High-throughput Experiments in Passive Microrheology

Overview

Microparticle rheology techniques measure the modulus and viscosity of a medium by monitoring the passive or active transport of particles through that medium. Over macroscale techniques such as cone and plate (CAP), microrheology offers the advantages of physiologically relevant probe sizes, a smaller sample size, and greater sensitivity to specimen heterogeneity. In this workshop exercise we include a general description of the physics involved when a spherical particle is pulled through a viscous medium. Using this theory and a high-throughput microscopy system named Panoptes, we will perform rheological measurements on biologically relevant materials. Figure 1 provides a summary of the lab and and outline of this document.

Why microrheology?

Microrheology is a type of rheological measurement that makes use of particle tracking methods for the observation and quantification of rheological phenomena in various materials. The technique requires that the experimenter observe, via microscopy, the passive or driven motion of probes within the material of interest, then use the properties of the microprobe path to determine properties of the material of interest. And therein lie two of the major challenges of microrheology: (1) The technique is limited to materials which are at least somewhat optically clear, and (2) the technique is computationally intense. Additionally, because the forces applied to microscale probes are typically significantly smaller than the forces which can be applied to bulk volumes by cone and plate tools, etc., microrheology is often ill-suited for evaluating very stiff or viscous materials (> 1 kPa, > 10 Pa s).

Yet, microrheology makes possible some measurements which are either not possible or extremely difficult using bulk measurement techniques. Perhaps most notably, the small size of the probes used in microrheology make it the principal technique for

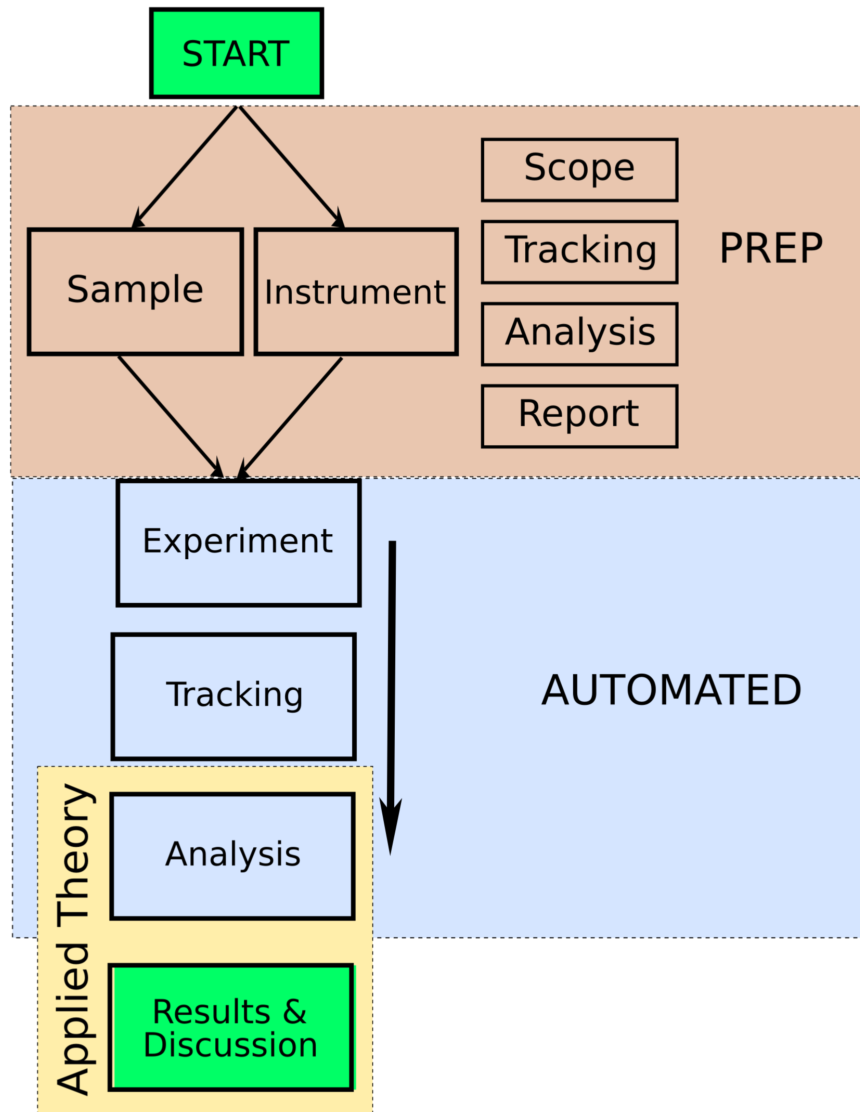

Figure 1: Flowchart of Experiment

studying materials with micrometer and sub-micrometer heterogeneity in addition to multiphase solutions. Bulk methods inherently average over all rheological processes occurring within a sample. Specifically, for highly heterogeneous biological samples such as mucus, and extracellular matrix, crucial system details are lost when bulk averaging methods are used. Microrheology also has the advantage of easily probing samples with low viscosities and low elasticities. This is possible because very slight deviations from Brownian motion can be detected using the statistics of diffusion paths as determined from carefully performed microrheology experiments and analysis. Additionally, microrheology requires very small volumes (on the order of 10 μL). This makes it the only realistic technique for studying materials which are either very expensive or, in many biological cases, very difficult to acquire or store in large ($>$ 1 mL) quantities. Finally, basic microrheology equipment is inexpensive and is widely deployed in standard biology, physiology, and biological physics laboratories.

Why high-throughput?

High throughput instrumentation measures many specimens in a single run, typically using software automation for data collection and analysis. Why is this so helpful? Biological materials (and many novel synthetic materials as well) are sensitive to a wide range of parameters. For materials like mucus, there is an interest in performing screens against libraries of small molecules – for example, to test the effect of potential drugs on mucus rheology. Similarly, cell culture experiments can use small interference RNA (siRNA) libraries that target thousands of genes to explore their effects on a particular cellular function. These types of broad searches are simply impossible without highly automated systems.

