

FAnalyze v0.1

User's Manual

Dan Valente
Mitra Lab
Cold Spring Harbor Laboratory

August 14, 2008

1 Introduction

FAnalyze is a suite of functions written in Matlab to aid in the analysis of animal behavior in the Open Field. It was written and tested primarily for fly exploratory behavior in a circular arena, and a description and illustration of the measures calculated by FAnalyze can be found in Ref. [1]. FAnalyze assumes that the user has tracked the (x,y) location of the animal to obtain the trajectory that the animal took over the course of the experiment. FAnalyze smooths the data, calculates the animal's velocity, and allows one to segment space and speed based on hard thresholds. With this inaugural version, FAnalyze only allows for segmentation into circular zones (it is assumed that the Open Field arena is circular). FAnalyze then allows the user to calculate and display probability distributions of the relevant variables to obtain a quantitative phenotypic characterization of the exploratory behavior. Please note that these calculations — the probability distributions — are the focus of the program.

Although it has been rigorously tested, being that this is version 0.1, please be aware that multiple bugs may still exist in FAnalyze. If you find them, feel free to fix them (if you are Matlab savvy), or send an email to the author. To use FAnalyze, a basic working knowledge of Matlab is assumed.

2 Placing FAnalyze in the Matlab path

In order to use FAnalyze, the FAnalyze folder must first be placed in the Matlab path.

1. Open Matlab.
2. Select **File >> Set Path...** from the toolbar. The Set Path dialogue will open up.
3. Click on the **Add Folder...** button.

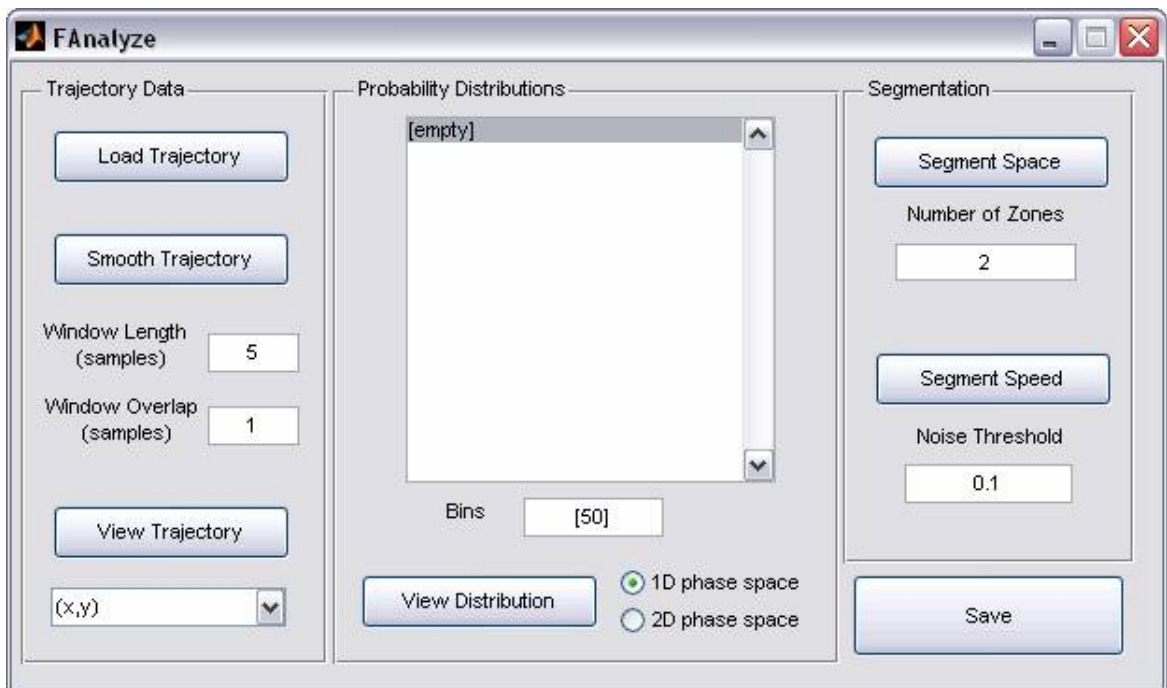


Figure 1: The FAnalyze GUI

4. Find the FAnalyze directory and select the **functions** folder. Click **OK**. The path and name of the FAnalyze functions folder should show up in the Matlab search path window.
5. Click **Save**, and then click **Close**. FAnalyze is now ready to use. To test if it is installed correctly, `help FAnalyze` in the Matlab command window. If an error is obtained upon this command, the folder was not correctly installed, so try to re-install. If the installation worked, you will be able to open the program or use any of the functions regardless of what directory you are currently in.

