Command Line Tool : DTI Fiber Tract Statistics

Summary:
Various tract-oriented scalar diffusion measures obtained from DTI brain images, are treated as a continuous function of white matter fibers' arc-length. To analyze the trend along a given fiber tract, a command line tool performs kernel regression on this data. The idea is to try out different noise models and maximum likelihood estimates within kernel windows (along the tract), such that they best represent the data and are robust to noise and Partial Volume effect.

Objective:
The results of this tool can be further used to model the water diffusion leading to population based analysis. The aim is to understand probabilistic models that can account for the behavior of water diffusion in white matter tracts. The long term goal is to use this to understand the changes in white matter structure with age, gender or a specific disease.

Motivation

There is an existing tool called **Fiber Viewer** which allows a number of operations on fibers obtained using fiber tractography. It allows the user to view fiber bundles and perform operations like clustering to remove outliers based on length, center of mass and distance measures; fiber utilities like cutting, spline interpolation; fiber analysis using parameters like Fractional Anisotropy, Mean Diffusivity and other parameters along the fiber and batch processing on several subjects.

The approach taken by the tool to analyze fibers using Fractional Anisotropy (FA) and other measures has certain limitations. The basics of the tool's approach are explained below:

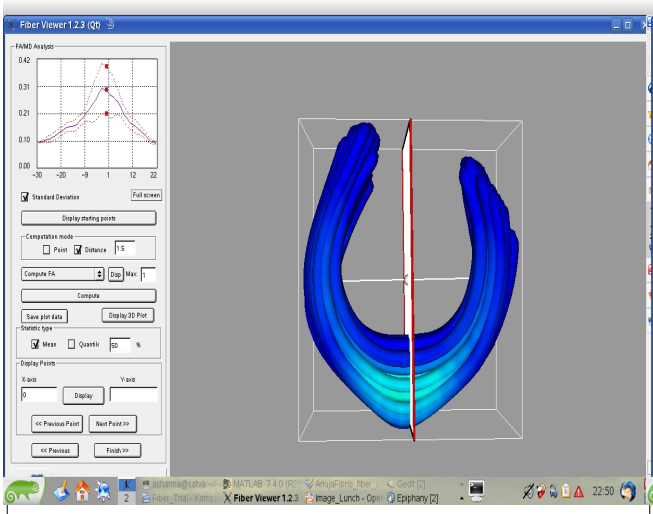
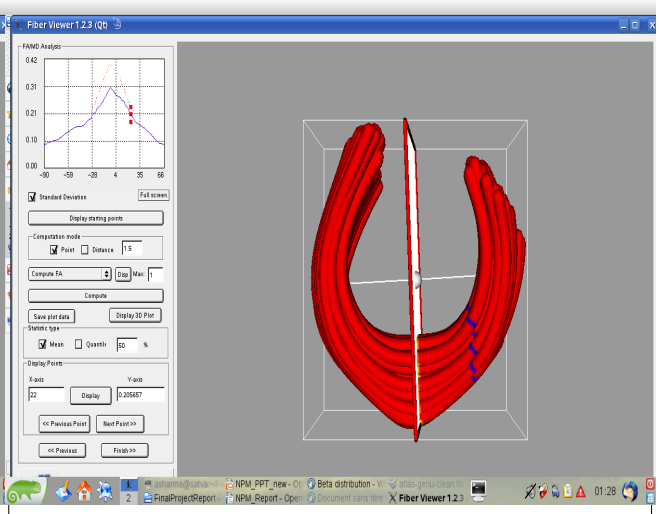
1. The origin for the operation is defined by the user in the form of a cut-plane. The starting point on the fiber is the point just right before or after the fiber bundle crosses the plane. If the fiber doesn't cross the plane at all, it is the point along the fiber which is the closest to the plane.

2. The tool moves along the fiber bundle arc length and computes a mean, standard deviation and quantiles for cross sections of the bundle from the starting point (chosen as the 0 of the curvilinear abscissa) until the extremities of the fiber. The values of FA at each cross section can be computed using two possible approaches:

    - The first method is the **point** method. All the starting points are considered as the 0. The mean of all the value of these points is associated to the 0. Then the algorithm moves forward to the next point on each fiber, and the mean is computed on these new points.
    - The second mode is the **distance**. The mean is not computed point by point anymore, but step by step (each step being a distance that can be set). The problem is that the value is not available at each location. It is only available on the points themselves. So from the starting point going forward step by step, if the step is not on a point (which is the case most of the time) the current value is that of the closest point.

3. These either deform the cross-section (which is not exactly perpendicular to the fiber length any more) along the fiber in the point approach or use the nearest neighbor's value in the distance approach which is not an accurate approximation.

Thus both the approaches have limitations which could introduce error in the observed mean, standard deviation and quantile curves (specially if the fiber bundle has a very high curvature).

Fiber Viewer: Mean and Standard Deviation curves for a fiber bundle. The Cut plane is user defined. The colors move from Blue to Red as the FA value increases. Thus, here the middle of the bundle has higher FA values than the extremes.	**Fiber Viewer:** Mean and Standard deviation curves with a different cut plane definition, using the **POINT** approach. The blue points to the right of the plane show an intermediate cross section. The irregularity of the sample locations deforms the cross section here which is not exactly perpendicular to the fiber bundle at that point.
	

Description

Since we do not know the actual distribution of any scalar diffusion values along the fibers, **Non-parametric Regression** is one of the possible solutions. This approach does not require any prior knowledge of the parameters and the analysis is performed based on the information derived from the data itself.

The basic approach is defined now:

1.1. Move along the fiber tracts and treat arc length as continuous instead of making it discrete using sample locations or distance based approach (as in the Fiber Viewer).

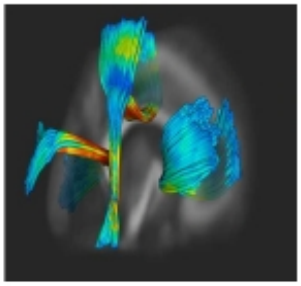
1.2. Using a cut plane, the origin for this arc length parametrization is defined along the fiber bundle. On each fiber in the bundle, we move from the closest point on the bundle where the plane cuts it, to both the ends of the bundle along the fiber. Thus, we generate an arc length vs scalar diffusion measure scatter plot where the arc length value is the distance of the sample points from the origin along the fiber.

1.3. This data is then used to analyze the trend of the diffusion measure along the fiber bundle. This approach minimizes any approximation error on sample locations or interpolation errors on the diffusion values at this stage.

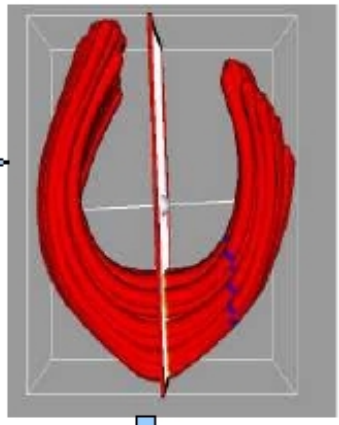
1.4. Non-Parametric regression is applied on this data by using a user defined step length across the Arc length axis which defines the cross section locations along the bundle. At each of these arc length values ,a kernel window is assumed and used to compute the estimated scalar diffusion value for that window. The window size is a user-defined

parameter and with a large or small window size, we may smooth the data as much as we need.