Theory

Microrheological measurements successfully subdivide into passive and active techniques. Passive microrheology measurements make use of the stochastic, thermal displacements of particles provided by kT, the energy that serves as the background thermal fluctuations of molecular motion (Rubinstein and Colby, 2003). Active measurements use a deterministic, directed, and controlled force of known magnitude that drives the particles through the medium in a manner that belies the medium's viscoelastic properties (Waigh, 2005).

In 1827 Robert Brown first documented the random motion of micron-sized particles in which he observed pollen grains "very evidently in motion ... [arising] neither from currents in the fluid, nor from its gradual evaporation" (Brown, 1828). Caused by a constant bombardment of solvent molecules, "Brownian motion" was not well under-

stood and characterized fundamentally until Einstein began to tackle it during his *Annus Mirabilis* in 1905 (Stachel and Raman, 1990; Einstein, 1905).

Several experimental methods in the literature take advantage of particle diffusion to measure microrheology, some of which include Dynamic Light Scattering (DLS) (Maret and Wolf, 1987), Diffusing Wave Spectroscopy (DWS) (Pine et al., 1988; Mason and Weitz, 1995), Single Particle Tracking (SPT) (Mason et al., 1997), Multiple Particle Tracking (MPT) (Apgar et al., 2000), and two-particle microrheology (TPM) (Crocker et al., 2000).

Developed first, DLS monitors time-dependent fluctuations from a molecular light scatterer and uses a detector such as a photomultiplier tube to record information about the particle size or surrounding solution viscosity. DWS, while based on DLS, does not share its single scatterer constraint. Both DLS and DWS offer rheology information across several orders of magnitude but only for solutions that are close to optically clear. Biological systems such as biopolymers or cell cultures become difficult to measure with these methods because they often contain scatterers of unknown size and/or shape (Gardel et al., 2005).

Passive, or thermal microparticle rheology (TMPR) is an umbrella technique that also estimates the response functions for materials. Traditionally we focus more greatly on those materials traditionally difficult to obtain in large quantities or for those with heterogeneities at small length scales. It can include single, multiple, and two particle methods. By using laser interferometric or video-based tracking techniques one can measure the displacement of micron-scale diameter spheres to within a few nanometers of resolution using only the thermal motion of the sphere as the driving force. By monitoring only the displacement of particles at constant temperature as a function of time, we can use mean-square displacement (MSD) and Generalized Stokes-Einstein Relation (GSER) methods to generate estimates of the viscoelastic response functions.

The remaining methods, single particle tracking (SPT), multiple particle tracking (MPT), and two-point microrheology (TPM) all use particle tracking methods in a video microscopy configuration. SPT monitors the displacement of single particles, traditionally fluorophores or microspheres (Saxton and Jacobson, 1997), while MPT handles simultaneous tracking of many particles in the same field of view, treating each as a distinct, isolated particle (Apgar et al., 2000). Particle tracking methods are more sensitive to heterogeneous materials, generating wide variances in the results that macroscale rheometry methods such as CAP would otherwise average across due to its vast contact area (Schmidt et al., 2000). While this is a limitation when trying to generate data comparable to CAP but with a much smaller specimen, it can also be considered an advantage when one wants to sample the heterogeneity in a material such as mucus for a given length scale.

Generally, the characterized length and time scales must be relevant to the phenomenon of interest. A variant on multiple particle tracking (MPT) is the two particle

microrheology (TPM) technique, which uses the correlated motion between pairs of particles. TPM recovers and reproduces macroscale results more faithfully because the probed length scales become the intervening distances between particles and not the sizes of the particles themselves (Crocker et al., 2000).

Random walks describe many of these stochastic processes where molecules diffusing in solvent codify the lower bound on chemical reaction rate. The Stokes-Einstein relation describes the diffusion rate for any size or shape particle in solution (Eq. 4), which depends on the temperature of the solution and the viscous drag the solution imparts on particle motion. (Berg, 1993).

The root-mean-square displacement, $\langle r \rangle = \sqrt{\langle r^2 \rangle}$, defines a characteristic length scale for a diffusing particle with radius a and is a relationship that functions at every observed time and length scale, a consequence of the fractal nature of stochastic processes and the ergodic theorem. The mean-squared displacement (MSD), $\langle r^2 \rangle$, of a diffusing particle in a Newtonian fluid will vary in time according to the statistics of a random walk,

$$\left\langle r^2 \right\rangle = 2dD\tau \,, \tag{1}$$

where d defines the dimensionality of the *observed* process ($d = 2$ for either diffusion in two dimensions or for the two-dimensional projection of diffusion in three dimensions, such as that measured by a camera). D is the diffusion coefficient, and τ describes a window of time, or period of duration (Rubinstein and Colby, 2003).

To compute the MSD across varying τ, we use the raw position data of diffusing particles as a function of time, i.e.

$$\Delta\bar{r}^2(\tau) = (N - \tau) \sum_{k=\tau}^{N-\tau} \left(r_{k+\tau} - r_k\right)^2 \,, \tag{2}$$

where r_k defines the position of the particle by

$$r_k = \sqrt{\left(x_k - x_o\right)^2 + \left(y_k - y_o\right)^2} \,. \tag{3}$$

Einstein described the diffusion coefficient in a general form, i.e.

$$D = kT/\zeta \,, \tag{4}$$

where ζ is a quantity that defines the Stokes drag interaction between a solvent with viscosity, η, and a diffusing particle's size and shape (Einstein, 1905). For a diffusing sphere with radius, a, $\zeta_s = 6\pi a\eta$, making the specific diffusion coefficient

$$D = \frac{kT}{6\pi\eta a} \,, \tag{5}$$

commonly called the Stokes-Einstein relation (Rubinstein and Colby, 2003). Expanded, this makes the MSD equation,

$$\langle r^2 \rangle = \frac{2dkT\tau}{6\pi\eta a}\,. \tag{6}$$

It is important to notice that this equation describes an averaged, statistical value for $\langle r^2 \rangle$ in a Newtonian fluid, as is the case when a single particle is monitored for an extended period of time, or when a large number of particles are tracked and their squared displacements averaged together. The MSD will be a linear function of τ, and the average displacement will be zero (Rubinstein and Colby, 2003). Because of its linear dependence on τ (shown in Equation 6), the MSD as a function of τ for a Newtonian fluid will have a slope, α, equal to one in log space. When the test material is viscoelastic α decreases to < 1 for some time scales and for purely elastic materials α becomes decreases even further to become zero for all time scales (Fig. 2).

