Segmentation and Statistical Analysis of Imaged and Simulated 3D Chromosome Territories

Hannah Varney
University of Maine, hannah.varney@maine.edu

Follow this and additional works at: https://digitalcommons.library.umaine.edu/etd

Part of the Other Biomedical Engineering and Bioengineering Commons

Recommended Citation
https://digitalcommons.library.umaine.edu/etd/3416

This Open-Access Thesis is brought to you for free and open access by DigitalCommons@UMaine. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of DigitalCommons@UMaine. For more information, please contact um.library.technical.services@maine.edu.
SEGMENTATION AND STATISTICAL ANALYSIS OF IMAGED AND SIMULATED 3D CHROMOSOME TERRITORIES

By

Hannah Varney

B.S. University of Maine, 2020

A THESIS

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science (in Biomedical Engineering)

The Graduate School
The University of Maine
August 2021

Advisory Committee:

Dr. Andre Khalil, Professor of Biomedical Engineering, Advisor

Dr. Michael Mason, Professor of Biomedical Engineering

Dr. Karissa Tilbury, Assistant Professor of Biomedical Engineering
SEGMENTATION AND STATISTICAL ANALYSIS OF IMAGED AND SIMULATED 3D CHROMOSOME TERRITORIES

By Hannah Varney

Thesis Advisor: Dr. Andre Khalil

An Abstract of the Thesis Presented in Partial Fulfillment of the Requirements for the Degree of Master of Science (in Biomedical Engineering)
August 2021

With advances in microscopy in the 21st century, more accurate measurements of the morphology and position of chromosome territories (CTs) during interphase within the cell nucleus can be obtained. These measurements have determined that CTs are non-randomly positioned, and their position and shape can change during diseases like breast cancer, pancreatic cancer, and Huntington’s Disease. The 2.5D Wavelet Transform Modulus Maxima segmentation method uses multi-scale edge detection to accurately segment objects in 3D images to find variables such as surface area, volume, and diameter without assuming a spherical or ellipsoidal shape. Measurements of centroid distance, shell distance and radial distance along with calculations of filament index, elongation and rugosity were analyzed to determine positional and morphological trends in mouse chromosomes 1, 2, 9, 11 and 14 within embryonic stem (ES) cells and lymphocytes. Scatterplots were created where the best-fit trendline of either linear, log-logistic, or exponential decay was fit to each dataset. These trends showed that the CTs were long with indented cavities and tended to reside in the periphery of the nucleus. Boxplots by chromosome type showed that the CTs of the ES cells were longer, smoother and farther from the centroid of the nucleus than their counterparts in the lymphocyte cells. To confirm this conclusion, a simulated model was created that placed a randomly positioned and randomly oriented
ellipsoidal CT within a spherical nucleus. This simulation used parameters gained from the imaged CTs to produce a realistic ellipsoid and then measured the radial distance. The radial distance of the simulated data was about half that of the imaged data, indicating that the CTs are not randomly placed. From linear plots using Pearson p-values, it was determined that the lymphocyte cell CTs had more occurrences of significance between variables, while the ES cell CTs had none, likely due to their undifferentiated state. Ideally the model will place four chromosomes within the nucleus, although this was too computationally taxing without a supercomputer. The model should also be expanded to allow for centroid distance and shell distance measurements, and to measure possible shape configurations if position is kept constant.
DEDICATION

To Troy and Dulsie Varney, together forever.

To Mom, Dad, Abby and Ethan: may you continue to motivate, love and support me through the rest of my endeavors as you did for this one, and may I return the favor in the future. Thank you for letting me hide for days at a time to code and write, for attending Zoom seminars to watch me present even though you got lost after slide 2, and for pretending to listen as I explain my thesis for the thousandth time. There is no success without support.
ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Andre Khalil, for suggesting a 15-month degree would even be possible and then making sure I achieved it. Thank you to my committee members, Dr. Karissa Tilbury and Dr. Michael Mason, who always have questions to help me rethink and strengthen my analysis. Also, thank you to Zach Marin, Aaron Tanenbaum, and Clara Bradley, who worked on this project before me and gave me a great starting point. Thank you to the CompuMAINE lab members for coding advice and presentation practice. Thank you to the Department of Chemical and Biomedical Engineering for providing me with a Teaching Assistant position to help finance my education. I would also like to thank my family and friends who have been my support system, especially through the last year. It’s been a roller coaster, but I have the best seatmates.
# TABLE OF CONTENTS

DEDICATION ........................................................................................................ ii

ACKNOWLEDGEMENTS ................................................................................ iii

LIST OF TABLES ................................................................................................ vi

LIST OF FIGURES ........................................................................................... vii

1. INTRODUCTION ....................................................................................... 1

2. CHROMOSOME TERRITORY IMAGING, SEGMENTATION, AND
   STATISTICAL METHODS ................................................................................ 8
   2.1 Confocal Laser Scanning Microscopy ......................................................... 8
   2.2 The 2D Wavelet Transform Modulus Maxima Method ............................... 10
   2.3 The 2.5D Wavelet Transform Modulus Maxima Method ............................ 15
      2.3.1 Segmentation of a 3D Image ................................................................. 15
      2.3.2 3D Analysis ......................................................................................... 16
   2.4 Measured Data and their Categories ......................................................... 18
   2.5 Statistics ................................................................................................... 20

3. ANALYSIS OF IMAGED CHROMOSOME TERRITORIES ..................... 23
   3.1 Morphological and Positional Relationships ............................................ 23
### 4. ANALYSIS OF SIMULATED CHROMOSOME TERRITORIES

4.1 Realistic Chromosome and Nucleus Shapes

4.2 One Chromosome Simulated Approach

4.2.1 Significance of Linear Scatterplots

4.3 Four Chromosome Simulated Attempt

### 5. CONCLUSION

5.1 Conclusions

5.2 Future Work

REFERENCES

APPENDIX A – C++ SIMULATION CODE

APPENDIX B – R CODE FOR SIMULATION PARAMETERS

APPENDIX C – IMAGED DATA PLOTS

APPENDIX D – SIMULATED DATA PLOTS

APPENDIX E – LINEAR SIMULATED DATA PLOTS

BIOGRAPHY OF THE AUTHOR
**LIST OF TABLES**

Table 2.1  Breakdown of Images Received ............................................. 19

Table 4.1  Scatterplots with p-values less than 0.001 ............................... 41

Table 4.2  Scatterplots with p-values greater than 0.001 and less than 0.1..... 42

Table 4.3  Scatterplots with p-values greater than 0.1 .............................. 43
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Mutually exclusive chromosome territories from a chicken fibroblast nucleus</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Non-neoplastic pancreatic ductal epithelium (A) and neoplastic pancreatic ductal adenocarcinoma epithelium (B) where the nuclear counterstain is blue and the DNA probe for chromosome 8 is green</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Comparison of automated thresholding of an image in B, E and H to WTMM segmentation in C, F and I of simulated CT images with increasing levels of white noise in A, D and G</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Maxima chains at 50 scales showing accurate edge detection of two objects</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>The algorithm selects only the largest two connected chains at each scale</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Chromosome territory outlined by maxima chain edge detection</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>2.5D WTMM method where up to 5 candidate objects are detected in the $xy$ (A), $yz$ (B), and $xz$ (C) planes. The intersection of the objects found in the planes is shown in D. The result of an erode-dilation step followed by a smoothing step is shown in E. Of the remaining objects, only the two largest are kept (F)</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>3D View of a Segmented Image, where the nucleus is in blue, one set of CTs is in green and a second set of CTs is in red</td>
<td>17</td>
</tr>
<tr>
<td>2.6</td>
<td>Visual 2D Interpretation of Variables</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>FID, Elongation and Rugosity versus Volume for all imaged data</td>
<td>24</td>
</tr>
</tbody>
</table>
Figure 3.2  FID versus Volume for ES and L CTs................................................. 25

Figure 3.3  FID, Elongation, Rugosity and Volume versus Centroid Distance for
all CTs........................................................................................................... 26

Figure 3.4  FID, Elongation, Rugosity and Volume versus Shell Distance for all
CTs .............................................................................................................. 27

Figure 3.5  FID, Elongation, Rugosity and Volume versus Radial Distance for all
CTs .............................................................................................................. 28

Figure 3.6  Centroid Distance versus FID for ES and L Chromosome 11 ............... 29

Figure 3.7  Rugosity Boxplots for All and ES Chromosomes............................. 30

Figure 3.8  FID Boxplots for All Chromosomes, ES Chromosomes, and L
Chromosomes and by Cell Type................................................................. 31

Figure 3.9  Volume Boxplots for All Chromosomes, ES Chromosomes, and L
Chromosomes and by Cell Type................................................................. 32

Figure 3.10 Radial Distance Boxplots for All Chromosomes, ES Chromosomes,
and L Chromosomes and by Cell Type ..................................................... 33

Figure 4.1  Radius Values Based on Volume and Surface Area Calculations......... 35

Figure 4.2  Visual explanation of C++ Simulation ........................................ 37

Figure 4.3  FID, Elongation, Rugosity and Volume vs Simulated Radial Distance
for all CTs................................................................. 38

Figure 4.4  FID, Elongation, Rugosity and Volume vs Simulated Radial Distance
for all CTs................................................................. 39
In eukaryotic cells, genetic information is stored in the nucleus and divided into chromosomes that carry specific genes. Chromosomes come in pairs, called homologous chromosomes, and were originally numbered in order of decreasing size of deoxyribonucleic acid (DNA) base pairs, the building blocks of chromosomes. During mitosis, or cell division, these chromosomes can easily be seen as a highly condensed entity, but their shape during interphase was unknown. The term "chromosome territory" was first used by Boveri in a paper published in 1909, and refers to the distinct space a chromosome occupies within the nucleus during interphase (Timme et al., 2011). This concept wasn’t confirmed until advances in microscopy at the turn of the 20th century, specifically with Fluorescent in situ hybridization (FISH) (Parada & Misteli, 2002).

One of the first images of multiple chromosomes territories (CTs) was from a chicken fibroblast nucleus that clearly showed homologous chromosomes in separate locations in the nucleus, which is shown in Figure 1.1 (Cremer & Cremer, 2001). This paper proposed two different models of how this structure allowed proteins to move within the nucleus, which are not mutually exclusive. The IC (interchromatin compartment) model proposes that the space between chromatin compartments is largely chromatin-free (Cremer & Cremer, 2001). This space exists between the CTs and inside an individual CT, which supports the evidence that large proteins can move through CTs. The CT-IC (chromosome-territory-interchromatin-compartment) model hypothesizes that transcription complexes can only diffuse within the IC space (Cremer & Cremer, 2001). This implies that the IC must have physical access to the transcriptionally active genes, and genes that require long-term silencing must be in the interior of the CTs. These models are supported by research which found that active transcription sites are scattered throughout CTs, not concentrated at the surface, which indicates the presence of interconnected channels.
throughout the CT (Parada & Misteli, 2002). As a result, chromatin remodeling events that change the positioning of genes are essential for gene-activation and gene-silencing mechanisms. This sparked the idea that the shape and position of the CT could have clinical relevance in identifying some diseases.

Chromosome territories exist in non-random positions within the nucleus, and each CT’s radial position, volume and shape correlate with their gene expression status. In humans, chromosome 1 contains the most base pairs and chromosome 22 has the least base pairs. However, there is not a linear correlation between number of base pairs and gene density of a CT. For example, human chromosome 18 has a low gene density, but human chromosome 19 has a high gene density. Gene density has been linked to both position and shape of the CT. Very gene-rich CTs were more irregular in shape, while very gene-poor CTs were more regularly ellipsoid-like. For CTs with intermediate gene densities, it was found that there was significant correlation between gene density and the degree of shape regularity (Sehgal et al., 2014).