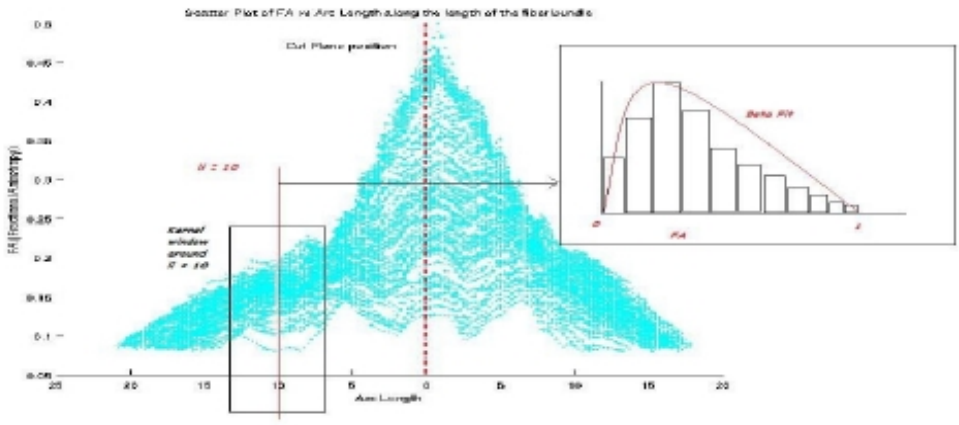
Visual explanation of the flow



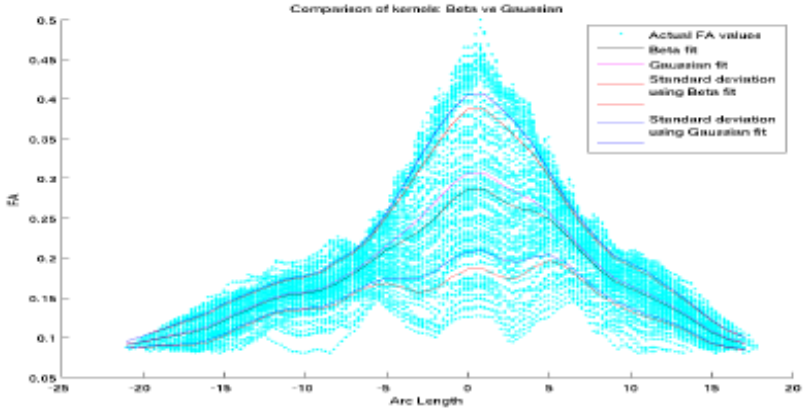
Fiber Tractography done on DTI Images gives white matter fiber tracts in the brain



A fiber bundle is then parametrized by arc length along the tract. The cut-plane defines the origin for this parametrization. As we move along the tract on eitehr side of the origin, we get cross sections across the bundle (marked in blue) with each fiber in the bundle contributing a sample point to this cross section. Each sample point has a coordinate location and scalar diffusion measures associated with it



As we move along the fiber tracts, we get the corresponding scalar diffusion value at each point in each cross-section, giving us the above scatter plot of arc length vs diffusion measure (eg FA). Kernel regression is now done along the arc length. Each kernel window has a distribution of the scalar measure within it (shown in the sub-figure on top-right of the plot). Choosing a noise model and a Maximum Likelihood Estimate for representing the distribution, we get a single point best representing the value of diffusion measure within each window.



The result of the kernel regression is shown here with different choices of the noise models, along with the curves for the resulting diffusion measure plus and minus the standard deviation calculated for each kernel window.

Cut Plane Origin: 64.5328 85.0293 27.8196
Cut Plane Normal: -0.984806 0.164354 0.0560822
Noise Model: Gaussian Statistics: Mean
Arc Length parametrization (Step size): 1.5 Standard Deviation for kernel window: 1.5
Parameter chosen for regression: MD
Number of samples along the bundle: 27

Arc_Length	#_fiber_points	Parameter_Value	Std_Dev	Param+Std_Dev	Param-Std_Dev
-20.4993	74	0.00132388	5.34331e-06	0.00132922	0.00131854
-18.9993	305	0.00132965	1.09856e-05	0.00134063	0.00131866
-17.4993	694	0.00133157	2.04357e-05	0.00135201	0.00131114
.					
.					
.					
-2.4993 1272	0.00129976	0.000108229	0.00140799	0.00119153	
-0.999301	1272	0.00132797	0.0001127	0.00144067	0.00121527
0.500699	1272	0.00135921	0.000102959	0.00146217	0.00125625
2.0007	1272	0.00129285	0.000112524	0.00140537	0.00118032
.					
.					
.					
15.5007828	0.00136584	4.80725e-05	0.00141391	0.00131776	
17.0007275	0.00135681	4.23942e-05	0.00139921	0.00131442	
18.500720	0.00137058	2.09224e-05	0.0013915	0.00134966	

The resulting curve is stored as a tab separated text file, including information like the plane details, noise model and MLE used, arc length vs diffusion measure and the corresponding standard deviation.

NOTE: This file is the actual output of the tool. The plots above (scatter plots and the fitted curves) have been generated using Matlab scripts and the below output file. Any simple plotting option can be used to visualize and further study the results below.

Tool Implementation/Execution Details

This is a command line tool which needs the following inputs:

<Executable> <Input_fiber_file> <World space 'W' or Image space 'I'> <Plane_fvp_file or 'Auto' option> <output_file> <scalar_diffusion_parameter> <arc_length_step_size> <bandwidth> <noise_model> <kernel_window_number> <Maximum_likelihood_estimate> <Quantile_percent>

<Executable> **help**
gives the above brief summary of the expected inputs and their order.

Example:
./dti_fiber_tract_statistics ~/Desktop/atlas-genu-clean-FV.vtk W Self ~/Desktop/result.fvp FA 1.5 1.5 Gaussian 10 Mean 60

Explanation of each command line input

<Executable> : This is the executable file of the command line tool

<Input_fiber_file> : Contains the fiber tracts obtained as a result of tractography on a DTI Image. The tracts can be stored with as much information as possible but the mandatory information includes :

- <x,y,z> coordinate of each sample point on a tract
- Information about which points contribute to which tract, thus forming a whole fiber bundle
- # of fibers in the bundle and # of points on each fiber in the bundle
- Spacing (World vs Image space) or Origin coordinate offset, if any
- The tensor matrix (3x3) associated with each sample point

Currently the tool accepts the UNC/UTAH **.fib** file format and the **.vtk** (polydata) file format as inputs. (Note: The .vtk file format needs the complete 3x3 symmetric, tensor matrix. The code needs to be modified to handle the input if only the upper or lower diagonal tensor matrix elements are given for each point).

NOTE: Even though the .vtk format might have the scalar diffusion measures and the eigen values for each point, already written in the input file; the tool only picks up the tensor matrix and computes the eigen values (using ComputeEigenValues function in ITK). These eigen values are then used to compute all the scalar diffusion measures as needed.
On the other hand, if the input is a .fib file, any information other than tensor matrix is also picked up directly from the file (like eigen values, FA, FRO etc).

<World space 'W' or Image space 'I'> : The results can be obtained in either of the 2 coordinate spaces. The option needs a 'W' for world space and 'I' for image space.

<Plane_fvp_file or 'Auto' option> : For arc length parametrization along a fiber bundle, we need a reference plane cutting the bundle approximately in the middle (or elsewhere, as needed). he point where the plane intersects with the bundle, is then taken as the origin (zero arc length). The arc length then increases (is positive) on one side of the origin as we move along the bundle, and decreases (is negative) on the other side. As we move along the fiber tract, any scalar diffusion measure given at a point along the tract, is treated as a function of the arc length that this point has wrt the origin of the fiber bundle as defined by the intersecting plane.
The plane is defined using a point on the plane (plane_origin) and the plane_normal.

There are 2 possible ways to provide this information:

- .fvp file: the first 2 lines in the header give the plane details :
  Cut Plane Origin: 6.453285e+01 8.502935e+01 2.781960e+01
  Cut Plane Normal: -9.848058e-01 1.643542e-01 5.608222e-02

  This approach requires a GUI where the user can visually set the plane's intersection with the bundle as needed. The resulting plane can be stored as a file with the first 2 lines as shown above.