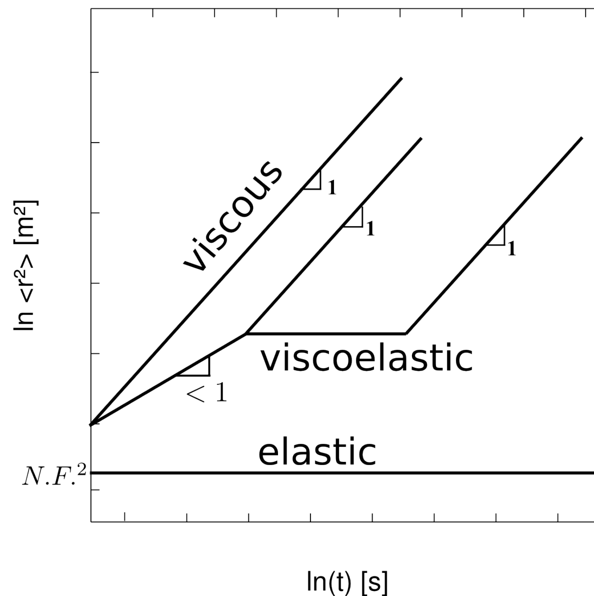

Figure 2: Expected MSD power laws for spheres diffusing in viscous, viscoelastic, and elastic materials

In 1995 Mason and Weitz proposed a generalized Stokes-Einstein relation (GSER) that extracted linear viscoelastic moduli for DLS (Mason and Weitz, 1995). By 2000 the technique had spread to cover single and multiple particle tracking techniques (Mason, 2000). Briefly, the GSER generalizes the Stokes-Einstein relation by making the diffusion coefficient, D, a function of complex frequency which propagates into the complex shear

modulus, $G^*(\omega)$,

$$G^*(\omega) = \frac{kT}{6\pi a D^*(\omega)}\,, \tag{7}$$

where $D^*(\omega)$ is a frequency-dependent complex diffusion function. Mason estimates the transform from the time to the frequency domain first with a power law expansion about the frequency, ω

$$\left\langle \Delta r^2(t) \right\rangle \approx \left\langle \Delta r^2(1/\omega) \right\rangle (\omega t)^{\alpha(\omega)}\,, \tag{8}$$

where $\alpha(\omega)$ is the power-law slope, defined by Mason as

$$\alpha(\omega) \equiv \frac{d \ln \langle \Delta r^2(t) \rangle}{d \ln t}\,, \tag{9}$$

evaluated with $t = 1/\omega$. The Fourier transform for the power law is approximately

$$i\omega \mathscr{F}\left\{ \left\langle \Delta r^2(t) \right\rangle \right\} \approx \left\langle \Delta r^2(1/\omega) \right\rangle \Gamma[1+\alpha(\omega)] \cdot i^{-\alpha(\omega)}\,, \tag{10}$$

where the well-known gamma function is $\Gamma(x) = \int_0^\infty t^{(x-1)} e^{-t} dt$ (Mason, 2000; Greenberg, 1998). Finally, the complex shear modulus, $G^*(\omega) = G'(\omega) + iG''(\omega)$ with its components become equal to

$$G'(\omega) = |G^*(\omega)| \cos\left(\pi\alpha(\omega)/2\right) \tag{11}$$

$$G''(\omega) = |G^*(\omega)| \sin\left(\pi\alpha(\omega)/2\right)\,, \tag{12}$$

where $G'(\omega)$ is the storage modulus (a measure of “solidness”) and $G''(\omega)$ is the loss modulus (a measure of “liguidness”). The scalar magnitude of $G * (\omega)$ becomes equal to

$$|G^*(\omega)| = \frac{kT}{\pi a \langle \Delta r^2(1/\omega) \rangle \Gamma\left[1+\alpha(\omega)\right]}\,. \tag{13}$$

A consequence of this methodology extends the slope from only the diffusive range with $\alpha(\omega) = 1$ across all time scales down to $\alpha = 0$ corresponding to a material that is completely elastic (Gardel et al., 2005). The limits for each of these slopes, their terminal values in both displacement and time scales are all predictable according to polymer physics models such as the Doi-Edwards tube model of polymer reptation (Rubinstein and Colby, 2003).

In summary, by tracing particle paths in the material, the GSER provides an adequate estimate for frequency-dependent relationships between the viscous and elastic moduli for a given material without the large amount of material that macroscale measurements such as CAP require. However, because the nature of the process is stochastic, large numbers of particles must be tracked, and because of the linear relationship between $D^*(\omega)$ and $\langle r^2 \rangle$, the method is limited to the linear viscoelastic regime. Size-dependence properties which may be inherent in the material are measurable by this method but is limited when one wants to make large length scale measurements. Even with these precautions the GSER can have artifacts at extremes of frequency due to data truncation (Mason, 2000).

Protocol

Sample Preparation

1. We will use 500 nm red-fluorescent volume-labelled, –COOH surface coated beads. These beads have an excitation wavelength of 580 nm and an emission wavelength of 605 nm.

2. Before the workshop, we prepared the following solutions, adding beads at 1:500 v/v dilution from stock.

   - PGM at 8%, 4%, 2%, 1% w/w concentrations
   - Agarose at 0.25% w/w
   - Sucrose at 2 M, and 2.5 M concentrations

3. To make dilutions, we titrated the stock solution with a bead-laden buffer solution as the diluent.

4. Load 96-square-well plate with 50 μL of each sample. Cut the pipette tip to increase diameter if sample is too thick. Whilst immersed in the sample well, touch the pipette tip around all sides to ensure a completely wetted surface. Keep your independent variables in mind while loading the plate. Once loaded, grease the well tops and seal with a single glass coverslip.