Chromosome territories with a lower gene density, like human chromosome 18, are often found in a more peripheral position within the nucleus than CTs with a higher gene density, like human chromosome 19 (Timme et al., 2011). This arrangement is called radial
positioning (Parada & Misteli, 2002). Another study found that in spherical nuclei, 
gene-rich CTs are positioned towards the center of the nucleus, while in ellipsoidal nuclei, 
larger CTs are positioned towards the periphery of the nucleus (de Castro et al., 2021). 
Position is often measured by radial distance, which is the distance from the centroid of the 
CT to the centroid of the nucleus. Radial distance is considered to be evolutionarily 
conserved; in lymphocytes and fibroblasts there is evidence of conservation from over 300 
million years and the radial positioning of human CTs 18 and 19 have been found to be 
conserved from Old-World monkeys, spanning 30 million years (Parada & Misteli, 2002; 
Timme et al., 2011). This implies that there is a functional reason for non-random CT 
positioning. Another possible arrangement pattern for CTs is preferential relative 
positioning. This is observed in mitotic rosettes, where CTs arrange themselves in a ring 
during metaphase. Homologous CTs are separated by at least 90° and are often directly 
opposed. For some cells this chromosomal arrangement was also found during interphase, 
although it was not seen in chicken cells (Parada & Misteli, 2002). Other factors like 
tethering to the nuclear envelope, lamina or nuclear pores could impact chromosomal 
movement during interphase and could play a role in CT positioning.

Some studies, including (Mayer et al., 2005), have suggested that both CT size and 
gene density play a role in the arrangement of CTs, but others have found that only CT 
size had a linear correlation with radial distance (Marella, Bhattacharya, Mukherjee, Xu, & 
Berezney, 2009). In looking at three-dimensional organization of gene loci, it was 
determined that gene loci on the same CT significantly differ from each other and from 
their CT in terms of radial nuclear position (Hepperger, Mannes, Merz, Peters, & Dietzel, 
2008). They concluded that there was no correlation between radial distribution and gene 
density, so gene density is not a strong factor in determining the position of the CTs, or the 
individual loci in each CT. Additionally, (Stadler et al., 2004) found that as differentiation 
occurs, the CTs change shape, so the repositioning of loci within each CT may actually
occur due to morphological changes rather than transcriptional changes. Better imaging techniques and accurate measurements are needed to continue this research.

With the confirmation of non-random CT positioning, pathologies were investigated to determine if the shape or position of human CTs changed from the normal state. It has been established that in many cancers chromosomes 18 and 19 are repositioned, and in malignant breast epithelial cells chromosomes 1, 4, and 16 are larger while the nuclear volume stays consistent (de Castro et al., 2021). Chromosome 8 (CT8) was compared in a non-neoplastic human pancreatic ductal epithelium and in a pancreatic ductal adenocarcinoma to determine malignant transformation. Although there were no statistical differences found in radial distance, volume or surface measurements, there were differences in the normalized CT roundness values (Timme et al., 2011). In normal epithelium CT8 is very spheroidal, while in carcinoma epithelium CT8 is more irregularly shaped. This correlates with the observation that carcinoma cells and nuclei have decreased roundness as seen in Figure 1.2.

Another paper studied the differences between chromosome pairs 1, 4, 11, 12, 16, and 18 in the normal human lung fibroblast cell line, the normal human breast epithelial cell line and the malignant human breast cancer cell line (Marella et al., 2009). The CT volumes measured in the malignant breast cells were 1.4-2.0 fold higher than the normal breast cells without any significant increase in nuclear volume. A higher CT volume may correspond to the deregulation and higher expression levels often seen in cancer cells because genes usually in the interior of the CT may now be accessed. Additionally, interchromosomal associations were measured by determining the shortest edge-to-edge distance between CTs. By generating random simulations of CT positions, it was determined that the measured values of association were significantly higher than the randomly generated values (Marella et al., 2009). This supports the idea of non-random CT positioning with cell type specificity since the measurements were different between the breast and lung cells.
In determining the relationship between gene density and size and shape, 12 different CTs were studied from normal diploid human lung fibroblast cells. The CTs studied were 19, 17, and 11 for the gene rich category; Y, 13, and 18 for the gene poor category; 22, 1, 12, and X for the intermediate gene densities; and 22, 15, 21, and 13 for the nucleolar associated chromosomes (Sehgal et al., 2014). There was a negative correlation between the percent distribution of CTs in the regular category and their gene densities, and a positive correlation between percent distribution in the moderately to highly irregular category and their gene densities. There was no correlation between CT size and shape regularity. They found that homologous chromosomes have a similar degree of shape regularity, except in chromosome X, where in females one X chromosome is active and one is inactive, so these results were expected. They concluded that CT shape is likely due to numerous factors.
such as open chromatin levels, gene density and activity, positioning within the nucleus, and associations with other nuclear proteins (Sehgal et al., 2014).

Huntington’s Disease is a fatal progressive neurodegenerative disorder caused by an autosomal dominant mutation in the huntingtin gene (HTT) that causes disruptions of gene expression, nuclear invaginations and nuclear transport. The nuclei of human peripheral blood mononuclear cells (PBMCs) from early or moderate Huntington’s Disease patients were both significantly larger and more elongated than the healthy controls (de Castro et al., 2021). Both chromosomes 19 and 22 became significantly more spherical in HD PBMCs, which supports the theory that chromosome sphericity is associated with gene repression. Chromosome 19 also had significant differences in size, that was correlated with the increase in nuclear volume, and in a smaller radial distance, indicating that it moves closer to the center of the nucleus in diseased cells (de Castro et al., 2021). The results of this study indicate that blood samples could be used as an identifier of both the early and moderate stages of the disease. If cures could be found for this disease at an early stage, patients could be diagnosed and receive life-saving treatments sooner. This technique could also be researched for earlier detection in other diseases as well.

As the importance of identifying shape, size and position of CTs becomes apparent for identifying diseases at earlier stages, a method to determine these variables becomes essential. Most labs in this area of research have developed their own segmentation and analysis methods, including those detailed in (Marella et al., 2009) and (Sehgal et al., 2014) which use Otsu thresholding and visual inspection respectively. The method used in this lab, the Wavelet-Transform Modulus Maxima segmentation method, has been proven to accurately segment a simulated CT even when generated white noise prevents visual inspection or Otsu thresholding as seen in Figure 1.3 (Khalil et al., 2007). White noise is a natural part of imaging, and in other segmentation techniques it tends to reduce the accuracy of edge detection. The success of the WTMM method in this experiment confirms that the most accurate measurements will be achieved. Having a widely accepted and
Figure 1.3. Comparison of automated thresholding of an image in B, E and H to WTMM segmentation in C, F and I of simulated CT images with increasing levels of white noise in A, D and G

Source: (Khalil et al., 2007)

accurate CT segmentation method could entice more researchers to investigate CT morphology and positioning in different cell types and disease states.
CHAPTER 2
CHROMOSOME TERRITORY IMAGING, SEGMENTATION, AND STATISTICAL METHODS

2.1 Confocal Laser Scanning Microscopy

Steffen Dietzel provided confocal image stacks of Mus musculus (MMU) cells to the CompuMAINE lab. Procedures from the (Mayer et al., 2005) paper from Dietzel’s lab were closely followed for cell culture, isolation, and fixation, DNA and FISH hybridization, and confocal microscopy imaging. The reader is referred to the (Mayer et al., 2005) paper for details.

For cell culture, isolation, and fixation, MMU cells were morphologically preserved and formaldehyde fixed. EB-5 embryonic stem cells (ES-cells) were cultivated in DMEM with 15% FCS with additional supplements as described elsewhere under 5% CO$_2$. The ES-cells grew in gelatinized flasks without feeder cells. For 3D preparations, glass cover slips were coated with gelatin (pork skin gelatin, Sigma, Deisenhofen, Germany) by incubation and then air-dried. ES-cells in gelatin were incubated to allow attachment. 3D ES-cell growth in unregulated environments can lead to irregularly shaped nuclei. Only spherical nuclei were imaged.

MMU lymphocytes were isolated from pooled blood, provided by Dr. Manuela Mohr, Lehrstuhl fr molekulare Tierzucht und Biotechnologie, Ludwig-Maximilians-Universitt Mnchen, on a Ficoll gradient. Lymphocytes were cultivated in RPMI with 15% fetal bovine serum and stimulated with concanavalin A. The cells were centrifuged and then resuspended in 50% FCS/50% RPMI. For 3D preparations, glass cover slips were coated with poly-L-lysine by incubation. Cover slips were then water washed and air-dried. The lymphocyte cell suspension was incubated to allow attachment.

Using confocal microscopy, image stacks were generated from optical sections collected via Leica TCS 4D (100x, N.A. 1.4 Plan Apo Objective) and Zeiss LSM 410 (63x, N.A. 1.4
Plan Apo Objective) confocal microscopes. Voxel sizes were 80nm or below in $x$ and $y$ and 240nm or below in $z$ (Mayer et al., 2005).

For imaging in 3D, having a precise $z$ measurement is very important. Confocal scanning laser microscopy was first used to image the three-dimensional shape of chromatin in the nuclei of mouse neuroblastoma 2A cells in 1985 (Brakenhoff, Van Der Voort, Van Spronsen, Linnemans, & Nanninga, 1985). That technology had a localization volume of 220nm in the $x$ and $y$ directions and 730nm in the $z$ direction. Since then technology has rapidly improved the resolution in the lateral ($x$ and $y$) and axial ($z$) directions. If the images used for this project were retaken to maximize the accuracy of the measurements, the current method to use would be 3D Stochastic Optical Reconstruction Microscopy (STORM). By using this method of microscopy, imaging can reach a resolution of 30nm laterally and 50nm axially. This would require using the dye Alexa-647 and a solution of cyclooctatraene to have a sufficient number of photons with photoswitching capability (Olivier, Keller, Gönczy, & Manley, 2013). While other methods could be considered, like electron microscopy for a localization volume up to 0.2nm, or 4-pi microscopy for localization volume increase with a faster acquisition time, for the goals of this project, 3D STORM would be the most effective. A sensitivity analysis was performed using the equation:

$$\sigma = \sqrt{\frac{s^2}{N} + \frac{a^2}{N} + \frac{4\sqrt{\pi}s^2b^2}{aN^2}} \quad (2.1)$$

where $s =$ standard deviation of the Gaussian function approximating the point-spread function, $N =$ number of photons, $a =$ pixel size, and $b =$ standard deviation of the background (Allen, Silfies, Schwartz, & Davidson, 2016). This revealed that increasing the number of photons from 10 to 1000 would dramatically reduce the standard error of the localization precision, while increasing the number of photons from 1000 to 4000 would only slightly decrease the standard error. Therefore for this project, the ideal imaging
system would be a 3D STORM system using Alexa-647 to capture 1000 photons and achieve a localization volume of 30nm laterally and 50nm axially.

2.2 The 2D Wavelet Transform Modulus Maxima Method

The Wavelet Transform Modulus Maxima (WTMM) method was originally developed by A. Arneodo as a multifractal formalism to analyze highly complex 1D signals (Muzy, Bacry, & Arneodo, 1993), 2D images (Khalil et al., 2009; Khalil, Joncas, Nekka, Kestener, & Arneodo, 2006), and 3D images (Kestener & Arnéodo, 2004). A detailed description of the 2D methodology can be found in (Arnéodo, Decoster, & Roux, 2000). Beyond the multifractal formalism, two additional types of analyses were spun off from the 2D WTMM method in the 2000s: an anisotropy method and a segmentation method.

The 2D WTMM segmentation method was first introduced to perform accurate object segmentation of a specific number of mouse chromosome territories (Khalil et al., 2007). The methodology was generalized to segment any number of objects when characterizing ultra-thin gold aggregates in atomic force microscopy (Thibault et al., 2009). The next step was then to further generalize the technique to segment any number of objects in three dimensions. This was done by developing an intermediate technique referred to as a 2.5D segmentation analysis. A comparison of these techniques can be seen later in Figure 2.4. This allowed investigation into the three-dimensional radial positioning of GFP fluorescence associated with HPL-2/HP1 foci in *c. elegans* nuclei (Grant et al., 2010). The 2.5D *c. elegans* segmentation analysis was performed in an idealistic environment since the objects of interest were embryonic nuclei, which are well known to be spherical. Given that spheres are symmetrical in nature from all axes of revolution, the segmentation procedure only required analysis of the *xy* image sections. For CTs, this is not the case, but the 2.5D WTMM segmentation method is still effective since the *yz* and *xz* planes are taken into account, not just the *xy* plane.
Anisotropic analysis indicates where the intensity varies within an image depending on the direction of measurement. This analysis can be done using the WTMM method by recording the position of the local maxima of the gradient, and its angle $A$. This will provide the largest local variations of signal. This method has been used to study zebrafish, to determine the effect of Fibronectin disruption on muscle development (Snow, Peterson, Khalil, & Henry, 2008). The application could also be adapted to 3D space, with the use of a two angle system.

