Model-Checking w Simulations AND Down-sampling and High-frequency error

Ian Seim

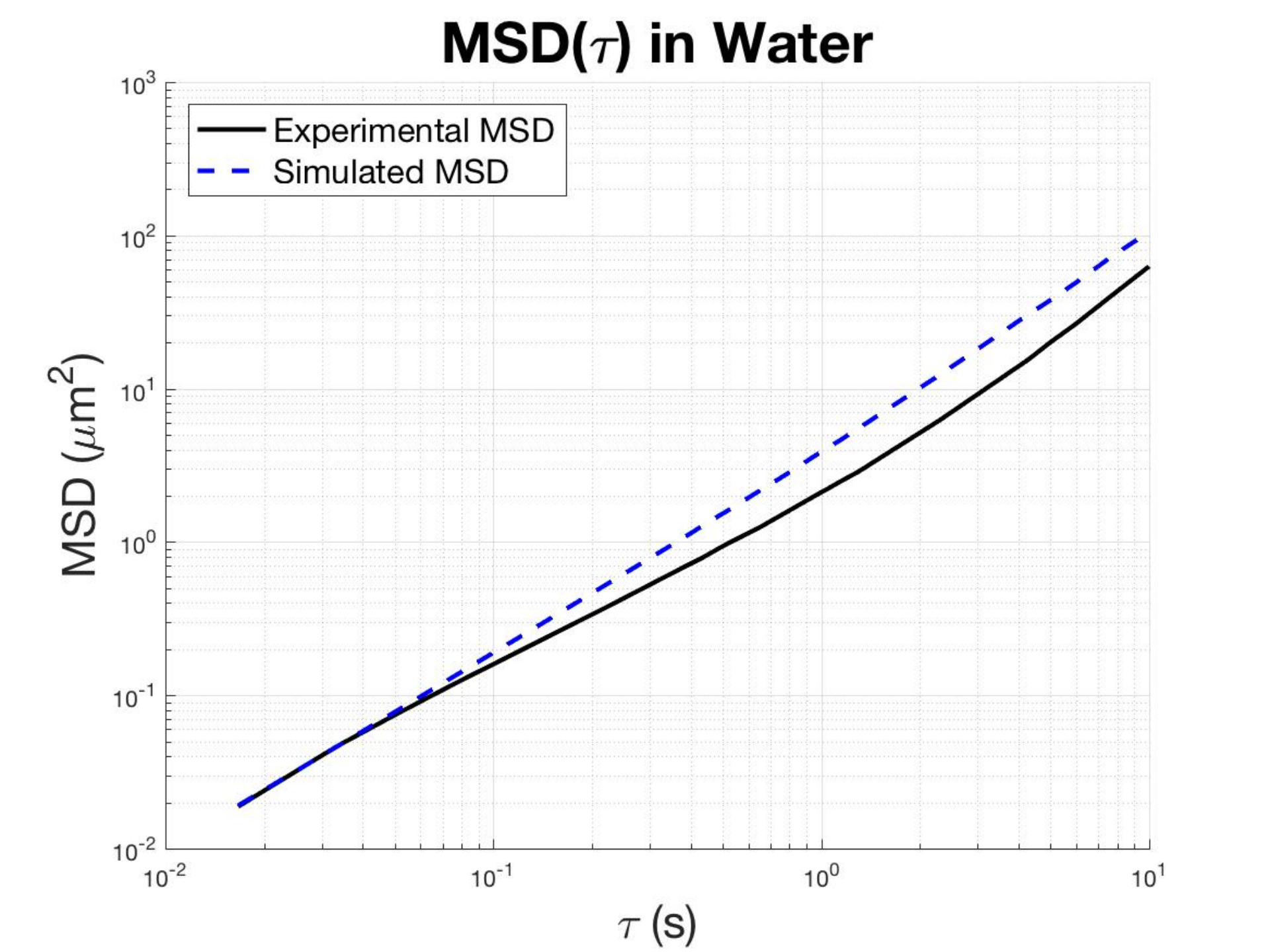
2/23/17

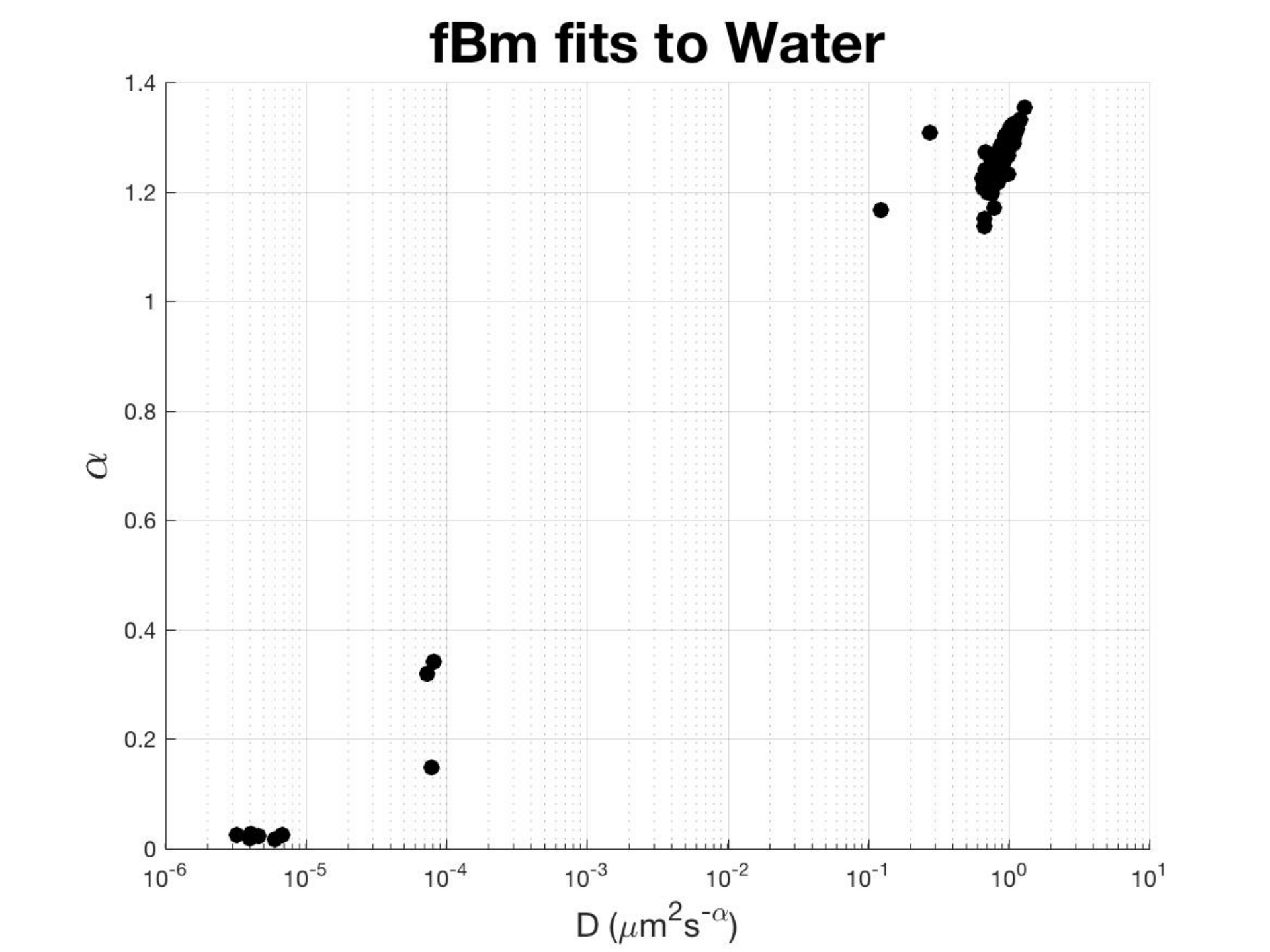
Simulating fits to Experimental Data

- For each path, we fit α, D, and a linear drift velocity, μ_x and μ_y, in each coordinate, respectively, using fbmMLE.m
- Using these parameters, we can simulate an experimental dataset, path-by-path, and compare MSD statistics of the experimental dataset to the simulated one(s) to validate our model fits
- Use fBmXY_HD.m to simulate the fBm component, inputting α and D from fbmMLE.m, and the observation time and fps for the given experiment/path
- Use the "Beta" output of fbmMLE.m to simulate linear drift in each coordinate (more on next slide)

Simulating paths (cont.)