Scope Preparation

1. Mount correct objective (40x, length calibration of 179 nm/pixel).

2. Put sample tray onto Ludl microscope stage.

3. Turn on Ludl stage along with the correct LED light source (BF = 1, FL = 2).

4. Adjust lighting to get an image on the camera.

5. Use joystick to find an appropriate XY position.

6. Find a focal plane by adjusting the z-axis micrometer.

Instrument/Software Preparation

1. On **magnesium**, double-click *in order*, the following icons, to load

   - `PanopticNerve VIS`
   - `PanopticNerve test_imager client`
   - `cmd PanopticNerve`

2. Use the joystick to find sample volume for any given well.

3. Find the ideal height across sample wells at which to record videos of your samples. This height should record valid data for as many wells as possible on your well plate.

4. Edit the `wells.txt` file to comply with the current experimental parameters (located in `D:\schubert\PanopticNerve`). Currently four parameters comprise the `wells.txt` file.

   - Next to `experiment` type in a name or short description of the experiment. Each new data file name will contain this string.
   - To define which portions of the experiment are automated, change the values next to `video`, `vst`, and `analysis`. A value of `1` indicates that step will be handled automatically while `0` indicates that step will be manually handled.
   - The duration of the video is defined as `length` in the `wells.txt` configuration file. Change this value to whatever period of time is necessary for the experiment, measured in milliseconds, e.g. 10000 corresponds to 10 seconds.
   - The coordinates in each well where video is recorded can also be changed by adjusting the values next to the header `offset`. Values for the offset are measured in "ticks" of the Ludl stage, and, at this time, are not convenient to change.
   - Change the values under the header `wells` to correspond to the wells that you wish to record. The 96 well plates are numbered so that the 'a' row contains wells 1-12, the 'b' row 13-24, the 'c' row 25-36 etc.

5. **NOTE:** Be sure to delete all legacy video files located in this directory or the software will crash!

Tracking Configuration

First, open a typical video and play with tracking parameters, using the table below as a guide. The auto-tracking configuration resides in the `autofind.bat` file, with common values for 1 μm beads listed below. When you have picked parameters that produce satisfactory results, open `autofind.bat` and set the parameters that will be used for auto-tracking.

```
set r=8
set samp_spac=0.7
set intens_lost_sens=0.04
set border_zone=1
set tracker_zone=1
set maintain=50
set threshold=1.6
set window=15
```

Tracking Parameters Defined

NOTE: Next to parameter descriptions are typical values used for 500 nm beads.

`r`	size of the tracker in pixels (8)
`samp_spac`	tracking step resolution (0.7)
`intens_lost_sens`	specifies how bright the particle must be relative to the background (0.04)
`border_zone`	delete trackers within the border zone surrounding each tracker (1)
`tracker_zone`	delete trackers surrounding each tracker within this pixel radius (8)
`maintain`	stop looking for new trackers when there are already this many trackers on the screen (50)
`threshold`	selectivity of the tracker-finding algorithm where a lower value finds more particles to track (1.6)
`window`	density of the search through the frame when finding new particles to track (15)

Tracking manually...

1. To track the typical video, open it first by dragging the `vrpn` file to the `video spot tracker` program icon and choose the appropriate settings for radius, sample spacing, and precision. For 1 micron carboxyl beads one should use values of radius 12, sample spacing of 0.7, and precision of 0.005.

2. Check the box to optimize and then proceed to check the logging box. Create a folder to place all the logging files in from that day. When saving your log files select a name for your files that differs by a number, for example for PGM `PGM_1`, `PGM_2`, etc. and for sucrose `Sucrose_1`, `Sucrose_2`, etc. This is important later.

3. Select `play video` and allow spot tracker to run over all the frames.

4. Repeat steps 1 through 3 for all the videos.

5. After finishing step 4 you should have a folder full of `.vrpn` files. Select all your files and right click and scroll down to `Send To...` and select `vrpnLogToMatlab 04.01`. For organizational purposes it is useful here to save all the `.vrpn.mat` files in their own folder.

Analysis Configuration

At this time, the configuration settings for the data analysis pipeline are all set as the default values defined by each function involved in the computation. Listed below is a summary for these settings.

```
calibum = 0.15
framerate = 30
minFrames = 50
minPixels = 0
window = [1:10 12 16 21 29 42 60 100]
```

Analysis Parameters Defined

NOTE: Next to parameter descriptions are typical values used for 500 nm beads.

`calibum`	length calibration for objective and camera, measured in microns per pixel `(0.15)`
`framerate`	camera frame rate in fps `(30)`
`minFrames`	deletes tracker paths with fewer than this minimum number of frames `(50)`
`minPixels`	delete tracker paths with fewer within the border zone surrounding each tracker `(0)`
`window`	each entry defines a number of frames (and thus, time) to skip when computing the MSD `([1:10 12 16 21 29 42 60 100])`

To analyze manually...

1. Open MATLAB and type in the command `evt_GUI`.
2. Click `load` and select your first MATLAB `.vrpn.mat` file (i.e. `PGM_1.vrpn`).
3. Check off the `Custom Frame Rate` box and enter your value for fps.
4. If any beads were not tracked properly and are more than just a diamond-shaped marker on the `XY` displacement map delete them by scrolling through the data using the arrows in the box labelled `Select...` above the `Closest Dataset` button until the rogue bead is highlighted and then proceed to click the `Edit Data` button in the box labelled `Delete...` making sure the `selected dataset` is selected.
5. Under the box labeled `Displacement Units` change the microns per pixel value that which corresponds to the lens used in the experiment.
6. Under the `Drift Subtraction` box select `Center of Mass` and click the `Remove Drift` button.
7. Change the name in the `Output` line to indicate that the drift has been removed (I would suggest `PGM_1_NoDrift` and click the `Save` button.
8. Repeat step 7 for all your `.vrpn` files.
9. Load your `PGM_1_NoDrift` file using the `Load` button next to the `Input` line.
10. Copy the line from the `Output` line and paste it into the `Input` line.

