APPENDIX A

Methodological Details for the Pre-Analysis

The pre-analysis for a single voxel follows the flowchart in Fig A1. First, a spherical analysis window is created around the voxel of interest such that all voxels at a distance less than the radius of the spherical window (winRad) from the voxel of interest are included in the window of analysis. The initial winRad is set to double the minimum voxel dimension (minVoxDim). For example, for an image with an x-y resolution of 0.3 μm and a z-step size of 0.5 μm, the initial window radius would be 0.6 μm. In each spherical window, all non-zero voxels which are not connected to the voxel of interest at the center of the window through a series of adjacent faces, edges, or corners of non-zero voxels (i.e., 26-connected) are set to zero. The removal of non-connected voxels prevents a source of error from fibers which are in close proximity but not in contact with one another.

After forming the spherical window of analysis, two metrics are determined: the number of objects on the surface of the window, and the volume ratio of the surface objects to the full surface of the spherical window. To calculate these metrics, the intensity of all voxels which do not have a face or edge adjacent to the outer surface of the spherical window is set to zero, leaving only the regions of the object within the window which form the outer surface of the window. The number of distinct non-connected objects after this procedure (numObjs) are counted and used in the subsequent decisions. Possible values for numObjs include all non-negative integers. Additionally, the ratio between the sum volume of all the surface objects and the volume of the full surface of the spherical window (surfRat) is calculated and is similarly used in subsequent decisions. The value for surfRat can be any real number between zero and one. The pre-analysis flowchart recognizes five possible combinations of numObjs and surfRat, which are each depicted in the main body of the paper in Fig 2. The first possibility is where surfRat > 0.5, indicating that the spherical window is too small for the fiber. The second possibility is where numObjs = 0 and surfRat ≤ 0.5, indicating that the voxel of interest is part of a smaller object which is not classified as a fiber. The third possibility is where numObjs = 1 and surfRat ≤ 0.05, which indicates the voxel of interest is at the end of a fiber. The fourth possibility is where numObjs = 1 and 0.05 < surfRat ≤ 0.5, indicating that the spherical window is too small to contain the entire fiber, as is the case when surfRat > 0.5. The fifth and final possibility is where numObjs > 1 and surfRat ≤ 0.05, which indicates that the spherical window contains a full cross-section of the fiber. Note that the pre-analysis does not distinguish between a full fiber (numObjs = 2) and a fiber intersection (numObjs >2) but instead groups these categories together as numObjs > 1. Fiber intersections are identified in the subsequent full analysis.

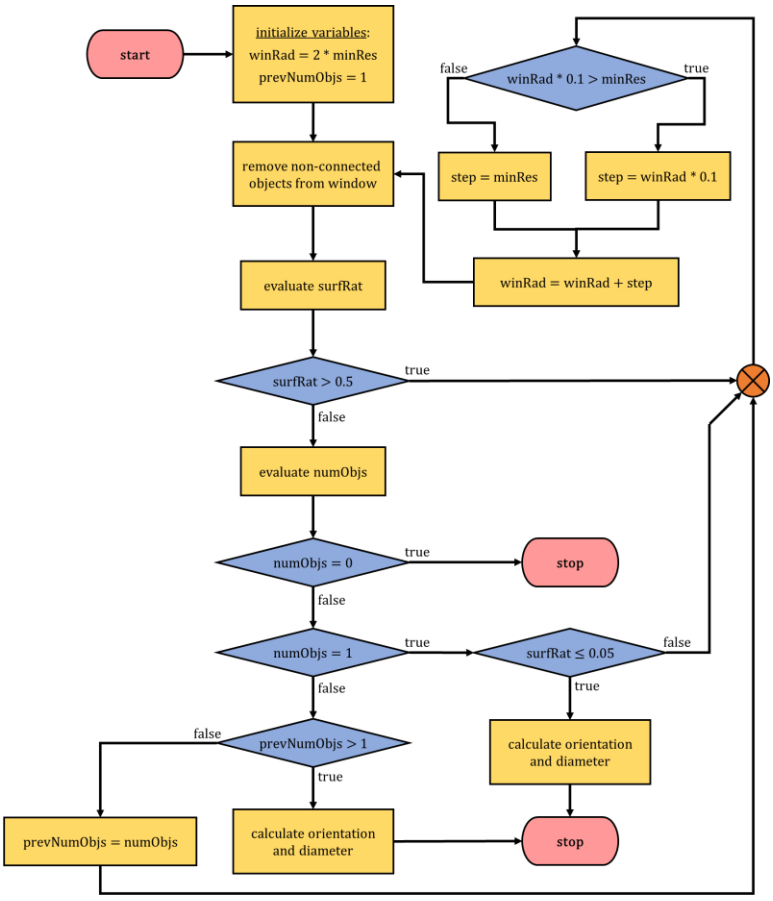


Fig A1. Flowchart depicting the steps of the pre-analysis portion of the algorithm.