- In the absence of such a GUI, the user can instead give the 'Auto' option. This would compute the details of the plane which is approximately cutting the middle of the fiber bundle. Its calculated by taking the plane_origin as the average <x,y,z> location of all the points in the complete fiber bundle. So, unless the bundle is very irregular, this will come to be approximately in the middle of the bundle. Next, 30% of the points towards both the ends of the bundle are ignored for finding the normal. Considering the 'middle' 60% points, the point 'on' the bundle, closest to the calculated plane_origin is found. This approximates the 'center' point 'on' the bundle. The plane normal is approximated as the vector going from a point 3 positions to the right of this center point to a point 3 position to the left.

  Note: The idea behind leaving out 30% of the points on either end is to avoid situations where the bundle has a high curvature. In this case, the plane origin could be far away from the actual bundle, located somewhere in the middle of the semi-circular part of the bundle. In this case, the 'center' point on the bundle could incorrectly be calculated as some point lying at the ends but being very close to it. (A more intuitive and feasible location for the plane is somewhere in the middle. This exceptional case might lead to a plane which intersects the bundle towards the end and might cut the bundle twice, leading to an ambiguity in the origin of the arc-length parametrization.

<output_file> : Two outputs are written.

- One is the exact file name as provided in <output_file> and contains the result for the chosen scalar diffusion measure, as per the chosen noise model and Maximum likelihood estimate.
  This output file looks as below:

  Cut Plane Origin: 64.5328 85.0293 27.8196
  Cut Plane Normal: -0.984806 0.164354 0.0560822
  Noise Model: Gaussian Statistics: Mean
  Arc Length parametrization (Step size): 1.5 Standard Deviation for kernel window: 1.5
  Parameter chosen for regression: MD
  Number of samples along the bundle: 27

Arc_Length	#_fiber_points	Parameter_Value	Std_Dev	Param+Std_Dev	Param-Std_Dev
-20.4993	74	0.00132388	5.34331e-06	0.00132922	0.00131854

  The first few header lines describe the options as given by the user- Plane details, Noise model, MLE or the statistics being used, Arc length step size, Bandwidth or standard deviation used in a given kernel window, scalar diffusion parameter chosen, final # of arc length steps after parametrization and kernel regression
  The rest of the file shows columns as the arc length value along with the # of points considered in the current kernel window for regression, the associated parameter value (after kernel regression), standard deviation in the current kernel window (based on overall variability of the diffusion measure values in that window), the final value + and - the standard deviation.

The # of points in a window help in removing the ends of the fiber bundle where the accuracy of the regression is severely affected because the kernel window does not have enough samples because of some fibers being shorter in length than others. These can be removed, if needed, prior to further analysis.

- The second output file is generated with the same name (and same location) as given in <output_file> but with '_all' appended to the name at the end.

  This file contains the results for all the allowed scalar diffusion parameters namely FA, MD, FRO, lambda 1, lambda 2, lambda 3, axial and radial diffusivity. But here there is no flexibility in the noise model chosen or the MLE used. Kernel regression for all scalar measures, except FA, uses Gaussian nose model with Mean as the MLE. FA uses Gaussian noise model but allows the flexibility of using a given quantile (given by quantile_percent input). This is because FA being a normalized measure, is highly sensitive to Partial Volume effects and quantiles might prove to be more robust in reducing the PV effects as compared to a simple Mean.

  This file looks as below:
  Cut Plane Origin: 64.5328 85.0293 27.8196
  Cut Plane Normal: -0.984806 0.164354 0.0560822
  Noise Model: Gaussian
  Statistics: Quantile(for FA), Mean (for other diffusion measures)
  Quantile Percent(for FA): 60

  Arc Length , FA , MD , FRO , l1 , l2 , l3, AD, RD

  -20.4993,0.0908912,0.00132388,0.00229939,0.00144178,0.0013289,0.00120097,0.00144178,0.00126493

So based on the most important scalar diffusion measure, the user can try different noise models and MLE for this measure while still getting a general idea (Gaussian with Mean) for all the other measures simultaneously.

Both the files are in .fvp format which is basically either a tab or comma separated file with a few lines of header. These are very easily imported into a spreadsheet environment for further analysis of the results and plotting options.

<scalar_diffusion_parameter> : Currently the code can generate results for the following scalar diffusion measures: FA, MD, FRO, Lambda 1, Lambda 2, Lambda 3 (where Lambda 1>=lambda 2>= lambda 3.), axial diffusivity (=lambda 1) and radial diffusivity (average of lambda 2 and lambda 3). More measures can easily be added to the tool for analysis.

<arc_length_step_size> : During arc length parametrization, this value would tell the step size to be taken to define the kernel window location along the fiber tract. This is defined in units of arc length itself(based on whether we are in world (eg 2 mm) or image space (2 voxels)). A very high value would give a very sparse sampling along the tract while a very small value may be too dense, not allowing enough points to lie in a given kernel window. Usually a value of 1.5 works well to avoid over or under sampling.

<bandwidth>: Given a kernel window location given by the arc length step size, this value defines the width of the window. The kernel window is taken to be 1 standard deviation (or bandwidth) on each side of the arc length location. This value should be equal to or slightly higher than the arc

length step size to avoid situations where samples are missed due to very small windows and a large arc length step size, creating gaps between kernel windows. For arc length step size of 1.5, a bandwidth of 2 is good. It is defined in units based on Image or world space (just like step size).

<noise_model> : The tool currently allows two noise model choices- **Gaussian and Beta**. The comparison between these two models has been done later in a separate section.

<kernel_window_number> : For visualization purpose, user can give a kernel window number for which the exact distribution needs to be studied. The distribution of sample points in this window are saved in text files which can then be viewed using Matlab or other plotting softwares. (This feature exists in the code but is yet to be tested for interoperability).

<Maximum_likelihood_estimate> : Possible options are Mean, Mode, Quantile (including the 50% quantile or the median). Currently the following combinations are allowed:

- Gaussian with Mean
- Gaussian with Mode (same as Gaussian with mode)
- Gaussian with any percentage quantile (50% gives the median)
- Beta with Mean
- Beta with Mode

<Quantile_percent>: The percentage quantile (50 gives the median value)

Compilation requirements - Needs ITK, VTK packages

Details of the concept and Comparison between Noise Models and MLE options

1. I first tried applying only a Gaussian kernel with a **Gaussian-distributed-noise** model as below:

   $Y_i = f(X_i) + \varepsilon_i$,i=1,2,....,n

   where ε_i is normally distributed noise.

   For a parameter like Fractional Anisotropy, the Gaussian-distributed-noise assumption has certain straightforward problems:

   - The support of the Gaussian distribution is $(-\infty, \infty)$ while FA values are always in the range [0,1].
   - The normal distribution is symmetric about the mean. This may not be true for FA even if we work with the assumption that FA curves are unimodal.

1. Applying **Gaussian smoothing with a Beta-distributed-noise** model.

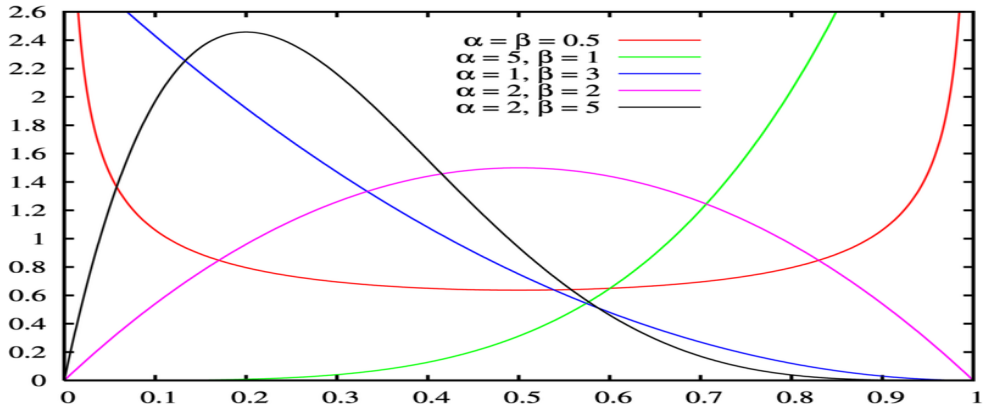
Here are some reasons as to why a Beta noise assumption is more appropriate than a Gaussian noise assumption:

- The support of Beta function is [0,1] which is exactly the same as the FA range, making it a more viable option.
- Using different α **,**β pairs, we can make the curve skewed, thus removing the limitation of being symmetric about the mean. For $\alpha = \beta$, the beta curve becomes symmetric about mean just like a Guassian curve.
- Referring to a presentation by Matt Gribbin and Michele Poe on Statistical Analysis of DTI data, FA is equivalent to a departure from sphericity and can be represented as a variable $B= \sqrt{(1- FA^2)}$, where B is beta distributed.
- Another finding shows that FA values when considered throughout the brain region, are beta distributed. Thus it could be possible locally too. Using a beta noise model would give a chance to study this possibility locally, in different regions of the brain.

