

Parallel Geometric Classification of Stem Cells by Their 3D Morphology

Derek Juba^{1,2}, Antonio Cardone^{1,2}, Cheuk Yiu Ip¹, Carl G. Simon Jr.³, Christopher K. Tison^{3,4}, Girish Kumar^{3,5}, Mary Brady² and Amitabh Varshney¹

¹ UM Institute for Advanced Computer Science (UMIACS), University of Maryland, College Park, MD 20742

² Software & Systems Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

³ Biosystems & Biomaterials Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

⁴ Biomedical Technologies Group, Luna Innovations Inc., Charlottesville, VA 22903

⁵ Division of Biology, Office of Science & Engineering Laboratories, Center for Devices and Radiological Health, U.S. Food & Drug Administration, 10903 New Hampshire Ave, Silver Spring, MD 20993

E-mail: `derek.juba@derek.juba.name`

Abstract. There is a need for tools to classify cells based on their 3D shape. Cells exist in vivo in 3D, cells are frequently cultured within 3D scaffolds in vitro, and 3D scaffolds are used for cell delivery in tissue engineering therapies. Recent work indicates that the physical structure of a tissue engineering scaffold can direct stem cell function by driving stem cells into morphologies that induce their differentiation. Thus, we have developed a rapid method for classifying cells based on their 3D shape. First, random lines are intersected with 3D Z-stacks of confocal images of stem cells. The intersection lengths are stored in histograms, which are then used to train a Support Vector Machine (SVM) learning algorithm to distinguish between stem cells cultured on differentiation-inducing 3D scaffolds and those cultured on non-differentiating flat substrates. The trained SVM is able to properly classify the “new” query cells over 80% of the time. The algorithm is easily parallelizable and we demonstrate its implementation on a commodity Graphics Processing Unit (GPU). Use of a GPU to run the algorithm increases throughput by over 100-fold as compared to use of a CPU. The algorithm is also progressive, providing an approximate answer quickly and refining the answer over time. This allows further increase in the throughput of the algorithm by allowing the SVM classification scheme to terminate early if it becomes confident enough of the class of the cell being analyzed. These results demonstrate a rapid method for classifying stem cells based on their 3D shape that can be used by tissue engineers for identifying 3D tissue scaffold structures that drive stem cells into shapes that correlate with differentiation.

1. Introduction

There is a need for tools to classify cells based on their 3D shape. Since there is a well-established link between cell shape and cell function [4, 10, 19, 26, 27, 32, 43, 46], cell shape is frequently used to assess cell state. However, cell culture experiments are typically performed in a 2D format where cells are cultured on flat tissue culture polystyrene plates and the cell shape is assessed using 2D imaging and image analysis. In contrast, cells *in vivo* reside in a 3D microenvironment composed of an extracellular matrix. 3D scaffolds are being advanced for *in vitro* culture in order to mimic the *in vivo* environment and are also being used for cell delivery in tissue engineering therapies [28]. Thus, 3D cell shape analysis is required to understand how 3D scaffold microenvironments influence cell shape and function.

Herein, we introduce a machine learning approach for classifying stem cells based on their 3D shape. Lines are intersected with Z-stacks of 3D confocal images of stem cells and the intersection lengths are used to generate a histogram. The histograms are used to train a Support Vector Machine (SVM) learning algorithm to recognize stem cells cultured on differentiation-inducing 3D scaffolds.

Tissue engineering has been defined as an “interdisciplinary field that applies the principles of engineering and the life sciences toward the development of biological substitutes that restore, maintain, or improve tissue or whole organ function” [21]. Recent advances include many tissues and organs, including heart [34], bladder [5], and urethra [37].

The two critical parts of a tissue-engineered implant are stem cells and 3D tissue scaffolds [28]. Stem cells have the ability to differentiate down multiple lineages for regeneration of different organs and tissues. Tissue scaffolds are porous structures, made from polymers, ceramics or natural materials such as collagen, that provide a 3D template for stem cells to adhere to, differentiate, and generate new tissue.

Traditionally, control of cell function has been considered in terms of soluble factors, biochemical signaling and paracrine effects (one cell secretes a factor that stimulates nearby cells). However, recent work shows that the physical properties of the cellular microenvironment can also influence cell function. In particular, the chemistry [6, 24, 36], mechanics [18, 35], and structure [13, 26, 27] of the cell niche are important. The cell niche is the local microenvironment in which individual cells reside, and *in vivo* the cell niche is composed of extracellular matrix proteins such as collagen and fibronectin. A primary goal in regenerative medicine is to engineer scaffolds to provide a 3D microenvironment that enhances tissue regeneration.

Human bone marrow stromal cells (hBMSCs) are a mix of cells isolated from bone marrow which contains a population of multipotent adult skeletal stem cells that can differentiate into bone, fat and cartilage [14]. When cultured with appropriate stimuli, such as differentiating supplements, hBMSCs start down a pathway going from a multipotent stem cell towards a more differentiated cell type, such as an osteoblast, chondrocyte or adipocyte.

Previously, we demonstrated that 3D PCL (poly(ϵ -caprolactone)) nanofiber scaffolds drive hBMSCs into an elongated and highly-branched morphology that induces them to differentiate down an osteogenic lineage [26]. Osteogenic differentiation on nanofibers was marked by calcium deposition (positive Alizarin red staining) and a gene expression profile that matched control hBMSCs incubated with osteogenic supplements (measured by microarray analysis). Electrospinning is a process for making porous polymer scaffolds that have a nanofibrous structure that mimicks extracellular matrix [44]. When hBMSCs were cultured on 2D flat PCL films (spun-coat), they assumed a well-spread, polygonal morphology, that supported cell proliferation but did not induce osteogenic differentiation. Spin-coating is a process for applying uniform, polymeric, flat films to substrates where centrifugal forces are used to spread polymer solutions. Electrospinning and spin-coating can be used to make nanofiber scaffolds and flat films, respectively, from the same material, PCL. These results indicated that the structure of the scaffold can be designed to drive cells into morphologies that direct their differentiation down a desired lineage. Images of the 2D films, 3D nanofibers and hBMSCs are given in figure 1.

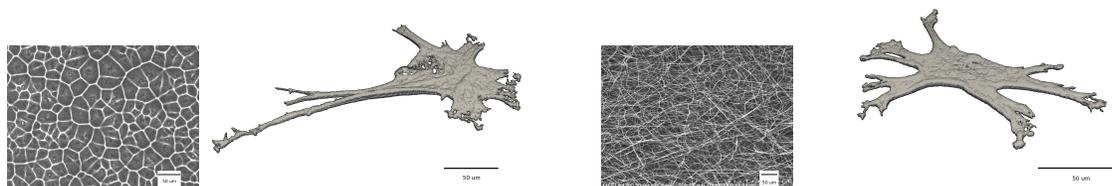


Figure 1: Left to right: SEM (Scanning Electron Micrograph) of 2D PCL spun-coat film, an hBMSC cultured on a 2D PCL spun-coat film, SEM of 3D PCL nanofiber scaffold, and an hBMSC cultured in a 3D PCL nanofiber scaffold. hBMSC images are reconstructed from 3D confocal fluorescent scans of fluorescently stained actin.

