Last Updated May 2021

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 first described in the *bioRxiv* preprint:

"Frequency-independent biological signal identification (FIBSI): A free program that simplifies intensive analysis of non-stationary time series data", by Cassidy RM, Bavencoffe AG, Lopez ER, Cheruvu SS, Walters ET, Uribe RA, Krachler AM, Odem MA. bioRxiv 2020.05.29.123042; doi: https://doi.org/10.1101/2020.05.29.123042

An earlier version of this tutorial that corresponds to the *bioRxiv* preprint can be found online:

https://github.com/rmcassidy/FIBSI_program/tree/master/bioRxiv%20June%202020

The FIBSI source code, readme file, and tutorial data are provided online:

https://github.com/rmcassidy/FIBSI_program

Table of Contents

Getting started

Prerequisites

1. The core analysis program
    a. **FIBSI_v1_0_1.py**
    b. **FIBSI_v1_0_1_calcium.py**
        i. Line 557 in item 1a changed from “if n>=0:” to “if n>0:”; this modification keeps the event detection rules consistent between the two methods to re-normalize to the signal peaks versus the signal troughs, respectively
        ii. Originally used to measure calcium $[Ca^{2+}]_i$ fluctuations in cultured spinal neurons (Alles et al., *Sci Rep*, 2021; doi: https://doi.org/10.1038/s41598-021-81269-6)
2. **README.md** – A description of usage, flags, explanations, and background related to FIBSI
3. **ZfishTutorial_AB_Fed#10.csv** – Raw fluorescence time series collected from a zebrafish larva fed with paramecia
4. **Batch for FIBSI_v1_0_1 Tutorial_v6 – Zfish AB_Fed#10.bat** – Windows batch file with input commands
5. 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**

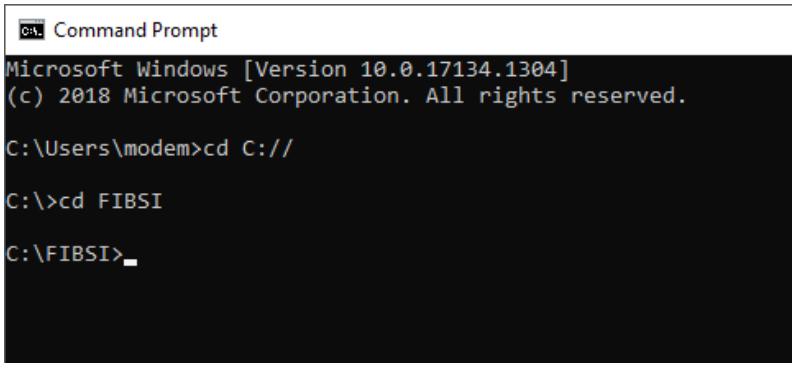
Items 1-4 can be downloaded online at https://github.com/rmcassidy/FIBSI_program. Items in 5 can be downloaded online at https://www.anaconda.com/ and https://www.python.org/.

