



VERSION 2

FEB 17, 2023

OPEN  ACCESS

DOI:

dx.doi.org/10.17504/protocols.io.rm7vzbb75vx1/v2

Protocol Citation: Clark Fritsch 2023. Selecting a Region of Interest for Hidden Markov Modeling using ebFRET. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.rm7vzbb75vx1/v2>
Version created by [Clark Fritsch](#)

License: This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working
We use this protocol and it's working

Created: Feb 17, 2023

Last Modified: Feb 17, 2023

PROTOCOL integer ID:
77154

Selecting a Region of Interest for Hidden Markov Modeling using ebFRET V.2

Clark Fritsch¹

¹University of Pennsylvania



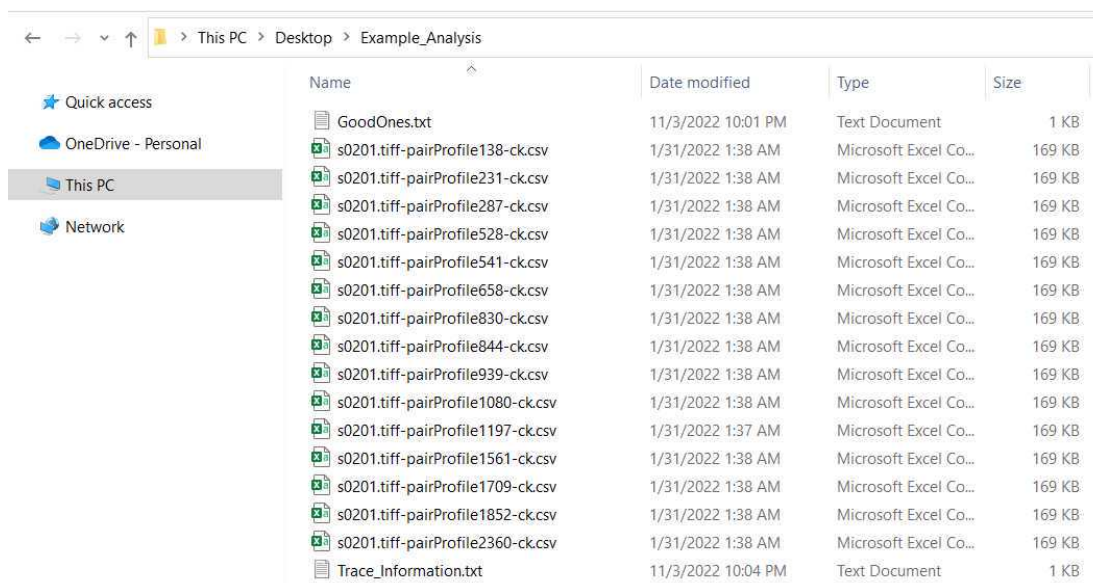
Clark Fritsch

University of Pennsylvania

ABSTRACT

This protocol follows from the "Basic Analysis Protocol" and is the first step towards analyzing your data using Hidden Markov Modeling. In this protocol, you select traces containing features that you wish to analyze using Hidden Markov Modeling and isolate only the regions of interest from each of the traces.

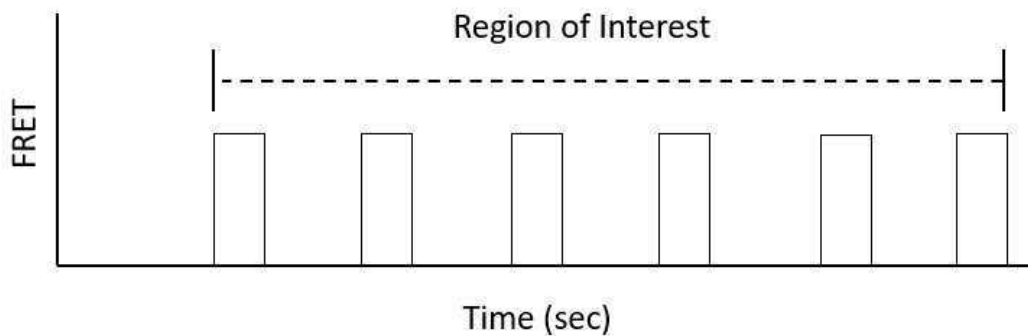
- 1 After generating a catalog of traces that you consider worthy of further analysis and saving these traces to your "GoodOnes.txt" file for a given directory (as described in the "Basic Analysis Protocol") you will have the opportunity to measure specific parameters in your traces by manually selecting regions of interest in each trace for downstream analysis.
- 2 For the purposes of this protocol, I will be using the following traces and GoodOnes.txt file:



Name	Date modified	Type	Size
GoodOnes.txt	11/3/2022 10:01 PM	Text Document	1 KB
s0201.tiff-pairProfile138-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile231-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile287-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile528-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile541-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile658-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile830-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile844-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile939-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile1080-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile1197-ck.csv	1/31/2022 1:37 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile1561-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile1709-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile1852-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
s0201.tiff-pairProfile2360-ck.csv	1/31/2022 1:38 AM	Microsoft Excel Co...	169 KB
Trace_Information.txt	11/3/2022 10:04 PM	Text Document	1 KB

 Example_Analysis.zip

- 3 For the purposes of this protocol, I am interested in determining the time between FRET events in my traces and the duration of the FRET events. Because of this, I will select the region of interest as the region between two distinct FRET events, as described below:

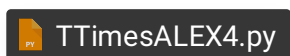


Questions:

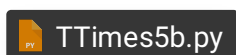
1. What is the lifetime of each FRET event in the region of interest?
2. What is the time between each FRET event in the region of interest?

4 You can select your region of interest using one of the following programs:

TTimesALEX4.py - Use this program if you have generated traces from an ALEX movie.



TTimes5b.py - Use this program if you have generated traces from a non-ALEX movie.



Functionally, these programs are exactly the same. They both take your GoodOnes.txt file as input and then read in each trace in your GoodOnes.txt file and allows you to select specific regions of the traces for further analysis. The TTimesALEX4.py program simply generates an additional space in each plot to display the signal intensity that comes from the direct excitation of the acceptor fluorophore that you obtain from ALEX measurements.

Note that for these programs to work, you need to have your GoodOnes.txt file in the same directory as your traces of interest, as shown in the image in Step 2 of this analysis protocol.

5 To run these programs, simply open them in Spyder (or your preferred Python IDE) and enter the path to the pair profiles that you want to analyze between the " " for the path variable, as shown below:

To run these programs, simply open them in Spyder (or your preferred Python IDE) and enter the path to the GoodOnes.txt file and traces that you want to analyze between the " " for the path variable, as shown below:

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Nov 6 07:02:58 2019
4
5 @author: goldman3g
6 """
7 import numpy as np
8 import matplotlib.pyplot as plt
9 import os
10 import csv
11 import sys
12 import re
13 from scipy import signal
14
15 #path = r"D:\Dropbox\Crew\Clark\PRE-6"
16 #path = r"D:\Dropbox\Crew\Clark\TransLocation"
17 #path = r"D:\Dropbox\Crew\Clark\120222"
18 #path = r"D:\Dropbox\Crew\Clark\TransLocation\More_Examples"
19 #path = r"D:\Dropbox\Crew\Clark\PRE-6_TrypIRES"
20 #path = r"D:\Dropbox\Crew\Clark\TransLocation\Good_Translocation_Traces"
21 path = r"C:\Users\Clark\Desktop\Example_Analysis1"
22
23 #####
24
25 def onclick(event):
26     global nct
27     global i
28
29
30     if (event.button == 1):
31         print("button0d, x=d4, y=d4, xdata=%2f, ydata=%2f" % (
32             event.button, event.x, event.y, event.xdata, event.ydata))
33
34     if (event.button == 3):
35         nct += 1
36         it = int(event.xdata/dt*0.5)
37         pte = it*dt
38         print("Click %d, Index: %d, Time: %.1f" % (nct, it, pte))
39         if (nct == 1):
40             file = open(path + "\\\" + 'LastI.txt', 'w')
41             file.write(str(nct)+"\n")
42             file.write("%d %d\n" % (nct, it))

```

Note that you also need to change the frame interval used in the program to match the frame interval that you used in your experiments. For example, for these example traces I used a frame interval of 40 ms. Because of this, I will go to Line 81 of the program and change the "dt" variable to equal 0.04, as shown below:

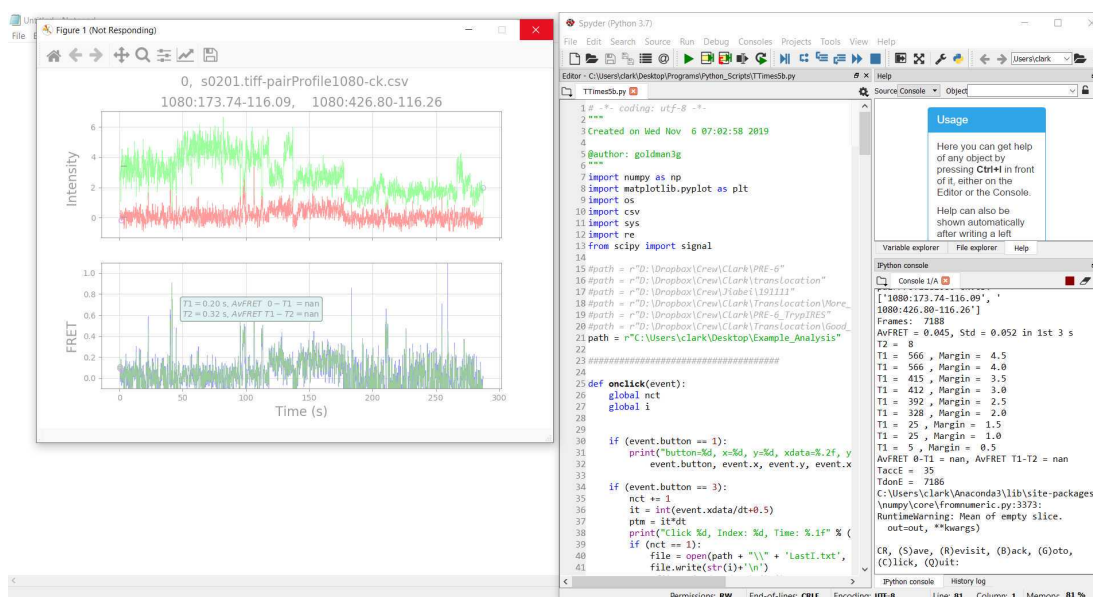
```