The use of scaffold structure to control stem cell function is attractive because scaffold structure is stable, has a low regulatory burden, and is relatively easy and inexpensive to control [31]. Covalent functionalization of scaffold devices with biochemically active molecules such as cell-adhesive peptides or growth factors is difficult, hard to characterize, and expensive. Loading scaffolds with growth factors to guide regeneration is challenging because proteins are hard to manufacture and highly unstable. In addition, including biomolecules or growth factors in a device increases the regulatory costs.

Cell shape and function are known to be intricately linked [10, 19] and recent work has shown that this premise holds true for hBMSCs [26, 27, 32, 43]. hBMSC differentiation can be directed by properly tuning the cellular microenvironment to drive the cells into the proper shape. Cell shape has the added value of being an early predictor of cell fate. Biochemical assays require weeks or months of culture for differentiation markers to become detectable, whereas cells attain a stable morphology within a day of culture that can be an indicator for their future behavior [43]. Though

previous methods for assessing cell shape have focused on 2D cell shape data [17, 43], tissue engineering aims to use 3D scaffolds to control cell shape function. Culture of cells in complex 3D microenvironments is likely to require 3D cell shape analysis in order to establish meaningful relationships. Rapid throughput is also desirable due to the large number of parameters that must be tested to identify scaffolds that promote the desired biologic response [30, 40].

In addition to the tissue engineering field, rapid 3D methods for measuring cell shape are likely to be needed by the pharmaceutical industry as it moves towards 3D scaffold systems for drug screening [11, 41]. 3D *in vitro* culture models are less expensive than animal models and may be more predictive of human clinical outcomes. Drug screens typically involve thousands of compounds and cell shape analysis is a parameter frequently used to determine toxicity response. Thus, high-content 3D methods for cell shape classification are required if pharma is to use 3D scaffold technologies to improve the predictive nature of *in vitro* testing [25, 45]. Herein, we have developed a machine-learning-based algorithm that can rapidly classify stem cells based on their 3D shape.

The remainder of this paper is organized as follows. In Section 2, we give an overview of the algorithm and describe some parameters that need to be set. In Section 3, we describe how we characterize a cell by intersecting it with random lines. In Section 4, we describe several ways in which our algorithm can be parallelized for implementation on a GPU (Graphics Processing Unit). We present timing and accuracy results in Section 5. Finally, we discuss our results and applications of our algorithm in Section 6, provide suggestions for future work in Section 7, and give our conclusions in Section 8.

2. Technical Approach

hBMSCs (human Bone Marrow Stromal Cells) were cultured on two types of substrates: a spun-coat substrate with a flat, 2D surface, and a nanofiber substrate with a fibrous, 3D surface. hBMSCs cultured on the nanofiber substrate underwent osteogenic differentiation and adopted a more spikey, branched appearance, while hBMSCs cultured on the spun-coat substrate did not differentiate and retained a smoother appearance. Our goal is to use 3D geometric properties of the hBMSCs to distinguish between hBMSCs cultured on the nanofiber substrate and hBMSCs cultured on the spun-coat substrate. Many methods for classifying three-dimensional objects can be found in the literature. For a survey of these methods, see Cardone *et al.* [8].

Our general workflow is as follows. First, confocal microscopy was used to collect 3D image data sets of individual cells. Next, we use an algorithm to intersect each cell with many lines in 3D and record the lengths of each of the partial line segments that run through the interior of the cell. From this line length data, we generate a line length histogram describing each cell. Finally, we use the histograms to train a machine learning technique which is used to classify new cells as coming from a 3D, fibrous nanofiber scaffold or a 2D, flat spun-coat film. An overview of this pipeline is

given in figure 2.

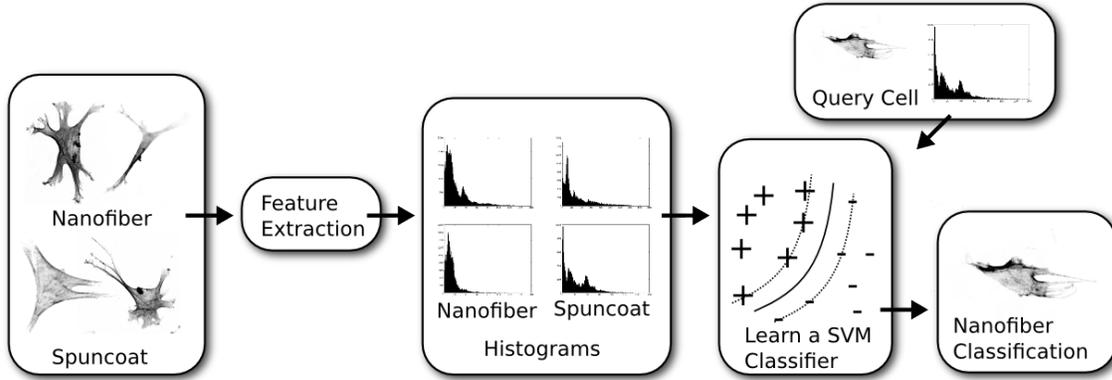


Figure 2: Overview of our cell data processing pipeline.

The 3D cell images that were analyzed in the current work were collected in previous experiments described in Kumar *et al.* (2011) [26]. Primary human bone marrow stromal cells (hBMSCs) isolated from iliac crest of healthy donors were cultured for 1 day on PCL (poly(ϵ -caprolactone)) spun-coat films or PCL nanofiber scaffolds. 2D PCL spun-coat films were made by spin-coating. 3D PCL nanofibers were made by electrospinning. Nanofiber diameter was measured by imaging with scanning electron microscopy (mean nanofiber diameter = 910 nm, S.D. = 526 nm, $n = 100$). 2D spun-coat films and 3D nanofiber scaffolds were made from the same material, PCL, so that the effects of scaffold chemistry could be discerned from effects of scaffold structure. For 3D imaging, cellular actin was stained with Alexa-Fluor-546-phalloidin and imaged by confocal fluorescence microscopy. Z-stacks of images of individual hBMSCs on spun-coat films for nanofiber scaffolds were collected using a 1 μm step size.

In Kumar *et al.* (2011) [26], osteogenic differentiation was assessed by measuring matrix calcification and by measuring gene expression by mRNA microarrays. hBMSCs cultured on PCL nanofibers underwent osteogenic differentiation while hBMSCs on PCL spun-coat films did not. Thus, it was not necessary to measure differentiation by osteogenic markers for the current manuscript since this work has already been published for the confocal Z-stack data sets analyzed herein. The current work focuses on classifying these previously collected hBMSC confocal Z-stacks by their 3D cell shape (not 2D shape analysis as was previously performed) using a new SVM algorithm.