Limitations with the beta-noise model are:

- The beta distribution curve varies highly in shape based on the values of α **,**β. To understand this, the below plot shows the shape of the beta curve for different values of α **,**β. For $0<\alpha$ **,**$\beta<1$, the curve is convex. For $1< \alpha$ **,**$\beta<2$, the curve is either continuously increasing or decreasing. For α**,**$\beta>=2$, the curve is unimodal. For $\alpha = \beta$, the curve is symmetric about 0.5.

Shape of the Beta distribution curve for different values of (α ,β) pairs.



Thus if we use a beta noise model, we must clamp the α ,β lower limits to be >= 2 (assuming for now that FA curves are unimodal).

Considering the above advantages, I have used a beta noise model for the FA data. Since I still don't know the actual distribution of FA values, this is just one of the possible approaches. Specially for cases with non-unimodal curves, this model will not be appropriate. Hence, this project's results are open to further discussion and critique.
Moreover, most of these advantages don't hold as we move to other scalar diffusion measures like MD or lambda values. So for these, the beta noise model might not be the best option.

1. **<u>Applying Non Parametric Regression on the input data</u>**

For my project, the data set that I have used is a fiber bundle from the genu region of the brain (the one which connects the left and the right hemispheres of the brain). It is a bundle taken from an atlas of brain images and has been pre-filtered to remove major outliers.

The non-parametric noise model used is:

$Y_i = \text{Beta}(\alpha(l_i) , \beta(l_i))$,i=1,2,....,n for n pairs of (l_i,Y_i) iid random variables

where Y_i gives the estimated FA value for the i_{th} cross section along the fiber bundle.

Here Beta stands for the Beta distribution, with alpha and beta as the two shape parameters and l_i are the arc length values where the cross sections would be defined.

This model assumes that the noise is Beta distributed. To find the FA estimate, the below algorithm flow is used:

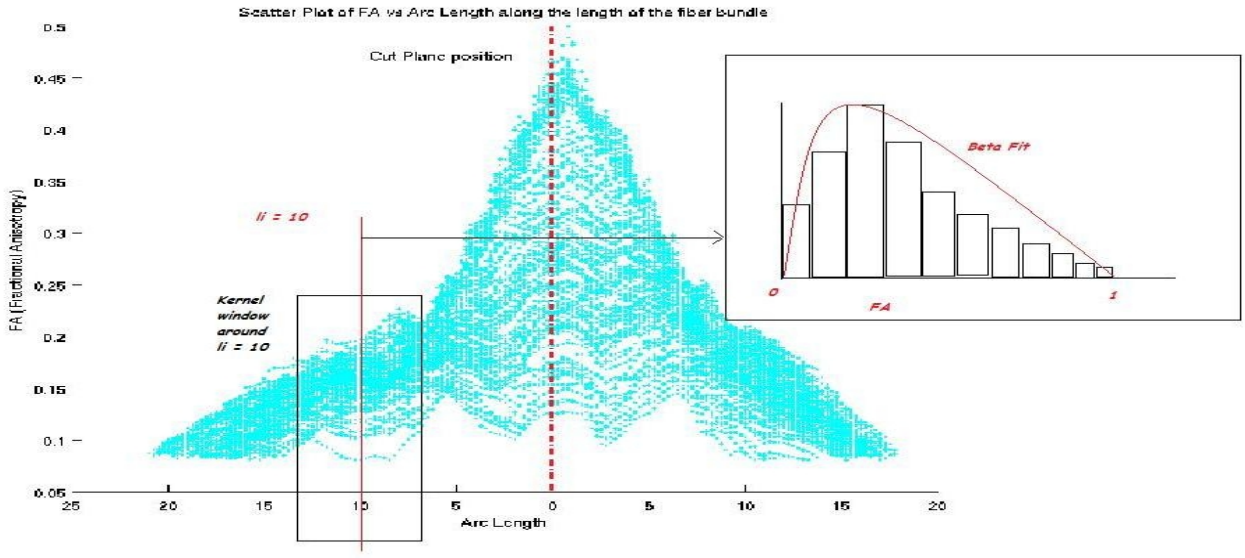
1. First, a **Gaussian kernel** is used along the Arc Length axis for smoothing using weighted accumulation.
2. Using these weighted values, **alpha and beta parameters** are estimated using parameter estimation, for each kernel window (that is, at each l_i) .

3. These estimated alpha, beta values can be used to find the FA estimate for the window assuming that the noise in that window is beta distributed (with respect to the FA values).

4. **Mode of the beta distribution** in a window is used as the Maximum Likelihood Estimator for the FA. This is because if we consider a cross section of a fiber bundle, the fibers lying towards the boundary may not be the ones we want. This would cause partial voluming mainly at the edges of the bundle. So when we get a beta fit for a cross section, the ends might show lower FA values due to the partial volume effect. Thus, taking the mode of the beta curve is a better estimation than taking the mean (specially when the beta curve is skewed).

Visual Explanation of the Algorithm

The below image explains the algorithm. The black box around $l_i = 10$ shows the kernel window. Gaussian weights are applied within this window. These weights can be thought of as forming a weighted histogram at $l_i = 10$ with respect to the FA values. This is shown in the side window. A beta fit is applied to this distribution and the Mode of this beta curve gives the FA estimate for the window at $l_i = 10$.
Similar procedure applied to all values of l_i gives the final fitted curve using Gaussian smoothing and beta noise model.



The detailed algorithm is explained as below:

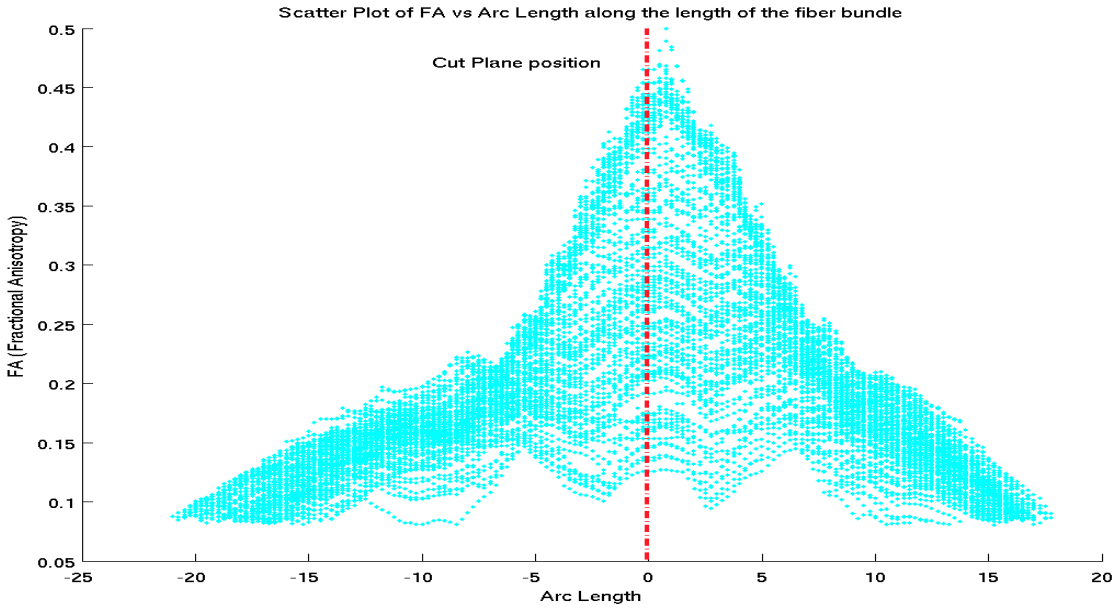
1. The program accepts the below command line inputs:
   - a fiber file- this file contains the sample points on all fibers in a fiber bundle with their (x,y,z) coordinate locations, the corresponding FA, Mean Diffusivity and other parameter values.
   - parameter to be analyzed- The parameter to be analyzed has been limited to being FA for this project
   - step size for arc length- Will decide the spacing between centers of the kernel windows.
   - standard deviation for the Gaussian kernel- Will decide the amount of smoothing achieved.