11. Whilst still in the `Input` line, replace the `1` in the file name with an `*` and delete one of the `.evt` extensions. So the file might look like:

```
C:\Workshop2010_day1\PGM_*_NoDrift.vrpn.evt.mat
```

12. Click `Load` and you should see an XY displacement figure containing all the beads you used in that one sample concentration of PGM. From here use can use the `AUX Plot Options` and display MSDs, GSERs or whatever else might be available. NOTE: "Lines" in the XY plot indicate the sample had problems with substantial drift.

13. Repeat steps 9-12 for your other solutions and/or concentrations and proceed to compare and analyze data across different the different independent variables, here namely solution type and polymer/solute concentration.

Initiate Experiment

1. If one is not available, open a new Panoptic nerve command prompt by double-clicking on the `cmd PanopticNerve` icon.

2. In the new command prompt type in `RunNerve.bat` and press `Enter`. This batch file starts the `PanopticNerve` software and executes the commands according to the `wells.txt` file. Initially, `PanopticNerve` will calibrate its XY position and then prompt you to press `Enter` to start data acquisition.

3. After `PanopticNerve` acquires video in every well as described by `wells.txt`, it will launch the `video_spot_tracker` program and begin autotracking beads according to the parameters specified in the `autofind.bat` file.

4. Finally, `PanopticNerve` will attempt to analyze the collected bead traces, calculating the MSD and reporting the slient results.

5. When this is finished running there should be a `.vrpn` file for every well you specified recording. They will be named `experimentname_well#.vrpn`, where the `#` indicates the well in which the data from that file were taken. These `vrpn` files can be read and analyzed by `video spot tracker`.

Discussion

What have we done and how have we done it?

In this workshop exercise we have shown how nanoparticle diffusion experiments can be prepared and data collected, tracked, and analyzed from eight different samples using Panoptes. Importantly, this process can be scaled up to 384 different samples, allowing for truly high-throughput automated experiments.

Figure 3: Photograph of Panoptes

This has been accomplished by mixing probes (500nm fluorescent particles) with materials of interest (PGM, agarose, and sucrose), dosing these samples into the wells of a 96-well tray, and acquiring video of the Brownian motion of the probes, sequentially, in each of the eight materials. Automating data collection allows the user to perform other tasks as the microscope collects the required data.

Automating analysis of the collected video data requires automatically finding and spot-tracking probes in sequential wells, removing drift from the probe traces, and quantifying probe motion. A figure showing the Auto Video Spot Tracker environment is shown in Figure 4. The current implementation of Video Spot Tracker (manual spot tracking) can be found here: http://cismm.cs.unc.edu/downloads/. Automated Video Spot Tracker is currently only available by request. Drift removal and quantification of

probe motion is performed using in-house Matlab code, examples of which can be found at the end of this document.

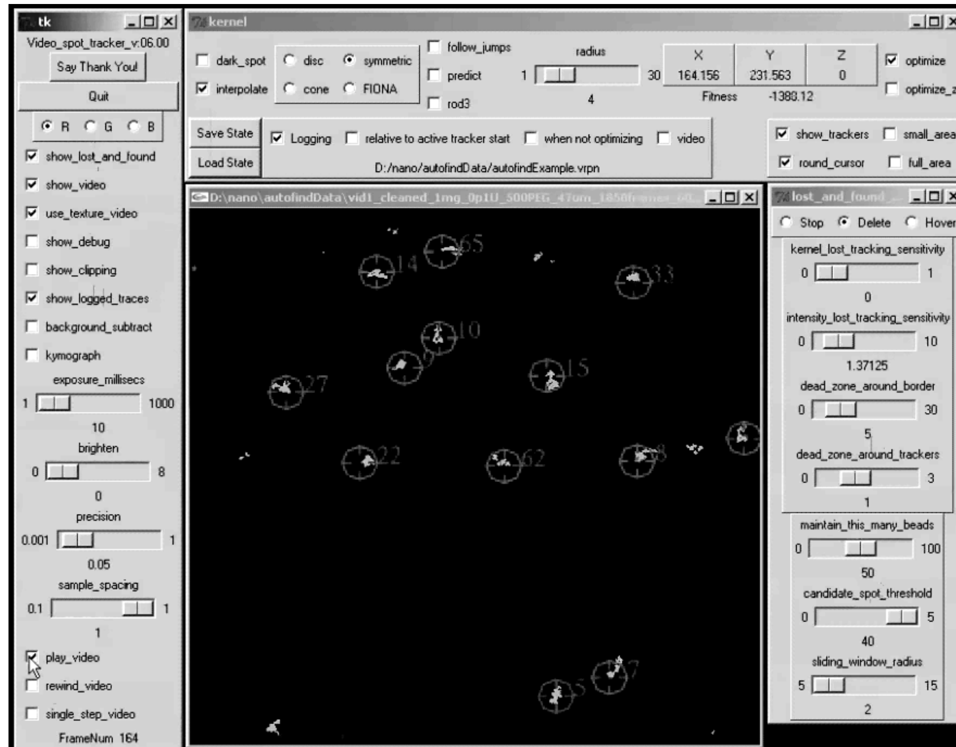

Figure 4: A screen shot of Panoptes Auto Video Tracking in operation

What is the impact?

In our lab we're using Panoptes to study diffusion in three specific systems, each system motivated by a broader biological motivation.

1. Fibrin is a protein responsible for clotting in blood and, consequently, the mechanical properties and diffusion characteristics of fibrin gels is significant. Abnormalities in fibrin gels may lead to a lack of clotting in situations where clotting is desired (wound healing) or the occurrence of clotting where it is undesired (deep vein thrombosis). These physiological events and the medical implications of each make the study of diffusion in fibrin particular interesting. In another Carolina Biophysics Workshop station you learned about our group's efforts in characterizing the mechanical properties of fibrin fibers using atomic force microscopy. We

are currently studying the diffusivity of fibrin gels (analogous to a fibrin clot) by observing nanoparticle diffusion in these gels. We hope to use this system to answer questions regarding how gel fiber properties correlate with the ability of particles to move diffusively through the gel.

2. Our lab also studies the rheological properties of mucus from both healthy individual and those diagnosed with cystic fibrosis. One could imagine other experimental modalities where a system like Panoptes might flourish, such as garnering time-dependent information while acquiring dose-response curves. Experiments such as this provides the means to study new treatments quickly and in parallel.