Previous analysis of hBMSC shape using traditional 2D shape analysis tools demonstrated that hBMSCs on 3D nanofiber scaffolds have a more branched structure than hBMSCs on the 2D spun-coat films [26]. We want to leverage this shape difference to distinguish between hBMSCs that were cultured on the different substrates. To accomplish this, we first intersect the cells with a set of randomly generated lines. We measure the lengths of the portions of the lines that passed through the cell and put these lengths into histograms. Our hypothesis was that the cells on the nanofiber substrates

would have more short segments since these cells had more long, thin branches. Analysis of the average segment length for the two cell types supports this hypothesis.

A question that arises is how to choose the random lines that are used to intersect with the cells. We want a method of line selection whose results do not depend on cell orientation. We have tried two different methods of line selection. The first is to select pairs of points on a bounding sphere around the cell and generate lines that pass through both points. The results of this method should converge to the same result regardless of the cell’s orientation. The second method is to select pairs of points on the curved surface of a cylinder rather than the surface of a sphere. This method takes advantage of the particular structure of our data set.

To perform our algorithm we need to classify each voxel in the three-dimensional data set as either being inside or outside the cell. Each voxel contains an integer that represents how much light was received from that location by the microscope. These values are not calibrated to any particular units. The fluorescently stained cell gives off a large amount of light from locations that are inside the cell, while only a small amount of noise is received from locations outside the cell. We classify the voxels by selecting a threshold value and labeling voxels with values below this threshold as being outside the cell and voxels with values above this threshold as being inside the cell. If the threshold value is chosen too low then noise in the data set may be labeled as part of the cell. If the threshold value is chosen too high, then significant geometric features of the cell may be eroded. We therefore expect that, as the threshold value is increased, the performance of the algorithm will improve up to a certain point and then worsen. The selection of the threshold value can be considered part of the training phase of the algorithm and the value can be chosen based on the training data set. For this work, the threshold value was chosen by testing a series of different values and selecting the one that produces the greatest percentage of correct classification. Based on the data presented in Table 1, we selected a threshold value of 600.

Threshold	300	400	500	600	700	800	900	1000
% Correct	75.6	78.0	82.9	82.9	82.9	75.6	75.6	73.2

Table 1: Percent correct classification with various threshold values. The voxels in the data set contained values in the range $[0, 4095]$, 12 bits of precision. Parameters used were Cylinder method, 10^6 lines intersected, minimum gap length of $8 \mu\text{m}$, polynomial SVM, ten-fold cross validation (see Section 5.2).

During data collection the confocal microscope images were each focused on a single cell, but often captured portions of other cells. To correct for this we clean the data volumes by identifying connected components and throwing out all components except the largest. We also fill in any holes (empty regions completely surrounded by cell voxels) we find in the component.

After these steps there can still be some noise in the data set that causes voxels that should be inside the cell to be labeled as empty space. In our line shooting algorithm

we correct this by ignoring gaps in the line segments that are smaller than a certain length, instead counting it as one continuous segment. If this minimum gap length is set too low then it will allow gaps that are actually noise or artifacts, while if it is set too high then it will exclude gaps that are actually part of the cell geometry. We therefore expect that, as the minimum gap length is increased, the performance of the algorithm will improve up to a certain point and then worsen. As with the previously mentioned threshold value, the selection of the minimum gap length can be considered part of the training phase of the algorithm and can be performed based on the training data set. For this work, we chose our value for the minimum gap length by once again testing a series of values and selecting the one that produced the greatest percentage of correct classification. Based on the data presented in Table 2, we selected a minimum gap length of 8 μm .

Min Gap Length (μm)	1	3	5	7	8	9	11	13	15
% Correct	73.2	78.0	82.9	82.9	82.9	82.9	82.9	80.5	78.0

Table 2: Percent correct classification with various minimum gap lengths. Parameters used were Cylinder method, 10^6 lines intersected, threshold value of 600, polynomial SVM, ten-fold cross validation (see Section 5.2).

We store the lengths of the segments which pass through a cell in a histogram with 256 bins. We found that this number was large enough that increasing it did not seem to increase the classification accuracy. To improve performance, it may be useful to experimentally determine the lowest number of histogram bins required to still achieve good classification results. It may also be useful to experiment with using histograms with non-uniform bins.

3. Geometric Characterization of the Cell

As described above, our analysis of the cells involves generating a set of random lines that intersect each cell. The lines were generated using several methods. In the first method, we pick pairs of points on the surface of a bounding sphere and generate a line that intersects these points. This is the method used by Juba and Varshney [22] and is described by Li *et al.* [29] as the Chord Model. Uniformly distributed points (x, y, z) on a sphere can be generated from pairs (u, θ) of uniformly distributed random numbers by using the formula

$$(x, y, z) = ((1 - u^2)^{\frac{1}{2}} \cos \theta, (1 - u^2)^{\frac{1}{2}} \sin \theta, u) \quad (1)$$

where u is in $[-1, 1]$ and θ is in $[0, 2\pi)$ [47]. A slightly more computationally efficient formula is given by Rovira *et al.* [38] which generates the points (x, y, z) from pairs

(ξ_1, ξ_2) of uniformly distributed random numbers

$$\begin{aligned} \cos \theta &= 1 - 2 * \xi_1 \\ \sin \theta &= \sqrt{1 - (\cos \theta)^2} \\ \varphi &= 2 * \pi * \xi_2 \\ (x, y, z) &= (\sin \theta * \sin \varphi, \cos \theta, \sin \theta * \cos \varphi) \end{aligned} \quad (2)$$

where ξ_1 and ξ_2 are in $[0, 1)$. Points generated by either of these methods have equal density over the entire surface of the sphere and are not clustered at the poles.

In the second method, we pick pairs of points on the curved surface of a cylinder and generate a line that intersects these points. The cylinder is oriented such that the central axis is perpendicular to the plane of the 2D microscope images that were stacked together to form the 3D volume. Each of the two points that define the line is defined by an angle θ around the circumference of the cylinder and a height z along the central axis. This is illustrated in Figure 3. The formula for points (x, y, z) on the curved surface of a cylinder is

$$(x, y, z) = (\cos \theta, \sin \theta, z) \quad (3)$$

where z is in $[-1, 1]$ and θ is in $[0, 2\pi)$.

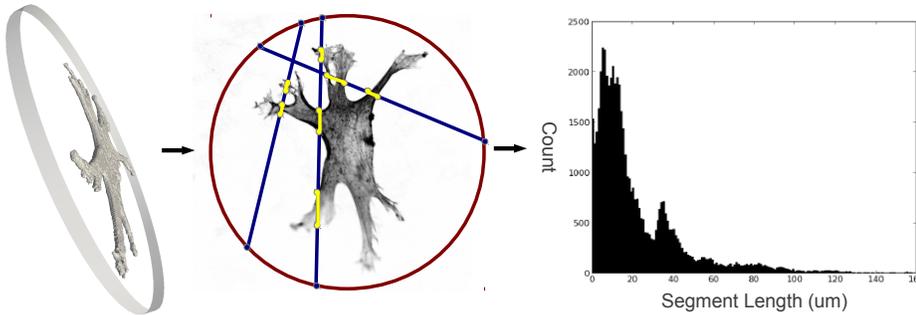


Figure 3: Top-down view of lines defined by points on a cylinder enclosing a cell. The lengths of the intersections of these lines with the cell are stored in a histogram.

