DISC User Manual

April 2019

v1.1.0

Welcome to DISC!

DISC (divisive segmentation and clustering) is an open source[1] `MATLAB` package for time series idealization (i.e. identifying significant states and transitions). We enhance change-point detection algorithms with divisive clustering to accelerate the typical speed of time-trajectory analysis by orders of magnitude while retaining or improving accuracy. We have designed the DISC framework to be a flexible approach, with applications primarily in single-molecule data, such as smFRET and force spectroscopy.

Please cite "White, D.S., Goldschen-Ohm, M.P., Goldsmith, R.H., Chanda, B.C., "High-Throughput Single-Molecule Analysis via Divisive Segmentation and Clustering" if using this software[2].

To ensure you have the most up-to-date version of DISC (of which this manual should be reflective of) please see https://github.com/ChandaLab/DISC.

Any questions, comments, or potential bugs can be reported to David S. White at dwhite7@wisc.edu.

---

[1] See https://www.gnu.org/licenses/gpl-3.0.en.html

[2] White, D., Goldschen-Ohm, M., Goldsmith, R. & Chanda, B. High-Throughput Single-Molecule Analysis via Divisive Segmentation and Clustering. *bioRxiv*, 603761 (2019).

Contents

1. Getting Started

1.1 Installing DISC

DISC is written entirely in `MATLAB` (2017B and above) and does not require installation beyond obtaining `MATLAB`. The `MATLAB Statistics and Machine Learning Toolbox` is required.

1.2 Sample Data

For this guide, we will be using the simulated sample data provided in `DISC/sample_data/sample_data.mat.`

2. Running DISCO

DISCO is the name we have given the graphical user interface (GUI) for running the DISC algorithm.

To open DISCO:

1. Open `MATLAB`
2. Add DISC folder to your file path
3. Type `DISCO` into the Command Window

or right click → run `DISCO.m` in MATLAB's file viewer.

DISC can be run outside of the GUI using `runDISC.m`. See section **8**.

3. Loading Data

Once DISCO is executed, it will open a window with a prompt to load a data set. Navigate to `DISC/sample_data` and load `sample_data.mat`. See **7.1** for data formats.

Once a data set is loaded, the GUI will initialize and display the first trajectory (**Figure 1**).

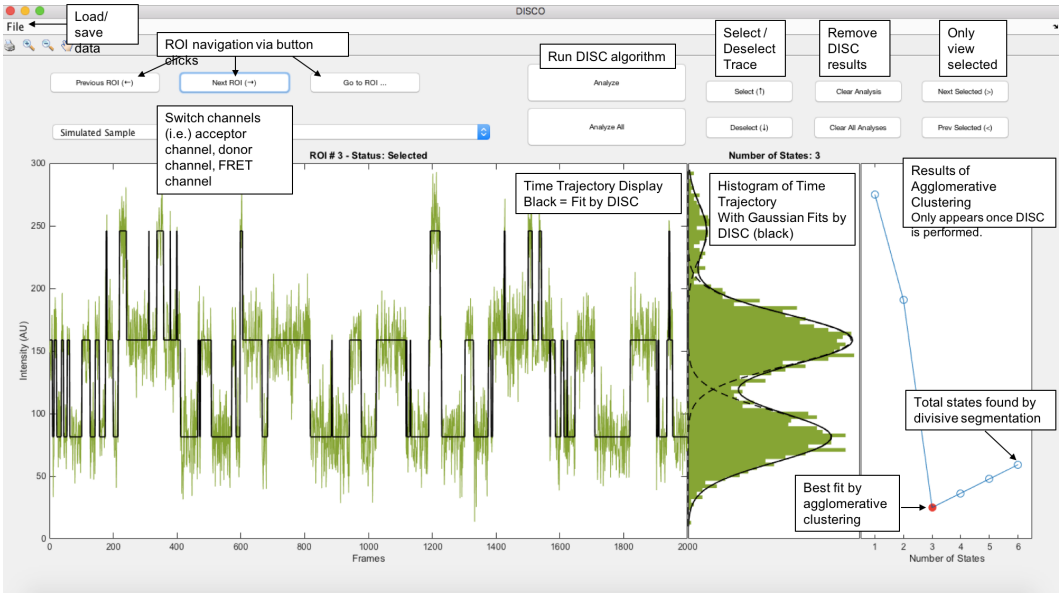

Figure 1: DISC GUI Overview

If DISC is already open and you wish to load a different data set:
File → Load Data

4. Navigation in DISCO

Trajectories can be navigated by either arrow clicks in the GUI (i.e. "Next ROI (→)) or by the keyboard.