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

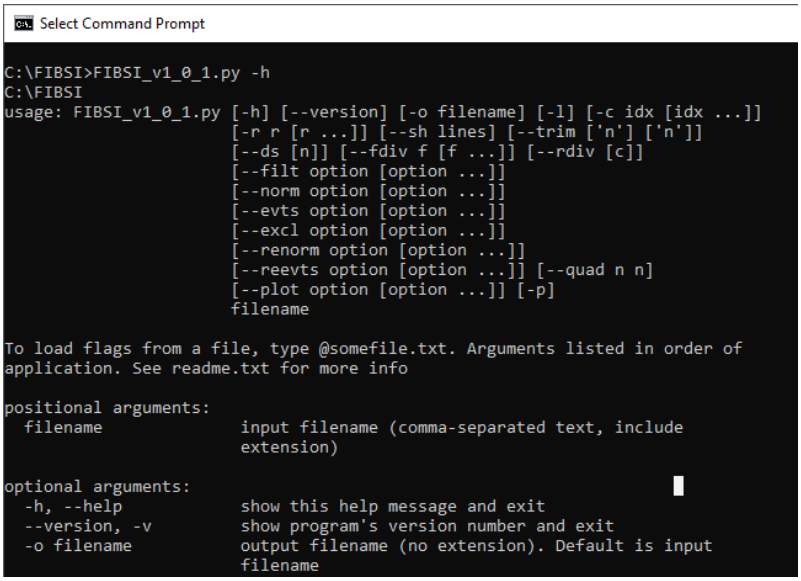
Installation

1. Choose whether to use item 5a or items 5b-c, download and install versions appropriate for your operating system
   Note: If prompted during installation, include Python and/or Anaconda to your PATH variable
2. Create a new folder on a drive of your choosing (e.g., FIBSI folder on the C: drive)
3. Drop FIBSI.py, README.md, the tutorial .csv, and the tutorial .bat file into the new folder

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



5. Test the FIBSI.py help message to ensure it is installed and invoked properly by typing "FIBSI_v1_0_1.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_v1_0_1.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 gut motility analysis performed in our 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
- 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:

- The X and Y data must be found within distinct columns
- Sampling interval
    - Must be consistent for a given X-Y series

    - Can differ between series if the series are input using separate commands (using either the same source file or separate source files)
    - Must be consistent if inputting two or more series from the same source file using a single command
- 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_1.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):

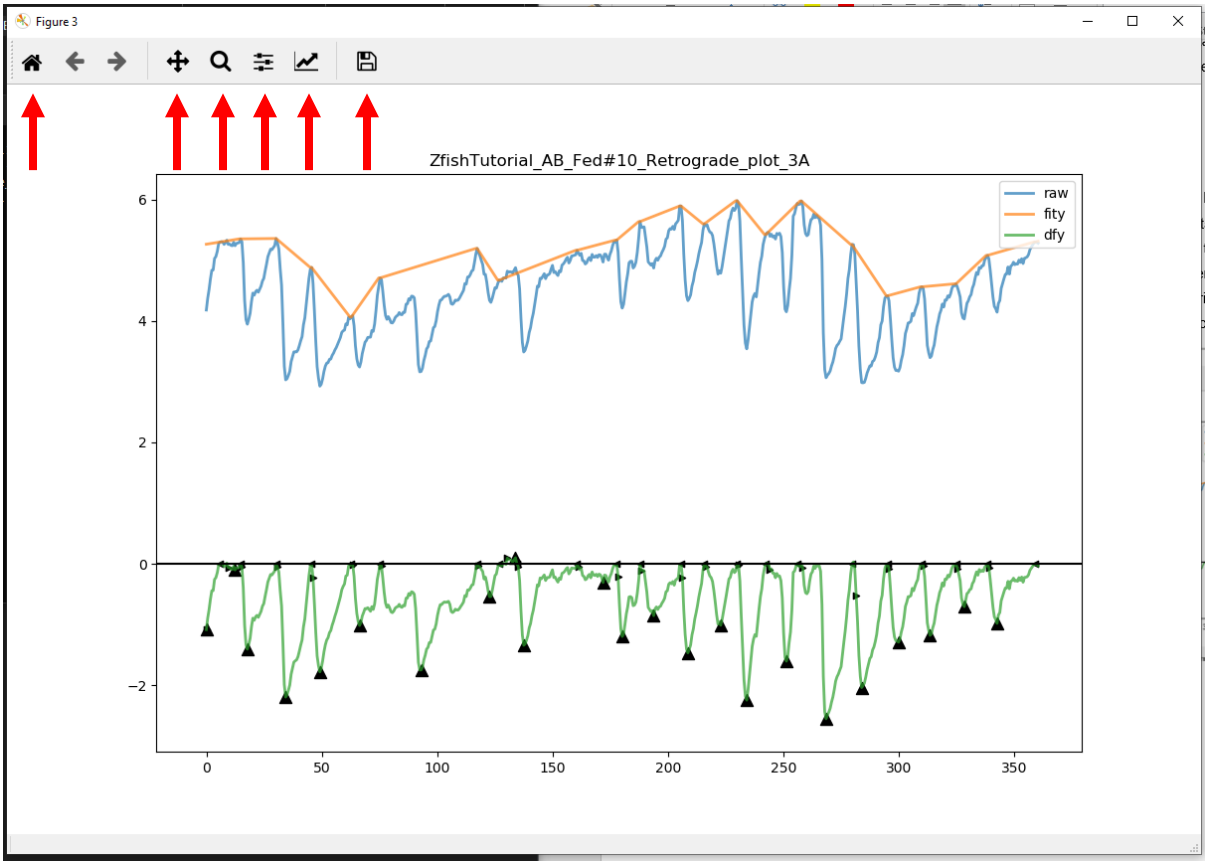
1. Invoke FIBSI_v1_0_1.py to analyze filename.csv and all results will be dumped into the root folder with the filename_results moniker
2. The X data are located in column 15 and Y data are in column 1 – give X, then Y
3. There are 4 series in column 0 with row IDs Cell1, Cell2, Cell3, and Cell4 – give 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 the 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 example(s) using real biological data