Image segmentation with continuous wavelets is based on the derivative of a 2D smoothing function (Gaussian filter) acting as an “edge detector” (Arnéodo et al., 2000). Let

$$\Phi_{GAU}(x, y) = \exp\left(-\frac{|x|^2}{2}\right),$$

(2.2)

where $|x| = \sqrt{x^2 + y^2}$, be the Gaussian function so that

$$\Psi_1(x, y) = \partial \Phi_{GAU}(x, y)/\partial x \quad \text{and}$$

$$\Psi_2(x, y) = \partial \Phi_{GAU}(x, y)/\partial y$$

(2.3)

are defined as the partial derivatives with respect to $x$ and $y$. For any 2D function $f$, such as an image, the continuous wavelet transform (WT) of $f$ with respect to $\Psi_1$ and $\Psi_2$ has two components:

$$T_{\Psi_1}[f] = \frac{1}{a^2} \int d^2x \Psi_1 \left(\frac{x - b}{a}\right) f(x) \quad \text{and}$$

(2.4)

$$T_{\Psi_2}[f] = \frac{1}{a^2} \int d^2x \Psi_2 \left(\frac{x - b}{a}\right) f(x)$$

where $b$ and $a$ represent parameters of position and scale, respectively. The WT is therefore the gradient vector of $f(x)$ smoothed by dilated versions of the Gaussian filter $\Phi_{GAU}(x/a)$. The WT can be written in polar coordinates, i.e. terms of its modulus and argument.
\[
T_\Psi[f](\mathbf{b}, a) = (M_\Psi[f](\mathbf{b}, a), A_\Psi[f](\mathbf{b}, a))
\]

(2.5)

where

\[
M_\Psi[f](\mathbf{b}, a) = \sqrt{T_{\Psi_1}[f]^2 + T_{\Psi_2}[f]^2} \quad \text{and} \quad A_\Psi[f](\mathbf{b}, a) = (T_{\Psi_1}[f] + iT_{\Psi_2}[f]).
\]

For a detailed description, the reader is referred to a previous work (Khalil et al., 2007).

At each size scale \(a\), the WTMM or intensity gradient maxima are defined by the positions where the modulus of WT, \(M_\Psi[f](\mathbf{b}, a)\), is locally maximal. These WTMM are automatically organized as maxima chains, which act as contour lines of the smoothed image at the considered scales. At a given scale, the algorithm scans all the boundary lines that correspond to the highest values of the gradient, i.e. the maxima chains. The algorithm then keeps only those maxima chains that are connected, consistent with the CT borders we want to detect. To find the first CT, the maxima chain with the highest average (mean) modulus is selected. This process is repeated across a wide range of size scales. The range of scales is limited by the resolution of the microscope and the size of the image. Figure 2.1 shows maxima chains over 50 scales, and clearly shows how the edges of the two CTs are accurately found as the scale becomes smaller.

The maxima chain with the highest mean modulus over all the scales is selected as the first CT bounding object. The algorithm again finds all the maxima chains at a given scale, across all scales, but now selects the maxima chain with the second highest mean modulus. The algorithm checks that the area bounded by this maxima chain is neither contained by the area bounded by nor contains the first object’s maxima chain. If this is not true, the algorithm selects the maxima chain with the next highest mean modulus and repeats the process until a maxima chain is found outside the maxima chain of the first object. This second object will be contained by the maxima chain with second highest
mean modulus in the image that is outside the area contained by the maxima chain bounding the first object. This process can be continued for any number of objects.

For our purposes, the algorithm finds up to five potential CT objects. If after a fixed number of objects (e.g. three or four) the algorithm can no longer find a maxima chain with a mean modulus lower than all previously found mean moduli and which is not contained within another maxima chain, the algorithm will stop and return only the number of objects it has found. This process can be seen in Figure 2.2, where the WTMM method was used on a simulated nucleus with two circular CTs. The end result of an imaged nucleus can be seen in Figure 2.3, and illustrates that assuming CTs have an ellipsoidal shape is not accurate, although it is more accurate than assuming a spherical shape for the nucleus.

Figure 2.1. Maxima chains at 50 scales showing accurate edge detection of two objects
Source: (Khalil et al., 2007)
Figure 2.2. The algorithm selects only the largest two connected chains at each scale
Source: (Khalil et al., 2007)

Figure 2.3. Chromosome territory outlined by maxima chain edge detection
Source: (Khalil et al., 2007)
Figure 2.4. 2.5D WTMM method where up to 5 candidate objects are detected in the $xy$ (A), $yz$ (B), and $xz$ (C) planes. The intersection of the objects found in the planes is shown in D. The result of an erode-dilation step followed by a smoothing step is shown in E. Of the remaining objects, only the two largest are kept (F).

2.3 The 2.5D Wavelet Transform Modulus Maxima Method

2.3.1 Segmentation of a 3D Image

The CT images are 3D data cubes represented as confocal microscopy stacks. The stack is a series of 2D images representing cross-sectional slices of a nucleus with stained CTs. The raw stacks have a voxel size of 80nm or below in $x$ and $y$ and 240nm or below in $z$ (Mayer et al., 2005). As such, the stacks are rescaled in $z$ using bicubic average interpolation in Fiji in order to obtain cubic voxels. The stack is then sliced into series of 2D images in three planes, $xy$, $xz$, and $yz$. Every image is segmented using the 2D WTMM segmentation method discussed above. This result can be seen in panels A, B and C of Figure 2.4.
After 2D segmentation, the maxima chains of each 2D slice are combined into 3D tube-like objects, with each “tube” representing the plane projection of a CT object. The “tubes” are filled in and points contained in the xy- and xz- and yz-generated “tubes” are kept. The “tubes” are each dilated twice, rechecking the intersection with other “tubes” after each dilation. The resulting point cloud containing the points of “tube” intersection are then eroded twice to compensate.

Since there are only two CT objects per confocal stack, usually only the “tubes” representing these CTs achieve 3D point intersection with the rest of the “tubes” lacking overlap. In case there are any residual points of intersection, only the two largest point clouds are kept. The result is the two 3D CT objects imaged in the confocal stack. This can be seen in panels D, E and F of Figure 2.4.

Separately, the DAPI-stained nucleus is segmented by Fiji’s auto-threshold feature. The WTMM segmentation was done by other members of the lab. These segmented images can be viewed in Figure 2.5 where the nucleus is blue, and there are four CTs, two that are green and two that are red.

2.3.2 3D Analysis

The bounding points or shell of the 3D CT objects were converted to a manifold triangular mesh using Point Cloud Library’s (PCL) Marching Cubes Hoppe algorithm (Hoppe, DeRose, Duchamp, McDonald, & Stuetzle, 1992). The surface area of the mesh was calculated via summing the areas of the mesh triangles. The volume of the CT was calculated by summing the signed volume of the tetrahedrons generated by the origin and the mesh triangles (Zhang & Chen, 2001). The sign of each tetrahedron is given by the triangle normals, which are calculated using the convention that all normals point away from the CT centroid. This was done by a previous member of the lab.

The volume, surface area, and diameter measurements performed by the 3D analysis software were calibrated against manually generated point clouds of known size (data not
Figure 2.5. 3D View of a Segmented Image, where the nucleus is in blue, one set of CTs is in green and a second set of CTs is in red shown). This calibration quantified the inherent uncertainty in the 3D measurements. The inherent uncertainty in ImageJ’s 3D measurement techniques is unknown. The filament index (Khalil, Joncas, & Nekka, 2004) of a 3D object,

\[ FID = \frac{SD}{6V}, \]

where \( S \) is the surface area, \( D \) is the diameter, and \( V \) is the volume of the object, indicates a shape’s departure from a spherical shape. A sphere has a filament index \( FID = 1 \).

Additionally, calculations were done to determine measurement of elongation and rugosity. Elongation is a representation of how long and thin an object is, while rugosity is a representation of the amount of cavities that are present on the surface of the object. A FID value of 9 could mean that the object is really long but smooth, or it could mean that the object is spherical but its membrane is rough and bumpy. Elongation and rugosity break FID down so that this distinction can be made. The elongation of an object can be calculated by

\[ E = \frac{\pi D^3}{6V}, \]
and the rugosity of an object can be calculated by

\[ R = \frac{S}{\pi D^2}. \]

These equations show that

\[ E \ast R = FID. \]

### 2.4 Measured Data and their Categories

From this segmentation, analysis was done on the images and measurements were taken. These measurements were concatenated into .txt files, which contained the following information: stack number, chromosome number, x-position, y-position, z-position, surface area, volume, diameter, convex hull area, convex hull volume, nucleus area, nucleus volume, shell distance to nucleus, centroid distance to nucleus, distance to other object, shell distance to other object, radial distance, O1C1O2C1, O1C1O1C2, O1C1O2C2, O2C1O1C2, O2C1O2C2, and O1C2O2C2. Convex hull area and convex hull volume are the area and volume of the smallest shape the 3D object can fit inside that has all convex angles, or less than 180 degrees. The shell distance to the nucleus is the distance from the outer edge of the CT to the nucleus. The radial distance is the distance from the centroid of the CT to the centroid of the nucleus. The distance to the other object is the distance from the centroid of one CT to the centroid of the second CT. The shell distance to the other object is the smallest distance between the edges of the two CTs. The centroid distance to nucleus is the largest distance from the centroid of a CT to the outer edge of the nucleus. The last six variables refer to how much the CTs overlap with each other. For example, O1C1O2C1 measures the area overlapped by object 1 chromosome 1 and object 2 chromosome 1. If the two CTs do not overlap then it is given a value of -1. If the objects do overlap, the volume that the two CTs share is recorded. All of these variable measurements can be visualized in 2D in Figure 2.6. The .txt files were split into two folders, one labeled ES-Zellen (which stands for embryonic stem cells) and one labelled Lymphocytes. ES will be used to
reference embryonic stem cell data and L will be used to reference lymphocyte cell data in the remainder of this paper. In the ES-Zellen folder there were 3 .txt files labeled blobs-eb5_29, blobs-eb5_114 and blobs-eb5_x11. In the Lymphocyte folder there were 5 .txt files labeled blobs-mly11xbios, blobs-mly11xdigs, blobs-stacksmly114, blobs-stacksmly114ii, and blobs-stacksmly29. The numbers in these file names indicate the chromosome numbers that were stained in the images. The combination of CTs are: 1 and 14, 2 and 9, X and 11. For example, blobs-eb5_29 contains information where blobs labeled object 1 correspond to chromosome 2, and blobs labeled object 2 correspond to chromosome 9. A more visual explanation can be seen in Table 2.1. The X chromosome data was not used because even in females with homologous chromosomes, there are confirmed volume and shape differences between the active and inactive chromosome. Therefore, these CTs would not contribute to the focus of this research.

In a .csv file, a professional experienced in chromosome identification went through each image and categorized them as Good or Bad. Bad was assigned if the chromosome was not there or appeared as two objects and so was given a label of either 0 or 2. Good was assigned if there was one chromosome and was given a label of 1. This was used so that the program only used good data to draw conclusions. This programming is not sophisticated enough to determine if data is good or bad on its own. This visual check ensures that all findings are based on good data.
2.5 Statistics

Once the good data has been separated and the variables have been determined, it is important to ascertain the relationship between the variables. For this project, three relationships were chosen to determine the best fit equation: linear, log-logistic and exponential decay. For the linear best fit, the RStudio function `lm` was used, and the coefficients produced by the summary of the model were temporarily saved for comparison to the other models. The equation used by the `lm` function is written as:

$$Y = b_0 + b_1 X$$ \hspace{1cm} (2.6)

where $b_0$ is the value of $Y$ when $X=0$, $b_1$ is the slope, and $X$ and $Y$ have a direct relationship (Onofri, 2019).