2. The fiber file is read and using the (x,y,z) coordinates and the definition of the plane (in

terms of the plane origin and the plane normal), the origin is computed as the closest point of intersection of the plane with the bundle. The point-to-plane distance formula has been used here.

3. Moving from the origin to each end of each fiber in the bundle, the arc length (distance from the origin to the current sample point) and the corresponding FA value is stored an a data structure. **NOTE:** Care needs to be taken to have all the fibers oriented in the same way, that is, with their start and end points defined coherently with respect to each other. If this is not taken into account, then some fibers may appear shifted in the final plot due to opposite orientations.

4. The Arc length versus FA list is sorted by length. The scatter plot of the FA versus Arc length values is shown below. The Red line indicates the position of the cut plane as defined by the user.



5. A **Gaussian kernel** is used along the Arc Length axis for smoothing using weighted accumulation. For each cross section defined on l_i :

   - Define a kernel window by a range of arc length values which are 1 standard deviation away from l_i .
   - Find Gaussian weights for each sample point in the current window at l_i using:

$$w_k = (\exp -(l_k - l_i)^2/2*std\_dev^2) / (std\_dev * (\sqrt{2}*pi))$$

   where k=1,2...n and n is the number of sample points lying in that window

   - Normalize the weights generated for each point by dividing them by the sum of all weights for the current window.

1. Now find the **Weighted Sample Mean** using normalized weights w_k:

$$y`(l_i) = (\Sigma^{n}_{k=1} w_k * y_k) / (\Sigma^{n}_{k=1} w_k)$$

where y_k are the FA values of all the sample points in the window and $y`(l_i)$ is the weighted sample mean of the FA values.

2. Find the **Weighted Sample Variance** using:

$$v`(l_i) =[ \Sigma^{n}_{k=1} w_k * (y`(l_i) - y_k)^2] / (1 - V2)$$

where $V2 = \Sigma^{n}_{k=1} w_k * w_k$,
$y`(l_i)$ is the weighted sample mean as computed above,
$v`(l_i)$ is the weighted sample variance

NOTE: The sample variance computed is the **unbiased** estimator for the population variance. The factor of (1 – V2) in the denominator makes it unbiased.

3. The probability density function of a Beta distribution is defined as:

$x^{(\alpha-1)} (1-x)^{(\beta-1)} / B(\alpha ,\beta)$ where x lies in [0,1] and $\alpha ,\beta >0$.

$B(\alpha ,\beta)$ is a normalization constant to ensure that the total probability integrates to 1. It is the integral of the pdf from 0 to 1.

4. Thus to find the pdf, we need to find an estimate for α ,β for the current window. This is done by parameter estimation using method-of-moments.

$\alpha (l_i) = x` * [ (x`(1-x`)/v`) - 1]$
$\beta (l_i) = x` * [ (x`(1-x`)/v`) - 1]$
where x` is the weighted sample mean and v` is the weighted sample variance of the population.

So using the weighted sample mean and variance calculated above, we get the below equations to **estimate $\alpha (l_i)$ and $\beta (l_i)$.**

$$\alpha (l_i) = y`(l_i) * [ (y`(l_i) * (1- y`(l_i)) / v`(l_i)) - 1]$$
$$\beta (l_i) = y`(l_i) * [ (y`(l_i) * (1- y`(l_i)) / v`(l_i)) - 1]$$

where $y`(l_i)$ is the weighted sample mean and $v`(l_i)$ is the weighted sample variance as calculated in the above steps.

5. Since we are assuming the curve to be unimodal, so we clamp the $\alpha (l_i)$, $\beta (l_i)$ estimates to be >= 2 in cases where they are <2.

6. Considering the Partial voluming at the boundaries of the fiber bundle, we use the **mode** of the beta distribution as the **maximum likelihood estimate** (instead of the mean).

For $\alpha (l_i)$, $\beta (l_i) >1$, mode is defined as

$$mode(l_i) = (\alpha (l_i) - 1) / (\alpha (l_i) + \beta (l_i) -2)$$

This estimated mode is the value of estimated FA for the current window (cross section) defined at l_i.

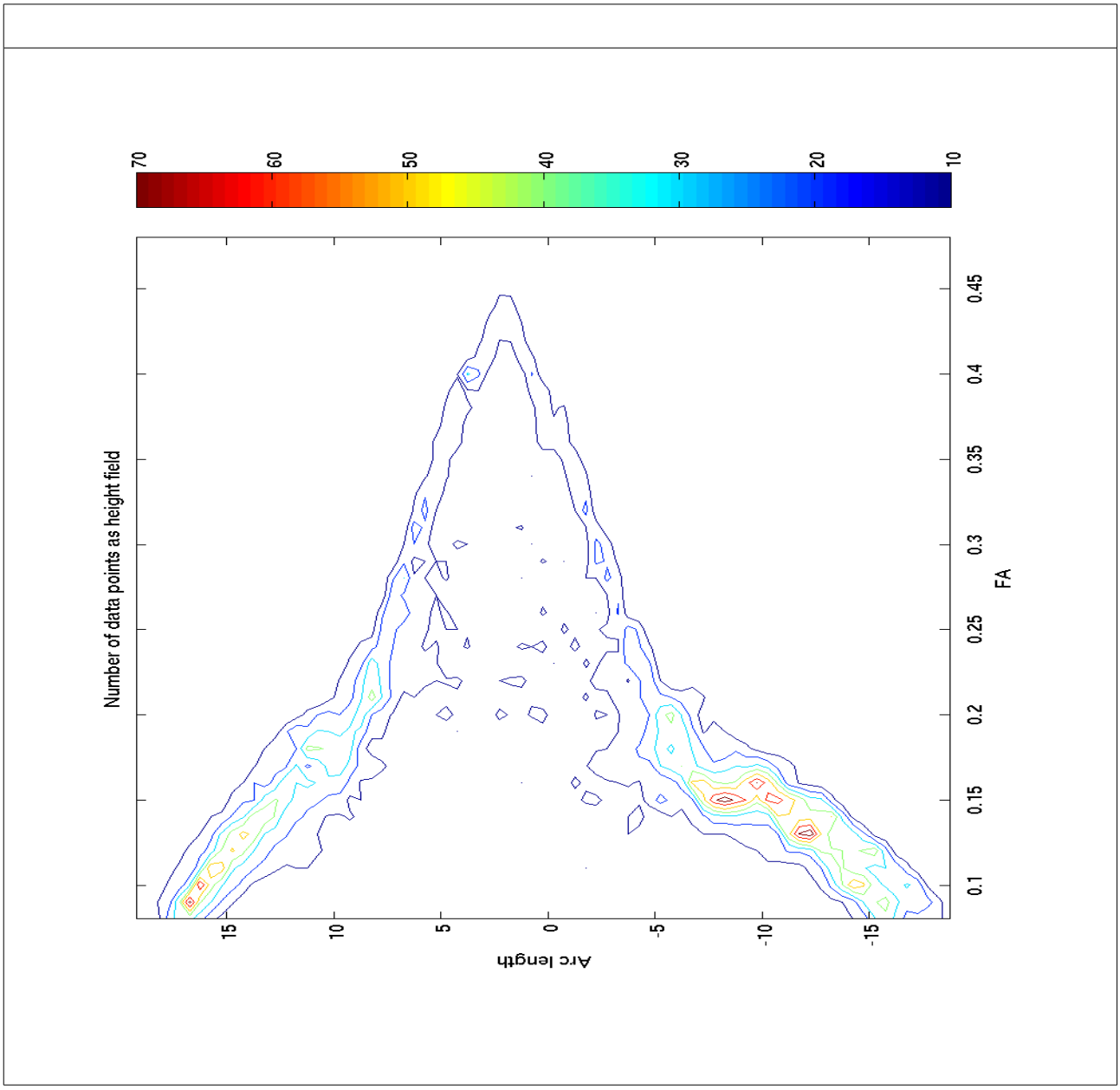
7. The above steps are repeated for all l_i values which gives us the final curve as (mode(l_i), l_i).