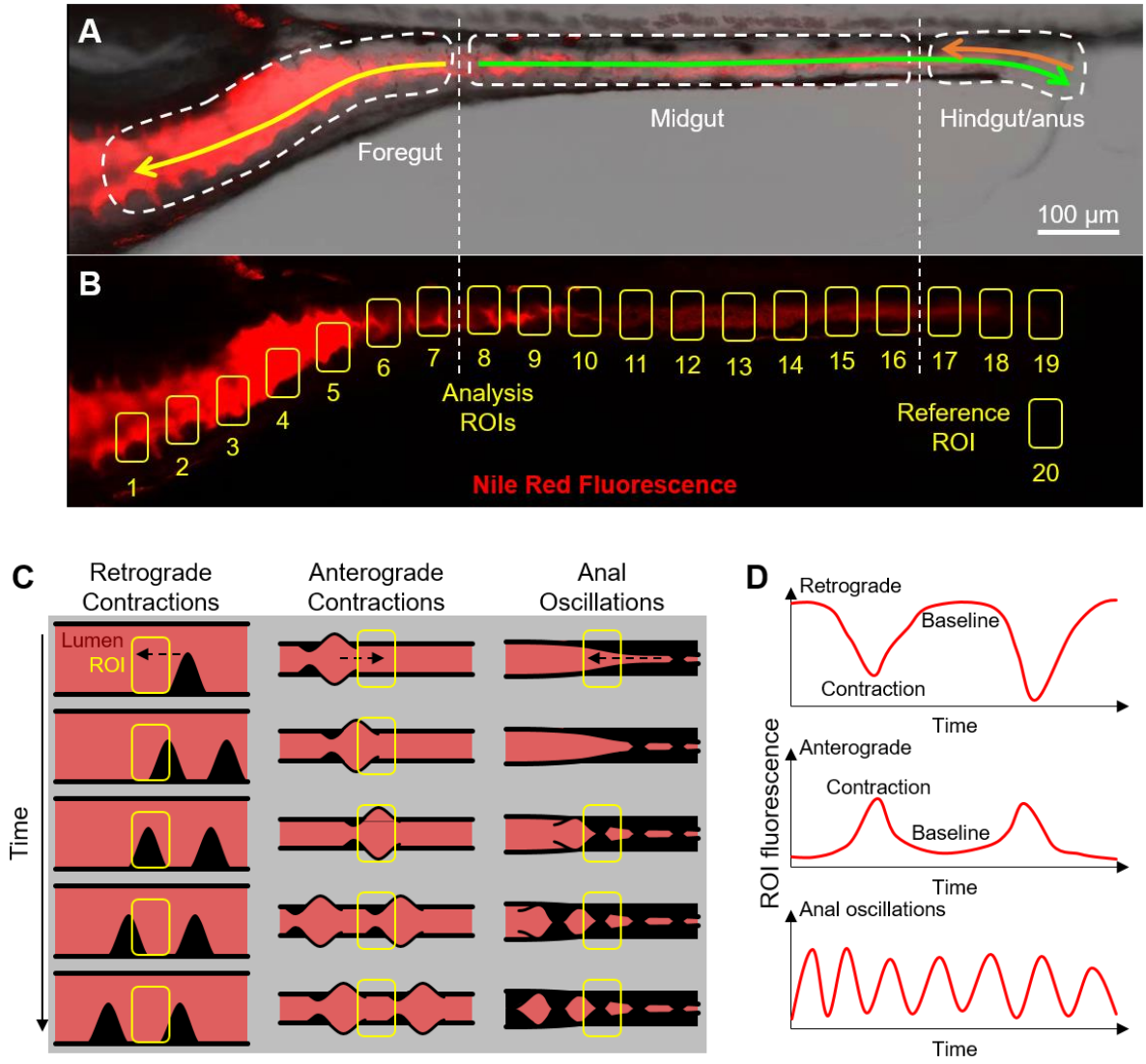
Here we will demonstrate the post-acquisition workflow to analyze gut motility in larval zebrafish. We recommend walking through this example and using the other raw fluorescence time series provided online to familiarize yourself with the workflow so you may begin analyzing your own data.

