3D VISUALIZATION OF GENETIC MUTATIONS IN PANCREATIC INTRAEPITHELIAL NEOPLASIA

by
Ting I Wang

A thesis submitted to Johns Hopkins University
in conformity with the requirements for
the degree of Master of Arts

Baltimore, Maryland
March, 2022

© 2022 Ting I Wang
All Rights Reserved
Pancreatic ductal adenocarcinoma (PDAC) is one of the deadliest forms of cancer in the United States and is often diagnosed in advanced stages with poor prognosis. A new workflow called CODA that uses machine learning to reconstruct pancreas pathology, from precursor lesions to PDAC, has been established to study PDAC in humans in three-dimensions. Although the genetic mutations that drive PDAC are known, there exists little information regarding 3D spatial distribution of these mutations. Once defined in 3D, these mutations would need to be visualized in a clear and organized way.

The application of genetic sequencing to 3D-constructed precursor lesions in the human pancreas afforded a novel opportunity to develop tools to visualize complex genetic changes in three dimensions. Each lesion was subdivided for deeper resolution of lesion heterogeneity. The visualization developed took a 3D scatter plot approach. Genetic mutations were represented by mapped objects spaced equally throughout the precursor lesions. Each genetic mutation was assigned a color. Object size was used to represent prevalence of each genetic mutation in 4 distinct 3D precursor lesions in each gene sequencing region. Four visualization outputs were created, including still images, turntable videos, an interactive platform, and a promotional image. The interactive platform includes a 3D interactive model that a user can rotate and scale,aggable genetic mutation representations, and a switch between “prevalence” and “no prevalence” modes. Modeling was done using 4D® and ZBrush®. Unity was used for lighting, materials, and creation of the 3D interactive platform.

This thesis project experimented with ways in which data commonly visualized in a 2D manner could be visualized in a 3D space. The visualization represents a first step in understanding tumorigenesis in three dimensions and its contributing factors as related to tumor microenvironments in human.

Ting I Wang
CHAIRPERSONS OF THE SUPERVISORY COMMITTEE

Ralph Hruban, M.D., preceptor

Director of the Sol Goldman Pancreatic Cancer Research Center
Director, Department of Pathology
Professor of Pathology
Professor of Oncology
Johns Hopkins University School of Medicine

Denis Wirtz, Ph.D., preceptor

Vice Provost for Research
Theophilus Halley Smoot Professor of Engineering Science
Professor of Chemical and Biomolecular Engineering
Professor of Pathology
Professor of Oncology
Johns Hopkins University Whiting School of Engineering

Jennifer Elizabeth Fairman, M.A., faculty advisor

Associate Professor of Art as Applied to Medicine
Johns Hopkins University School of Medicine
ACKNOWLEDGMENTS

I would like to extend my utmost gratitude for the great many who had guided and aided me throughout the undertaking of this study:

**Jennifer Fairman**, my advisor, for your guidance, support, encouragement, and feedback throughout the entire process. You treaded unknown territories with me in discovering the different possibilities and yet was able to guide me down the path with best solutions.

**Dr. Ralph Hruban** and **Dr. Denis Wirtz**, my preceptors, for this unique opportunity to visualize cutting-edge research. I was greatly inspired to create a visualization that was just as exciting. You kept an open mind for me to experiment with visual different solutions and your knowledge on the subject proved invaluable in shaping the final outcomes.

**Ashley Kiemen** and **Alicia Braxton**, who provided the PanIN models and mutational data for the study. Your input made the process smooth-sailing and your feedback most valuable.

**Dr. Laura Wood**, **Dr. Pei-Hsun Wu**, and other research team members in the engineering and pathology collaboration group for welcoming me to their monthly meeting and for giving me a more well-rounded understanding of the project. **Ellen Winslow** and **Michelle Crum** for your responsive communications during scheduling.