The pre-analysis begins with a small spherical window and proceeds by incrementally increasing winRad until the algorithm identifies a small non-fiber object (numObjs = 0), a fiber end (numObjs = 1 and surfRat ≤ 0.01), or the most general case where a full fiber cross-section is contained within the spherical window (numObjs > 1 for two successive winRad). While winRad is less than 10*minVoxDim, winRad increases by minVoxDim for each iteration. When winRad exceeds 10*minVoxDim, winRad increases by 0.1*winRad for each subsequent iteration. Because of the voxelated nature of the spherical window and potential irregularities in the fiber shape, it is possible to have an aberrant situation where numObjs > 1 without fully containing a cross-section of the fiber within the window. Consequently, the flowchart requires that numObjs > 1 for two successive values of winRad in order reach the terminus of the flowchart. Where a fiber is identified, initial estimates of orientation and diameter are calculated using the current winRad. While some voxels fit a criterion for being classified as a fiber end during the pre-analysis, this information is not saved or passed on to the full analysis.

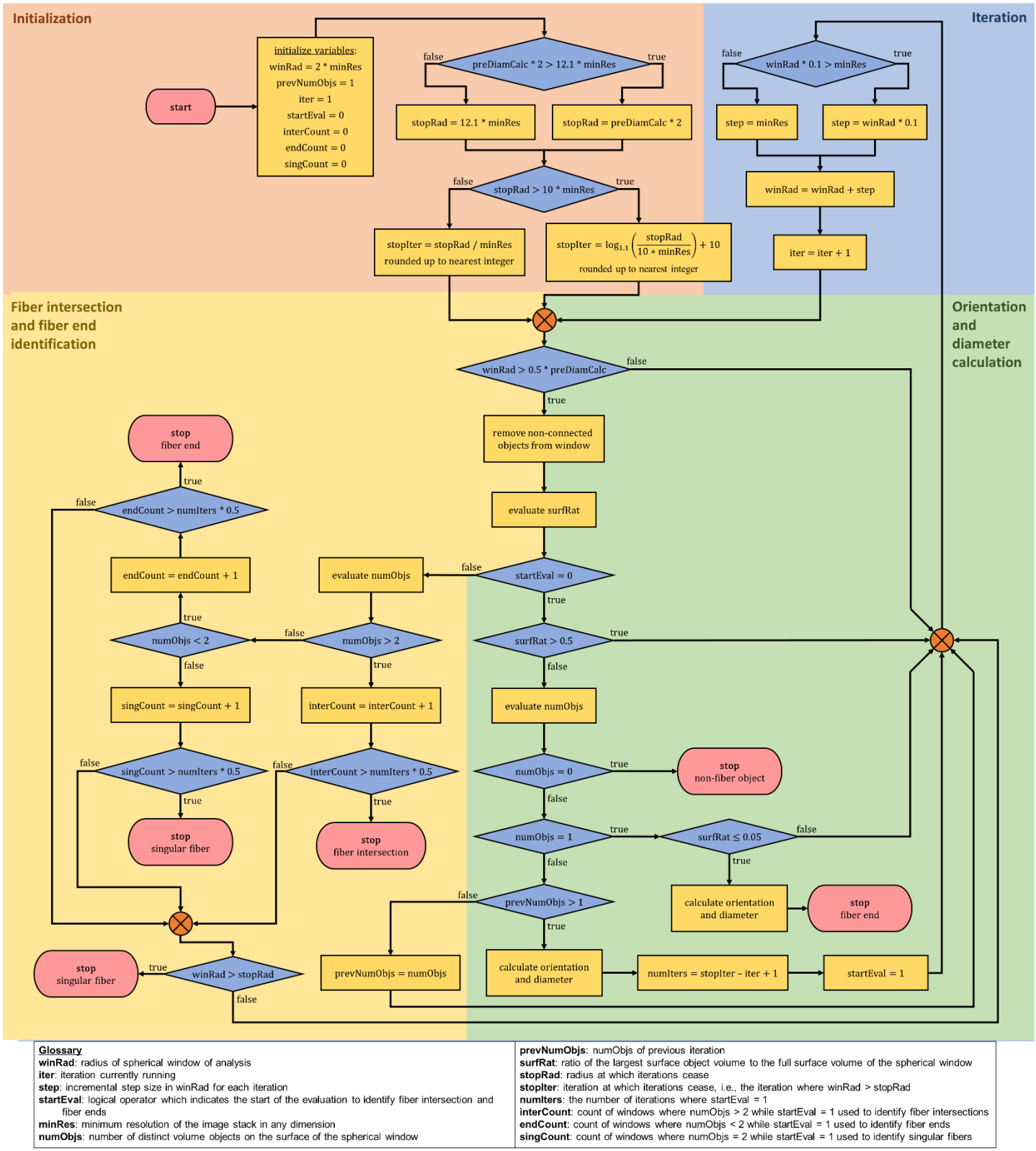
Methodological Details for the Full Analysis