For the log-logistic best fit, the RStudio function `drm` with the factor `LL.3()` was used and the coefficients of the summary were again temporarily saved for comparison. The equation used for log-logistic curves is written as:

$$Y = c + \frac{d - c}{1 + exp(b[log(X) - log(c)])}$$ \hspace{1cm} (2.7)
where $b$ is the slope around the inflection point, $c$ is the lower asymptote, $d$ is the higher asymptote, and $e$ is the $X$ value that produces a response halfway between $d$ and $c$ (Onofri, 2019). LL.3 specifies a three-parameter logistic, where the lower limit, $c$, is equal to zero and the function is symmetric around the inflection point (Onofri, 2019). This simplifies the equation to:

$$Y = \frac{d}{1 + \exp(b[\log(X) - \log(e)])} \quad (2.8)$$

For the exponential decay best fit, the RStudio function drm with the factor EXD.3() was used and the coefficients of the summary were temporarily saved. The equation used by the EXD.3() function is written as:

$$Y = ae^{kX} \quad (2.9)$$

where $a$ is the value of $Y$ when $X=0$, and $k$ represents the relative change in $Y$ due to a change in $X$. If $k$ is positive then there is exponential growth, and if $k$ is negative then there is exponential decay (Onofri, 2019). The coefficients of the function summary represent how accurate the data matches the best fit line. Code was written so the lm, LL.3() and EXD.3() functions were run for each data set, and the function summary with the lowest coefficients, or the more accurate fit, was selected to produce that dataset’s plot. This was an accurate way of determining which regression trend best represented each individual data set.

To compare the significance of the relationship between the variables for each linear plot, p-values were calculated. A p-value is used in statistical analysis to represent the correlation between two sets of data and the values range from 0 to 1. Values above 0.05 are considered to be statistically similar and values below 0.05 are considered to be statistically different. The p-values of interest for this project are the values below 0.05. In addition, there are levels of significant difference below the 0.05 threshold. A p-value less than 0.05 is significant and a p-value less than 0.001 is highly significant. This system is typically given an asterisk ranking system where $P<0.05^*$ is given one asterisk, $P<0.01^{**}$
is given two asterisks and \( P<0.001 \)*** is given three asterisks and \( P<0.0001 \)**** is given four asterisks. In this paper when the p-values are color coded, green is for ****, pink is for ***, light blue is for **, yellow is for *, and white is for values above 0.05. In these plots the Pearson method was used to calculate the p-values. This method measures the linear dependence between the two variables plotted and assumes a normal distribution of the variables. While this method is effective for the linear plots, Pearson p-values are only relevant when variables have a linear correlation.
CHAPTER 3
ANALYSIS OF IMAGED CHROMOSOME TERRITORIES

3.1 Morphological and Positional Relationships

Scatterplots were made to determine the relationship between the variables volume, FID, elongation, rugosity, centroid distance, shell distance, and radial distance. This produced a total of 166 scatterplots of which 91 had a linear best-fit correlation, 53 had an exponential decay best-fit correlation and 22 had a log-logistic best-fit correlation. These can be viewed in Appendix C. Additionally, box-plots were created to show the relationship of FID, elongation, rugosity and volume by chromosome number and by chromosome type. The sets of data by chromosome were compared using the Wilcoxon test and p-values between individual comparisons and the global ANOVA p-value was printed onto the box-plots. These can be viewed at the end of Appendix C.

The first three scatterplots were FID versus Volume, Elongation versus Volume, and Rugosity versus Volume, which can be seen in Figure 3.1. The data for FID and volume had outliers removed if FID>10 or if volume>0.1, and this can be seen in the axis limits. There is a linear correlation between these two variables, but the p value is 0.1296, so there is no statistical difference between them. For the elongation data points there is one noticeable outlier above 35, but the majority of the points are below 15. For a perfect sphere, elongation would equal 1.0, so a value greater than 1.0 indicates that the object is long but skinny. It is interesting to note that rugosity was below 1.0 for all but one data point. For a perfect sphere, rugosity would equal 1.0, while for a flat disk, rugosity would equal 0.25. The majority of the data falls between these two numbers, indicating that the CT shape is long with cavities. Both elongation and rugosity had exponential decay relationships with volume, so the statistical significance of the relationship can’t be concluded from p-values.
and volume have an indirect relationship (although not significant), while for L cells there are

Figure 3.1. FID, Elongation and Rugosity versus Volume for all imaged data

Another important comparison is between ES cells and L cells. The FID vs Volume plots of these two categories can be seen in Figure 3.2. Both of these plots are linear, so the p-values can reveal the significance of the relationships. FID and volume for ES cells is not significantly different, but they are for L cells. It can also be seen that for ES cells, FID and volume have an indirect relationship (although not significant), while for L cells there is a direct relationship. This same comparison cannot be made for the elongation versus volume and rugosity versus volume plots, because although for ES cells both of these plots
are linear and have statistical significance, for L cells these plots have a log-logistic relationship.

For centroid distance, there was a linear relationship between it and FID, rugosity and volume, and an exponential decay relationship between it and elongation, which can be seen in Figure 3.3. For all three linear relationships there was a direct correlation, but there was only statistical significance between centroid distance and FID and between centroid distance and volume. A majority of the data points had a centroid distance of between 0.1 and 0.2, indicating all the chromosomes were fairly close to the edge of the nucleus.

For shell distance, there was a linear relationship between it and FID, elongation and volume, and an exponential decay relationship between it and rugosity, which can be seen in Figure 3.4. For the linear relationships, they all had an indirect relationship, and there was only significance between shell distance and volume. Most of the data points had a value for shell distance between 0.0 and 0.1, with many appearing to be exactly zero, which indicates that the CTs are very close or touch the edge of the nucleus.
rugosity, and radial distance and volume. Most of the data points had a value of radial distance and FID, radial distance and rugosity, which can be seen in Figure 3.5. There is an indirect relationship for each comparison, and there is statistical significance between radial distance and FID, radial distance and rugosity, and radial distance and volume. Most of the data points had a value of radial distance between 0.25 and 0.45. Since these distances are normalized and the radial distance is larger than the shell and centroid distances, it can be concluded that most of the CTs exist in the periphery of the nucleus.

Figure 3.3. FID, Elongation, Rugosity and Volume versus Centroid Distance for all CTs

For radial distance, there was a linear relationship between it and all four variables, which can be seen in Figure 3.5. There is an indirect relationship for each comparison, and there is statistical significance between radial distance and FID, radial distance and rugosity, and radial distance and volume. Most of the data points had a value of radial distance between 0.25 and 0.45. Since these distances are normalized and the radial distance is larger than the shell and centroid distances, it can be concluded that most of the CTs exist in the periphery of the nucleus.
Figure 3.4. FID, Elongation, Rugosity and Volume versus Shell Distance for all CTs

One factor that affects statistical analysis is population size. As can be seen in Figure 3.6, there are very few data points for chromosome 11, for both the ES cells and the L cells. Most of the data for chromosome 11 was removed during the Good or Bad categorization, where most were given a score of 0 because the chromosome did not appear in the image. This could be due to errors in the image labeling process. Since there are so few data points, the log-logistic trend could be inaccurate if more good data was added.

The box-plots of FID, elongation and rugosity vs chromosomes collectively have very low occurrences of significance. The plot Rugosity vs Chromosomes has the most
Different it is likely due to Figure 3.7. This lack of significance indicates that a majority of the chromosomes have a occurrences of significance with 6, and the highest level of significance is 0.001 < P < 0.01 (**). Additionally, the ANOVA p-value is only statistically significant for the Rugosity vs Chromosomes and the Rugosity vs ES Chromosomes, which can be seen in Figure 3.7. This lack of significance indicates that a majority of the chromosomes have a shape that is statistically similar to the others and if they are different it is likely due to the roughness of their membrane surface, not their length. It can be noticed that between cell types, the FID is significantly higher in ES CTs than in L CTs, as seen in Figure 3.8. The same trend occurs in elongation, and the opposite trend occurs in rugosity, and all
three relationship are very significant. This indicates that ES cell CTs are longer and smoother than L cell CTs.

In the box-plots of volume vs chromosomes, chromosome 11 does not follow the anticipated convention. Chromosomes were originally numbered so that chromosome 1 had the most Mega-basepairs (Mb) and the size decreased in order as the chromosome number became larger. By this convention, chromosome 11 should contain less Mb and therefore have a smaller volume than chromosome 9. When viewing the box-plots for volume in Figure 3.9, it can be seen that chromosome 11 has a larger median volume than chromosome 9, and is the only one to deviate from the expected pattern. Additionally, in the volume vs ES chromosomes it can be seen that the chromosomes do not appear to follow the expected pattern at all. Chromosome 2 has the smallest median volume, while chromosome 9 and 14 have similar median volumes and chromosome 11 has a larger median volume than those three. This could be due to the undifferentiated state of embryonic stem cells, in that their CTs are not developed completely like a differentiated cell. There is a

Figure 3.6. Centroid Distance versus FID for ES and L Chromosome 11
high statistical significance between cell type, where the volume of ES cell CTs is lower than that of L cell CTs, possibly a result of ES cells undifferentiated state as well.

There was very little significance in the boxplots of radial distance. Only the ANOVA p-value for L CTs was significant, and the highest level of significance between chromosome numbers is $0.001 < P < 0.01 (**). Comparing the cell types, it can be seen that there is a high degree of statistical significance, although visually it does not appear so. This indicates that ES cell CTs have a higher radial distance, so they prefer the periphery of the nucleus more than L cell CTs do.
Figure 3.8. FID Boxplots for All Chromosomes, ES Chromosomes, and L Chromosomes and by Cell Type
Figure 3.9. Volume Boxplots for All Chromosomes, ES Chromosomes, and L Chromosomes and by Cell Type
Figure 3.10. Radial Distance Boxplots for All Chromosomes, ES Chromosomes, and L Chromosomes and by Cell Type
CHAPTER 4
ANALYSIS OF SIMULATED CHROMOSOME TERRITORIES

4.1 Realistic Chromosome and Nucleus Shapes

Once the morphological and positional relationships were found with the imaged data, a computer simulation was written to make randomized measurements. This was done so that the imaged data could be compared with purely random data for statistical significance. Volume, FID, elongation and rugosity were kept constant so that positional relationships could be studied. To keep these variables constant, the surface area, volume and diameter measurements of the imaged data were used to find the parameters for the simulation. An ellipsoidal shape was assumed for the CTs, so values for the ellipsoid axis ratios $a, b,$ and $c$ needed to be found for each imaged CT. The following equations were used to accomplish this task:

\[
S = 4\pi \left( \frac{((a \ast b)^p + (a \ast c)^p + (b \ast c)^p)}{3} \right)^{\frac{1}{p}} \tag{4.1}
\]

\[
V = \frac{2\pi}{3} abc \tag{4.2}
\]

\[
D = 2c \tag{4.3}
\]

where $p = 1.6$, $S$ is surface area, $V$ is volume, and $D$ is diameter. Since $c$ was a known value from the diameter equation, the equations for surface area and volume where placed in a for loop to find the smallest possible difference based on varying values of $a$ and $b$. Once the best combination was found, this was determined to be the most realistic ellipsoidal shape based on the imaged CT.

A spherical shape was assumed for the nucleus. This allowed the values for volume and surface area to be used to find realistic radius values based on the following equations:

\[
r = \left( \frac{3V}{4\pi} \right)^{1/3} \tag{4.4}
\]

\[
r = \left( \frac{S}{4\pi} \right)^{1/2} \tag{4.5}
\]
where \( r \) is radius. To determine if the nucleus could accurately be considered spherical, the two values were graphed against each other. If the nucleus was mostly spherical, the two radii would follow the \( y = x \) line, which can be seen in Figure 4.1. This justified the spherical assumption for the simulation.

\section*{4.2 One Chromosome Simulated Approach}

Once the parameters were found using the measurements from the imaged data, C++ code was written to place an ellipsoidal CT inside a spherical nucleus. The code randomly placed and randomly oriented the CT inside the nucleus, and then checked to ensure the whole CT was inside the nucleus. The radial distance was then measured and saved and the program restarted. If the code did not successfully place the CT within the nucleus boundaries after 100 tries, the program was restarted as the centroid of the CT was too close to the edge of the nucleus to produce a successful result. Within Rstudio, the C++ code was run until 250 successes were recorded for each set of parameters calculated for an imaged CT. This took between 4.8 and 9.2 hours depending on how many imaged CTs there were in the original .txt files. Of the 250 successes, a random value of radial distance
was taken to compare to the corresponding imaged CT. A graphic of this explanation can be seen in Figure 4.2 below.