Both of these line generation methods require a set of uniformly distributed random numbers. Typically a pseudo-random sequence of random numbers is used for this purpose. However, it has been shown that a so-called quasi-random sequence (also called a low-discrepancy sequence) has better properties, including a lower error bound in numeric integration [29]. For this work, we use the Niederreiter quasi-random sequence [7], which can be found in the GNU Scientific Library [1]. For Equation 1, we generate quasi-random points (a, b, c, d) in four dimensions and use the first and second coordinate pairs (a, b) and (c, d) to generate the (u_1, θ_1) and (u_2, θ_2) . For Equation 3, we generate quasi-random points (a, b, c) in three dimensions and use the three coordinates for θ_1 , θ_2 , and z (in our implementation both points of the line are at the same altitude z).

The sphere method of generating lines is good for general data sets in which no direction should be treated differently than any other. In the data we are working with,

however, most of the cell structure variation is in the xy plane of the microscope images. We therefore chose to generate lines using the cylinder method with the central axis of the cylinder aligned with the z axis of the image stack when computing most of the results reported in this paper. A comparison of the performance between these two methods is given in Section 5.2.

Once a line is generated, the next step is to compute its intersections with the cell. We do this by stepping along the line at uniform intervals and at each point checking if that point is inside or outside the cell. Initially, the line starts outside the cell. If two adjacent points are inside and outside the cell respectively (or vice versa), we know that we have either entered or left the cell. Whenever we leave the cell we compute the length of the line segment that was inside the cell and store this in a histogram.

4. High-Throughput Processing

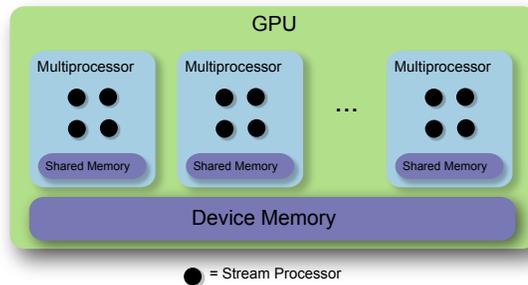


Figure 4: GPU memory and multiprocessor layout.

A GPU (Graphics Processing Unit) is a SIMT (Single Instruction, Multiple Thread) processor capable of executing many identical instruction threads in parallel on different sets of input data. It has a large number of stream processors, each with some local memory and registers. These are grouped into several multiprocessors, each containing a small amount of fast memory that is shared within the multiprocessor. It also has a large amount of slower memory that is accessible by all thread processors. A diagram of the processors and memory is given in Figure 4.

The SIMT architecture is similar to the traditional SIMD (Single Instruction, Multiple Data) except that in SIMT, threads can take diverging branches. On the NVIDIA CUDA [2] capable hardware on which we implemented our algorithm, threads are divided into groups of 32 called warps. Threads within a warp cannot execute different instructions at the same time. If threads in a warp go down diverging branches, the GPU will first disable all the threads going down the second branch and execute the instructions of the first branch, then vice versa. Greater efficiency can therefore be achieved by ensuring that as often as possible all threads in a warp follow the same branches. Threads in different warps are free to go down different branches without penalty.

If an algorithm can be mapped to this architecture, the GPU can generally execute the algorithm much faster than it could be executed on a CPU. This is because the SIMT nature of the GPU allows more of its transistors to be dedicated to arithmetic operations rather than tasks such as caching and flow control.

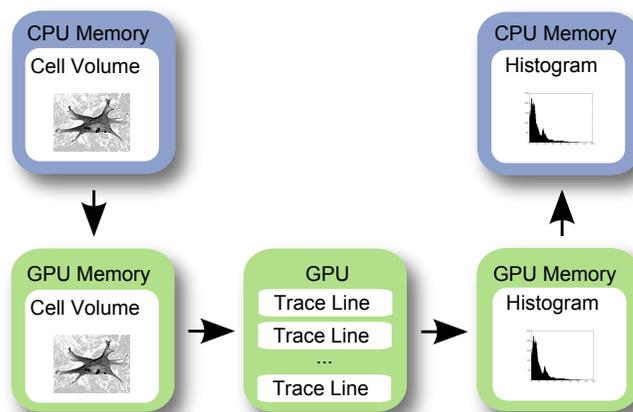


Figure 5: Use of CPU and GPU to generate segment length histograms from cell volumes using the Atomic Operations algorithm.

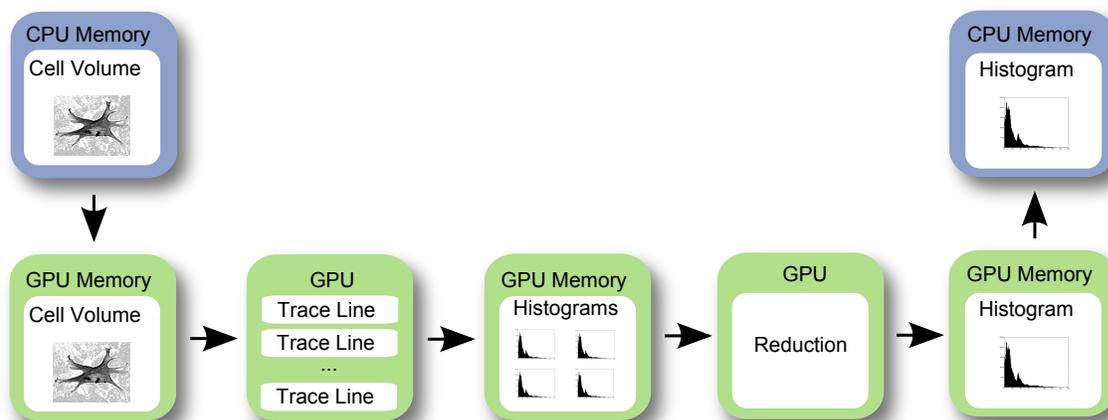


Figure 6: Use of CPU and GPU to generate segment length histograms from cell volumes using the Parallel Reduction algorithm.

We mapped our line intersection algorithm to the GPU by using one thread to compute the intersections of each line with the cell. A straightforward implementation of this algorithm involves each thread writing the lengths of the intersecting segments into a shared histogram (see Figure 5). However, this requires synchronization of the threads or the use of slow atomic operations. One way around this requirement is to give each thread its own 256 bin histogram to store its results in, and then merge the histograms at the end using a parallel reduction operation (see Figure 6). This approach