Right arrow	Navigate to next trajectory
Left arrow	Navigate to previous trajectory
Up arrow	Select trajectory
Down arrow	Deselect trajectory
Period / >	Navigate to next 'selected' trajectory
Comma / <	Navigate to previous ''selected trajectory

Adding or modifying existing keys can be done in
`/src_GUI/roiViewerGUI.m` via `figure1_WindowKeyPressFcn`.

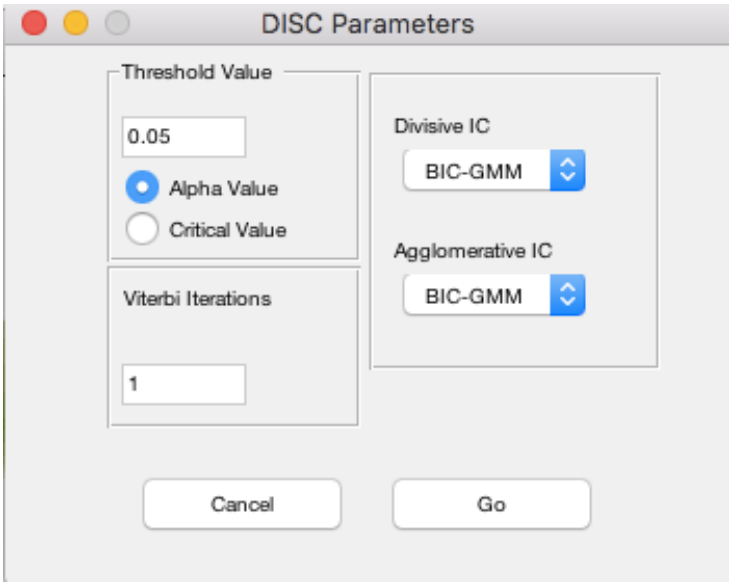

Figure 2: DISC Parameters window

5. Analyzing Trajectories with DISCO

5.1 Single Trajectory Analysis

Trajectories can be analyzed either in or individually at once for a given channel. Let's start by analyzing the first trajectory.

1. Click "Analyze"
2. Input the parameters for the DISC algorithm or used the suggested default values
3. Click "Go"

The trajectory will be analyzed. Black lines appear on the time-trajectory and the histogram plot showing the fit by DISC. In addition, a plot to the right will appear showing the total number of states found and the overall best number of states.

Note: Deciding which parameters to use will depend on the sort of data you are analyzing. We will provide better benchmarks as DISC gains more use. For now, it is valuable to try different parameters and see what appears to work best.

5.2 Analyzing an Entire Data Set

Now let's analyze all 100 traces:

1.Click "Analyze All"
2. Input the parameters for the DISC algorithm or used the suggested default values
3. Click "Go"

A wait-bar will pop up during this process. Once it closes, the analysis is complete.

Note: This is our suggested method of use for DISC, as this scheme analyzes all trajectories with the statistical assumptions, rather than changing confidence intervals or objective functions per trace to get a desired answer.

5.3 Clear Analysis

The results of DISC can be removed either per trace using "Clear Analysis" or for every trajectory using "Clear all Analyses".

5.4. Trace Selection / Deselection

Trajectories can be selected by pressing the up arrow or click the "select" button in the GUI. This will add a field to the rois structure (see **7.1**) to be indexed by further analysis by DISC. For example, ROI #24 has a low signal-to-noise ratio and you may not be confident in the analysis results, unlike ROI #22 which has a very high signal-to-noise. Therefore, you may want to Select ROI #24 and Unselect ROI #22 for further analysis.

By default, all traces are "Unselected" (data.rois(1,1).status == 0). Once traces are selected, you can navigated between only the selected traces using "," or "." on the keyboard or by clicking "Next Selected (>)" or "Prev Selected (<)" in the GUI.

6. Saving Data

Simply click File → Save Data. See **7.2** for information on DISC output.

7. Data Format

7.1 Input Data Format

Notation:
"rois" = regions of interest.
N = number of data points

Data is formatted in data structures with the following required fields:

data	Data structure to describe the entire data set
data.rois	Data structure for a specific region of interest (roi)
data.names	Cell to name the channels (strings)
data.rois.time_series	Array[N,1] of observed time series for the roi to be analyzed by DISC

For example, in our simulated data there are 100 trajectories each with 2000 data points. Therefore,

size(data.rois) = [100 2]
size(data.rois(1,1).time_series) = [2000 1]

where data.rois(1,1) indexes the first trajectory of the first channel

In our simulated data, we have 2 channels, shown by:

data.names{1} = 'Simulated Sampled'
data.names{2} = 'Simulated Sampled with True Fits'

This allows different trajectories to be collected from the same roi. For example, in smFRET experiments, it is common to collect the emission time-trajectories of the acceptor, the donor, and the calculated FRET efficiency. If data.names is not provided, default channel names will be given.