3 The FAnalyze GUI

The FAnalyze graphical user interface (GUI) was written to facilitate use of the analysis functions for quick data exploration. It is not intended for large scale data analysis—the individual FAnalyze Matlab functions are best for this—although the user may find the GUI useful in high-throughput studies. The interface is shown in Fig. 1 and is divided into three general sections: Trajectory Data, Probability Distributions, and Segmentation. The functionality of these sections is described below.

Loading a Trajectory

To begin, the user must load a trajectory file for analysis by clicking the **Load Trajectory** button. FAnalyze assumes that the data is contained in a `.mat` file. Within that file *must* be variables labeled `x`, `y`, and `t` describing the spatial coordinates and time of every point in the trajectory. FAnalyze permits analysis of only one file per session. Once the file is loaded, a message is displayed in the Command Window informing the user of the chosen file's name.

Instructions for Smoothing

Due to jitter in the object's location caused by non-translational object movements and artifacts of the tracking method, it is typically good practice to smooth the resulting trajectory. The smoothing is executed using the function `runline` from the Chronux neural data analysis software package (<http://www.chronux.org>), which performs a local linear regression on the data¹.

1. Select the length of the running window (in samples) to be used in the smoothing process.
2. Select the step-size (in samples) that the window will take as it moves across the data (also known as the Window Overlap.) Enter these numbers into the appropriate boxes.
3. Click the **Smooth Trajectory** button. In the Command Window, a message informs the user that the smoothing process is underway². When smoothing has finished, a message will be displayed declaring a successful completion of the smoothing algorithm.

Because smoothing can significantly eliminate fluctuations in the data, one is advised to investigate the effects of changing the smoothing parameters on the resulting trajectory. This will allow you to decide what amount of fluctuations are physically relevant in your videos. For instance, if the data is smoothed too drastically, a short but visible stop of the object can be entirely smoothed into an apparent movement!

Viewing the Trajectory

Once the data is smoothed, the user is able to view the resulting trajectory. Simply select the plot of interest from the pull-down menu in the Trajectory Data section and click **View Trajectory**. The user has access to view the following data (note that a circular arena is assumed, hence the availability of data in polar coordinates):

¹`runline` is included with FAnalyze, so having Chronux is not a pre-requisite for use of this program.

²Smoothing may take a long time, depending on the length of the trajectory.

<code>(x,y)</code>	The complete position trajectory
<code>x</code>	The x-coordinate as a function of time
<code>y</code>	The y-coordinate as a function of time
<code>r</code>	The radial coordinate as a function of time
<code>theta</code>	The angular coordinate as a function of time
<code>vx</code>	The x-velocity as a function of time
<code>vy</code>	The y-velocity as a function of time
<code>v</code>	The speed as a function of time
<code>vtheta</code>	The direction of the velocity vector as a function of time

Creating and Viewing Probability Distributions

The philosophy behind FAnalyze is as follows³: We regard the trajectory as a stochastic process $\mathbf{x}(t)$. This process would be fully characterized if the joint distributions $P(\mathbf{x}(t_1), \mathbf{x}(t_2), \dots, \mathbf{x}(t_n))$ were specified for all choices of time points (t_1, t_2, \dots, t_n) . Unfortunately, the full distribution, $P(\mathbf{x}(t_1), \mathbf{x}(t_2), \dots, \mathbf{x}(t_n))$, is difficult (if not impossible) to measure. However, by examining joint distributions of position and velocity along with distributions of path curvature, reorientation angle, and event durations, we can obtain a convenient summary of the animal’s behavior in the arena and its interaction with the environment.

The distributions are estimated using histograms of the data, so it is recommended that the organism be studied for a long period of time for “clean-looking” distributions (a “good” length of time will depend on the activity of the animal and the frame rate that the video was taken at). These probability histograms are calculated with the functions `ProbDist1D`, `ProbDist2D`, and `JointDist`.