Before the full analysis begins, orientation diameter estimates are assigned to each non-zero intensity voxel which was not analyzed during the pre-analysis. Specifically, the diameter estimate is the maximum diameter value computed for a voxel at a distance less than half of the estimated diameter for that voxel; or, if no voxels meet the former criterion, the estimated diameter is set to the estimate from the nearest voxel analyzed in the pre-analysis. The orientation estimate is the value for the nearest voxel from the pre-analysis.

Then the full analysis begins following the flowchart in Fig A2. After initializing variables, the first steps of the of the full analysis are identical to the pre-analysis with the exception that winRad starts at half of the estimated diameter for the voxel of interest in order to reduce computational time. The winRad increases incrementally until a small non-fiber object (numObjs = 0), a fiber end (numObjs = 1 and surfRat ≤ 0.01), or a full fiber (numObjs > 1 for two successive winRad) is identified, as in the pre-analysis. If a fiber is identified at this point in the process, the fiber orientation and diameter are calculated as described in the main body of the paper. The analysis then continues by evaluating numObjs for every incrementally increased winRad until winRad is greater than double the initial diameter estimate or until a stop criterion is reached to classify the voxel. If the majority of numObjs within this range are greater than 2, then the voxel of interest is identified and labeled as a fiber intersection. Similarly, if the majority of numObjs within this range are less than 2, then the voxel of interest is identified and labeled as a fiber end. When neither of these conditions are met, the voxel of interest is identified and labeled as a singular fiber.

When the full image stack has been analyzed, each individual voxel or group of voxels classified as a fiber intersection is evaluated to determine the number of adjacent singular fiber objects. Where there is only one adjacent fiber object, the classification for that group of voxels is changed from an intersection to a singular fiber. This step serves to remove small groups of voxels classified as intersections which may be near an actual intersection but not connected to other voxels at the intersection, thereby cleaning the image and leaving only intersections which separate distinct fiber objects.

Lastly, voxels which are classified as a fiber intersection, a fiber end, or where the calculated diameter is greater than the distance between the voxel and the edge of the image stack have their respective orientation and diameter removed from the final output. Voxels identified as intersections have their calculated orientation and diameter removed because a singular orientation or diameter is undefined where multiple fibers are present. At fibers ends, the lack of a continuous fiber through the window of analysis can cause elevated errors in the calculations of orientation and diameter, and therefore these values are also excluded. Similarly, the lack of complete information near the edges of the image stack where parts of fibers are outside of the image area can elevate errors; therefore, exclusion of orientation and diameter calculations near the edges of the image is warranted.



Glossary

winRad: radius of spherical window of analysis
iter: iteration currently running
step: incremental step size in winRad for each iteration
startEval: logical operator which indicates the start of the evaluation to identify fiber intersection and fiber ends
minRes: minimum resolution of the image stack in any dimension
numObjs: number of distinct volume objects on the surface of the spherical window
prevNumObjs: numObjs of previous iteration
surfRat: ratio of the largest surface object volume to the full surface volume of the spherical window
stopRad: radius at which iterations cease
stopIter: iteration at which iterations cease, i.e., the iteration where winRad > stopRad
numIters: the number of iterations where startEval = 1
interCount: count of windows where numObjs > 2 while startEval = 1 used to identify fiber intersections
endCount: count of windows where numObjs < 2 while startEval = 1 used to identify fiber ends
singCount: count of windows where numObjs = 2 while startEval = 1 used to identify singular fibers

Fig A2. Flowchart depicting the steps of the full analysis portion of the algorithm. The analysis can be divided into four components: (1) initialization, (2) iteration, (3) orientation and diameter calculation, and (4) fiber intersection and fiber end identification.