63 def findTime (var, thresh, ud): #find the time at which
64     if ud > 0:
65         for k in range(5,len(var)-1):
66             if (var[k] > thresh):
67                 print("B lip lip = ", k)
68                 if (np.mean(var[k:k+5]) > thresh):
69                     break
70             else:
71                 for k in range(5,len(var)-1):
72                     if (var[k] < thresh):
73                         print("B lip lip = ", k)
74                         if (np.mean(var[k:k+5]) < thresh):
75                             break
76         return k
77
78 #####
79
80 #change frame rate here
81 dt = 0.04
82 print ("Non-ALEX time resolution = ", dt, " s")
83
84 plt.close('all')
85
86 T1 = 0
87 T2 = 0
88 TaccE = 0
89 TdonE = 0
90
91 file_list = gfn()
92 print ("files: ", len(file_list))
93 i = 0
94 nct = 0
95 plist = []
96 print ("Stored click times:")
97 for fn in file_list:
98     if "LastI" in fn:
99         file = open(path + "\\\" + 'LastI.txt', 'r')
100         i = int(file.readline(3)) #index of last opened pfile
101         i = int(file.readline(10))
102         for jct in range(4):
103             try:
104                 clkTime = file.readline(10) #should be 0 or 4 of these
105                 spl = clkTime.split(',')

```

6 Then you simply run the program by clicking the green triangle button on the top bar of the Spyder tool bar and the program will begin processing your files. You can see the program's controls in

the IPython Console in the lower right corner of the Spyder IDE, as shown below:



Once you run the program, you will see a window pop up that contains the plotted traces (shown individually) that are contained in your path directory and are contained within the GoodOnes.txt file. You will also see the controls for the program in the IPython Console in the lower right hand corner of the IDE:

(S)ave: Saves the selected trace and selected region of interest into an external file (SavedData.csv)

(R)everisit: Not relevant for this analysis.

(B)ack: Moves the index backwards by one index so that you can see the previous trace.

(G)oto: Allows you to enter in a trace index and go to that index automatically without having to view every trace one at a time. For example, to go to the fifth trace in your directory, you would simply press "G" and then enter the number "5" in the prompt that appears in the IPython Console.