**Patrick McCarthy**, a unity developer, for your expertise and generosity that made the interactive platform possible.

**The Vesalius Trust** for helping fund this study.

**Corinne Sandone, David Rini, Tim Phelps, Lydia Gregg, Jeff Day, Juan Garcia, Anne Altemus, Dacia Balch, Norman Barker, Sara Poynton, Ian Suk, Karen Klinedinst, Dan Hermansen**, and **Lauren Rakes** for your wisdom and support.

**Kelly Lim, Gaby Rivera Del Rio, Shirley Li, Jennifer Wang**, and **Jason Brady** for your amazing talents that kept me inspired and for your companionship.

My **family** and **friends** for your love and support.
# Table of Contents

Abstract ......................................................................................................................................... ii

Chairpersons of the Supervisory Committee .......................................................................... iii

Acknowledgments ....................................................................................................................... iv

Table of Contents .......................................................................................................................... v

List of Tables .............................................................................................................................. viii

Index of Figures ............................................................................................................................ ix

Introduction ................................................................................................................................... 1

Overview ....................................................................................................................................... 1

Genetic Profile of PDAC and PanINs ............................................................................................ 1

CODA ....................................................................................................................................... 2

DNA Sequencing and Genetic Mutation Identification ................................................................. 3

3D Data Visualizations .................................................................................................................. 4

Interactivity in Data Visualization ................................................................................................. 5

Thesis Objectives .......................................................................................................................... 5

Audience ....................................................................................................................................... 6

Materials and Methods ............................................................................................................... 7

Raw PanIN Volume Data and Other Related Structures ................................................................. 7

Software ....................................................................................................................................... 11

Raw Volume Data Reconstruction in Cinema 4D® ................................................................. 11

Model Optimization in ZBrush® ................................................................................................. 14

Data Mapping in ZBrush® ............................................................................................................ 17

Object Size to Represent Prevalence Information ......................................................................... 19

Model Organization ...................................................................................................................... 20

UV mapping in ZBrush® .................................................................................................................. 20

Exporting Models from ZBrush® ................................................................................................... 22
LIST OF TABLES

Table 1. Genetic mutations and their prevalence in each PanIN and region; \( P^* = \) prevalence.... 10

Table 2. Relevant structures in “TC 92” and their coordinates for point 1 and point 2 .......... 12

Table 3. Length, width, and height of structures in “TC 92” .................................................. 13

Table 4. Exported models for “TC 92” .................................................................................. 22

Table 5. C# scripts for Unity; *attached to the main camera .................................................. 28

Table 6. Color guide for genetic mutations and other structures .......................................... 31
INDEX OF FIGURES

Figure 1A. Raw volume data of TC 92 showing 4 distinct PanINs labeled A (orange), B (yellow), C (blue), D (pink) .......................................................... 8

Figure 1B. Raw volume data of TC 92 showing normal pancreatic ducts (grey), cancer region (red), other small PanIN (green) ......................................................... 8

Figure 2. H&E-stained cross-sectional slide for TC 92 ...................................................................... 9

Figure 3. PanINs divided into 5 equal regions in the z plane of the tissue block for gene sequencing: Region 1 (orange), 2 (light green), 3 (teal), 4 (blue), 5 (purple) .............................................. 9

Figure 4. Diagram showing point 1 (solid white) and point 2 (solid black with white stroke) on the body-diagonal (blue dotted line) of the cuboidal bounding box ........................................... 12

Figure 5. Reconstruction of all structures in “TC 92” sitting on top of the H&E slide and showing accurate relative size and spatial relationships with each other ........................................ 13

Figure 6A. Pre-processed surface details of models showing staircasing artifacts .................. 15

Figure 6B. Smoothed surface details of models after application of “polish by features” ........ 15

Figure 6C. Polyframe of pre-decimated and pre-optimized surface details showing densely packed polygons with high active point count .......................................................... 16

Figure 6D. Polyframe of surface details after decimation, optimization, and projection showing more regularly packed polygons and a lower active point count ................................ 16

Figure 7A. Surface showing before polygons have been split into 3 different polygroups .......... 18

Figure 7B. Surface showing after polygons have been split into 3 different polygroups .......... 18

Figure 7C. Surface showing after objects have been mapped to the polygons using NanoMesh .................................................................................................................................... 19