Measuring gut motility in larval zebrafish using fluorescence contrast imaging

Methodology

In Cassidy et al., *JOURNAL*, YEAR, the total fluorescence intensity X-Y time series were extracted as from the image time series of the zebrafish larvae guts. Analysis ROIs 1-19 were positioned along the gut and reference ROI 20 was positioned outside of the gut. The fluorescence intensity of the Nile Red dye increases or decreases depending on how the gut moves during muscle contractions. In Cassidy et al. we described 3 different kinds of contraction events:

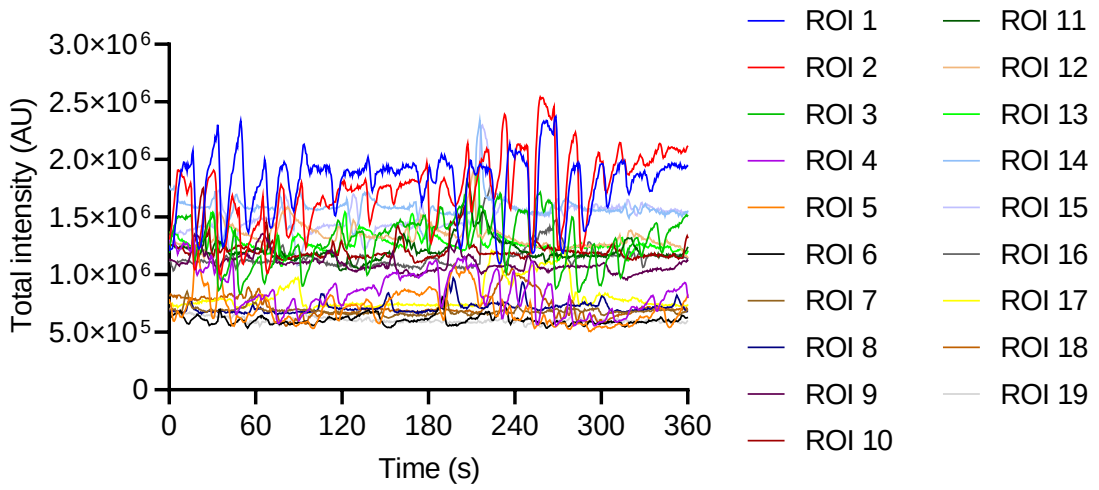
- Narrowing of the gut
    - Decrease in fluorescence intensity
    - Predominantly observed in the foregut ROIs (ROIs #1-7 in the below example)
- Dilation of the gut
    - Increase in fluorescence intensity
    - Predominantly observed in the midgut/hindgut ROIs, sometimes observed in distal foregut ROIs (ROIs #8-19)
- Undulations in the anus
    - Generates an oscillation in the fluorescence intensity
    - Observed in the hindgut ROIs (ROIs #17-19)



The type of movement by the gut influences whether the contraction 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 flag for contractions that decrease fluorescence, and vice versa). 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 in the foregut while smaller window sizes are better for a more fluid baseline in the midgut and hindgut/anus. Choosing the best-fit approach and optimal window size will require observation of gut motility videos synced with the fluorescence time series data.