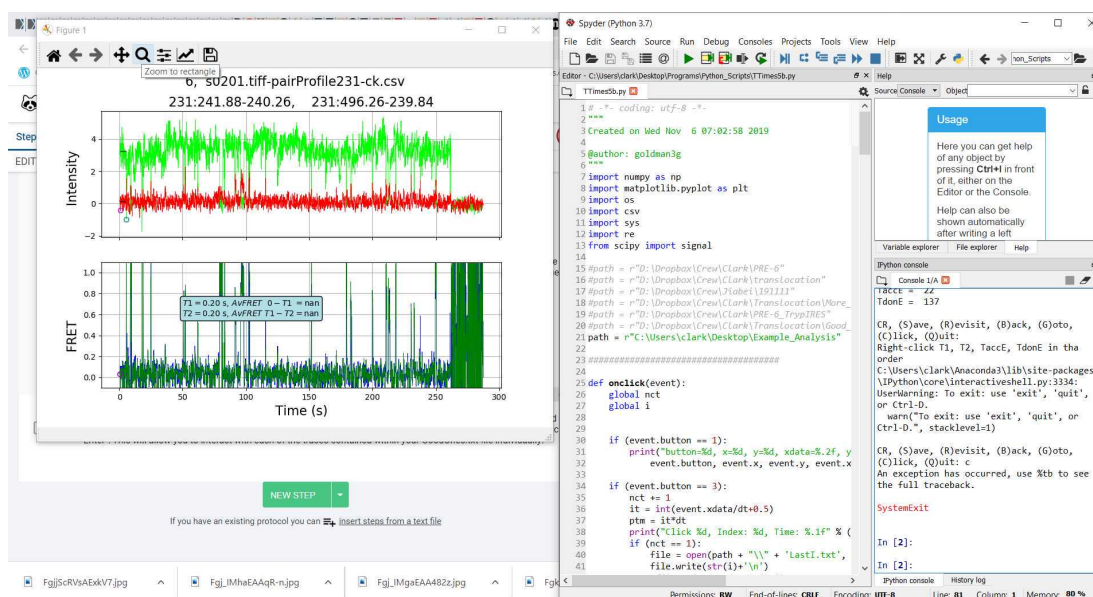
(C)lick: By entering "C" into the IPython Console and pressing enter, you are able to manually interact with the trace that the program is displaying.

(Q)uit: By entering "Q" into the IPython Console and pressing enter, you can quit the program.

I recommend dragging the IDE to the right hand of the screen and dragging the trace plot to the left hand of the screen. It will make it easier to cycle through the traces quickly.

7 As stated previously, for the purposes of this analysis, we want to select a region of interest in each of the displayed traces for further analysis. To do this, simply enter "C" into the IPython

Console while the program is running and click "Enter". This will allow you to interact with each of the traces contained within your GoodOnes.txt file individually:



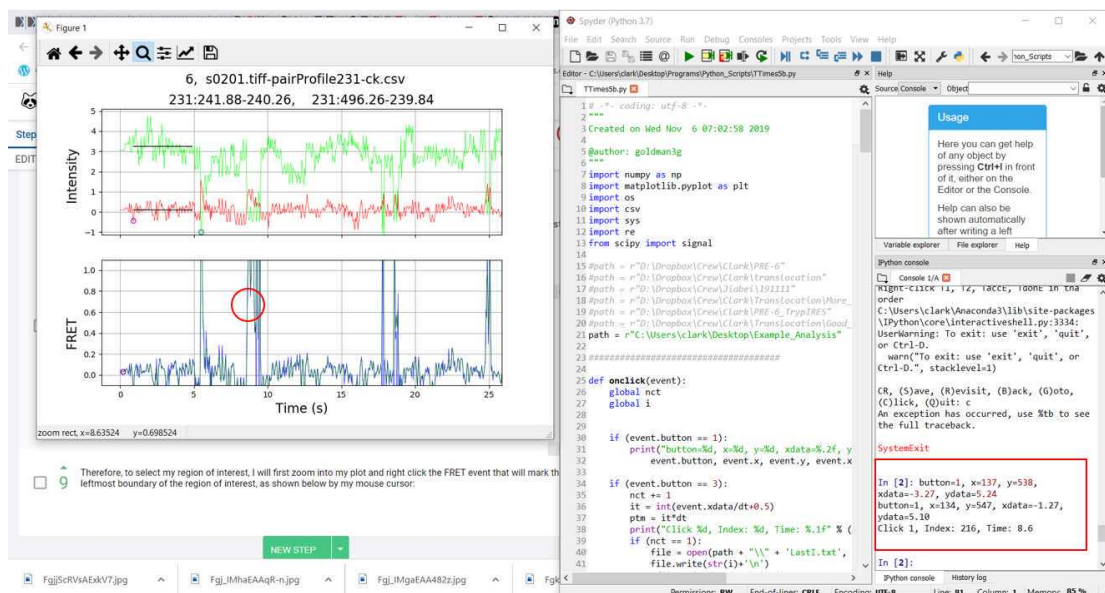
Once you enter "C" into the console and click "Enter", you will notice that the system exits the program and that the plot of your trace becomes available for you to interact with. The two options available to you in the plot that are most useful are the magnifying glass that allows you to zoom into your plot and the movement key, which is positioned immediately to the left of the magnifying glass. The movement key allows you to scroll through the trace as necessary.

- 8 Once in this state, the program allows you to RIGHT CLICK the selected plot four times. Each click will save the position of the click into an external output file (SavedData.csv). The clicks that you do represent the following information:

Right Click 1: The leftmost boundary of your region of interest
 Right Click 2: The rightmost boundary of your region of interest
 Right Click 3: The photobleaching time of your sensitized emission signal
 Right Click 4: The photobleaching time of your donor emission signal

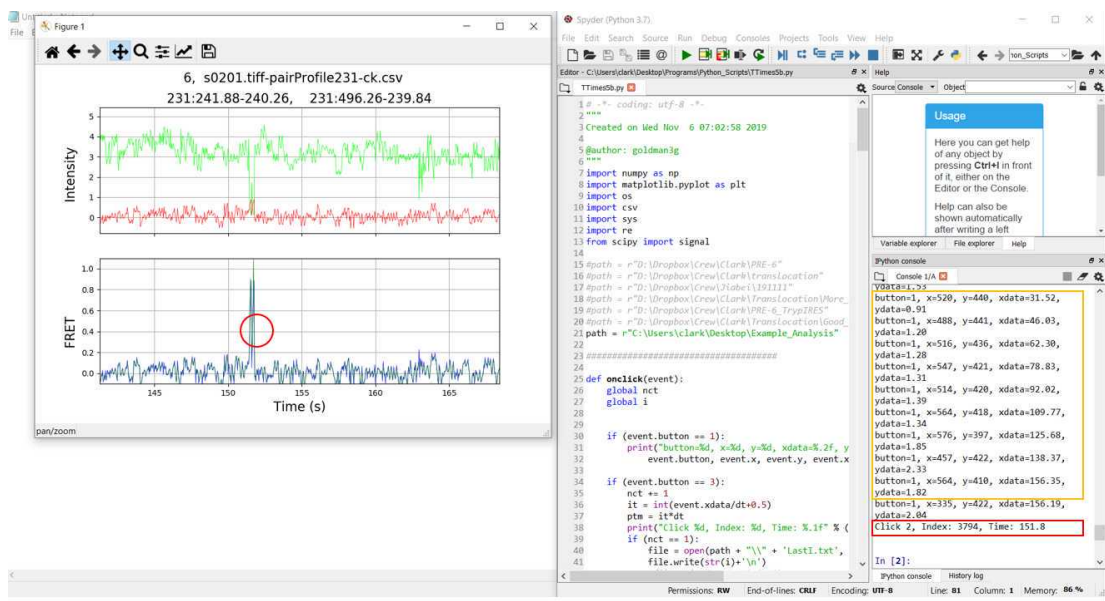
For the purposes of this analysis, we will not be assigning values to Click 3 or Click 4. Only Click 1 and Click 2 are necessary for this analysis.

- 9 Therefore, to select my region of interest, I will first zoom into my plot and right click the FRET event that will mark the leftmost boundary of the region of interest, as shown below by my mouse cursor:



Once you have clicked the leftmost boundary of your region of interest, you can see that the information for the click (index and time of the click in the trace) appears in the IPython Console, as marked by the red box.

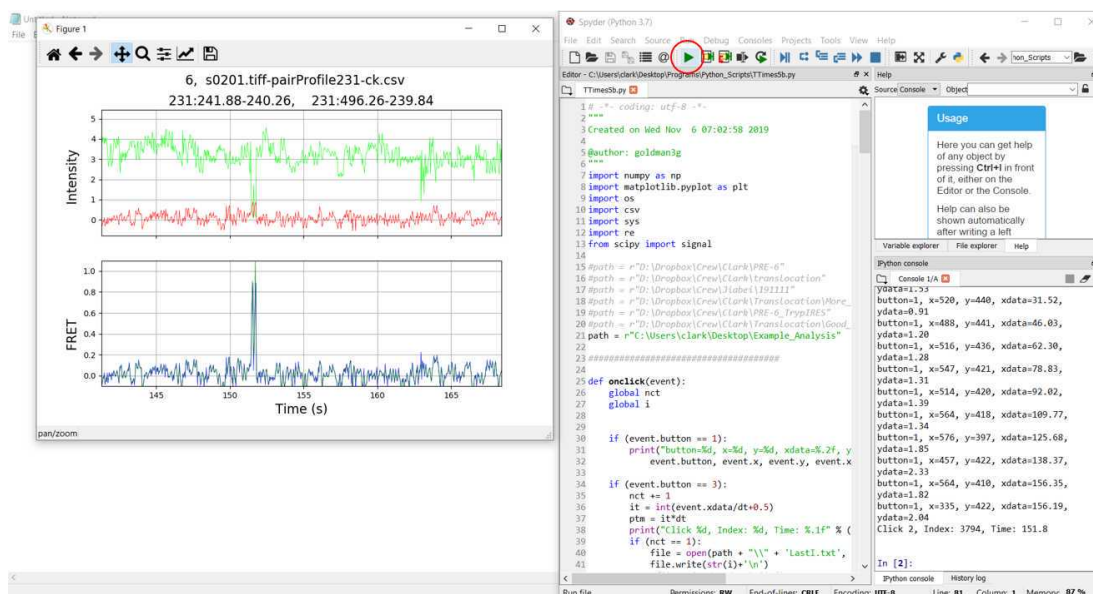
10 Next, I need to select the rightmost boundary of my region of interest and so I will scroll through my trace (still zoomed in) and find the rightmost FRET event that I am interested in. I will then right click the event, as shown below by the red circle in the plot:



As stated above, once you have clicked the rightmost boundary of your region of interest, the information regarding the click will appear in the IPython Console, as shown in the red box in the above image. Additionally, it is important to note that when you are scrolling through your plot to find the different regions of interest (scroll by left clicking and dragging the plot), the information

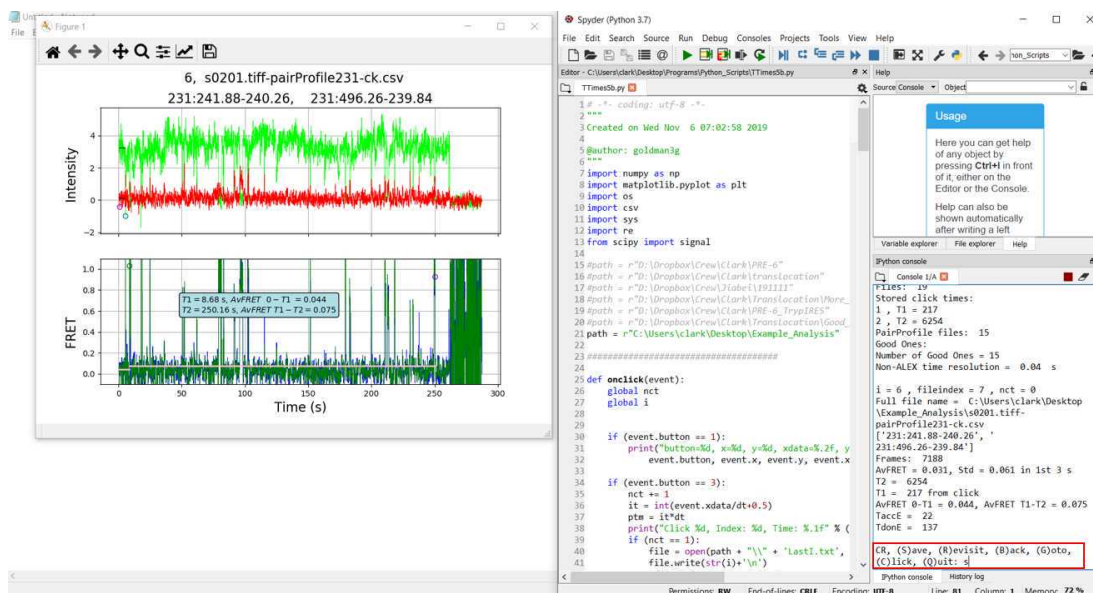
for where you are left clicking will also appear in the IPython Console (as shown in the orange box in the above region). This information is not important and is not stored or used in future analysis. It is just displayed in the console.

- 11 Once you have selected your leftmost and rightmost boundaries for your region of interest, you can re-initiate the program by clicking the green triangle button on the top bar of the Spyder tool bar, as highlighted by the red circle in the below image:



You will then see that the IPython Console has started functioning again as it was before you clicked "C" to interact with your plot.

- 12 To save the "clicked" data to an external file for your trace, you then want to input "S" into the IPython Console and press "Enter", as shown below in the red box:

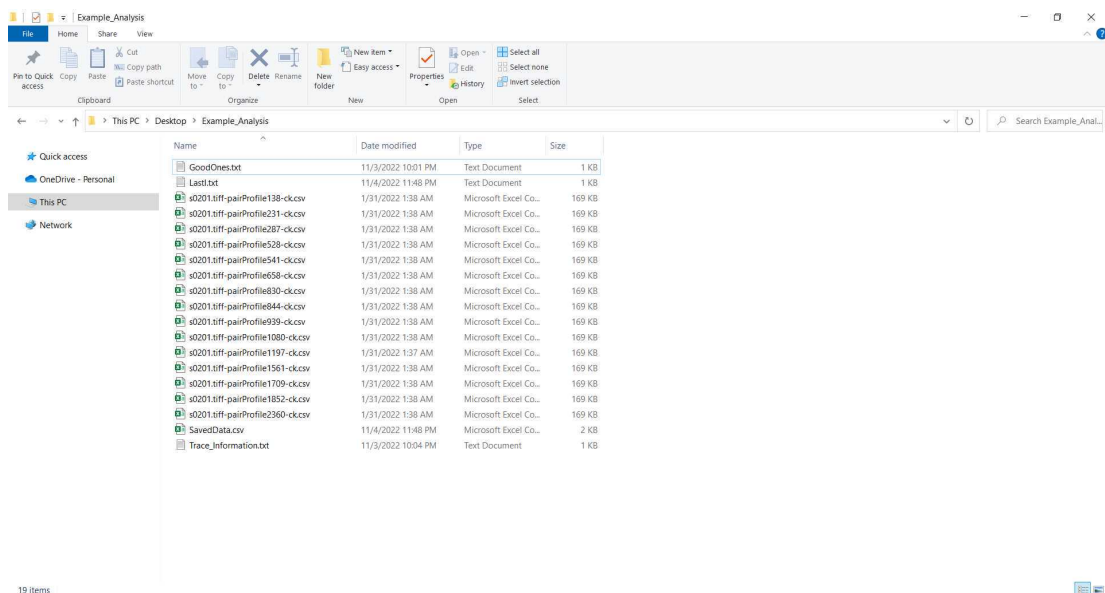


13 Once you press "Enter", the program will save your clicked data and move onto the next trace. You can then repeat this process for each of your traces. Note that, depending on your analysis and the number of traces that you need to analyze, this process can become extremely time consuming.

Once you have finished with your traces, simply type "q" and press "Enter" to exit the program.