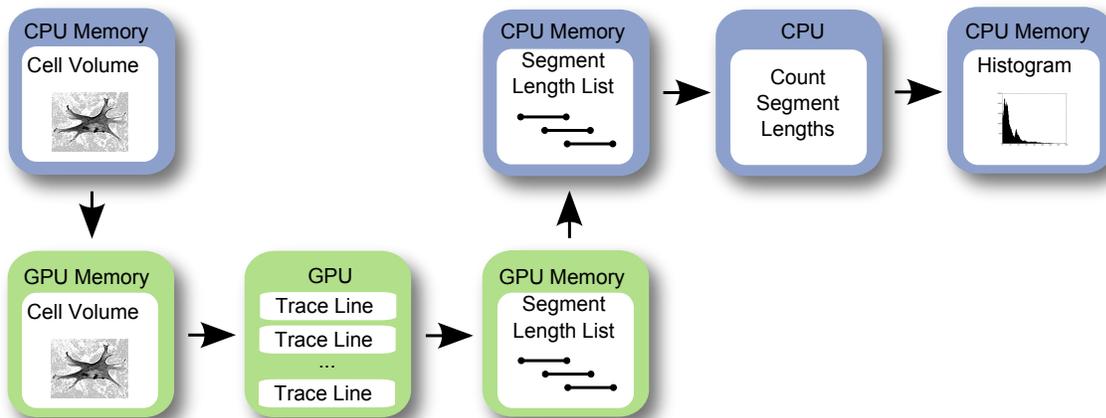


Figure 7: Use of CPU and GPU to generate segment length histograms from cell volumes using the Segment Length List algorithm.

is wasteful, however, since each thread would likely write only a few segment lengths into each histogram. Instead, each thread writes its intersected segment lengths into a fixed size list, and simply ignores any intersected segments that occur after the list is full. We found that a list of length 10 is sufficient to produce results that are almost identical to the CPU version of the algorithm. Once the lists are constructed they are read back to the CPU and the lengths are counted to generate the histogram (see Figure 7). A comparison of performance data for these three algorithms is given in Section 5.

5. Results

5.1. Running Time

We tested our algorithm on a data set collected by Kumar *et al.* [26] consisting of 21 cells grown on a fibrous nanofiber substrate and 20 cells grown on a flat spun-coat substrate. For each cell, we had a stack of 12-bit grayscale confocal microscopy images of resolution 2048×2048 which we merged together into a single volume. The number of images in each stack varied from 11 to 20. The dimensions of each voxel were approximately $0.1 \times 0.1 \times 1$ (in μm).

We measured the performance of our algorithms when intersecting a cell with various numbers of lines, ranging from 10^3 to 10^6 . The results are given in Table 3. For each algorithm the first step was to generate the random lines with which to intersect the cell. This step is independent of the data and can be done once as a pre-process, with the same set of lines then being used to intersect each cell. The line generation time is therefore not included in the running times for the algorithms.

For each number of lines, we measured the times the algorithms took to compute the intersection counts for a single cell both on the CPU and in the parallel implementations on the GPU. The timing for each algorithm is broken down into the steps listed in Figures

Num lines	10^3	10^4	10^5	10^6
Generate lines (ms)	0.1408	1.251	13.08	129.6
CPU Algorithm				
Total time (ms)	200.3	2030	19970	200100
GPU Algorithm – Atomic				
Total time (ms)	12.25	25.92	191.1	1863
Speedup factor	16.35	78.32	104.5	107.4
GPU Algorithm – Reduction				
Trace lines (ms)	0.4551	0.5024	1.594	15.51
Do reduction (ms)	21.72	36.76	210.5	1987
Total time (ms)	22.62	38.98	227.6	2147
Speedup factor	8.855	52.08	87.74	93.20
GPU Algorithm – Lists				
Trace lines (ms)	0.3208	0.4902	1.458	14.97
Count lengths (ms)	11.35	24.36	180.3	1748
Total time (ms)	12.12	26.53	197.1	1908
Speedup factor	16.53	76.52	101.3	104.9

Table 3: Results of intersecting a cell with different numbers of lines. The CPU algorithm was run on an Intel Xeon X5260 (using only one core) with 10 GB of RAM. The GPU algorithms were run on an NVIDIA Tesla C2050. The dimensions of the cell volume data were $2048 \times 2048 \times 20$. The threshold was 600 and the minimum gap length was $8 \mu\text{m}$. The random lines were generated using the Cylinder method—running times when using the Sphere method were very similar.

5, 6, and 7. For the GPU Reduction algorithm we measured the time required to send the lines to the GPU and trace them (“Trace lines”), as well as the time required to perform the parallel reduction operation and read back the resulting histogram (“Do reduction”), which was implemented using the CUDA Thrust library [20]. For the GPU Lists algorithm we again measured the time required to send the lines to the GPU and trace them (“Trace lines”), as well as the time required to read the segment length lists back to the CPU and convert them into a histogram (“Count lengths”). The total time includes these times as well as the time required for any other miscellaneous tasks. In addition to these times, all algorithms took about 4520 ms to load the cell volume data from the disk. We also list speedup factors which show how many times faster each parallel GPU implementation is over the serial implementation on the CPU. Note that the best parallel GPU implementation can be over two orders of magnitude faster than the serial CPU implementation.

On the current-generation GPU listed in the caption of Table 3, the best performing algorithm was Atomic, followed closely by Lists. In addition to this GPU, we also tested the algorithms on several older-generation GPUs, an NVIDIA Quadro NVS 285 and an

NVIDIA Quadro NVS 290. On these GPUs the Lists algorithm was actually slightly faster than Atomic. We suspect the reason for this is that the implementation of atomic operations has been improved in the current generation of GPUs.

5.2. Validation

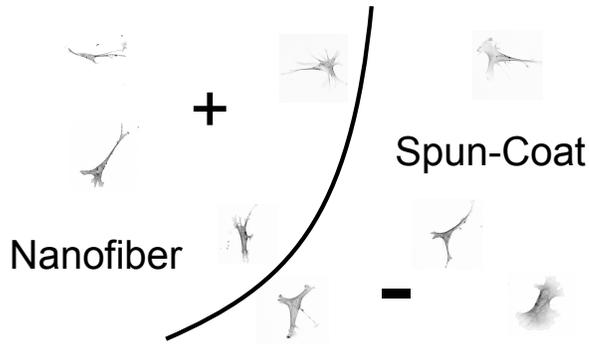


Figure 8: Trained Support Vector Machine (SVM). The hyperplane that separates the two classes of histograms is the zero level-set of a decision function that divides space into a positive half and a negative half. Histograms that fall in the positive half are labeled as one class (here, Nanofiber) while histograms that fall in the negative half are labeled as the other (here, Spun Coat). Note that this figure is an example only—it does not depict the actual hyperplane and the positions of the cell images do not correspond to their actual positions in the feature space.

Once the cell histograms have been produced they are either used to train a Support Vector Machine (SVM) classifier (if they are in the training set) or are classified by a trained SVM. The training phase of the SVM treats each histogram as a point in high-dimensional space and tries to compute the hyperplane that best separates the points corresponding to differentiated cells and the points corresponding to non-differentiated cells. The orientation of the hyperplane is determined only by those points close to it, which are referred to as the support vector. Once the algorithm is trained, additional histograms can be classified as being cultured on a differentiating (3D nanofiber scaffold) or non-differentiating (2D flat spun-coat film) substrate by testing where the corresponding point lies in relation to the hyperplane. An illustration of a trained SVM is given in Figure 8. Additional information on Support Vector Machines can be found by referring to Cortes and Vapnik [12].