When examining histogram estimates of probability distributions, one needs to exercise care about phase space factors in order to obtain accurate estimates. For example, if the animal is moving in two dimensions, the probability density for the speed v along with the phase space factors is given by $p(v)v dv d\theta$ (where θ is polar angle of the point (v_x, v_y) in velocity space). Therefore, binning data in bins of size $\Delta v \Delta \theta$ would yield an estimate for $p(v)v$. When this is the case, we eliminate the need to divide by v (which could be an unstable calculation for small v) by binning in v^2 , since $p(v)v dv d\theta \sim p(v^2/2)d(v^2/2)d\theta$. For one-dimensional motion, such as movement along the arena boundary, there are no phase space factors and it is sufficient to bin the data in v . FAnalyze allows the user to select whether to calculate the distributions assuming a 1D or 2D phase space. For 2D phase space calculations, the user should take note of the non-constant bin widths of these histograms.

Therefore, as soon as the data is smoothed (as well as when the data is segmented), the Probability Distributions list box will become populated with the variables that are available for analysis. The naming convention for variables in the list box is described in Appendix A. To calculate and view the probability distributions of interest, proceed as follows:

³This is adapted from Ref. [1]

1. For a single variable marginal distribution, highlight the variable of interest by clicking on it. For a joint distribution of two variables, select the first variable of interest, hold down the CTRL button on the keyboard, and select the second variable of interest.
2. Enter the **Bins** to use for the calculation. This field will accept any bin description that the Matlab `hist` or `hist3` commands accept. See the help files of those functions for details, and make sure that brackets commas and other necessary punctuation are used. Also note that no other options available to `hist` or `hist3` are available in the FAnalyze functions with this release (v0.1).

As an example, if you wish to calculate a joint distribution having 100 bins in the first variable and 150 in the second, you would enter: `[100 150]`. If, instead you wanted bin centers from 0 to 10 in steps of 0.1 for the first variable, and bin centers from 2 to 4 in steps of 0.3 for the second you would enter: `{ [0:0.1:10] [2:0.3:4] }`. (Note the curly brackets).

3. Select whether the variable of interest exists in a one or two-dimensional phase space (see above).
4. Click the **View Distribution** button. The distribution will be calculated and a plot of the distribution will be displayed.

Every calculation that is performed is held in memory until FAnalyze is closed. At any point in time, you may click the Save button to save your calculations (structure of data is described below). Unfortunately, at this point, the user cannot access the calculated data from the command window until after the data is saved and reloaded.

Segmenting Space

Often, an examination of the joint distributions $p(x, v)$ or $p(r, \theta)$ will show that the animal has a spatial preference for some part of the arena. FAnalyze allows the user to segment the arena into any number of circular spatial “zones.” Please note that only concentric circular zones are allowed (or, rather, toroidal zones). To segment space:

1. Enter the **Number of Zones** that you wish to segment the arena into.
2. Click the **Segment Space** button. A window will pop up asking the user to input relevant information.
3. Input the location of the threshold defining the boundary between zones 1 and 2, in terms of the radial distance from center. Enter names for these zones. Click **OK**.
4. If more than two zones are requested, another window will pop up asking for similar information. Make sure that the first zone name in this window is the same as the second name in the last window and that the second threshold

is larger than the first; otherwise, you will get an error (zone 2 must have a consistent name, and the second threshold must be further than the first threshold).

5. Repeat this for all the zones you requested.
6. Once the segmentation is finished, a message will be displayed in the command window, and the Probability Distributions list box will be populated with variables available for analysis.

Segmenting Speed

Similar to the spatial distributions, when examining the speed distribution $p(v)$, the user may find that the distribution appears to be a mixture of a few different types of motion. Because of this, investigators often find it useful to segment the speed into distinct modes of motion. In FAnalyze, speed segmentation is performed almost exactly as the space segmentation. To segment speed:

1. Enter the **Noise Threshold** seen in your data. The noise threshold is the lowest speed that you can accurately resolve. It can be obtained by examining the speed vs. time plot and noting the maximum speed attained in regions where the animal is visibly stationary. Velocity and speed points below this threshold are assigned a value of 0.
2. Click the **Segment Speed** button. A window will pop up asking the user to input relevant information *for each spatial zone*.
3. Input the location of the threshold defining the boundary between segments 1 and 2, in terms of the absolute speed. Enter names for these segments. Click **OK**.
4. If more than two segments are requested, another window will pop up asking for similar information. Make sure that the first segment name in this window is the same as the second name in the last window and that the second threshold is larger than the first; otherwise, you will get an error (segment 2 must have a consistent name, and the second threshold must be further than the first threshold).
5. Repeat this for all the segments and zones you requested.
6. FAnalyze now segments the data according to the user's requests, as well as calculating where the animal has stopped (points below the noise threshold). Stops are considered a "segment."
7. Once the segmentation is finished, a message will be displayed in the command window, and the Probability Distributions list box will be populated with variables available for analysis.

