Drift Markov Fitter Help File

Important Files:

DriftMarkov2.jar (java file that computes)

DriftMarkovFitter.ipf (igor procedure)

DriftMarkov(Folder)

GetRidOfUseWave.jar

Instructions:

1. Place the MarkovFitter Folder in C:\ so that the path to the folder is C:\DriftMarkov
2. Ensure that this folder contains the latest version of DriftMarkov2.jar and GetRidOfUseWave.jar
3. Take Igor Procedure DriftMarkovFitter and place in Igor procedures folder and then open in Igor.
4. Use the procedure in Igor.

How to Use Procedure:

The java jar file takes in theory as many inputs as you want but this particular case only uses nine and needs at least 2. The Igor procedure takes care of passing the arguments to the jar file so you only have to give them to the Igor procedure. The first arguments that the file takes is a txt file containing a wave. Enter a wave as the first argument in DriftMarkovFitter and the procedure will take care of turning it into a txt file. The next input that the jar file takes is a state count which is basically the number of potential states that the input data can take. Just enter an integer into the second input into DriftMarkovFitter. The third input to the jar file is a column number but that has been hardcoded to zero because that it how the wave file created is structured. The next few arguments that the jar file takes are mode count, time step, drift bound, sigma bound, transition bound, and iteration count. Enter the values you want to use in that order into the input for DriftMarkovFitter. Please be sure to have state count, mode count, and iteration count as integers. The time it takes for the program to run is affected by the inputs the most important factors being mode count, iteration count, and the size of the wave. **Please ensure that the units of your points and the values you enter are the same (i.e. nm vs m).** This affects driftBound, sigmaBound, and transitionBound.

**Function DriftMarkovFitter( UseWave, stateCount, modeCount, timeStep, driftBound, sigmaBound, transitionBound, iterationCount)**