For this work we used the SVM implementation in LIBSVM [9] version 3.12 with all options set to default (degree = 3, gamma = 1/256, coef0 = 0) except for Kernel Type, which we set to Polynomial. Training a polynomial SVM on our test data set of 41 cell histograms took about 9.3 ms. Once the SVM is trained, new cell histograms can be classified in about 4.4 ms each.

In addition to running time, we also measured the classification correctness of our algorithm when intersecting the cells with different numbers of lines. To compute the

percentage of correct classification for each trial we used 10-fold cross validation. This means that we randomly distributed the 41 cells of the data set into 10 groups of approximately equal size, trained the machine learning algorithm on 9 of the groups, and measured the percent correct classification of the 10th. An example of one fold is given in Table 4. Note that the numbers of Nanofiber and Spun-Coat cells in a fold will not necessarily be equal. This was repeated 10 times, each time using a different group to test the classification correctness. The percent correct classification of all the groups was then averaged. The average percent correct classifications are given in Figure 9.

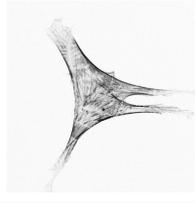
				
Actual	Nanofiber	Nanofiber	Spun-Coat	Spun-Coat
Classified as	Nanofiber	Spun-Coat	Spun-Coat	Spun-Coat

Table 4: Example of one fold of the validation of a trained SVM model. The lines were generated using the Cylinder method and intersected using the GPU Lists algorithm with a threshold of 600 and a minimum gap length of $8 \mu\text{m}$. An SVM was trained with 19 nanofiber cells and 18 spun-coat cells. An additional 2 nanofiber cells and 2 spun-coat cells were set aside to be classified by the model. Of these 4 cells, 3 were classified correctly.

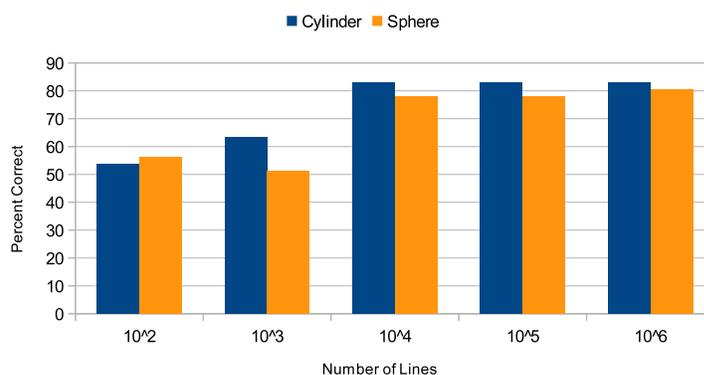


Figure 9: Percent correct classification of differentiated and non-differentiated stem cells. The lines were generated using the Cylinder method (left) and Sphere method (right), and intersected using the GPU Lists algorithm with a threshold of 600 and a minimum gap length of $8 \mu\text{m}$.

We found that, for this data set, we get better cell classification results with the Cylinder method than with the Sphere method. When using 10^4 lines, the Cylinder

method gives its maximum performance of 82.9% correct, while the Sphere method gives only 78.0% correct. With 10^6 lines, the Sphere method gives only 80.5% correct. This was expected, as for this data set most of the cell structure variation is in the xy plane, and so would be more efficiently detected by using lines which are more closely aligned with this plane.

This data demonstrates the progressive nature of our algorithm. For a small number of lines the classification accuracy is about 50%, which is what would be expected from random guessing. As the number of intersected lines increases, so does the classification accuracy until it levels off at a maximum. If, after a certain number of lines have been intersected, the user desires additional accuracy, then the already-computed lines can be re-used and only the additional lines will need to be intersected. The algorithm can therefore provide rough results quickly, which can then be improved to the desired accuracy with additional running time.

6. Discussion

We present an algorithm that can classify, with over 80% efficacy, hBMSCs (human bone marrow stromal cells) as having been grown on a differentiation-inducing 3D PCL nanofiber scaffold or on a non-differentiation-inducing 2D PCL spun-coat film. Although the algorithm has been tested on a relatively small data set of 41 z-stacks in the current work, this serves as the basis for future research where the size of Z-stack data sets will increase. High-content imaging for screening cell state is currently used to screen materials or drug libraries in 2D where cells on flat substrates are imaged in a single plane [23, 39, 42]. However, 3D scaffolds are required for regeneration of 3D tissues [26, 27, 28] and 3D in vitro cell culture models may be more predictive of clinical outcomes [11, 41]. As imaging technology rapidly advances, high-content 3D imaging will emerge for data collection and analysis in 3D scaffolds [15]. Indeed, Oh *et al.* [33] have collected thousands of cell nuclei Z-stacks to study cell growth and senescence and high-content confocal imaging systems are commercially available [16]. Thus, it is not unrealistic to anticipate that tens of thousands of z-stacks of 3D cell morphology data will require analysis by algorithms such as the one tested in this work.

It is widely accepted that 3D cell culture provides a more physiologic environment for cells than does a flat 2D surface. However, exactly what constitutes a 3D environment from the perspective of a cell has not been defined. This issue was highlighted as a critical roadblock for biomaterials research by the 2012 National Science Foundation Biomaterials Workshop [3] (p. 72). The lack of shape variation noted in the Z-direction for cells on nanofiber scaffolds may indicate that nanofiber scaffolds do not provide a 3D environment from the perspective of the hBMSCs. The algorithm developed in the current work, which classifies cells by 3D shape, can be used to determine if optimal cell performance requires a scaffold that allows the cells to adopt a 3D morphology. In addition, if nanofiber scaffolds do not provide a 3D microenvironment, the current algorithm can be used to design improved scaffolds. Since the algorithm classifies cells

by 3D shape, it will be sensitive to scaffold designs that cause changes in 3D shape, and could identify scaffolds that provide a more 3D-like microenvironment.

It is important to note that current algorithm does not indicate differentiation vs. non-differentiation, but instead classifies cells based on 3D shape. This work is an important step because 1) it establishes a tool (the SVM algorithm) to classify cells based on 3D shape and 2) demonstrates the effectiveness of the tool to discern cells grown on 3D nanofiber scaffolds from 2D spun-coat films. Future work can use this tool to identify new scaffold structures and formulations that drive hBMSCs into morphologies that correlate with differentiation, such as culture on 3D nanofiber scaffolds.

Stemming from seminal observations by Folkman and Moscona in 1978 [19], cell shape is widely accepted as an indicator of cell state [4, 10, 19, 26, 27, 32, 43, 46]. However, no single measurement can absolutely define cell state. Cell shape is not absolute and can be ambiguous. A panel of measurements is required, and even then, cell state is never known with 100% certainty. For example, a cell that has a rounded, spherical morphology may be dying, in suspension, attached to a poorly adherent substrate, or a chondrocyte, which has an inherently rounded shape. Thus, additional measurements are necessary to properly interpret cell shape results. For the case of hBMSCs, we have observed that several types of morphologies can lead to osteogenic differentiation, but that a well-spread, flattened morphology does not [26, 35, 27].