- If our experimental path has 1800 data points (30 seconds at 60fps, for example), we can simulate drift as follows:
  - driftx = [0:1799]'./60 * Beta(1);
  - drifty = [0:1799]'./60 * Beta(2);
- We then simply add these vectors to the x and y components of the fBm path output by fBmXY_HD.m to complete the simulation of a single path


MSD(τ) in Water
Experimental MSD
Simulated MSD
MSD (μm²)
τ (s)
10³
10²
10¹
10⁰
10⁻¹
10⁻²
10⁻²
10⁻¹
10⁰
10¹

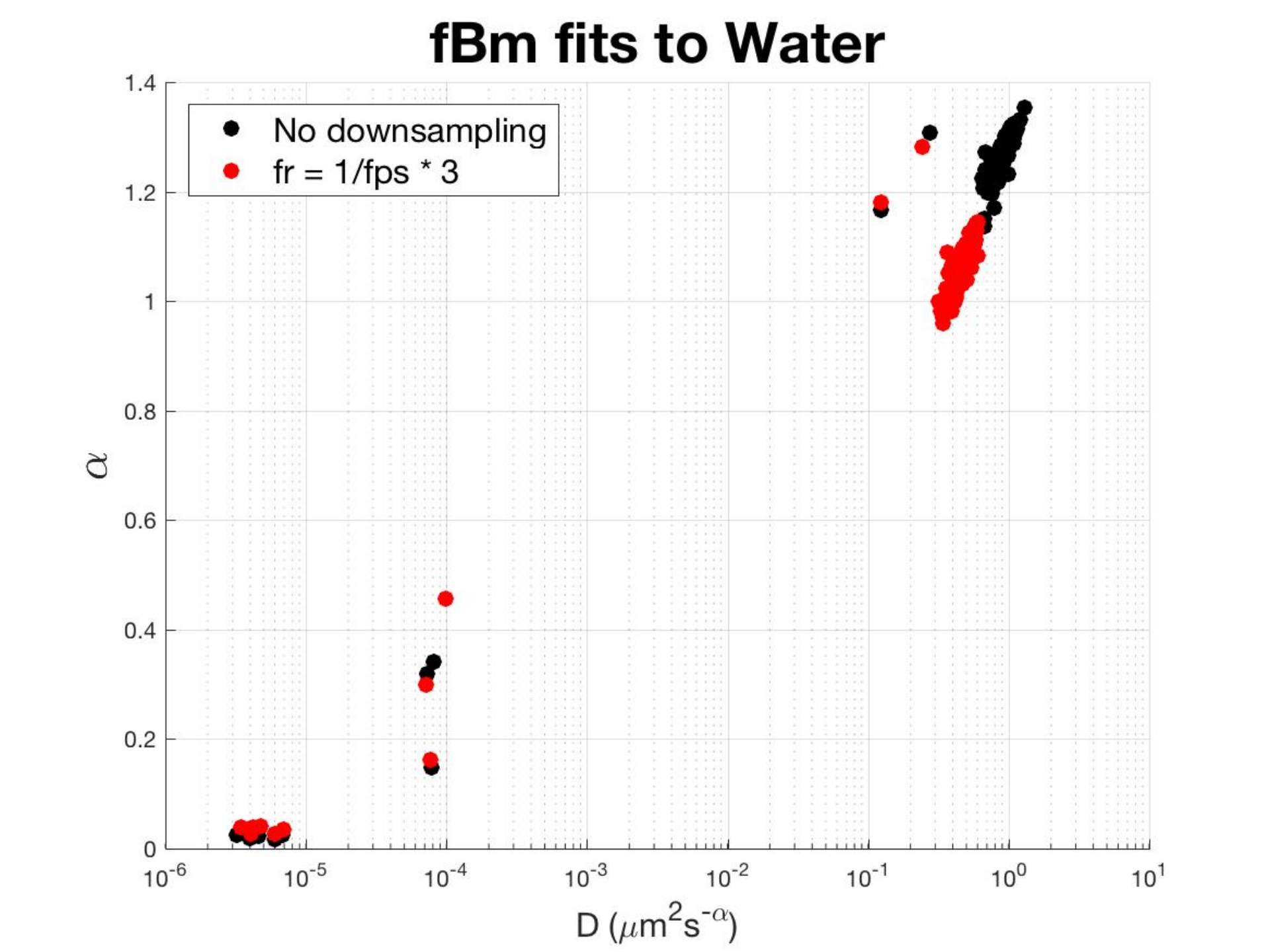

fBm fits to Water
1.4
1.2
1
0.8
0.6
0.4
0.2
0
α
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
D (μm^2s^-α)