This tutorial will mirror the methods used in Cassidy et al., *JOURNAL*, YEAR. The settings used to analyze **ZfishTutorial_AB_Fed#10.csv** (commands written in the **Batch for FIBSI_v1_0_1 Tutorial_v6 – Zfish AB_Fed#10.bat** file, raw fluorescence time series shown below):

- For retrograde contractions in ROIs #1-7
    - rmd window size = 50 X units with X-cutoff = 3 units
    - re-normalize to “above” with X-cutoff = 3 units; no Y-cutoffs
- For anterograde contractions in ROIs #8-19
    - rmd window size = 25 X units with X-cutoff = 3 units
    - re-normalize to “below” with X-cutoff = 3 units: no Y-cutoffs
- For anal oscillations in ROIs #17-19
    a. rmd window size = 10 X units with X-cutoff = 1 unit
    b. re-normalize to “above” with X-cutoff = 3 units; no Y-cutoffs

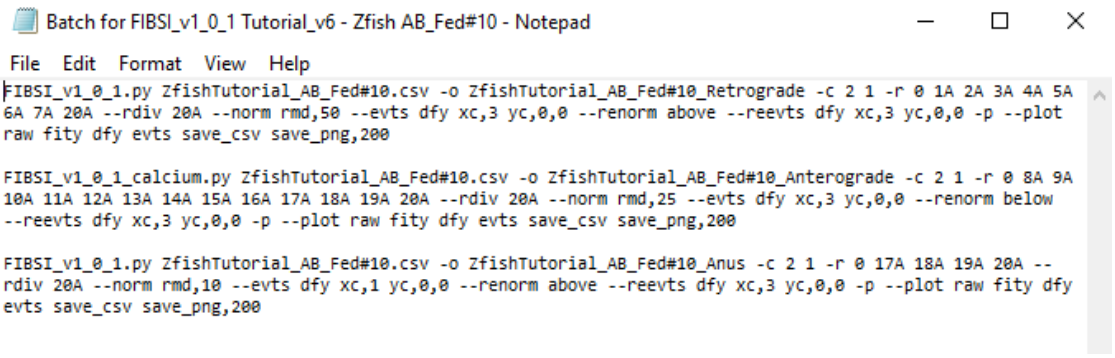


Processing the tutorial data

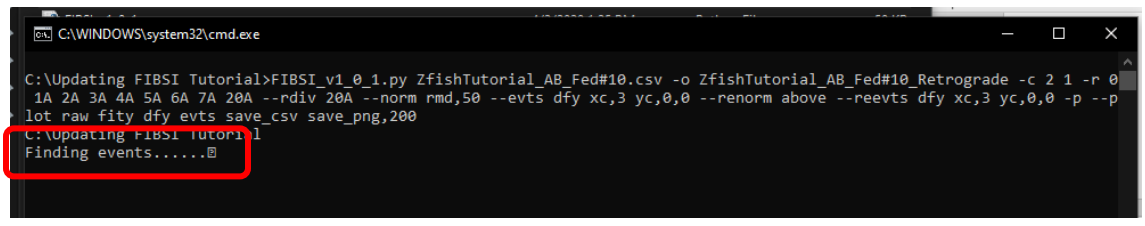
1. Drop the **ZfishTutorial_AB_Fed#10.csv** and **Batch for FIBSI_v1_0_1 Tutorial_v6 – Zfish AB_Fed#10.bat** files into your root folder
2. Inspect each file
    a. The .csv should have 4 columns, with 609 frames for ROIs 1A-20A