3. As a final application of Panoptes, we're studying how substrate stiffness affects cell differentiation. It is a well-known phenomenon that cells, and specifically stem cells, are responsive to the stiffness of their surrounding environment. In an attempt to probe cell differentiation using a variety of different substrates, we are studying the diffusion of beads in agarose, a common cell growth substrate. We hope to build more specific, finely tuned cell substrates, based on the quantification of agarose gel rheological properties.

Future User Interface for Panoptes

Anomalies in the data are currently handled manually, as each experiment and each sample may present its own set of unusual data. In order to handle data from experiments in which not all data should be kept we have designed a mockup of a graphical user interface (GUI), the Panoptes DataBrowser (5). Using this GUI we hope to be able to view and interact with data in the form of packets, each packet containing the information from a single well. As Panoptes collects very large sets of data (on the order of 100GB+ of video), this interface will make it possible to peruse all aspects of the data and analysis (video files, spot tracking, MSDs) serially.

Bringing your samples to Panoptes for experimentation

Currently, Panoptes is available for use and collaboration with other researchers seeking to study biopolymer mechanics / rheology. Currently, the following sample requirements must be met:

1. The sample must contain beads (not necessarily fluorescent) with a minimum bead diameter of 250nm.

2. If using fluorescent beads, the bead excitation must be approximately 580nm.

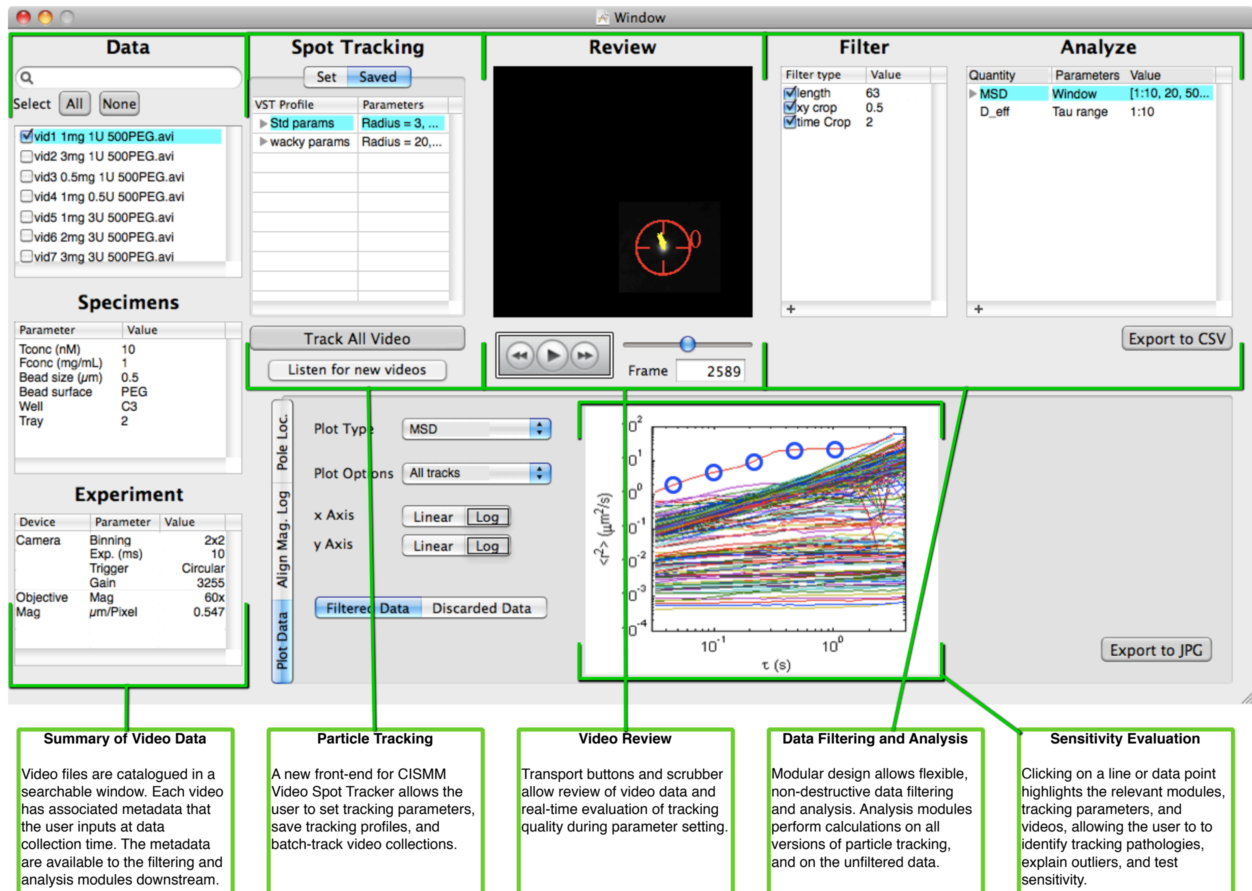

Figure 5: Proposed user interface for Panoptes DataBrowser

3. Samples must be pipettable.

Conclusions

In this workshop exercise we reviewed the literature and the methodology we use to determine the rheology of a material from the diffusion of embedded microparticles. After illustrating the basic principles of passive, thermal microbead rheology, we showed how the MSD and GSER methods produce the results of these rheology measurements and then combined them with a largely automated process through Panoptes, our high-throughput microscope system.