Any other fields in either *data* or *rois* is acceptable and will not have an effect on DISC. We find this to be a useful format for including information about our data at various levels. For example, in our sample data, we can retain information about how our data was simulated:

```
>> data =

  struct with fields:

              rois: [100×2 struct]
          fcAMP_uM: 1
          Q_matrix: [4×4 double]
    emission_states: [1 1 2 2]
   bound_intensites: [1×1 prob.LognormalDistribution]
    bound_variation: [1×1 prob.ExponentialDistribution]
         duration_s: 200
       frame_rate_s: 0.1000
              names: {'Simulated Sample'  'Simulated Sample with True Fits'}
```

At the data.rois level, we can include information such as DISC fits, signal to noise ratio (SNR), selection status, etc...

7.2 Analysis Output

`runDISC.m` (which DISCO calls) returns the structure `disc_fit` for each roi in data.rois.

For example:

```
>> data.rois(1,1).disc_fit

ans =

  struct with fields:

    components: [2×3 double]
         ideal: [2000×1 double]
         class: [2000×1 double]
      n_states: 2
       metrics: [2×1 double]
    all_ideal: [2000×2 double]
   parameters: [1×1 struct]
```

where:

components	[weight, mean, standard deviation] of each identified state
ideal	time_series fit described by the mean value of each state
class	time_ series fit described by the integer of unique states
n_states	number of states identified
metrics	all computed information criterion (i.e. BIC) values from agglomerative clustering.
all_ideal	all possible state sequences from the initial results of divisive segmentation and grouped by agglomerative clustering: all_ideal(:,1) = 1 state fit; all_ideal(:,2) = 2 state fit, etc...
parameters	all input values used for analysis in DISC (**Figure 2**)

8. Running DISC outside of the GUI

DISC can easily be called outside of the GUI using the `runDISC.m` function. This will be a faster since MATLAB uses quite a bit of memory to render the GUI.

To perform this analysis with default DISC values across a data set, one can simply write:

```
for i = 1:length(data.rois)
    data.rois(i,1).disc_fit = ...
        runDISC(data.rois(i,1).time_series);
end
```

Default values easily modifiable in `setDefault.m`.

Alternatively, one can input their own values outside of `setDefault.m` via:

```
disc_input = struct;
disc_input.input_type = 'alpha_value';
disc_input.input_value 0.05;
disc_input.divisive = 'BIC_GMM';
disc_input.agglomerative = 'BIC_GMM';
disc_input.viterbi = 1;
disc_input.return_k = 0;

for i = 1:length(data.rois)
    data.rois(i,1).disc_fit = ...
        runDISC(data.rois(i,1).time_series, disc_input);
end
```

where:

input_type	Either 'alpha_value' or 'critical_value' for use in change-point detection
Input_value	Value corresponding to input_type. i.e. 0.05 = 95% confidence interval when used with 'alpha_value'
divisive	Information criterion/ objective function for identifying states during the divisive phase (see `computeIC.m` for a list of available options). This value determines the max number of states possible for agglomerative clustering.
agglomerative	Information criterion/ objective function for identifying states during the agglomerative phase (see `computeIC.m` for a list of available options). This value determines the final number of states returned for fitting by Viterbi
viterbi	Iterations of the Viterbi algorithm to identify the most likely hidden state sequence. We recommend 1 or 2 iterations. 0 = do no run Viterbi.
return_k	Force the number of states you want `runDISC.m` to return. If return_k > # states identified, then the # of states identified will be returned. *Note: This is not the suggested use of DISC and is not an option in DISCO currently.*

9. Tutorial

1. Click "File" → "Load Data"
2. Select data set you would like to load
3. Click "Analyze All"
4. Enter the analysis parameters for DISC you would like to use.
6. Click "Go"
7. Check the fits of the data and either "Select" or "Unselect" traces as desired.
8. If more than one channel is present, switch channels with the drop-down box.
9. Repeat 3-8 as needed.
10. Click "File" → "Save Data"
11. Save the data in the location with the name you want.
12. DONE![3]

[3] Post-processing of the trajectories will still be required (i.e dwell time analysis or transition-density plots). Currently, DISCO only idealizes data although functionalities may be added later.