	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

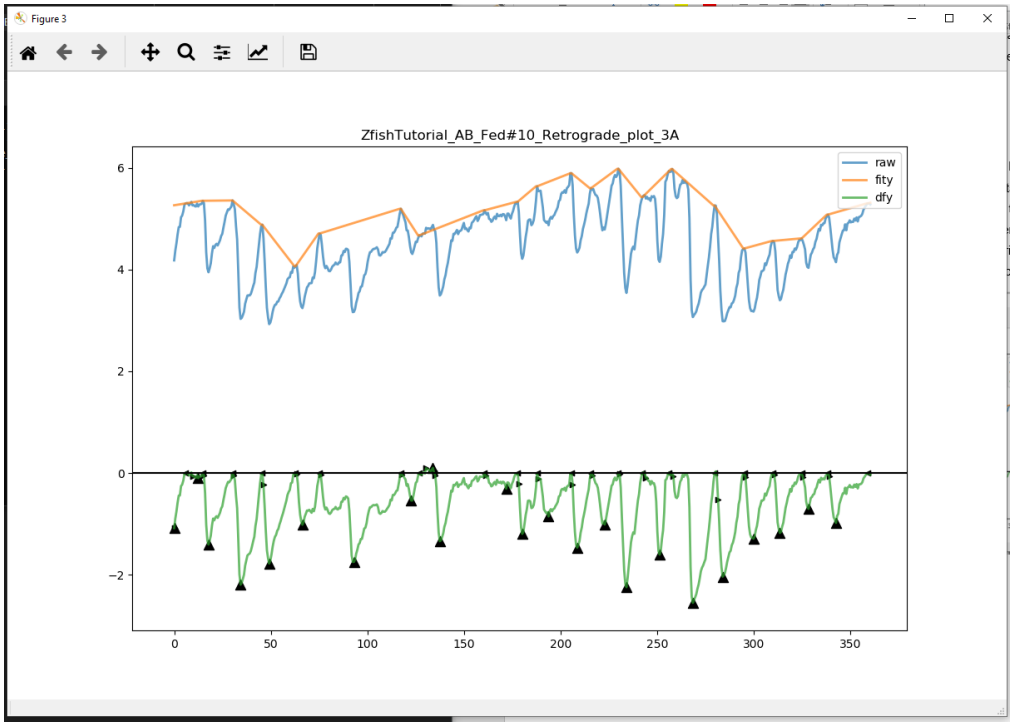


3. Run FIBSI
   a. If running FIBSI using Command Prompt, change your directory to your root folder then type each command and hit enter; If running FIBSI using the .bat file, double click the .bat file in the root folder
   b. If run correctly, a real-time status will update as each command is processed



c. The pyplot interactive viewer will open for each ROI in the order of commands processed

Note: Exit out of each pyplot viewer in order to proceed



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. 22 .csv files, 1 for each ROI analyzed – contains the Y values for the raw, fity, and dfy lines
    c. 22 .png files, 1 for each ROI analyzed – captures the default view for each pyplot viewer
    d. 22 .txt files ending in _evts, 1 for each ROI analyzed – 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 analysis.