APPENDIX B

Phantom image stacks were constructed to test the accuracy of the fiber analysis algorithm. For each phantom, single fibers were created by defining a line or curve in 3D space and assigning an intensity value to each voxel within a given distance from the line or curve. The intensity distribution across the width of a fiber was set to a Gaussian distribution to simulate what would be expected of a microscopy image. Where multiple fibers overlapped in a single voxel, the intensity for that voxel was defined from the fiber which would have the greatest intensity at that voxel. Any voxel containing overlapping or directly adjacent fibers was labeled as a fiber intersection. The actual fiber orientation and diameter were saved on a voxel-wise basis for each phantom.

As an initial test, and to visually demonstrate the algorithm, two phantoms were constructed with fiber orientations restricted to the x-y plane. One of these phantoms included parallel straight fibers with a range of diameters and a single perpendicular fiber to intersect each of the parallel fibers (Fig. 3a). This phantom was analyzed to qualitatively investigate how intersections between fibers of different size are identified. The second of these phantoms featured a fiber with non-constant diameter, curved fibers, and an intersection of more than two fibers (Fig. 3c). Again, this phantom was analyzed to qualitatively assess the effect of these features on the performance of the fiber analysis algorithm.

Sets of more complex phantom images were generated to quantitatively assess the accuracy of the fiber analysis algorithm and evaluate the effects of fiber diameter, density, and curvature on its performance. Each phantom was 256 x 256 x 256 voxels in size and contained fibers with randomly assigned locations, orientations, and lengths. To investigate fiber diameter, a set of ten phantoms containing forty straight fibers were created, where each phantom was assigned a fiber diameter from two to twenty a.u. in increments of two (Fig. 4a). For fiber density, a set of phantoms containing straight fibers (with a diameter of eight a.u.) were created, where each phantom included between twenty and two hundred fibers in increments of twenty (Fig. 5a). Fibers with 3D curvature which followed the path of a sine wave that oscillated in a direction defined by a helix were used in a set of phantoms to evaluate the effect of fiber curvature on the performance of the algorithm (Fig. 6a). This path of curvature was used because it represents a worst-case scenario with complex curvature with significant variation along the length of the fiber. For each phantom in this set, forty fibers were created with a diameter of eight a.u.. Fiber curvature was varied between phantoms by changing the amplitude of the sine wave: values between zero and twenty a.u. were analyzed in increments of two. Each phantom was analyzed using the fiber analysis algorithm to determine fiber orientation and diameter and to identify fiber intersections.

After independently evaluating the effects of fiber diameter, density, and curvature, these effects were combined to create a phantom image with high complexity to more closely resemble what may be encountered in images of biological tissue (Fig. 7a). This phantom included sixty curved fibers with randomly assigned location, orientation, length, diameter, wave amplitude (i.e., curvature), and intensity, where possible values for each metric fell within the ranges used in the previous phantoms. This phantom was manipulated in particular ways to synthetically mimic situations which arise in actual images. When acquiring an image stack using confocal or two-photon microscopy, the step size in the z-dimension is typically larger than the distance between voxels in the x- and y-dimension due to poorer resolution along z. To simulate this effect, the phantom was downsampled in the z-dimension and analyzed using a non-cubic voxel size corresponding to the downsampling rate (Fig 7(a)). In addition, varying levels of Gaussian noise were artificially added to the phantom (Fig 7(f)). After noise was added to the phantom, it was passed through a 3D Gaussian smoothing filter and thresholded based on intensity and object volume to isolate the fibers in the phantom from the background noise. The parameters for the gaussian filter and intensity threshold were also varied to determine the effect of fiber segmentation on algorithm performance. For each scenario, the manipulated phantom was analyzed to determine fiber orientation and diameter and to identify fiber intersections.

To demonstrate what types of distributions the axial spherical variance values represent, simpler phantoms containing straight fibers with various degrees of alignment were generated (Fig. 8a). Individual fiber orientations in these phantoms were defined by the weighted average of a randomly oriented vector and a user-defined

orientation vector, where the weights were varied accordingly for each phantom to produce a range of variances. The phantoms were analyzed using the fiber analysis algorithm, and the variance was calculated as described.