8. To find the **standard deviation** for the above fitted curve, we use the below algorithm:
   - For each l_i (that is, for each kernel window), we compute the sum of squared differences of the FA values for each sample point with the estimated FA value (as found above) for that window.
   - This sum is normalized with the number of sample points in that window to give the standard deviation for that window.
   - In the final result, apart from the fitted curve, I have also plotted the (curve value + standard deviation) and the (curve value – standard deviation) for each l_i.

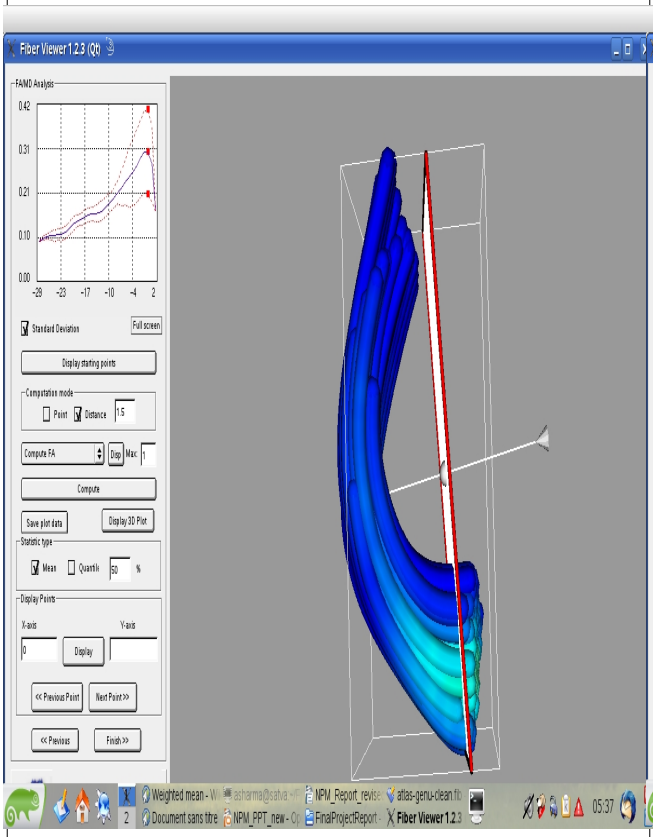
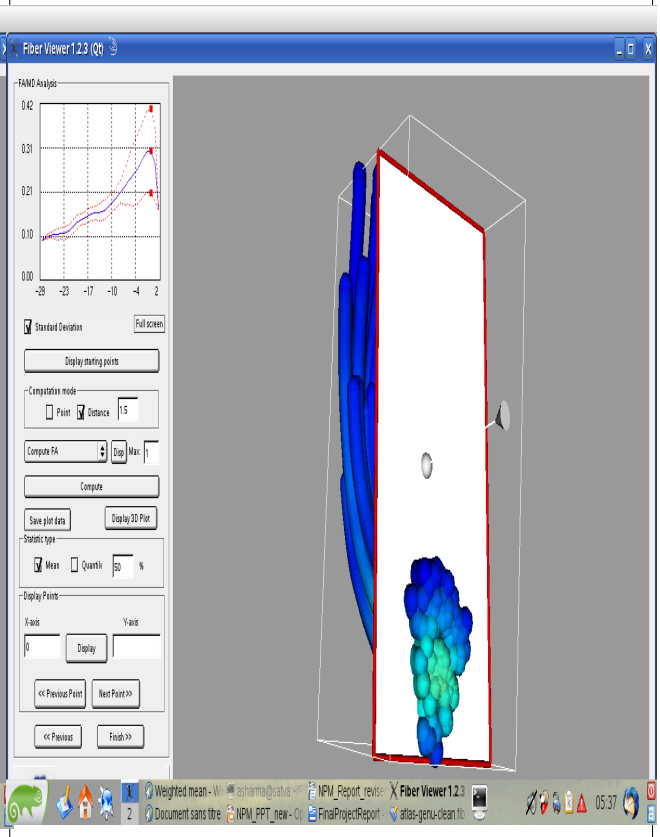
Experiments and Results

NOTE: The plots have been generated using Matlab and the .fvp files and other text files generated by the tool. The Matlab scripts are not currently part of this tool. The user can use any other plot options with these files to get similar results.
Also, most of the below sample results have been generated for FA.

1.5. The below figures show the number of data points plotted as a height field (the color code depicts the number of data points) for the arc length vs FA scatter plot. This visualization helps in understanding the accuracy of the fit since the 2D scatter plot fails to show the overlapped sample points. So areas with higher overlap (for instance, the extreme ends of the fibers) tend to show lower standard deviations when compared with the fitted values. This is expected since the samples agree more with each other causing more overlap in those regions. (This part of the code is not incorporated in the current version as it is still being tested).



1.6. To understand the Partial Volume effect on this bundle, I have cut the bundle close to the middle so as to view the cross section. Now when we compute the Mean FA value using Fiber Viewer, we see that the high FA values are mostly in the center of the cross section while the boundaries have low FA values. This explains the large variance of FA values close to the cut plane origin in the above scatter plots. At the ends, the scatter plot values mostly overlap which is expected since the Fiber Viewer results also show a similar color for all the fibers at the ends.

Result of viewing the fiber bundle in Fiber Viewer tool. Observe the change in color in the middle owing to higher FA values as compared to the fiber ends.	The fiber has been cut close to the middle. The Cross Section shows a range of colors here showing the Partial Volume effect and explaining the high variance observed near the origin in the scatter plots.
	

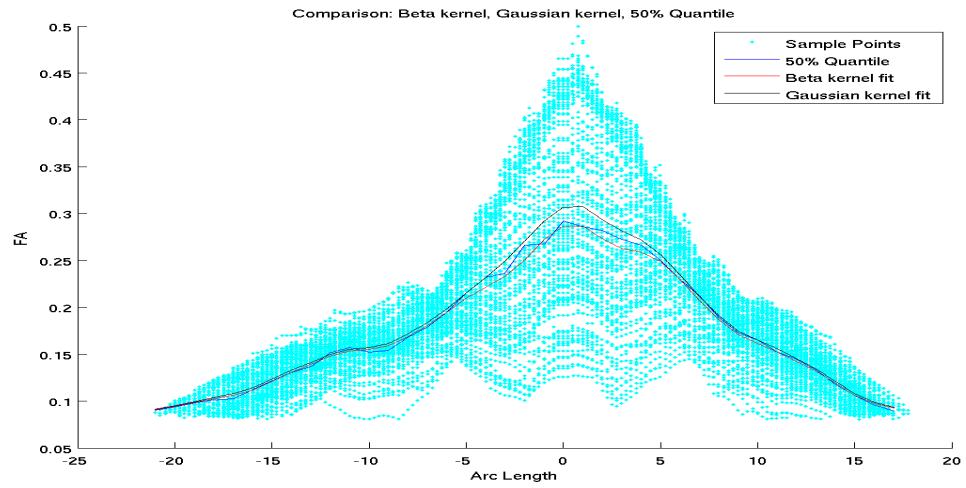
1.7. I first tried a simple Gaussian kernel to estimate the FA value for each window. This was to verify my code against the results of the Fiber Viewer tool which computes a simple mean for each cross section. My results looked similar but not exactly the same (as was expected since a kernel regression would theoretically give better estimates than a simple mean at each cross-section).