Scatterplots were generated where the code was allowed to pick the best fit line between linear, exponential decay and log-logistic and were plotted with the corresponding imaged data. These scatterplots can be seen in Appendix D. There are a total of 52 plots, where blue represents the imaged data and red represents the simulated data. There are 9 possible combinations of trendlines for the imaged and simulated data. Written as imaged-simulated, there were 23 linear-linear, 2 linear-exponential decay, 1 linear-log-logistic, 13 exponential decay-linear, 5 exponential decay-exponential decay, 7 log-logistic-linear, and 1 log-logisitic-log-logistic. There were no plots of exponential decay-log-logistic or log-logistic-exponential decay.

There was a linear-linear relationship for radial distance versus FID, elongation, rugosity and volume, which can be seen in Figure 4.4. Five of relationships had an inverse relationship, while simulated radial distance versus rugosity had a direct relationship. Additionally, imaged radial distance vs elongation was the only relationship that did not have statistical significance. In all four cases the simulated p-value was much more significant than the imaged p-value. Another interesting observation is that the simulated radial distance is about half that of the imaged radial distance. This confirms the hypothesis that the CTs observed in this study tend to prefer the periphery of the nucleus. There is a possibility that the shape of the ellipsoidal CT and spherical nucleus prevents the simulated radial distance from being any farther from the centroid of the nucleus, so future investigations should include making a simulation with more realistic shapes.

Box-plots were graphed to find any significant differences in the radial distances based on chromosome number or cell type. As seen in Figure 4.4, there is a lot of very significant statistical differences in radial distance. The radial distance measurements were normalized by dividing by the diameter of the nucleus, so varying nucleus sizes did not impact the
Figure 4.2. Visual explanation of C++ Simulation
Figure 4.3. FID, Elongation, Rugosity and Volume vs Simulated Radial Distance for all CTs

(a) Simulated Radial Distance vs FID
(b) Simulated Radial Distance vs Elongation
(c) Simulated Radial Distance vs Rugosity
(d) Simulated Radial Distance vs Volume

Figure 4.3. FID, Elongation, Rugosity and Volume vs Simulated Radial Distance for all CTs

statistics. Additionally, for all four plots, the ANOVA p-value was less than $2.2e^{-16}$, which is the lowest possible p-value produced by Rstudio.

4.2.1 Significance of Linear Scatterplots

Since most of the plots had a linear-linear combination, and there is the possibility that a lack of data points swayed the trendline choice, scatterplots where all the trendlines were forced to be linear were created, and can be seen in Appendix E. The p-values for the

38
Figure 4.4. FID, Elongation, Rugosity and Volume vs Simulated Radial Distance for all CTs. Linear scatterplots can be viewed below in Tables 4.2.1, 4.2.1, and 4.2.1, and can be used to determine the relationship between each data set. Compared to the imaged p-values, all the corresponding simulated p-values were more significant. Considering that the simulated data is randomized, the opposite trend was anticipated. It could be that the shape of the ellipsoidal CT does impact the radial position, or a random value of the 250 radial distances should be selected instead of the median value. For the imaged data, 10 plots had...
significance for L CTs but no plots for ES CTs had significance. This is hypothesized to be due to the undifferentiated state of ES cells.

4.3 Four Chromosome Simulated Attempt

To more accurately simulate the imaged data, four ellipsoidal CTs would be randomly placed inside a spherical nucleus. This code was written in C++, but it was found that it took too long computationally for the program to find the right orientation of four CTs so that they were all inside the nucleus and did not overlap with each other. For example, for one simulation, it took over 17 hours to get three successful results. Future work with this code should look into supercomputers in an attempt to reduce the computational time.
<table>
<thead>
<tr>
<th>Table 4.1. Scatterplots with p-values less than 0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 14 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 1 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 2 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 9 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 2 Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated Rugosity vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 14 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 14 Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 9 Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Volume vs Radial Distance</td>
</tr>
<tr>
<td>L Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 2 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Rugosity vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 14 Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 9 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 2 Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 1 Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 9 Elongation vs Radial Distance</td>
</tr>
<tr>
<td>L Chrom 14 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 1 Volume vs Radial Distance</td>
</tr>
<tr>
<td>L Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 1 Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 9 Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>L Chrom 9 Filament Index vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 9 Elongation vs Radial Distance</td>
</tr>
<tr>
<td>Simulated L Chrom 11 Volume vs Radial Distance</td>
</tr>
<tr>
<td>Simulated ES Chrom 14 Filament Index vs Radial Distance</td>
</tr>
</tbody>
</table>
### Table 4.2. Scatterplots with p-values greater than 0.001 and less than 0.1

<table>
<thead>
<tr>
<th>Scatterplot</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated L Chrom 14 Rugosity vs Radial Distance</td>
<td>0.001072594</td>
</tr>
<tr>
<td>L Chrom 1 Volume vs Radial Distance</td>
<td>0.00379353</td>
</tr>
<tr>
<td>L Chrom 2 Volume vs Radial Distance</td>
<td>0.004194049</td>
</tr>
<tr>
<td>L Chrom 9 Volume vs Radial Distance</td>
<td>0.004733385</td>
</tr>
<tr>
<td>Simulated ES Rugosity vs Radial Distance</td>
<td>0.005149298</td>
</tr>
<tr>
<td>Simulated ES Chrom 1 Filament Index vs Radial Distance</td>
<td>0.006649125</td>
</tr>
<tr>
<td>Simulated ES Chrom 1 Elongation vs Radial Distance</td>
<td>0.008581613</td>
</tr>
<tr>
<td>Simulated ES Chrom 1 Rugosity vs Radial Distance</td>
<td>0.009022006</td>
</tr>
<tr>
<td>Simulated L Chrom 1 Rugosity vs Radial Distance</td>
<td>0.01068318</td>
</tr>
<tr>
<td>L Elongation vs Radial Distance</td>
<td>0.01192875</td>
</tr>
<tr>
<td>L Chrom 9 Elongation vs Radial Distance</td>
<td>0.0135515</td>
</tr>
<tr>
<td>Simulated L Chrom 9 Rugosity vs Radial Distance</td>
<td>0.01459024</td>
</tr>
<tr>
<td>Simulated ES Chrom 9 Rugosity vs Radial Distance</td>
<td>0.0155437</td>
</tr>
<tr>
<td>Simulated ES Chrom 11 Filament Index vs Radial Distance</td>
<td>0.02401603</td>
</tr>
<tr>
<td>Filament Index vs Radial Distance</td>
<td>0.02994661</td>
</tr>
<tr>
<td>Simulated ES Chrom 11 Elongation vs Radial Distance</td>
<td>0.0338091</td>
</tr>
<tr>
<td>L Chrom 14 Filament Index vs Radial Distance</td>
<td>0.04243729</td>
</tr>
<tr>
<td>Simulated ES Chrom 14 Rugosity vs Radial Distance</td>
<td>0.04673189</td>
</tr>
<tr>
<td>Rugosity vs Radial Distance</td>
<td>0.04795123</td>
</tr>
<tr>
<td>Simulated ES Chrom 14 Elongation vs Radial Distance</td>
<td>0.0508473</td>
</tr>
<tr>
<td>L Chrom 2 Filament Index vs Radial Distance</td>
<td>0.05671417</td>
</tr>
<tr>
<td>L Chrom 14 Elongation vs Radial Distance</td>
<td>0.05974745</td>
</tr>
<tr>
<td>Simulated L Chrom 2 Rugosity vs Radial Distance</td>
<td>0.06310271</td>
</tr>
<tr>
<td>Scatterplots with p-values greater than 0.1</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>L Chrom 1 Filament Index vs Radial Distance</td>
<td>0.1014473</td>
</tr>
<tr>
<td>Simulated ES Chrom 11 Volume vs Radial Distance</td>
<td>0.1489866</td>
</tr>
<tr>
<td>Simulated ES Chrom 2 Rugosity vs Radial Distance</td>
<td>0.1563711</td>
</tr>
<tr>
<td>ES Chrom 14 Volume vs Radial Distance</td>
<td>0.1608595</td>
</tr>
<tr>
<td>ES Volume vs Radial Distance</td>
<td>0.1656788</td>
</tr>
<tr>
<td>ES Chrom 1 Rugosity vs Radial Distance</td>
<td>0.1662591</td>
</tr>
<tr>
<td>Simulated L Chrom 11 Rugosity vs Radial Distance</td>
<td>0.1752311</td>
</tr>
<tr>
<td>ES Chrom 9 Elongation vs Radial Distance</td>
<td>0.1791627</td>
</tr>
<tr>
<td>L Chrom 1 Rugosity vs Radial Distance</td>
<td>0.2146924</td>
</tr>
<tr>
<td>L Chrom 2 Rugosity vs Radial Distance</td>
<td>0.2165542</td>
</tr>
<tr>
<td>ES Chrom 2 Filament Index vs Radial Distance</td>
<td>0.2252128</td>
</tr>
<tr>
<td>ES Chrom 11 Filament Index vs Radial Distance</td>
<td>0.2496893</td>
</tr>
<tr>
<td>ES Chrom 1 Elongation vs Radial Distance</td>
<td>0.2736558</td>
</tr>
<tr>
<td>ES Chrom 9 Rugosity vs Radial Distance</td>
<td>0.2797954</td>
</tr>
<tr>
<td>ES Rugosity vs Radial Distance</td>
<td>0.3111347</td>
</tr>
<tr>
<td>ES Chrom 2 Elongation vs Radial Distance</td>
<td>0.3182785</td>
</tr>
<tr>
<td>ES Filament Index vs Radial Distance</td>
<td>0.3188412</td>
</tr>
<tr>
<td>ES Chrom 2 Volume vs Radial Distance</td>
<td>0.3217271</td>
</tr>
<tr>
<td>Simulated ES Chrom 11 Rugosity vs Radial Distance</td>
<td>0.3491551</td>
</tr>
<tr>
<td>L Rugosity vs Radial Distance</td>
<td>0.3608036</td>
</tr>
<tr>
<td>Simulated ES Chrom 2 Filament Index vs Radial Distance</td>
<td>0.3672757</td>
</tr>
<tr>
<td>ES Chrom 1 Volume vs Radial Distance</td>
<td>0.3740831</td>
</tr>
<tr>
<td>L Chrom 2 Elongation vs Radial Distance</td>
<td>0.4598888</td>
</tr>
<tr>
<td>ES Chrom 2 Rugosity vs Radial Distance</td>
<td>0.4660814</td>
</tr>
<tr>
<td>ES Chrom 11 Elongation vs Radial Distance</td>
<td>0.4715501</td>
</tr>
<tr>
<td>ES Chrom 11 Volume vs Radial Distance</td>
<td>0.4850437</td>
</tr>
<tr>
<td>Simulated L Chrom 11 Filament Index vs Radial Distance</td>
<td>0.5246377</td>
</tr>
<tr>
<td>ES Chrom 9 Filament Index vs Radial Distance</td>
<td>0.556076</td>
</tr>
<tr>
<td>L Chrom 11 Elongation vs Radial Distance</td>
<td>0.5710863</td>
</tr>
<tr>
<td>Elongation vs Radial Distance</td>
<td>0.5944848</td>
</tr>
<tr>
<td>L Chrom 11 Filament Index vs Radial Distance</td>
<td>0.6375397</td>
</tr>
<tr>
<td>ES Elongation vs Radial Distance</td>
<td>0.589196</td>
</tr>
<tr>
<td>ES Chrom 11 Rugosity vs Radial Distance</td>
<td>0.6844064</td>
</tr>
<tr>
<td>Simulated L Chrom 11 Elongation vs Radial Distance</td>
<td>0.7320977</td>
</tr>
<tr>
<td>L Chrom 14 Rugosity vs Radial Distance</td>
<td>0.7497118</td>
</tr>
<tr>
<td>L Chrom 11 Volume vs Radial Distance</td>
<td>0.7787473</td>
</tr>
<tr>
<td>ES Chrom 14 Rugosity vs Radial Distance</td>
<td>0.8111908</td>
</tr>
<tr>
<td>ES Chrom 9 Volume vs Radial Distance</td>
<td>0.8228885</td>
</tr>
<tr>
<td>L Chrom 1 Elongation vs Radial Distance</td>
<td>0.8293727</td>
</tr>
<tr>
<td>L Chrom 9 Rugosity vs Radial Distance</td>
<td>0.8388918</td>
</tr>
<tr>
<td>L Chrom 11 Rugosity vs Radial Distance</td>
<td>0.8398612</td>
</tr>
</tbody>
</table>
5.1 Conclusions

Using the 2.5D WTMM segmentation method, accurate measurements were taken of chromosome territories within ES and L cells. This allowed various variables to undergo statistical analysis to determine the typical morphological and positional relationships of the CTs. Ideally, this baseline in normal cells will be used to diagnose various diseases at early stages that have deviations in morphological or positional relationships.