14 Once you have finished, you can then go to the directory that contains your GoodOnes.txt file and your traces and you will see two new files:

1. SavedData.csv: Contains all of the information from each click for each of your traces.
2. LastI.txt: Contains information on the last visited index that you were on prior to exiting the program.



15 When you open the "SavedData.csv" file, you will see the data in the following format:

AutoSave Off SavedData.csv

Search (Alt+Q)

Clark Fritsch

File Home Insert Page Layout Formulas Data Review View Help

Clipboard Font Alignment Number Styles Cells Editing Analysis

Comments Share

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Index	File Index	Click1	Click2	TaccE	TdOne	AvFRET_Click1	AvFRET_Click2	AvDonBTaccE	AvDonaTaccE	AvAccbTaccE	AvAcceTaccE	filename							
2	6	7	215	6258	22	137	0.047	0.076	nan	3.26	nan	0.13	s0201.tiff-pairProfile1231-ck.csv							
3	0	1	565	2828	35	7186	0.014	0.039	nan	3.43	nan	0.06	s0201.tiff-pairProfile1080-ck.csv							
4	1	2	115	6914	63	1205	0.058	0.103	nan	3.13	0.4	0.23	s0201.tiff-pairProfile1197-ck.csv							
5	2	3	98	5273	36	171	0.032	0.038	nan	6.09	nan	0.21	s0201.tiff-pairProfile138-ck.csv							
6	3	4	81	2654	55	7186	0.07	0.077	2.97	2.58	0.36	0.44	s0201.tiff-pairProfile1561-ck.csv							
7	4	5	360	3206	15	5	0.257	0	nan	0.72	nan	-0.24	s0201.tiff-pairProfile1709-ck.csv							
8	5	6	936	7028	19	7186	0.027	0.071	nan	4.68	nan	0.06	s0201.tiff-pairProfile1852-ck.csv							
9	7	8	330	1158	34	1749	0.087	0.134	nan	2.33	nan	0.35	s0201.tiff-pairProfile2360-ck.csv							
10	8	9	198	4044	31	275	0.041	0.036	nan	4.52	nan	0.24	s0201.tiff-pairProfile287-ck.csv							
11	9	10	86	5702	18	1854	0.027	0.082	nan	1.95	nan	0.13	s0201.tiff-pairProfile528-ck.csv							
12	10	11	184	2870	18	4301	0.034	0.05	nan	4.93	nan	0.26	s0201.tiff-pairProfile541-ck.csv							
13	11	12	58	2638	36	3142	0.044	0.056	nan	4.77	nan	0.47	s0201.tiff-pairProfile658-ck.csv							
14	12	13	785	5373	5	3768	0.025	0.026	nan	5.98	nan	0.15	s0201.tiff-pairProfile830-ck.csv							
15	13	14	277	3502	29	418	-0.009	0.061	nan	3.5	nan	-0.14	s0201.tiff-pairProfile844-ck.csv							
16	14	15	60	2141	45	4086	0.052	0.079	nan	3.19	nan	0.43	s0201.tiff-pairProfile939-ck.csv							
17																				
18																				
19																				