Analyzing the processed tutorial data

The .txt files ending in _evts_formatted list the detected contractions/oscillations in sequential order, per ROI. Columns A and C-H will be used for analysis. The remaining columns are provided by default, but not necessary for this tutorial. Contractions/oscillations will be isolated from noise (i.e., signal artefacts and/or fluorescence fluctuations not caused by muscle contractions) using post-hoc filters applied to the .txt files.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1											ZfishTutorial_AB_Fed#10_Retrograde			
2	ROI	Quadrant	Direction	Start time	End time (	Peak time	Duration (	Amplitud	Midpoint	Half Durat	AUC			
3	1A	2	b	0	8.295672	1.777644	8.295672	-2.08949	4.147836	4.147836	-18.246			
4	1A	2	b	11.85096	15.9988	12.44351	4.147836	-0.29043	13.6286	2.073918	-1.25891			
5	1A	2	b	17.77644	33.77524	21.33173	15.9988	-3.15347	25.47956	7.999398	-43.4915			
6	1A	2	b	34.36778	49.77403	37.33052	15.40625	-3.7753	42.07091	7.703124	-51.2029			
7	1A	2	b	50.36658	66.36538	55.10696	15.9988	-3.04412	58.0697	7.999398	-43.732			
8	1A	2	b	66.95792	80.58653	70.51321	13.6286	-2.06011	73.47595	6.814302	-22.7541			
9	1A	2	b	81.17908	93.03004	86.51201	11.85096	-1.7036	87.10456	5.92548	-19.4433			
10	1A	2	b	94.80768	123.25	115.5469	28.4423	-1.60772	109.0288	14.22115	-25.9472			
11	1A	2	b	123.8425	138.0637	132.7308	14.22115	-0.41381	130.9531	7.110576	-5.71264			
12	1A	2	b	138.6562	147.5445	141.0264	8.88822	-0.89955	142.8041	4.44411	-5.75264			
13	1A	2	b	149.3221	151.0997	150.5072	1.777644	-0.13696	149.9146	0.888822	-0.21602			
14	1A	2	b	151.6923	154.0625	151.6923	2.370192	-0.18062	152.8774	1.185096	-0.3077			
15	1A	2	b	154.655	162.3582	157.0252	7.703124	-0.29708	158.2103	3.851562	-2.48856			
16	1A	2	b	162.9507	189.0228	175.3942	26.07211	-1.35818	175.9868	13.03606	-22.1321			
17	1A	2	b	189.6154	191.393	190.2079	1.777644	-0.08341	190.2079	0.888822	-0.13786			
18	1A	2	b	193.1706	196.1334	194.9483	2.96274	-0.14643	194.3557	1.48137	-0.34501			
19	1A	2	b	196.7259	209.1694	202.6514	12.44351	-2.62449	202.6514	6.221754	-29.8392			

ZfishTutorial_AB_Fed#10_Retrogr

Note: The X- and Y-cutoffs that are set as part of the re-normalization function (--reevts flag) in the .bat commands can be changed to mirror the post-hoc filters described below, if desired by the user. Be mindful that X-cutoff will need to be converted to match the X and Y units used (e.g., if using a ≥10 sec duration cutoff for a time series sampled at 0.5 frames/sec, change the X-cutoff to 20 X units).

- Isolate retrograde contractions in ROIs 1-7
    - Normalizing dfy to signal peaks means contractions are “below” the normalization
    - Column C “Direction” – Select the “below” events, filter the “above” events
    - Duration filter x ≥ 10 sec
    - No amplitude filter necessary
- Isolate anterograde contractions in ROIs 8-19
    - Normalizing dfy to signal troughs means contractions are “above” the normalization
    - Column C “Direction” – Select the “above” events, filter the “below” events
    - Duration filter x ≥ 10 sec
    - No amplitude filter necessary

- Isolate anal oscillations in ROIs 17-19
    - Normalizing dfy to signal peaks means oscillations are “below” the normalization
    - Column C “Direction” – Select the “below” events, filter the “above” events
    - Set duration window between 3 sec ≤ x < 10 sec
    - Set amplitude filter x ≥ 0.01 ΔF (AU)