The relationships between FID, elongation, rugosity, volume, centroid distance, shell distance and radial distance have been determined between categories of cell type and chromosome number. For the imaged data, L CTs have more significant relationships than the ES CTs do. Additionally, the FID of the L CTs is lower than that of the ES CTs, the elongation is lower, the rugosity is higher, the volume is larger, and the radial distance is smaller. This indicates that the L CTs are more spherical and closer to the center of the nucleus. This is probably due to the undifferentiated state of ES cells, where they are more elongated and can fit better along the periphery of the nucleus. Chromosome 11 had no significance in either the L or ES cells. This could be because most of the data for this category was thrown out before analysis. Before any conclusions can be made about chromosome 11, more data would have to be studied. It was determined that CTs tend to have cavities dipping inward on their surfaces, and they tend to exist in the periphery of the nucleus.

When determining the requirements necessary for a randomized model, it was concluded that a spherical nucleus was an adequate assumption because a plot of radius from the volume and surface area measurements followed the $y = x$ line closely. The randomized values of radial distance were about half of those of the measured radial distance, which implies that CTs tend to be closer to the periphery of the nucleus than
chance would allow. The simulated data had a lot of statistical significance, much more than that of the imaged data, when all the plots were forced to have a linear trendline. This may be due to effects of assuming an ellipsoidal CT in the model. The simulated data does support the positional conclusions of the imaged data.

5.2 Future Work

Although most of the plots did have a best fit trendline of a linear relationship, more images should be analyzed to determine if the assumption to force them all to have a linear relationship is justified or not. Although this allows for easier comparison by using p-values, if a linear relationship is incorrect, the p-values would be invalidated and could allow for misleading conclusions.

It would also be useful to measure centroid and shell distance within the simulated model so that all three of the positional distances studied with the imaged data could be compared to random values. This would increase the computational time necessary to successfully run the simulation, so further work should consider a supercomputer to help cut this time. The simulation model only compares the positional data to random values. Another avenue to explore would be to hold the positional distances constant and see what shape and size CT could fit in the space. This would allow the volume, FID, elongation and rugosity values to be compared to random values. This could also require a lot of computational time depending on the complexity of equations required to vary the shape of the CT, which would be another motivating factor to explore supercomputers.
REFERENCES


APPENDIX A
C++ SIMULATION CODE

/*
   **************************************************************************
   **************************************************************************
   Modified in 2021 by Hannah Varney to measure radial distance of one CT
   placed in the nucleus. Fails after a number of attempts loop_max, under
   the assumption that the centroid of the CT is too close to the edge of the
   nucleus.
   **** Last modified by H. Varney 2021 05 13 ******************************

   To compile:

   g++ 3DEllipsiods_1CT_2021_05_13.cpp -o 3DEllipsoids_1CT_2021_05_13

   */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream>
#include <unistd.h>
#include <sys/time.h>

using namespace std;

double **g_A1;
double **h1;
double **g_A2;
double **h2;
double ***M_A1;
double ***M_A2;

int main(int argc, char *argv[]) 
{
    /* initialize random generator */
    //srand ( time(0) );
    struct timeval tv; // C requires "struct timval" instead of just "timeval"
    gettimeofday(&tv, 0);

    // use BOTH microsecond precision AND pid as seed
    long int n = tv.tv_usec * getpid();
    srand(n);

    int i, j, k, x, y, z, u, v, w;
    double i_A, j_A, k_A;
    double A[2];
    double a_A1 = atof(argv[1]);
    double b_A1 = atof(argv[2]);
    double c_A1 = atof(argv[3]);
    double rad_nuc=atof(argv[4]);
    double size = 2*rad_nuc+4;
    double squish = 1;
int r, s;
int xmax = int(2*size);
int ymax = int(2*size);
int zmax = int(2*size);

g_A1 = new double *[xmax]; //allocates memory for the array g[x][y]
h1 = new double *[xmax];
g_A2 = new double *[xmax];
h2 = new double *[xmax];
for (r=0; r < xmax; r++) {
    g_A1[r] = new double [ymax];
    h1[r] = new double [ymax];
    g_A2[r] = new double [ymax];
    h2[r] = new double [ymax];
}
M_A1 = new double **[xmax];
for (r=0; r < xmax; r++) {
    M_A1[r] = new double *[ymax];
    for (s=0; s < ymax; s++) {
        M_A1[r][s] = new double [zmax];
    }
}
M_A2 = new double **[xmax];
for (r=0; r < xmax; r++) {
    M_A2[r] = new double *[ymax];
    for (s=0; s < ymax; s++) {
double urand, vrand;
double theta_x_A, theta_y_A;
double cosx_A, sinx_A, cosy_A, siny_A, dist_A;
double x_A1, y_A1, z_A1;

/*
 * Here's the construction of the chromosome territory
 */

A[0]=0;
A[1]=0;
A[2]=0;

int outside = 1;
int counter1 = 0;
int loop_max = 100;
while ( (outside == 1) && (counter1<loop_max) ) {
    A[0] = rand()/(double)RAND_MAX + 1)*(2*(rad_nuc)) + size/2-(rad_nuc);
    A[1] = rand()/(double)RAND_MAX + 1)*(2*(rad_nuc)) + size/2-(rad_nuc);
    A[2] = rand()/(double)RAND_MAX + 1)*(2*(rad_nuc)) + size/2-(rad_nuc);
    counter1++;
    if ( (sqrt( (A[0]-size/2)*(A[0]-size/2) + (A[1]-size/2)*(A[1]-size/2)
...
< (squish*rad_nuc-c_A1)) ) {
    urand = rand()/((double)RAND_MAX + 1);
    vrand = rand()/((double)RAND_MAX + 1);
    theta_x_A = 2*3.14159265*urand;
    theta_y_A = acos(2*vrand-1);
    cosx_A = cos(theta_x_A);
    sinx_A = sin(theta_x_A);
    cosy_A = cos(theta_y_A);
    siny_A = sin(theta_y_A);

    /* M_A1 and M_B1 tell us whether we are inside (=1) or outside the
      3D ellipsoid */
    outside = 0;
    for (i = 0; i < size; i++) {
        for (j = 0; j < size; j++) {
            for (k = 0; k < size; k++) {
                i_A = (cosx_A*cosy_A)*(i-A[0]) + (-sinx_A)*(j-A[1]) +
                    (cosx_A*siny_A)*(k-A[2]);
                j_A = (sinx_A*cosy_A)*(i-A[0]) + (cosx_A)*(j-A[1]) +
                    (sinx_A*siny_A)*(k-A[2]);
                k_A = (-siny_A)*(i-A[0]) + (cosy_A)*(k-A[2]);
                dist_A = i_A*i_A/a_A1/a_A1 + j_A*j_A/b_A1/b_A1 +
                    k_A*k_A/c_A1/c_A1;
                if (dist_A <= 1) {
                    M_A1[i][j][k]=1;
                }
            }
            if ( M_A1[i][j][k] == 1 ) {
                if (sqrt((i-size/2)*(i-size/2) + (j-size/2)*(j-size/2) +

53
if ( (counter1==loop_max) || (counter1==loop_max+1) ) {
    cout<<"Failed"<<’\n’;
} else {
    double radialDA1=sqrt((x_A1-size/2)*(x_A1-size/2) +
    (y_A1-size/2)*(y_A1-size/2) + (z_A1-size/2)*(z_A1-size/2));
    cout<<radialDA1<<’\n’;
}
return 0;
**APPENDIX B**

**R CODE FOR SIMULATION PARAMETERS**

# Author: Hannah Varney

# Convert Surface Area, Volume and Diameter to a, b, c values so that realistic
# ellipsoids can be made and find radius of nucleus

# Calculate the filament index
getFID <- function(blobs)
{
    return(blobs$Surface.Area*blobs$Diameter/(6*blobs$Volume))
}

# Set the working directory
setwd("/Users/hannah/Desktop/Masters/Thesis/Rscripts_from_Zach/CT_data")

# load in blobs files
blobs1 <- read.csv("ES-Zellen/blobs-eb5_114.txt",sep="\t")
blobs2 <- read.csv("ES-Zellen/blobs-eb5_29.txt",sep="\t")
blobs3 <- read.csv("ES-Zellen/blobs-eb5_x11.txt",sep="\t")
blobs4 <- read.csv("Lymphocytes/blobs-mly11xbios.txt",sep="\t")
blobs5 <- read.csv("Lymphocytes/blobs-mly11xdigs.txt",sep="\t")
blobs6 <- read.csv("Lymphocytes/blobs-stacksmly114.txt",sep="\t")
blobs7 <- read.csv("Lymphocytes/blobs-stacksmly114ii.txt",sep="\t")
blobs8 <- read.csv("Lymphocytes/blobs-stacksmyl29.txt",sep="\t")

# Make a column for the folder names
blobs1['Folder'] <- "EB5_114"
blobs2['Folder'] <- "EB5_29"
blobs3['Folder'] <- "EB5_X11"
blobs4['Folder'] <- "MLy11XbioS"
blobs5['Folder'] <- "MLy11XdigS"
blobs6['Folder'] <- "StacksMly114"
blobs7['Folder'] <- "StacksMly114II"
blobs8['Folder'] <- "StacksMy129"

# Add the chromosome numbers
blobs1['Chrom.Name'] <- "1"
blobs1[blobs1$Chromosome=="chrom2","Chrom.Name"] <- "14"
blobs2['Chrom.Name'] <- "2"
blobs2[blobs2$Chromosome=="chrom2","Chrom.Name"] <- "9"
blobs3['Chrom.Name'] <- "X"
blobs3[blobs3$Chromosome=="chrom2","Chrom.Name"] <- "11"
blobs4['Chrom.Name'] <- "11"
blobs4[blobs4$Chromosome=="chrom2","Chrom.Name"] <- "X"
blobs5['Chrom.Name'] <- "11"
blobs5[blobs5$Chromosome=="chrom2","Chrom.Name"] <- "X"
blobs6['Chrom.Name'] <- "1"
blobs6[blobs6$Chromosome=="chrom2","Chrom.Name"] <- "14"
blobs7['Chrom.Name'] <- "1"
blobs7[blobs7$Chromosome=="chrom2","Chrom.Name"] <- "14"
blobs8['Chrom.Name'] <- "2"
blobs8[blobs8$Chromosome=="chrom2","Chrom.Name"] <- "9"
# Remove volume outliers
blobs1["V/NV"] <- blobs1$Volume/blobs1$Nucleus.Volume
blobs1 <- subset(blobs1,blobs1$'V/NV'<0.10)
blobs2["V/NV"] <- blobs2$Volume/blobs2$Nucleus.Volume
blobs2 <- subset(blobs2,blobs2$'V/NV'<0.10)
blobs3["V/NV"] <- blobs3$Volume/blobs3$Nucleus.Volume
blobs3 <- subset(blobs3,blobs3$'V/NV'<0.10)
blobs4["V/NV"] <- blobs4$Volume/blobs4$Nucleus.Volume
blobs4 <- subset(blobs4,blobs4$'V/NV'<0.10)
blobs5["V/NV"] <- blobs5$Volume/blobs5$Nucleus.Volume
blobs5 <- subset(blobs5,blobs5$'V/NV'<0.10)
blobs6["V/NV"] <- blobs6$Volume/blobs6$Nucleus.Volume
blobs6 <- subset(blobs6,blobs6$'V/NV'<0.10)
blobs7["V/NV"] <- blobs7$Volume/blobs7$Nucleus.Volume
blobs7 <- subset(blobs7,blobs7$'V/NV'<0.10)
blobs8["V/NV"] <- blobs8$Volume/blobs8$Nucleus.Volume
blobs8 <- subset(blobs8,blobs8$'V/NV'<0.10)