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				

SavedData

Ready Accessibility: Unavailable

Note that normally, the program does not include column headings, but I have included them here to show the data that each column contains, as described below:

Index: Index of trace (Python uses zero-based indexing, meaning the first file or trace will have an index of 0 while the second trace will have an index of 1, etc.)

File Index: The actual position of the file or trace that you are looking at. The first file that python reads in will have a file index of 1, while the second file will have a file index of 2.

Click 1: The frame of your recording that you selected as your leftmost boundary for your region of interest. To convert frames to an actual time (in seconds), simply multiply the number here by your frame interval (0.04, in this practice protocol).

Click 2: The frame of your recording that you selected as your rightmost boundary for your region of interest. To convert frames to an actual time (in seconds), simply multiply the number here by your frame interval (0.04, in this practice protocol).

TaccE (Click 3): The frame of your recording at which your acceptor fluorophore (sensitized emission) undergoes photobleaching or loss of signal.

TdonE (Click 4): The frame of your recording at which your donor fluorophore undergoes photobleaching or loss of signal.

AvFRET_0-Click1: The average FRET value between the start of your trace and the position of Click 1 (the leftmost boundary of your ROI).

AvFRET_Click1-Click2: The average FRET value between the position of Click 1 (the leftmost boundary) and Click 2 (the rightmost boundary).

AvDonBTaccE: N/A

AvDonaTaccE: N/A

AvAccbTaccE: N/A

AvAccaTaccE: N/A

filename: The filename of the trace associated with the data in the other columns for a given row.

Note that TaccE and TdonE are useful information to have for some applications, but they are not necessary for what we are covering in this practice protocol. If you don't explicitly make a third and fourth click for each trace, it will simply autofill with an automatically generated value.

- 16** Once you have your "SavedData.csv" file, you can move on to further analysis using ebFRET or a similar Hidden Markov Modeling software package.