The new aspect of the current work is that a 3D cell shape algorithm is used to classify cells whereas 2D shape metrics were used previously [26, 43]. Although additional measurements besides cell shape will be required to determine cell state with greater assurance, 3D cell shape is informative and can be part of a panel of measurements used to determine cell state. Cell shape may not be 100% indicative of cell state, but cell shape and cell state are linked and cell shape is an indicator of cell state.

7. Future Work

Now that we have developed the algorithm, we will apply it to additional hBMSC Z-stack datasets in the future. hBMSCs cultured on different types of scaffolds or in the presence of osteogenic supplements will be useful. In addition, larger numbers of Z-stacks can be tested to improve the statistical confidence in the results.

Since the method we present can theoretically be applied to data sets of any dimensionality, it will be interesting to examine its performance on a 2D version of the stem cell data set. The results of this analysis can define the benefit of doing a 3D analysis and identify what additional information is provided by doing a 3D analysis.

Although we have only applied the algorithm to cell data, the algorithm is general enough that it can potentially be applied to any type of 3D data, such as CAD models or protein molecules. The use of the algorithm to classify other data types can be another interesting avenue for future work.

Another possibility is to take advantage of the progressive nature of the algorithm

during the SVM classification. Rather than simply generating the histogram using the full number of lines and then performing the classification, we can instead generate the histogram using some small initial number of lines and then check the certainty of the classification. If the classification was still doubtful then the histogram can be improved by intersecting the cell with more lines, while if the classification was sufficiently certain then the results can be returned immediately. For an SVM, the certainty of the classification can perhaps be measured by the distance of the query point from the hyperplane dividing the two regions of classification.

8. Conclusions

We present an algorithm that can classify hBMSCs (human Bone Marrow Stromal Cells) by their 3D shape and demonstrate how it can distinguish between hBMSCs grown on a differentiation-inducing 3D PCL nanofiber scaffold or on a non-differentiation-inducing 2D PCL spun-coat film. The algorithm takes 3D cell image data and intersects it with randomly generated lines that connect the sides of a cylinder that bounds the cell. The lengths of line segments that are within the cell are used to generate a histogram. These histograms can then be used as sample points to train a machine learning algorithm such as a support vector machine (SVM), which can then be used to classify future cells.

The algorithm is easily parallelizable and is also progressive, allowing it to provide a rough histogram quickly and then refine it as desired. The parallel GPU implementation can convert a cell into a histogram representation suitable for machine learning training or classification by intersecting it with 10^6 lines in about 1863 ms, representing an over 100-fold speedup from the serial CPU implementation. By applying the algorithm to our test data set of 41 cells, we were able to achieve 82.9% correct classification using 10-fold cross validation. This rapid 3D image analysis algorithm can be used to classify stem cells by their 3D shape to identify 3D tissue scaffolds that drive cells into morphologies that correlate with differentiation. The algorithm used 3D cell image data in order to take advantage of the benefits of 3D culture and to capture the effects of 3D scaffold structure on cell shape. The approach has been demonstrated using stem cell image data from 1-day cultures, which enables cell classification at a much earlier stage than is possible with osteogenic markers, which can require weeks of culture.

Acknowledgments

This work has been supported in part by the NSF grants: CCF 05-41120, CMMI 08-35572, CNS 09-59979 and the NVIDIA CUDA Center of Excellence. This work has also been supported in part by the NIST grant 70NANB9H9181. The primary human bone marrow stromal cells employed in this work were provided by the Tulane Center for Gene Therapy through a grant from NCRP of the NIH P40RR017447. The “standard deviation” (S.D.) is the same as the “combined standard uncertainty of the mean” for the purposes of this work. This article, a contribution of NIST, is not subject

to US copyright. Certain equipment and instruments or materials are identified in the paper to adequately specify the experimental details. Such identification does not imply recommendation by NIST, nor does it imply the materials are necessarily the best available for the purpose.

References

- [1] GNU Scientific Library. <http://www.gnu.org/software/gsl/>.
- [2] NVIDIA CUDA Zone. <http://developer.nvidia.com/category/zone/cuda-zone>.
- [3] *Biomaterials: Important Areas for Future Investment*, 2013. Report on a 2012 National Science Foundation Workshop. <http://nsfbiomatworkshop2012.caltech.edu/report/>.
- [4] C. W. Archer, P. Rooney, and L. Wolpert. Cell shape and cartilage differentiation of early chick limb bud cells in culture. *Cell Differentiation*, 11(4):245–251, 1982.
- [5] A. Atala, S. Bauer, S. Soker, J. Yoo, and A. Retik. Tissue-engineered autologous bladders for patients needing cystoplasty. *The Lancet*, 367(9518):1241–1246, 2006.
- [6] D. Benoit, M. Schwartz, A. Durney, and K. Anseth. Small functional groups for controlled differentiation of hydrogel-encapsulated human mesenchymal stem cells. *Nature Materials*, 7(10):816–823, 2008.
- [7] P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Trans. Model. Comput. Simul.*, 2(3):195–213, 1992.
- [8] A. Cardone, S. K. Gupta, and M. Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering*, 3(2):109–118, 2003.
- [9] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [10] C. Chen, M. Mrksich, S. Huang, G. Whitesides, and D. Ingber. Geometric control of cell life and death. *Science*, 276(5317):1425–1428, 1997.
- [11] F. Collins. Reengineering translational science: the time is right. *Science translational medicine*, 3(90):1–6, 2011.
- [12] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [13] M. Dalby, N. Gadegaard, R. Tare, A. Andar, M. Riehle, P. Herzyk, C. Wilkinson, and R. Oreffo. The control of human mesenchymal cell differentiation using nanoscale symmetry and disorder. *Nature Materials*, 6(12):997–1003, 2007.
- [14] M. Dominici, K. Le Blanc, I. Mueller, I. Slaper-Cortenbach, F. Marini, D. Krause, R. Deans, A. Keating, D. Prockop, and E. Horwitz. Minimal criteria for defining multipotent mesenchymal stromal cells. The International Society for Cellular Therapy position statement. *Cytotherapy*, 8(4):315–317, 2006.
- [15] T. Dorval and R. Grailhe. 3D high content imaging: high level phenotypic quantification new opportunity for drug discovery. In *BMC Proceedings*, volume 5, page 75. BioMed Central Ltd, 2011.
- [16] S. Duffy and V. M. Avery. Development and optimization of a novel 384-well anti-malarial imaging assay validated for high-throughput screening. *The American Journal of Tropical Medicine and Hygiene*, 86(1):84, 2012.
- [17] J. Elliott, M. Halter, A. Plant, J. Woodward, K. Langenbach, and A. Tona. Evaluating the performance of fibrillar collagen films formed at polystyrene surfaces as cell culture substrates. *Biointerphases*, 3(2):19–28, 2008.
- [18] A. Engler, S. Sen, H. Sweeney, and D. Discher. Matrix elasticity directs stem cell lineage specification. *Cell*, 126(4):677–689, 2006.
- [19] J. Folkman and A. Moscona. Role of cell shape in growth control. *Nature*, 273(5661):345, 1978.