# Remove FID outliers
blobs1["FID"] <- getFID(blobs1)
blobs1 <- subset(blobs1,blobs1$FID<10)
blobs2["FID"] <- getFID(blobs2)
blobs2 <- subset(blobs2,blobs2$FID<10)
blobs3["FID"] <- getFID(blobs3)
blobs3 <- subset(blobs3,blobs3$FID<10)
blobs4["FID"] <- getFID(blobs4)
blobs4 <- subset(blobs4,blobs4$FID<10)
blobs5["FID"] <- getFID(blobs5)
blobs5 <- subset(blobs5,blobs5$FID<10)
blobs6["FID"] <- getFID(blobs6)
blobs6 <- subset(blobs6,blobs6$FID<10)
blobs7["FID"] <- getFID(blobs7)
blobs7 <- subset(blobs7,blobs7$FID<10)
blobs8["FID"] <- getFID(blobs8)
blobs8 <- subset(blobs8,blobs8$FID<10)

# Make one list of all the blobs
csvlist <- rbind(blobs1,blobs2,blobs3,blobs4,blobs5,blobs6,blobs7,blobs8)
csvlist$Stack <- as.numeric(gsub("stack","",csvlist$Stack))
csvlist$Chromosome <- as.numeric(gsub("chrom","",csvlist$Chromosome))

good <- read.csv("chs.csv") # Load in the file list of good images

good <- sortBlobs(good)
row.names(good) <- NULL
csvlist <- sortBlobs(csvlist)
row.names(csvlist) <- NULL

# Get good blobs
goodList <- getGood(csvlist, good)

# In case the program screwed up where the visual check did not
goodList <- goodList[goodList$Volume != -1,]
row.names(goodList) <- NULL

# Get nuclear diameter estimates
goodList[, "Diameter.Nucleus"] = (((3/(4*pi))*goodList$Nucleus.Volume)^(1/3))
+((goodList$Nucleus.Area/(4*pi))^(1/2))

# Remove an lines with no chromosome
goodList = goodList[goodList$Diameter.Nucleus!=0,]
# Remove any X chromosomes
goodList = goodList[goodList$Chrom.Name!="X",]
row.names(goodList) <- NULL

goodList = na.omit(goodList) # Remove files with NA

# Create empty SA, Vol and Dia files for each loaded file
for (i in 1:length(goodList)) {
  assign(paste("S0_list", i, sep = ""), i)
  assign(paste("V0_list", i, sep = ""), i)
  assign(paste("D0_list", i, sep = ""), i)
}

# Assign the Surface area to the correct list
S0_list1=goodList[goodList$Folder=="EB5_114",]$Surface.Area
S0_list2=goodList[goodList$Folder=="EB5_29",]$Surface.Area
S0_list3=goodList[goodList$Folder=="EB5_X11",]$Surface.Area
S0_list4=goodList[goodList$Folder=="MLy11XbioS",]$Surface.Area
S0_list5=goodList[goodList$Folder=="MLy11XdigS",]$Surface.Area
S0_list6=goodList[goodList$Folder=="StacksMly114",]$Surface.Area
S0_list7=goodList[goodList$Folder=="StacksMly114II",]$Surface.Area
S0_list8=goodList[goodList$Folder=="StacksMy129",]$Surface.Area

# Assign the Volume to the correct list
V0_list1=goodList[goodList$Folder=="EB5_114",]$Volume
V0_list2=goodList[goodList$Folder=="EB5_29",]$Volume
V0_list3=goodList[goodList$Folder=="EB5_X11",]$Volume
V0_list4=goodList[goodList$Folder=="MLy11XbioS",]$Volume
V0_list5=goodList[goodList$Folder=="MLy11XdigS",]$Volume
V0_list6=goodList[goodList$Folder=="StacksMly114",]$Volume
V0_list7=goodList[goodList$Folder=="StacksMly114II",]$Volume
V0_list8=goodList[goodList$Folder=="StacksMy129",]$Volume

# Assign the diameter to the correct list
D0_list1=goodList[goodList$Folder=="EB5_114",]$Diameter
D0_list2=goodList[goodList$Folder=="EB5_29",]$Diameter
D0_list3=goodList[goodList$Folder=="EB5_X11",]$Diameter
D0_list4=goodList[goodList$Folder=="MLy11XbioS",]$Diameter
D0_list5=goodList[goodList$Folder=="MLy11XdigS",]$Diameter
D0_list6=goodList[goodList$Folder=="StacksMly114",]$Diameter
D0_list7=goodList[goodList$Folder=="StacksMly114II",]$Diameter
D0_list8=goodList[goodList$Folder=="StacksMy129",]$Diameter

# Assign the nucleus volume to a list
V0_Nuc_list1=blobs1$Nucleus.Volume
V0_Nuc_list2=blobs2$Nucleus.Volume
V0_Nuc_list3=blobs3$Nucleus.Volume
V0_Nuc_list4=blobs4$Nucleus.Volume
V0_Nuc_list5=blobs5$Nucleus.Volume
V0_Nuc_list6=blobs6$Nucleus.Volume
V0_Nuc_list7=blobs7$Nucleus.Volume
V0_Nuc_list8=blobs8$Nucleus.Volume

# Assign the nucleus surface area to a list
S0_Nuc_list1=blobs1$Nucleus.Area
S0_Nuc_list2=blobs2$Nucleus.Area
S0_Nuc_list3=blobs3$Nucleus.Area
S0_Nuc_list4=blobs4$Nucleus.Area
S0_Nuc_list5=blobs5$Nucleus.Area
S0_Nuc_list6=blobs6$Nucleus.Area
S0_Nuc_list7=blobs7$Nucleus.Area
S0_Nuc_list8=blobs8$Nucleus.Area

# Convert the surface area lists to data frames
S0_df1=data.frame(matrix(unlist(S0_list1),nrow=length(S0_list1),byrow=TRUE))
S0_df2=data.frame(matrix(unlist(S0_list2),nrow=length(S0_list2),byrow=TRUE))
S0_df3=data.frame(matrix(unlist(S0_list3),nrow=length(S0_list3),byrow=TRUE))
S0_df4=data.frame(matrix(unlist(S0_list4),nrow=length(S0_list4),byrow=TRUE))
S0_df5=data.frame(matrix(unlist(S0_list5),nrow=length(S0_list5),byrow=TRUE))
S0_df6=data.frame(matrix(unlist(S0_list6),nrow=length(S0_list6),byrow=TRUE))
S0_df7=data.frame(matrix(unlist(S0_list7),nrow=length(S0_list7),byrow=TRUE))
S0_df8=data.frame(matrix(unlist(S0_list8),nrow=length(S0_list8),byrow=TRUE))

# Convert the volume lists to data frames
V0_df1=data.frame(matrix(unlist(V0_list1),nrow=length(V0_list1),byrow=TRUE))
V0_df2=data.frame(matrix(unlist(V0_list2),nrow=length(V0_list2),byrow=TRUE))
V0_df3=data.frame(matrix(unlist(V0_list3),nrow=length(V0_list3),byrow=TRUE))
V0_df4=data.frame(matrix(unlist(V0_list4),nrow=length(V0_list4),byrow=TRUE))
V0_df5=data.frame(matrix(unlist(V0_list5),nrow=length(V0_list5),byrow=TRUE))
V0_df6=data.frame(matrix(unlist(V0_list6),nrow=length(V0_list6),byrow=TRUE))
V0_df7=data.frame(matrix(unlist(V0_list7),nrow=length(V0_list7),byrow=TRUE))
V0_df8=data.frame(matrix(unlist(V0_list8),nrow=length(V0_list8),byrow=TRUE))

# Convert the diameter lists to data frames
D0_df1=data.frame(matrix(unlist(D0_list1),nrow=length(D0_list1),byrow=TRUE))
D0_df2=data.frame(matrix(unlist(D0_list2),nrow=length(D0_list2),byrow=TRUE))
D0_df3=data.frame(matrix(unlist(D0_list3),nrow=length(D0_list3),byrow=TRUE))
D0_df4=data.frame(matrix(unlist(D0_list4),nrow=length(D0_list4),byrow=TRUE))
D0_df5=data.frame(matrix(unlist(D0_list5),nrow=length(D0_list5),byrow=TRUE))
D0_df6=data.frame(matrix(unlist(D0_list6),nrow=length(D0_list6),byrow=TRUE))
D0_df7=data.frame(matrix(unlist(D0_list7),nrow=length(D0_list7),byrow=TRUE))
D0_df8=data.frame(matrix(unlist(D0_list8),nrow=length(D0_list8),byrow=TRUE))

# Convert the nucleus volume lists to data frames
V0_Nuc_df1=data.frame(matrix(unlist(V0_Nuc_list1),nrow=length(V0_Nuc_list1),byrow=TRUE))
V0_Nuc_df2=data.frame(matrix(unlist(V0_Nuc_list2),nrow=length(V0_Nuc_list2),byrow=TRUE))
V0_Nuc_df3=data.frame(matrix(unlist(V0_Nuc_list3),nrow=length(V0_Nuc_list3),byrow=TRUE))
V0_Nuc_df4=data.frame(matrix(unlist(V0_Nuc_list4),nrow=length(V0_Nuc_list4),byrow=TRUE))

62
byrow=TRUE))
V0_Nuc_df6=data.frame(matrix(unlist(V0_Nuc_list6),nrow=length(V0_Nuc_list6),
byrow=TRUE))
V0_Nuc_df7=data.frame(matrix(unlist(V0_Nuc_list7),nrow=length(V0_Nuc_list7),
byrow=TRUE))
V0_Nuc_df8=data.frame(matrix(unlist(V0_Nuc_list8),nrow=length(V0_Nuc_list8),
byrow=TRUE))

# Convert the nucleus surface area lists to data frames
S0_Nuc_df1=data.frame(matrix(unlist(S0_Nuc_list1),nrow=length(S0_Nuc_list1),
byrow=TRUE))
S0_Nuc_df2=data.frame(matrix(unlist(S0_Nuc_list2),nrow=length(S0_Nuc_list2),
byrow=TRUE))
S0_Nuc_df3=data.frame(matrix(unlist(S0_Nuc_list3),nrow=length(S0_Nuc_list3),
byrow=TRUE))
S0_Nuc_df4=data.frame(matrix(unlist(S0_Nuc_list4),nrow=length(S0_Nuc_list4),
byrow=TRUE))
S0_Nuc_df5=data.frame(matrix(unlist(S0_Nuc_list5),nrow=length(S0_Nuc_list5),
byrow=TRUE))
S0_Nuc_df6=data.frame(matrix(unlist(S0_Nuc_list6),nrow=length(S0_Nuc_list6),
byrow=TRUE))
S0_Nuc_df7=data.frame(matrix(unlist(S0_Nuc_list7),nrow=length(S0_Nuc_list7),
byrow=TRUE))
S0_Nuc_df8=data.frame(matrix(unlist(S0_Nuc_list8),nrow=length(S0_Nuc_list8),
byrow=TRUE))
# Set a constant value for p
p=1.6

# Set a large value for minimum distance
min_dist=1000000000

# Function for finding the a,b,c values
findab <- function(D0,S0,V0){
    F0=S0*D0/6/V0 # Calculate the filament index
    c=D0/2 # Calculate the c value

    if (c>0 & !is.na(c)) { # Make sure c is a valid number
        for (a in seq(from=0, to=c, by=0.2)) {
            for (b in seq(from=0, to=c, by=0.2)) {
                # Calculate the Surface and Volume
                S = 4*pi*((a*b)^p + (a*D0/2)^p + (b*D0/2)^p)/3)^(1/p)
                V = (2*pi/3)*a*b*D0/2

                # Calculate a "distance" between the given V0 and S0 and the
                # S and V we just calculated for this particular pair of a and b
                dist = sqrt((V-V0)^2 + (S-S0)^2)

                # If that distance is smaller than the smallest one found so far,
                # set the minimum distance to it and assign a, b, V, S are the
                # critical values found.
                if (dist < min_dist) {

                }
            }
        }
    }
}

a_crit = a
b_crit = b
V_crit = V
S_crit = S
min_dist = dist

# Make a data frame of a, b, c and FID
abc <- data.frame(a_c=a_crit, b_c=b_crit, c_c=c, FID_c=F0)
return(abc)

# Calculate the radius from Nucleus Volume
findr_NucVol <- function(V0){
    r_NucVol = (3*V0/(4*pi))^(1/3)
    return(r_NucVol)
}

# Calculate the radius from Nucleus Surface Area
findr_NucSA <- function(S0) {
    r_NucSA = (S0/(4*pi))^(1/2)
    return(r_NucSA)
}