1.8. Then I tried Quantiles (which I had implemented as part of my research). The Quantile value in a given window uses the Gaussian weights to create the histogram (that is, weighted accumulation in the histogram rather than adding 1 to the bin value for each sample point). The normalized histogram was then used to find the 50 % Quantile for the same fiber bundle.

1.9. Then I applied the Beta noise model as discussed above. The three results have been combined here in a single plot for comparison.

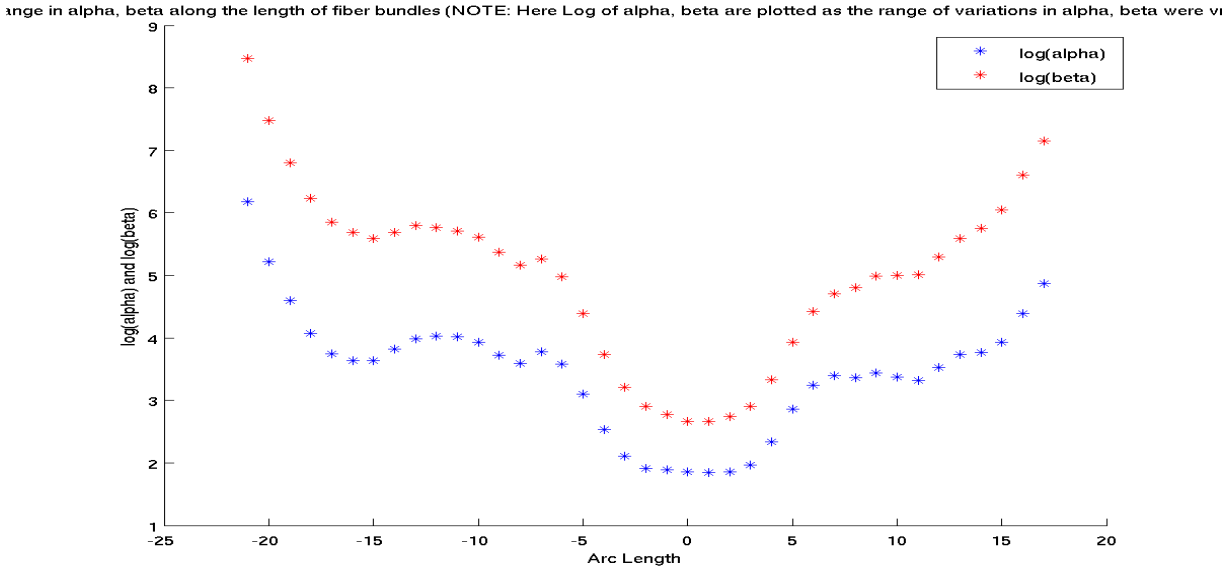
The plot shows the results of a fit with a simple weighted mean using Gaussian kernel, 50% Quantile using Gaussian weights, and A Beta noise model fit with Gaussian smoothing.

We can see here that the results disagree the most in the center of the plot where the variance of the data is very high. At the extreme ends of the bundle, the curves almost overlap. The comparison of these approaches using the standard deviations in them has been shown later.

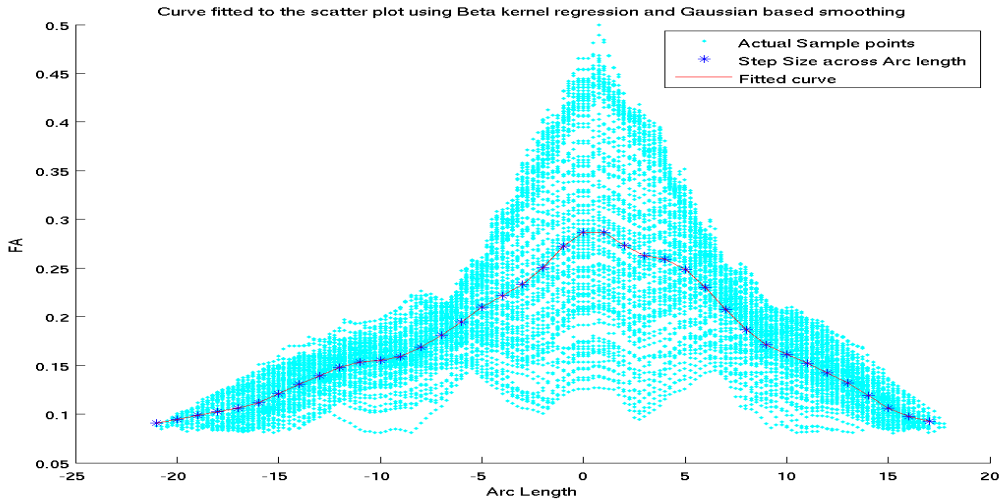


1.10. The below figures show the results of the Beta Noise model in detail.

The below figure shows the **$\alpha(l_i)$, $\beta(l_i)$ estimations** as generated from the real data using parameter estimation technique. Here I have plotted the log of these values to enable them to fit in a single plot with appropriate scaling. We can observe here that the $\beta(l_i)$ values are always higher than $\alpha(l_i)$ which means that for the given data, the beta curve in each window is skewed rather than being symmetric about the mean. (For reference see the figure with beta curves for different alpha, beta pairs.)

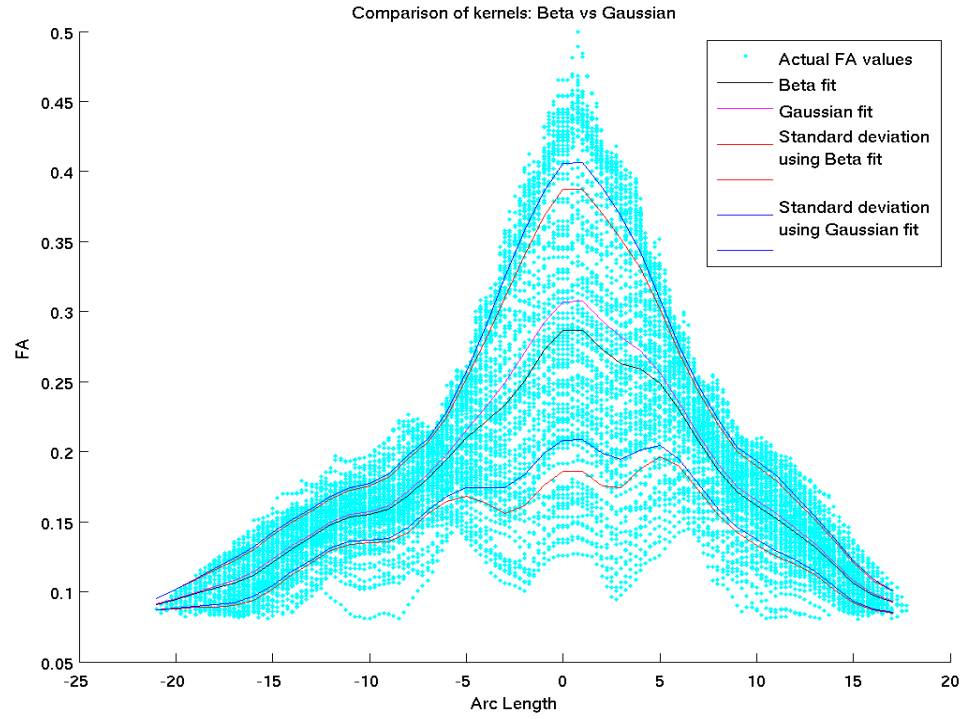


The below result shows the final fit using Gaussian smoothing and Beta noise model. The crosses show the locations of the cross sections along the fiber bundle length.