Figure 8A. UV map for mapped objects representing KRAS G12V (no prevalence) ............. 21

Figure 8B. Texture map for mapped objects representing KRAS G12V (no prevalence) .......... 21

Figure 9A. Colors representing different genetic mutations as seen with normal vision .......... 24

Figure 9B. Colors representing different genetic mutations as seen with protanopia-type color blindness ........................................................................................................ 24

Figure 9C. Colors representing different genetic mutations as seen with deuteranopia-type color blindness .......................................................................................................... 25

Figure 10. Material settings showing how texture maps are added to various channels ........... 25

Figure 11A. 3D visualization with texture maps, showing more three-dimensionality .......... 26
Figure 32. Mock journal cover with the 3D visualization.
INTRODUCTION

Overview

Pancreatic ductal adenocarcinoma (PDAC) is one of the deadliest forms of cancers in the United States (Leinonen et al., 2017). It accounts for more than 90% of malignancies of the pancreas, and with a five-year overall survival rate of only 10%, is the third leading cause of cancer-related death in United States (Kleeff et al., 2016, Siegel et al., 2018). Most patients with PDAC are diagnosed at advanced disease stages due to late onset of symptoms and effective early disease detection (Yonezawa et al., 2008, Cid-Arregui and Juarez, 2015). These patients often have poor prognosis because at the time of diagnosis, more than 80% of patients present with locally advanced, non-surgically resectable PDAC or distant metastases (Orth et al., 2019). Even primary tumors less than 2cm can metastasize (Yonezawa et al., 2008). A better understanding of tumor progression is critical for early diagnosis and improvement of patient outcomes.

Genetic Profile of PDAC and PanINs

PDAC arises from precursor lesions in the pancreatic ducts (Hruban et al., 2019). One of the most well-defined precursor lesions is pancreatic intraepithelial neoplasia (PanIN), small flat or papillary lesions that can either be classified as low or high-grade depending on their degree of dysplasia (Hruban et al., 2019, Lee et al., 2021). Though microscopic and clinically undetectable, PanINs can give rise to the most invasive pancreatic cancers, making them a desirable target for early cancer detection (Yonezawa et al., 2008).

On a molecular level, PanINs and other precursor lesions accumulate genetic mutations that eventually transform them into invasive cancer (Lee et al., 2021). Two genes that strongly associate with pancreatic cancers are KRAS and TP53, with KRAS being present in more than 90% of PDAC and TP53 in over 75% of PDAC (Lee et al., 2021). KRAS encodes a GTPase that plays a crucial role in proliferation, survival, and differentiation (Lee et al., 2021). TP53 encodes a protein that mediates cellular stress response by inhibiting cell growth and promoting cell death (Lee et al., 2021). Though mutations in KRAS drive PanIN development and can be seen
early in low-grade PanINs, mutations in TP53 are only observed in high-grade PanINs and PDAC (Lee et al., 2021, Opitz et al., 2021). A combination of mutations in KRAS and in TP53, together with other somatic mutations, drives PDAC development (Lee et al., 2021, Opitz et al., 2021).

Still, the question remains how to better profile the genetic heterogeneity of PanINs to aid with early detection of PDAC. Studies have shown that different mutations of the same driver gene can occur in the same precursor lesion (Kuboki et al., 2019). However, the small size of PanINs makes it difficult to perform genetic profiling on different regions of the same PanIN. Being able to more precisely profile the genetic heterogeneity of PanINs can help clarify how a PanIN lesion evolves, and answer questions if distinct PanINs can develop the same genetic lineage, or do distinct PanINs have different genetic mutations? How certain mutations change or drive other genetic mutations over the course of PanIN development and into onset of PDAC, and how these genetic mutations modify the surrounding TME to promote tumorigenesis remain to be elucidated.