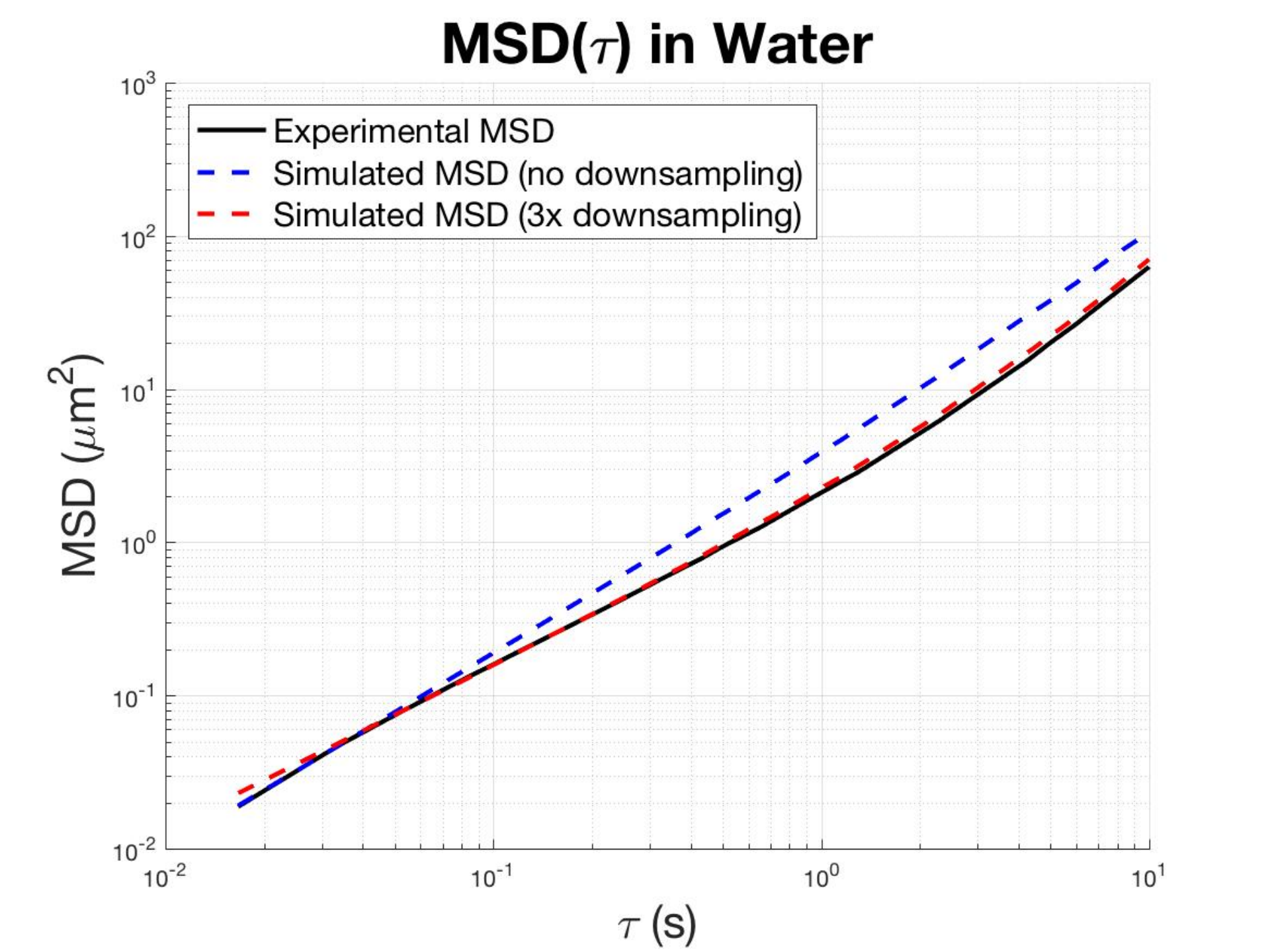
High-Frequency Error

- It turns out that in some experimental data, there is an error found only in the highest frequencies (shortest lagtimes), that falsely increases the correlations between increments
- We think this is due to dynamic tracking error in the camera
- Our collaborators (Martin Lysy and Yun Ling) having created a model for this camera error and will have it incorporated in the MLE fits soon, but in the meantime…

Down-sampling

- By down-sampling the data, we can bypass the noise in the highest frequencies of the path data
- Do this by inputting every other point of the path data to fbmMLE, and inputting 2x the original frame rate; or every third point and inputting 3x the original frame rate. Looks like:
  - mle = fbmMLE(path.Positions(1:2:end, :), 2/60, 2/60) (if the frame rate of the camera is 1/60)
  - In general… mle = fbmMLE(path.Positions(1:i:end, :), i/60, i/60)


fBm fits to Water
No downsampling
fr = 1/fps * 3
1.4
1.2
1
0.8
0.6
0.4
0.2
0
α
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
D (μm^2s^-α)


MSD(τ) in Water
Experimental MSD
Simulated MSD (no downsampling)
Simulated MSD (3x downsampling)
MSD (μm²)
τ (s)
10³
10²
10¹
10⁰
10⁻¹
10⁻²
10⁻²
10⁻¹
10⁰
10¹