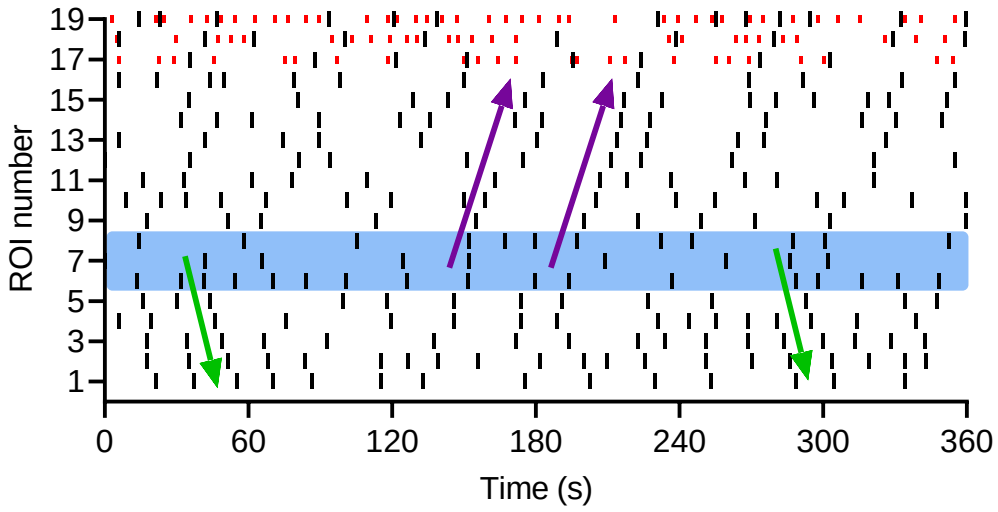
The isolated contractions/oscillations can be rasterized to visualize contraction rhythmicity. In Cassidy et al., *JOURNAL*, YEAR, FIBSI’s accuracy was assessed by superimposing the rasterized output onto spatiotemporal maps of gut movement corresponding to the same acquisition period. Raster plots should depict contraction waves passing through the discrete ROIs as a function of time. The plot will shift depending on which information is used (i.e., contraction start, peak, or end time). Peak times are used in Cassidy et al. and this tutorial. To generate a raster, plot the ROI #s along the Y axis and the peak times along the X axis (example generated using Prism software):

Table format: XY	X	Group A	Group B	Group C	Group
	Time (s)	Retrograde	Anterograde	Oscillations	Title
	X	Y	Y	Y	Y
93	Title	288.570900	6		
94	Title	298.051600	6		
95	Title	316.420600	6		
96	Title	331.234300	6		
97	Title	348.418200	6		
98	Title	0.000000	7		
99	Title	42.070910	7		
100	Title	65.772830	7		
101	Title	124.435100	7		
102	Title	152.284800	7		
103	Title	209.169400	7		
104	Title	259.536000	7		
105	Title	286.200700	7		
106	Title	302.199500	7		
107	Title	14.221150		8	
108	Title	58.069700		8	
109	Title	105.473500		8	
110	Title	152.284800		8	
111	Title	167.098500		8	
112	Title	179.542000		8	
113	Title	197.318500		8	
114	Title	232.278800		8	
115	Title	245.314900		8	
116	Title	287.385800		8	

Qualitatively assess the raster plot’s features to ensure FIBSI’s event detection accurately reflects the raw fluorescence time series and videos:

- Contraction rhythmicity (i.e., contiguous waves of **black ticks**) should be visible

- Contractions should originate from a site of pacemaker-like activity near the end of the foregut (ROIs #6-8, **blue highlight**)
- Contraction waves should travel in two directions from the origin
    - Retrograde contractions (**green arrows**)
    - Anterograde contractions (**purple arrows**)
- Anal oscillations (**red ticks**) may also exhibit rhythmicity



The rasterized output can also be used to calculate the contraction/oscillation frequency in each ROI and mean frequencies:

$$\overline{f_{retrograde}} = \frac{(f_{ROI1} = x_{ROI1}/t) + f_{ROI2} + \cdots + f_{ROI7}}{7}$$

$$\overline{f_{anterograde}} = \frac{(f_{ROI8} = x_{ROI8}/t) + f_{ROI9} + \cdots + f_{ROI9}}{12}$$

$$\overline{f_{oscillations}} = \frac{(f_{ROI17} = x_{ROI7}/t) + f_{ROI8} + f_{ROI9}}{3}$$

These frequencies can then be used to compare different zebrafish and experimental conditions. Results for **ZfishTutorial_AB_Fed#10.csv**: retrograde = 0.042 Hz, anterograde = 0.034 Hz, oscillations = 0.083 Hz.

Final notes

This tutorial will be updated to reflect any modifications to the gut motility analysis workflow that arise from the peer review process. Additional changes may reflect non-peer reviewed updates to the current workflow, new quantitative/qualitative metrics, and/or new methodologies, and will be noted as such.