental conditions.

1. Isolate low- and high-frequency contractions from noise
    a. Generate a XY scatter plot of all event durations and amplitudes in each .txt file
        i. For peak-to-peak renormalization
            1. Extract the "b" events, ignore the "a" events
            2. Transform amplitudes to the absolute values
        ii. For trough-to-trough renormalization
            1. Extract the "a" events, ignore the "b" events
            2. Transformation is unnecessary
    b. Set cutoffs
        i. Low-frequency peristalsis contractions passing through ROIs #1-19
            1. Amplitude ≥ 0.05
            2. Duration ≥ 10 s
            3. Separates data into quadrants (**Q1-Q4**)

            Note: If the X- and Y-cutoffs are set during command input using the --evts flag, then the "Quadrant" column will reflect the quadrants manually set here
        ii. High-frequency undulations in/near the cloaca at ROIs #17-19

1. Amplitude ≥ 0.01
2. Duration 5 s ≤ x ≤ 10 s

Note: **Q1-Q4** still apply for isolating low-frequency contractions in ROIs #17-19 before isolating high-frequency undulations (dotted black cutoffs) specific to the cloaca



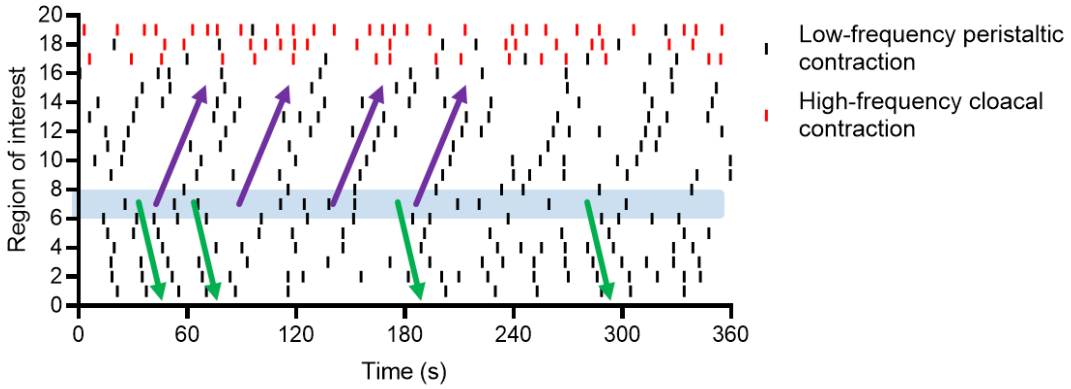
2. Generate a raster plot
    a. The raster plot will depict contraction waves passing through discrete ROIs as a function of time
        i. Low-frequency peristaltic contractions are found in **Q4** of ROIs #1-19
        ii. High-frequency cloacal contractions are found within the narrow duration window set for ROIs #17-19
    b. The raster plot will shift depending on which information is used (e.g., start, peak, or end time), it is recommended to plot the **peak times**
    c. To generate the raster, plot the ROI #s along the Y axis and the peak times along the X axis
3. Qualitatively assess the raster plot's features to ensure FIBSI's event detection accurately reflects the raw fluorescence and videos
    a. Contraction rhythmicity (i.e., waves of tick marks along the X-axis) should be visibly apparent

b. Peristalsis contractions originate from a site of pacemaker-like activity near the end of the foregut (ROIs #6-8, **blue highlight**)

c. Contractions travel in two directions from the origin

   i. Retrograde contractions (**green arrows**)

   ii. Anterograde contractions (**purple arrows**)



4. Calculate the contraction frequency (in Hz) for each ROI

   a. For peristalsis, divide the number of low-frequency contraction events in ROIs #1-19 by the duration of the recording (360 s)

   b. For the cloaca, divide the number of high-frequency contraction events in ROIs #17-19 by the duration of the recording (360 s)

   c. Results for **ZfishTutorial_AB_Fed#10.csv**

	A	B	C
1	ROI	# of low-freq events	Peristaltic contraction freq (Hz)
2	1	12	0.033
3	2	19	0.053
4	3	17	0.047
5	4	15	0.042
6	5	12	0.033
7	6	15	0.042
8	7	11	0.031
9	8	8	0.022
10	9	12	0.033
11	10	12	0.033
12	11	11	0.031
13	12	15	0.042
14	13	14	0.039
15	14	13	0.036
16	15	10	0.028
17	16	10	0.028
18	17	6	0.017
19	18	5	0.014
20	19	2	0.006
21			
22	ROI	# of high-freq events	Cloacal contraction freq (Hz)
23	17	16	0.044
24	18	16	0.044
25	19	27	0.075
26			

Final notes