Code examples

```
function [tau, msd, r2out] = msd(t, data, window)
% MSD computes the mean-square displacements (via the Stokes-Einstein
% relation) for a single bead
%
% 3DFM function
% specific\rheology\msd
% last modified 11/20/08 (krisford)
%
%
% [tau, msd] = msd(t, data, window);
%
% where "t" is the time
%       "data" is the input matrix of data
%       "window" is a vector containing window sizes of tau.
%
% Notes: - default window = [1 2 5 10 20 50 100 200 500 1000]
%


%initializing arguments
if (nargin < 1) || isempty(t)
    error('Input data needed.');
end;

if (nargin < 2) || isempty(data)
    error('Input data needed.');
end;

if (nargin < 3) || isempty(window)
    window = [1 2 5 10 20 50 100 200 500 1000 1001];
end;

% for every window size (or tau)
warning('off', 'MATLAB:divideByZero');

r2out = zeros(length(window), size(data,1)) * NaN;

    for w = 1:length(window)

      for k = 1:cols(data)

        A = data(1:end-window(w), k);
        B = data(window(w)+1:end, k);
```

```
            r = (B - A);

            if k == 1
                r2 = r.^2;
            elseif k > 1
                r2 = r2 + r.^2;
            end

          end

            tau(w, :) = window(w) * mean(diff(t));
            msd(w, :) = mean(r2);
            r2out(w,1:length(r2)) = r2;
        end

    warning('on', 'MATLAB:divideByZero');

    return;


    function vmsd = video_msd(files, window, frame_rate, ...
                              calib_um, make_plot)
    % VIDEO_MSD computes the MSD for an aggregate number of beads
    %
    % 3DFM function
    % specific\rheology\msd
    % last modified 11/20/08
    %
    % This function computes the mean square displacements for
    % an aggregate number of video tracked beads via the Stokes-
    % Einstein relation and allows for the option of plotting
    % MSD vs. window size (tau).
    %
    % [vmsd] = video_msd(files, window, frame_rate, ...
    %                    calib_um, make_plot)
    %
    % where "files" is the filename containing the video tracking
    %               data (wildcards ok).
    %       "window" is a vector containing window sizes of tau
    %               when computing MSD.
    %       "frame_rate" is the video tracking frame rate
    %               in [frames / second].
    %       "calib_um" is the microns per pixel conversion unit
    %       "make_plot" gives the option of producing a plot of
    %               MSD versus tau.
    %
    % Notes: - No arguments will run a 2D MSD on all .mat files
    %          in the current directory and use the default
```

```
%           window sizes.
%         - Use empty matrices to substitute default values.
%         - default files = '*.mat'
%         - default window = [1 2 5 10 20 50 100 200 500 1000]
%         - default calib_um = 0.152 microns/pixel
%         - default make_plot = yes
%


% initializing arguments
if (nargin < 1) || isempty(files)
    files = '*.mat';
end;

if (nargin < 2) || isempty(window)
    window = [1 2 5 10 20 50 100 200 500 1000 1001];
end;

if (nargin < 3) || isempty(frame_rate)
    frame_rate = [];
end;

if (nargin < 4) || isempty(calib_um)
    calib_um = 0.152;
end;


% define constants that identify the column headers
% for the current version of vrpn-to-matlab.
    TIME   = 1;
    ID     = 2;
    FRAME  = 3;
    X      = 4;
    Y      = 5;
    Z      = 6;
    ROLL   = 7;
    PITCH  = 8;
    YAW    = 9;
    RADIAL = 10;


% determine whether we have to load these data from disk
if ~isnumeric(files)
    % load video data
    v = load_video_tracking(files, frame_rate, 'pixels', 1, ...
                            'relative', 'yes', 'table');
else
    v = files;
```

```
end

% convert from pixels to meters
v(:,X:Z) = v(:,X:Z) * calib_um * 1e-6;


% for every bead
beadID = unique(v(:,ID))';
nbeads = length(beadID);
nwindow = length(window);

myr2 = NaN * zeros(nwindow, nbeads, max(v(:,FRAME)+1));
for k = 1 : length(beadID);

    b = data( data(:,ID) == k, :);
    framemax = max(b(:,FRAME));

    % use the MSD program to compute the MSD for each bead
    [tau(:, k), mymsd(:, k), r2] = ...
                        msd(b(:, TIME), b(:, X:Z), window);
    [ro, cl] = size(r2);
    myr2(1:ro, k, 1:cl) = r2;
end;


% trim data by removing window sizes that had no data
sample_count = sum(~isnan(mymsd),2);
idx = find(sample_count > 0);
tau = tau(idx,:);
mymsd = mymsd(idx,:);
sample_count = sample_count(idx);


% output structure
vmsd.tau = tau;
vmsd.r2  = myr2;
vmsd.msd = mymsd;
vmsd.n   = sample_count;

% creation of the plot MSD vs. tau
if (nargin < 5) || ...
    isempty(make_plot) || ...
    strncmp(make_plot,'y',1)

    plot_msd(vmsd, [], 'me');
end;
```

```
function v = ve(d, bead_radius, freq_type, plot_results);
% VE computes the viscoelastic moduli from the computed MSD
%
% 3DFM function
% specific\rheology\msd
% last modified 11/20/08
%
% ve computes the viscoelastic moduli from mean-square
% displacement data using the algorithm outlined in Mason's
% 2000 Rheologica Acta paper.  The output structure of ve
% contains four members: raw (which contains data for each
% individual tracker/bead), mean (contains means across
% trackers/beads), and error (contains standard error
% (stdev/sqrt(N) about the mean value, and N (the number of
% trackers/beads in the dataset.
%
%  [v] = ve(d, bead_radius, freq_type, plot_results);
%
%  where "d" is the output structure of video_msd.
%        "bead_radius" is in [m].
%        "freq_type" is 'f' for [Hz] (default)
%                       'w' for [rad/s]
%        "plot_results" plots a figure if 'y' (default)
%

if (nargin < 3) || isempty(freq_type)
    freq_type = 'f';
end

if (nargin < 2) || isempty(bead_radius)
    bead_radius = 0.5e-6;
end

if (nargin < 1) || isempty(d)
    error('no data struct found');
end

k = 1.3806e-23;
T = 298;

msd = d.msd;
tau = d.tau;

% N corresponds to the number of trackers at each tau
N = d.n(1:end-1);


A = tau(1:end-1,:);
```

```
B = tau(2:end,:);
C = msd(1:end-1,:);
D = msd(2:end,:);
alpha = log10(D./C)./log10(B./A);
MYgamma = gamma(1 + alpha);
% gamma = 0.457*(1+alpha).^2-1.36*(1+alpha)+1.9;

% because of the first-difference equation used to compute
% alpha, we must delete the last row of f, tau, and
% msd values.
msd = msd(1:end-1,:);
tau = tau(1:end-1,:);

% get frequencies all worked out from timing (tau)
f = 1 ./ tau;
w = 2*pi*f;

% compute shear and viscosity
gstar = (2/3) * (k*T) ./ ...
        (pi * bead_radius .* msd .* MYgamma);
gp = gstar .* cos(pi/2 .* alpha);
gpp= gstar .* sin(pi/2 .* alpha);

nstar = gstar .* tau;
np = gpp .* tau;
npp= gp  .* tau;

%
% setup the output structure
%
v.raw.f       = f;
v.raw.w       = w;
v.raw.tau     = tau;
v.raw.msd     = msd;
v.raw.alpha   = alpha;
v.raw.gamma   = MYgamma;
v.raw.gstar   = gstar;
v.raw.gp      = gp;
v.raw.gpp     = gpp;
v.raw.nstar   = nstar;
v.raw.np      = np;
v.raw.npp     = npp;

v.mean.f      = nanmean(f,2);
v.mean.w      = nanmean(w,2);
v.mean.tau    = nanmean(tau,2);
v.mean.msd    = nanmean(msd,2);
v.mean.alpha  = nanmean(alpha,2);
```

```
v.mean.gamma  = nanmean(MYgamma,2);
v.mean.gstar  = nanmean(gstar,2);
v.mean.gp     = nanmean(gp,2);
v.mean.gpp    = nanmean(gpp,2);
v.mean.nstar  = nanmean(nstar,2);
v.mean.np     = nanmean(np,2);
v.mean.npp    = nanmean(npp,2);

v.error.f     = nanstd(f,[],2)       ./ sqrt(N);
v.error.w     = nanstd(w,[],2)       ./ sqrt(N);
v.error.tau   = nanstd(tau,[],2)     ./ sqrt(N);
v.error.msd   = nanstd(msd,[],2)     ./ sqrt(N);
v.error.alpha = nanstd(alpha,[],2)   ./ sqrt(N);
v.error.gamma = nanstd(MYgamma,[],2) ./ sqrt(N);
v.error.gstar = nanstd(gstar,[],2)   ./ sqrt(N);
v.error.gp    = nanstd(gp,[],2)      ./ sqrt(N);
v.error.gpp   = nanstd(gpp,[],2)     ./ sqrt(N);
v.error.nstar = nanstd(nstar,[],2)   ./ sqrt(N);
v.error.np    = nanstd(np,[],2)      ./ sqrt(N);
v.error.npp   = nanstd(npp,[],2)     ./ sqrt(N);

v.n = N;

% plot output
if (nargin < 4)           || ...
   isempty(plot_results) || ...
   strncmp(plot_results,'y',1)

    fig1 = figure; fig2 = figure;
    plot_ve(v, freq_type, fig1, 'Gg');
    plot_ve(v, freq_type, fig2, 'Nn');
end;
```