Saving the Data

To save the data from the session as a `.mat` file, click on the **Save** button in the lower right-hand corner of FAnalyze. The user is asked to choose a location and a filename in which to save. The data is saved as two cells, `traj` and `P`.

`traj` is a $1 \times N$ cell, where N is the number of zones in the arena. The i^{th} cell contains a structure with the trajectory information from that zone. Within that structure, each variable is a cell itself containing structures for each speed segment. For example, data from the second zone is accessed by typing `traj{2}` and is organized as follows:

```
traj{2} =
  zone_label: {'CZ'}
  seg_label: {'all' 'stops' 'NZS' 'FSS'}
           t: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
           x: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
           y: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
           r: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  theta: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  vx: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  vy: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  v: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  vtheta: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  tau: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  kappa: {[1x1 struct] [1x1 struct] [1x1 struct] [1x1 struct]}
  beta: {[1x1 struct]}
```

The labels are fairly self explanatory. Note that `beta` can only be calculated if all the points in the trajectory are considered.

The j^{th} entry of each variable's cell contains a structure with a single field. This field is called `data`. So, if one is interested in extracting the x-position of the organism in zone 'CZ' (the second zone), while the organism walked in speed segments labeled by 'FSS' (the fourth segment), one would type

```
traj{2}.x{4}.data
```

The probability histograms are saved in the `P` cell. `P` is a $1 \times M$ cell, where M is the number of times the View Distributions button was pressed during the session. Each entry of the cell contains a structure that is organized as follows:

```
P{1} =
  label: 'x_Full Arena_all'
  phase_opt: 'phase1D'
  data: [1x50 double]
  bins: [1x50 double]
```

The field `label` is the name of the variable from which the probability distribution was calculated. `phase_opt` denotes whether the user chose to calculate the distribution assuming a 1D or 2D phase space. `data` contains the bin-by-bin data from the calculated histogram, and `bins` contains the bin centers. If an error occurs, the entry is completely empty.

Admittedly, this seems complicated, but the author felt it was a decent way to organize the data file.

4 Known Problems

There is only one known problem with FAnalyze at this point. Please be aware that you cannot choose to segment speed and *then* segment space. You must segment space first and *then* segment speed. You can, however, select to only segment speed or only segment space.

5 Concluding Comments

For those who wish to use the functions from the Matlab command line, complete descriptions of their use and workings can be found in the Matlab help files; simply type `help function_name`. Describing them in detail here would be superfluous. The scripts are also commented, and as such, they should be relatively easy to follow. Suggestions for improvements to the algorithms, the GUI or the coding style are highly encouraged! Comments on the ease of use of the GUI and functions are also important for refining this program. Since this is v0.1, FAnalyze needs quite a bit of testing in order to find all of the bugs (pun intended). Until then, please check and double-check any results that you obtain from this program, and make sure that they make sense!

Enjoy!

6 Appendix A: Variable Naming Convention

There are eleven variables that are available for analysis in FAnalyze. They are:

x	The x-coordinate as a function of time
y	The y-coordinate as a function of time
r	The radial coordinate as a function of time
theta	The angular coordinate as a function of time
vx	The x-velocity as a function of time
vy	The y-velocity as a function of time
v	The speed as a function of time
vtheta	The direction of the velocity vector as a function of time
tau	Duration of speed segments
kappa	Curvature of the path
beta	Reorientation angle

In the list box, these variable names are followed by the zone name and the speed segment name as given by the user. For example, for a zone named ‘RZ’ and a speed segment within that zone named ‘FSS’, the speed would appear as v_RZ_FSS. For the full arena, the label is automatically called (appropriately) ‘Full Arena’. If all the velocity points are included the speed segment label is ‘all’. Therefore, after smoothing and before any segmentation, the variables in the list contain the label ‘_Full_Arena_all’.

References

- [1] Valente, D., Golani, I., and P.P. Mitra, “Analysis of the trajectory of *Drosophila melanogaster* in a circular open field arena.” PLoS ONE 2(10), e1083 doi:10.1371/journal.pone.0001083, (2007)
- [2] Valente, D., Wang H., Andrews P., Saar S., Tchernichovski O., Benjanimi, Y., Golani I. and P.P. Mitra, “Characterization of animal behavior through the use of audio and video signal processing.” IEEE Multimedia, 14 (2), 32-41, (2007)