- [20] J. Hoberock and N. Bell. Thrust: A parallel template library, 2010. Version 1.5.3 <http://thrust.github.io/>.
- [21] A. Jaklenec, A. Stamp, E. Deweerd, A. Sherwin, and R. Langer. Progress in the tissue engineering and stem cell industry are we there yet?. *Tissue Engineering Part B: Reviews*, 18(3):155–166, 2012.
- [22] D. Juba and A. Varshney. Parallel, stochastic measurement of molecular surface area. *Journal of Molecular Graphics and Modelling*, 27(1):82 – 87, 2008.
- [23] S. B. Kennedy, N. R. Washburn, C. G. Simon Jr, and E. J. Amis. Combinatorial screen of the effect of surface energy on fibronectin-mediated osteoblast adhesion, spreading and proliferation. *Biomaterials*, 27(20):3817–3824, 2006.
- [24] J. Kohn, W. Welsh, and D. Knight. A new approach to the rationale discovery of polymeric biomaterials. *Biomaterials*, 28(29):4171, 2007.
- [25] E. Krausz, R. de Hoogt, E. Gustin, F. Cornelissen, T. Grand-Perret, L. Janssen, N. Vloemans, D. Wuyts, S. Frans, A. Axel, et al. Translation of a tumor microenvironment mimicking 3D tumor growth co-culture assay platform to high-content screening. *Journal of Biomolecular Screening*, 18(1):54–66, 2012.
- [26] G. Kumar, C. Tison, K. Chatterjee, P. Pine, J. McDaniel, M. Salit, M. Young, and C. Simon Jr. The determination of stem cell fate by 3D scaffold structures through the control of cell shape. *Biomaterials*, 32(35):9188–9196, 2011.
- [27] G. Kumar, M. Waters, T. Farooque, M. Young, and C. Simon. Freeform fabricated scaffolds with roughened struts that enhance both stem cell proliferation and differentiation by controlling cell shape. *Biomaterials*, 33(16):4022–4030, 2012.
- [28] R. Langer and J. P. Vacanti. Tissue engineering. *Science*, 260:920–925, 1993.
- [29] X. Li, W. Wang, and A. Bowyer. Using low-discrepancy sequences and the Crofton formula to compute surface areas of geometric models. *Computer-Aided Design*, 35(9):771–82, Aug. 2003.
- [30] J. Lovmand, J. Justesen, M. Foss, R. Lauridsen, M. Lovmand, C. Modin, F. Besenbacher, F. Pedersen, and M. Duch. The use of combinatorial topographical libraries for the screening of enhanced osteogenic expression and mineralization. *Biomaterials*, 30(11):2015–2022, 2009.
- [31] J. Makower, A. Meer, and L. Denend. FDA impact on us medical technology innovation: a survey of over 200 medical technology companies. *Arlington (Virginia): National Venture Capital Association*, 2010.
- [32] R. McBeath, D. Pirone, C. Nelson, K. Bhadriraju, and C. Chen. Cell shape, cytoskeletal tension, and rhoa regulate stem cell lineage commitment. *Developmental Cell*, 6(4):483–495, 2004.
- [33] J. Oh, A. Gertych, and J. Tajbakhsh. Nuclear DNA methylation and chromatin condensation phenotypes are distinct between normally proliferating/aging, rapidly growing/immortal, and senescent cells. *Oncotarget*, 4(3):474–493, 2013.
- [34] H. Ott, T. Matthiesen, S. Goh, L. Black, S. Kren, T. Netoff, and D. Taylor. Perfusion-decellularized matrix: using nature’s platform to engineer a bioartificial heart. *Nature Medicine*, 14(2):213–221, 2008.
- [35] S. Parekh, K. Chatterjee, S. Lin-Gibson, N. Moore, M. Cicerone, M. Young, and C. Simon. Modulus-driven differentiation of marrow stromal cells in 3D scaffolds that is independent of myosin-based cytoskeletal tension. *Biomaterials*, 32(9):2256–2264, 2011.
- [36] T. Petrie, J. Raynor, D. Dumbauld, T. Lee, S. Jagtap, K. Templeman, D. Collard, and A. García. Multivalent integrin-specific ligands enhance tissue healing and biomaterial integration. *Science Translational Medicine*, 2(45):45ra60, 2010.
- [37] A. Raya-Rivera, D. R. Esquiliano, J. J. Yoo, E. Lopez-Bayghen, S. Soker, and A. Atala. Tissue-engineered autologous urethras for patients who need reconstruction: an observational study. *The Lancet*, 377(9772):1175–1182, April 2011.
- [38] J. Rovira, P. Wonka, F. Castro, and M. Sbert. Point Sampling with Uniformly Distributed Lines. *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 109–118, 2005.

- [39] C. G. Simon Jr, N. Eidelman, S. B. Kennedy, A. Sehgal, C. A. Khatri, and N. R. Washburn. Combinatorial screening of cell proliferation on poly (l-lactic acid)/poly (d, l-lactic acid) blends. *Biomaterials*, 26(34):6906–6915, 2005.
- [40] C. G. Simon Jr, Y. Yang, V. Thomas, S. Dorsey, and A. Morgan. Cell interactions with biomaterials gradients and arrays. *Combinatorial Chemistry & High Throughput Screening*, 12(6):544–553, 2009.
- [41] J. Sung and M. Shuler. A micro cell culture analog (μ cca) with 3-D hydrogel culture of multiple cell lines to assess metabolism-dependent cytotoxicity of anti-cancer drugs. *Lab Chip*, 9(10):1385–1394, 2009.
- [42] D. L. Taylor. Past, present, and future of high content screening and the field of cellomics. In *High Content Screening*, pages 3–18. Springer, 2006.
- [43] M. Treiser, E. Yang, S. Gordonov, D. Cohen, I. Androulakis, J. Kohn, C. Chen, and P. Moghe. Cytoskeleton-based forecasting of stem cell lineage fates. *Proceedings of the National Academy of Sciences*, 107(2):610–615, 2010.
- [44] W. Tutak, S. Sarkar, S. Lin-Gibson, T. M. Farooque, G. Jyotsnendu, D. Wang, J. Kohn, D. Bolikal, and C. G. Simon Jr. The support of bone marrow stromal cell differentiation by airbrushed nanofiber scaffolds. *Biomaterials*, 34(10):2389–2398, 2013.
- [45] M. Vinci, S. Gowan, F. Boxall, L. Patterson, M. Zimmermann, C. Lomas, M. Mendiola, D. Hardisson, S. Eccles, et al. Advances in establishment and analysis of three-dimensional tumor spheroid-based functional assays for target validation and drug evaluation. *BMC Biology*, 10(1):29, 2012.
- [46] F. M. Watt, P. W. Jordan, and C. H. O’Neill. Cell shape controls terminal differentiation of human epidermal keratinocytes. *Proceedings of the National Academy of Sciences*, 85(15):5576–5580, 1988.
- [47] E. W. Weisstein. Sphere point picking. From MathWorld— A Wolfram Web Resource, <http://mathworld.wolfram.com/SpherePointPicking.html>. Last visited on 2013/4/29.