Bibliography

Apgar, J., Tseng, Y., Fedorov, E., Herwig, M. B., Almo, S. C., and Wirtz, D. (2000). Multiple-particle tracking measurements of heterogeneities in solutions of actin filaments and actin bundles. *Biophysical Journal*, 79(2):1095–1106.

Berg, H. C. (1993). *Random Walks in Biology.* Princeton University Press, Princeton, NJ.

Brown, R. (1828). On the existence of active molecules in organic and inorganic bodies. *Phil. Mag*, 4:162–173.

Crocker, J. C., Valentine, M. T., Weeks, E. R., Gisler, T., Kaplan, P. D., Yodh, A. G., and Weitz, D. A. (2000). Two-point microrheology of inhomogeneous soft materials. *Physical Review Letters*, 85(4):888–891.

Einstein, A. (1905). Motion of suspended particles on the kinetic theory. *Annalen der Physik*, 17(3):549–560.

Gardel, M., Valentine, M., and Weitz, D. (2005). Microrheology. *Microscale Diagnostic Techniques. Heidelberg, Springer-Verlag Berlin Heidelberg, 2005*, pages 1–49.

Greenberg, M. (1998). *Advanced Engineering Mathematics.* Prentice-Hall, Englewood Cliffs, NJ.

Maret, G. and Wolf, P. (1987). Multiple light scattering from disordered media. The effect of Brownian motion of scatterers. *Zeitschrift für Physik B Condensed Matter*, 65(4):409–413.

Mason, T. G. (2000). Estimating the viscoelastic moduli of complex fluids using the generalized stokes-einstein equation. *Rheologica Acta*, 39(4):371–378.

Mason, T. G., Ganesan, K., vanZanten, J. H., Wirtz, D., and Kuo, S. C. (1997). Particle tracking microrheology of complex fluids. *Physical Review Letters*, 79(17):3282–3285.

Mason, T. G. and Weitz, D. A. (1995). Optical measurements of frequency-dependent linear viscoelastic moduli of complex fluids. *Physical Review Letters*, 74(7):1250–1253.

Pine, D. J., Weitz, D. A., Chaikin, P. M., and Herbolzheimer, E. (1988). Diffusing wave spectroscopy. *Phys. Rev. Lett.*, 60(12):1134–1137.

Rubinstein, M. and Colby, R. H. (2003). *Polymer Physics*. Oxford University Press, Oxford, UK.

Saxton, M. J. and Jacobson, K. (1997). Single-particle tracking:applications to membrane dynamics. *Annual Review of Biophysics and Biomolecular Structure*, 26(1):373–399.

Schmidt, F. G., Hinner, B., and Sackmann, E. (2000). Microrheometry underestimates the values of the viscoelastic moduli in measurements on f-actin solutions compared to macrorheometry. *Physical Review E*, 61(5):5646–5653.

Stachel, J. and Raman, V. (1990). The Collected Papers of Albert Einstein. Vol. 2. The Swiss Years: Writings, 1900- 1909. *American Journal of Physics*, 58:700.

Waigh, T. A. (2005). Microrheology of complex fluids. *Reports on Progress in Physics*, 68(3):685–742.