# Create a new folder for the results
dir.create("/Users/hannah/Desktop/Masters/Thesis/Rscripts_from_Zach/CT_data/ABC_values_nuc_good_take2")
# Reset the working directory
setwd("/Users/hannah/Desktop/Masters/Thesis/Rscripts_from_Zach/CT_data/ABC_values_nuc_good_take2")

start_time <- Sys.time() # Record the time the program started

# For blobs 1 aka eb5_114
results_y = data.frame() # Make an empty data frame
results_z = data.frame() # Make an empty data frame
for (i in seq(from=1,to=nrow(D0_df1), by=1)) {
  x<-findab(D0_df1[i,],S0_df1[i,],V0_df1[i,]) #Find the abc values for chrom
  if (!is.null(x)){ # If the values were valid
    r_NucVol=findr_NucVol(V0_Nuc_df1[i,]) # Find the radius of nuc from vol
    r_NucSA=findr_NucSA(S0_Nuc_df1[i,]) # Find the radius of nuc from SA
    avgr_Nuc=(r_NucVol+r_NucSA)/2 # Average the two radius values
    Chromosome=blobs1[i,]$Chromosome # Find which chromosome it is
    # Add the results to the chrom data
    results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
    results_z<-rbind(results_z,results_y) # Bind all data together in a table
  }
}
write.table(results_z, file="eb5_114_abc.txt",sep = "\t",
            row.names = TRUE,col.names = NA) # Convert table into a txt file
# For blobs 2 aka eb5_29

results_y = data.frame() # Make an empty data frame
results_z = data.frame() # Make an empty data frame

for (i in seq(from=1,to=nrow(D0_df2), by=1)) {
    x<-findab(D0_df2[i,],S0_df2[i,],V0_df2[i,]) # Find the a, b, c values
    if (!is.null(x)){ # If the values were valid
        r_NucVol=findr_NucVol(V0_Nuc_df2[i,]) # Find the radius of nuc from vol
        r_NucSA=findr_NucSA(S0_Nuc_df2[i,]) # Find the radius of nuc from SA
        avgr_Nuc=(r_NucVol+r_NucSA)/2 # Average the two radius values
        Chromosome=blobs2[i,]$Chromosome # Find the chromosome number
        # Add the results to the chrom data
        results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
        results_z<-rbind(results_z,results_y) # Bind all data together in a table
    }
}

write.table(results_z, file="eb5_29_abc.txt",sep = "\t",
            row.names = TRUE,col.names = NA) # Convert table into a txt file

# For blobs 3 aka eb5_x11

results_y = data.frame() # Make an empty data frame
results_z = data.frame() # Make an empty data frame
for (i in seq(from=1,to=nrow(D0_df3), by=1)) {
    x<-findab(D0_df3[i,],S0_df3[i,],V0_df3[i,]) # Find the a, b, c values
    if (!is.null(x)){ # If the values were valid
        r_NucVol=findr_NucVol(V0_Nuc_df3[i,]) # Find the radius of nuc from vol
        r_NucSA=findr_NucSA(S0_Nuc_df3[i,]) # Find the radius of nuc from SA
        avgr_Nuc=(r_NucVol+r_NucSA)/2 # Average the two radius values
Chromosome=blobs3[i,]$Chromosome # Find the chromosome number
# Add the results to the chrom data
results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
results_z<-rbind(results_z,results_y) # Bind all data together in a table
}
}
write.table(results_z, file="eb5_x11_abc.txt",sep = "\t",
row.names = TRUE,col.names = NA) # Convert table into a txt file

# For blobs 4 aka mly11xbios
results_y = data.frame() # Make an empty data frame
results_z = data.frame() # Make an empty data frame
for (i in seq(from=1,to=nrow(D0_df4), by=1)) {
  x<-findab(D0_df4[i,],S0_df4[i,],V0_df4[i,]) # Find the a, b, c values
  if (!is.null(x)){ # If the values were valid
    r_NucVol=findr_NucVol(V0_Nuc_df4[i,]) # Find the radius of nuc from vol
    r_NucSA=findr_NucSA(S0_Nuc_df4[i,]) # Find the radius of nuc from SA
    avgr_Nuc=(r_NucVol+r_NucSA)/2 # Average the two radius values
    Chromosome=blobs4[i,]$Chromosome # Find the chromosome number
    # Add the results to the chrom data
    results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
    results_z<-rbind(results_z,results_y) # Bind all data together in a table
  }
}
write.table(results_z, file="mly11xbios_abc.txt",sep = "\t",
row.names = TRUE,col.names = NA) # Convert table into a txt file
# For blobs 5 aka mly11xdigs
results_y = data.frame()  # Make an empty data frame
results_z = data.frame()  # Make an empty data frame
for (i in seq(from=1,to=nrow(D0_df5), by=1)) {
  x<-findab(D0_df5[i,],S0_df5[i,],V0_df5[i,])  # Find the a, b, c values
  if (!is.null(x)){  # If the values were valid
    r_NucVol=findr_NucVol(V0_Nuc_df5[i,])  # Find the radius of nuc from vol
    r_NucSA=findr_NucSA(S0_Nuc_df5[i,])  # Find the radius of nuc from SA
    avgr_Nuc=(r_NucVol+r_NucSA)/2  # Average the two radius values
    Chromosome=blobs5[i,]$Chromosome  # Find the chromosome number
    # Add the results to the chrom data
    results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
    results_z<-rbind(results_z,results_y)  # Bind all data together in a table
  }
}
write.table(results_z, file="mly11xdigs_abc.txt",sep = "\t",  
           row.names = TRUE,col.names = NA)  # Convert table into a txt file

# For blobs 6 aka stacksmly114
results_y = data.frame()  # Make an empty data frame
results_z = data.frame()  # Make an empty data frame
for (i in seq(from=1,to=nrow(D0_df6), by=1)) {
  x<-findab(D0_df6[i,],S0_df6[i,],V0_df6[i,])  # Find the a, b, c values
  if (!is.null(x)){  # If the values were valid
    r_NucVol=findr_NucVol(V0_Nuc_df6[i,])  # Find the radius of nuc from vol
    r_NucSA=findr_NucSA(S0_Nuc_df6[i,])  # Find the radius of nuc from SA
    avgr_Nuc=(r_NucVol+r_NucSA)/2  # Average the two radius values
  }
Chromosome=blobs6[i,]$Chromosome # Find the chromosome number
# Add the results to the chrom data
results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
results_z<-rbind(results_z,results_y) # Bind all data together in a table
}
}
write.table(results_z, file="stacksmly114_abc.txt",sep = "\t",
           row.names = TRUE,col.names = NA) # Convert table into a txt file

# For blobs7 aka stacksmly114ii
results_y = data.frame() # Make an empty data frame
results_z = data.frame() # Make an empty data frame
for (i in seq(from=1,to=nrow(D0_df7), by=1)) {
  x<-findab(D0_df7[i,],S0_df7[i,],V0_df7[i,]) # Find the a, b, c values
  if (!is.null(x)) { # If the values were valid
    r_NucVol=findr_NucVol(V0_Nuc_df7[i,]) # Find the radius of nuc from vol
    r_NucSA=findr_NucSA(S0_Nuc_df7[i,]) # Find the radius of nuc from SA
    avgr_Nuc=(r_NucVol+r_NucSA)/2 # Average the two radius values
    Chromosome=blobs7[i,]$Chromosome # Find the chromosome number
    # Add the results to the chrom data
    results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
    results_z<-rbind(results_z,results_y) # Bind all data together in a table
  }
}
write.table(results_z, file="stacksmly114ii_abc.txt",sep = "\t",
           row.names = TRUE,col.names = NA) # Convert table into a txt file
# For blobs 8 aka stacksmy129

results_y = data.frame() # Make an empty data frame
results_z = data.frame() # Make an empty data frame

for (i in seq(from=1, to=nrow(D0_df8), by=1)) {
    x<-findab(D0_df8[i,],S0_df8[i,],V0_df8[i,]) # Find the a, b, c values
    if (!is.null(x)){ # If the values were valid
        r_NucVol=findr_NucVol(V0_Nuc_df8[i,]) # Find the radius of nuc from vol
        r_NucSA=findr_NucSA(S0_Nuc_df8[i,]) # Find the radius of nuc from SA
        avgr_Nuc=(r_NucVol+r_NucSA)/2 # Average the two radius values
        Chromosome=blobs8[i,]$Chromosome # Find the chromosome number
        # Add the results to the chrom data
        results_y<-cbind(x,r_NucVol,r_NucSA,avgr_Nuc,Chromosome)
        results_z<-rbind(results_z,results_y) # Bind all data together in a table
    }
}

write.table(results_z, file="stacksmy129_abc.txt", sep = "\t",
            row.names = TRUE, col.names = NA) # Convert table into a txt file

end_time <- Sys.time() # Record the time the program finished
run_time=end_time-start_time # Calculate the time it took to run
print(run_time) # Print the run time
APPENDIX C
IMAGED DATA PLOTS

C.1. Plots of Imaged Data
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued

74
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
ES Chrom 14 Radial Distance

ES Chrom 11 Radial Distance

ES Chrom 2 Radial Distance

ES Chrom 14 Filament Index

ES Chrom 11 Filament Index

ES Chrom 2 Filament Index

ES Chrom 14 Elongation

ES Chrom 11 Elongation

ES Chrom 2 Elongation

ES Chrom 14 Rugosity

ES Chrom 11 Rugosity

ES Chrom 2 Rugosity

Figure C.1. Plots of Imaged Data Continued

\[ f(x) = 0.34643/(1+\exp(95.04131(\log(x)/(0.37045)))) \]

\[ f(x) = 0.34813+(0.26459)\exp(-x/0.34813) \]

\[ f(x) = 0.26459+(0.48911)\exp(-x/0.48911) \]

\[ f(x) = 0.00255x+0.41231 \]

\[ f(x) = 0.02312x+0.42302 \]

\[ p=0.2985877 \]

\[ p=0.24968926 \]

\[ p=0.29660517 \]
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
Anova, $p = 2 \times 10^{-7}$

Volume vs Chromosomes

Volume vs ES Chromosomes

Anova, $p = 0.26$

Volume vs L Chromosomes

Anova, $p = 8.8 \times 10^{-8}$

Volume vs Cell Type

Wilcoxon, $p = 9.3 \times 10^{-5}$

Figure C.1. Plots of Imaged Data Continued
Figure C.1. Plots of Imaged Data Continued
APPENDIX D
SIMULATED DATA PLOTS

D. 1. Plots of Imaged and Simulated Data
Figure D.1. Plots of Imaged and Simulated Data Continued
Figure D.1. Plots of Imaged and Simulated Data Continued
Figure D.1. Plots of Imaged and Simulated Data Continued
Figure D.1. Plots of Imaged and Simulated Data Continued
Figure D.1. Plots of Imaged and Simulated Data Continued
APPENDIX E
LINEAR SIMULATED DATA PLOTS

E.1. Linear Plots of Imaged and Simulated Data
Figure E.1. Linear Plots of Imaged and Simulated Data Continued
Figure E.1. Linear Plots of Imaged and Simulated Data Continued

105
Figure E.1. Linear Plots of Imaged and Simulated Data Continued
Figure E.1. Linear Plots of Imaged and Simulated Data Continued
Figure E.1. Linear Plots of Imaged and Simulated Data Continued
Hannah Varney was born in Lewiston, Maine on February 16th, 1998. She was raised in Turner, Maine and graduated from Leavitt Area High School in 2016. She graduated from the University of Maine in 2020 with a Bachelor of Science degree in Biomedical Engineering and a minor in Chemistry. Hannah participated in research in the Department in Chemistry that lead to a publication titled “Synthesis and characterization of (RPh3P)3[Bi3I12] (R=Me, Ph) iodobismuthate complexes for photocatalytic degradation of organic pollutants” in Springer Nature, October 2019. She is also a member of National Society of Leadership and Success and Tau Beta Pi. In her free time Hannah enjoys reading, canoeing, fishing, hunting, camping and hiking. After graduating Hannah will attend New York Institute of Technology College of Medicine - Arkansas to pursue her D.O. degree. Hannah Varney is a candidate for the Master of Science degree in Biomedical Engineering from the University of Maine in August 2021.