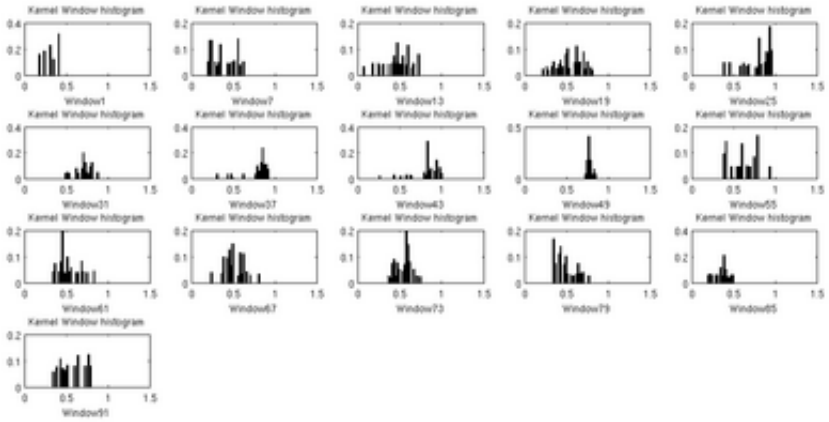


1.11. The following results will be comparing the fit obtained using weighted mean with a Gaussian kernel with that of a Beta noise model with Gaussian smoothing.

The below plot shows the fit with a simple Gaussian kernel and the fit with the beta noise model. It also plots the standard deviations (Fitted value + Standard deviation) and (Fitted value - Standard deviation). We see that at the ends the standard deviations are very low and they are very high in the middle of the fiber where FA values show a large variance. The large variance of FA values could be due to partial voluming in the middle of the fiber length causing low FA values at the boundaries of the cross sections, as explained previously.



The below image shows how we can visualize individual kernel windows and the distribution of the scalar diffusion measure along the fiber tract. This image shows every 5th window along the tract. This helps in understanding the distribution of the measure and better understand which noise model or MLE would best represent the trend in teh data.

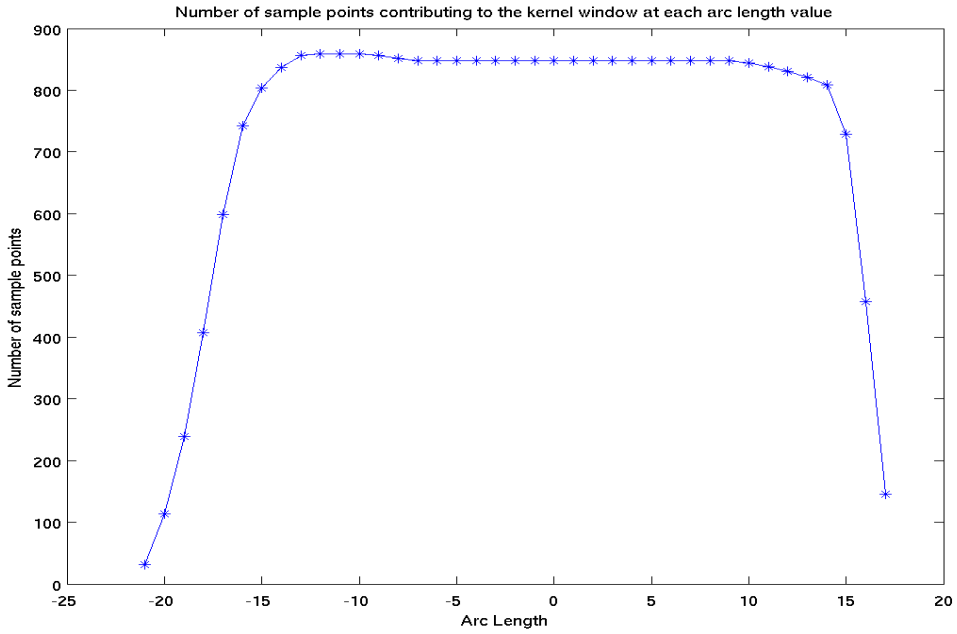


2. Issues to consider

2.1. One of the findings show that the FA values when considered throughout the brain region are beta distributed. This leads us to an important point. The beta distribution observed throughout the brain could just be a result of partial voluming effects because of considering a very heterogeneous region rather than the FA values being actually beta distributed. (Partial volume effect occurs in medical imaging when a single voxel contains a mixture of multiple tissue values).Thus assumption of a beta noise may not be true locally for specific fiber bundles in the brain.

2.2. Since the fibers in a bundle may have different lengths, so at the ends of the bundle, the number of sample points may be very low. This may effect the robustness of the estimations. The below figure shows the histogram of the number of points contributing to the kernel window at each l_i value (that is, at each cross section). For instance, at the left end, the number of values to the right side of l_i would be much more than those towards the left side. Hence, the right side samples would bias the result. This puts a question mark on the validity of the kernel statistics at the end of the fiber bundle.

A histogram showing the number of sample points contributing to the kernel window at each cross section. We see that at the fiber ends, the number drops very low thus making the validity of the kernel regression results doubtful.



2.3. For a specific fiber bundle, the FA distribution might not be unimodal making it tough to estimate the shape parameters α ,β for a beta distribution. In case the curve is bimodal, the beta fit would not be the appropriate one.

2.4. Clamping α, β to be >=2 removes any chances for the curve to have a continuously increasing or decreasing shape. Moreover for [0,1] the curve actually becomes convex thus giving 2 possibilities of a maximum value. Thus for [0,1), the mode formula does not hold true.

2.5. This code needs to be modified for parameters other than FA since they may have a different support for the function. For a function with support in the range of [l,h], the sample mean x` would be (x`-l)/ (h-l) and the sample variance v` would be v`/(h-l)^2. Apart from this, for non-FA parameters, the beta noise might in fact not be the appropriate model at all.

References

Much of the information comes from the **Wikipedia website**, specially on the below topics:

1. Non Parameteric Regression: http://en.wikipedia.org/wiki/Nonparametric_regression
2. Kernel Regression: http://en.wikipedia.org/wiki/Kernel_regression
3. Beta Distribution: http://en.wikipedia.org/wiki/Beta_distribution
4. Gaussian Distributions: http://en.wikipedia.org/wiki/Gaussian_distribution
5. Formulae for Weighted Sample Mean and Unbiased Weighted Sample Variance: http://en.wikipedia.org/wiki/Weighted_variance#Weighted_sample_variance

Other references:

A Presentation on "Statistical Analysis of DTI data" by Matt Gribbin, Michele Poe, in collaboration with Dr. Keith Muller and Meagan Clement, Oct 10, 2005

Isabelle Corouge, P. Thomas Fletcher, Sarang Joshi, Sylvain Gouttard, Guido Gerig, "Fiber Tract-oriented statistics for quantitative diffusion tensor MRI analysis", Medical Image Analysis 10 (2006), 786-798

Denis Le Bihan, Jean-Francois Mangin, Cyril Poupon, Chris A. Clark, Sabina Pappata, Nicolos Molko, Hughes Chabriat, "Diffusion Tensor Imaging: Concepts and Applications", Journal of Magnetic Resonance Imaging 13:534-546 (2001)

R. Douglas Fields, "White Matter", Scientific American, March 2008

Documentation on Fiber Viewer:
http://www.ia.unc.edu/dev/tutorials/FiberViewer/main.htm#StartingFV

"Statistical Computing with R" by Maria L. Rizzo