Other considerations when profiling genetic heterogeneity of PanINs include the morphology. PanINs can take on the same complex branching pattern as the pancreatic ducts they develop in. PanINs can be highly branched or architecturally simple; spatially distinct PanINs can also develop less than centimeters away from each other (Kiemen et al., 2020). Studying PanINs in two-dimensional (2D) using histopathological slides though allows for studies at single-cell resolution. However, it is impossible to tell if two seemingly spatially distinct PanIN lesions in a single slide may be connected in three-dimensional (3D) space.

Studies in three dimensions are needed to accurately define the true heterogeneity of the genetic changes in PanIN lesions.

**CODA**

In recent years, 3D in vitro and in vivo techniques have been explored to better understand the complex tissue architecture in the context of pancreatic cancers and TME. Specifically, the Wirtz and Hruban labs at the Johns Hopkins University employed a novel workflow named
CODA that reconstructs cm-scale 3D human pancreatic tissues containing cancer precursors at single-cell resolution (Kiemen et al., 2020). The tissue block is serially sectioned and every third slide is stained with hematoxylin-and-eosin (H&E). Intervening slides can be saved for genetic and immunolabeling studies (Kiemen et al., 2020). The serial H&E slides are digitized at high resolution (Kiemen et al., 2020). Deep learning is used to identify up to 10 distinct tissues types in the slides including normal ductal epithelium, pancreatic precancers, pancreatic cancer, islets of Langerhans, vessels, nerves, acini, stroma, lymph nodes, and fat (Kiemen et al., 2020). Images of labelled cells are then registered to create cm-scale digital volumes of different tissues (Kiemen et al., 2020). This workflow allows for analysis of composition of tissue subtypes and number of cells in each tissue subtype (Kiemen et al., 2020).

**DNA Sequencing and Genetic Mutation Identification**

Because intervening slides are available for somatic genetic analyses, CODA can be used to visualize the genetic heterogeneity of PanIN lesions in 3D. Cells labelled as PanINs by deep learning semantic were cut from the intervening slides using laser capture microscopy (LCM) and collected for DNA sequencing according to the PanIN and region they are located in. DNA sequencing was performed for each distinct PanIN. Since multiple slides need to be harvested to obtain enough DNA for somatic genetic analysis, slides in the z-dimension were grouped in into 5 equal regions. In doing so, each distinct 3D PanIN lesion analyzed in a maximum of five regions.

The Wirtz and Hruban labs identified multiple genetic mutations across 4 genes, including *KRAS*, *GNAS*, *RET*, and *TP53*. *RET* encodes a receptor tyrosine kinase that plays a role in growth factor signaling (Amit et al., 2019). It is a proto-oncogene that has been implicated in tumorigenesis of several other types of tumors and could be a co-driver for PDAC (Amit et al., 2019). *GNAS*, on the other hand, encodes the stimulatory α-subunit (Gsα) of heterotrimeric G-proteins (Hosoda et al., 2015). Mutations in *GNAS*, however, are more implicated in intraductal papillary mucinous neoplasms (IPMNs), another precursor lesions to PDAC and other types of pancreatic cancers, than PDAC (Kuboki et al., 2019). Like *KRAS*, *GNAS* is likely amongst the earliest set of mutated genes during development of precursor lesions (Kuboki et al., 2019). Since not all PanINs progress to invasive cancer, or that IPMNs
still make up a significant minority of precursor lesions PDAC, it is not surprising that \textit{GNAS} mutations are found in PanINs, whether the PanINs become cancerous or take on a similar developmental trajectory as IPMNs or not (Hruban et al., 2019, Kuboki et al., 2019).

Once all the data are collected, the major challenge is how to visualize the complex spatial distribution of these mutations simultaneously in reconstructed PanINs in a clear and organized way.

\textbf{3D Data Visualizations}

Many previous visualizations of genomic data focus on analyzing and comparing sequence data (Nusrat et al., 2019). Most of these approaches display the sequential data either in a 2D linear or circular fashion, with either a parallel or serial layout (Nusrat et al., 2019). There exists little information regarding how gene mutation data can be visualized in 3D models with complex topology while preserving the 3D topology of the models. Djekidel et al. experimented with mapping gene sequence data directly onto 3D models, but Djekidel’s visualization focused more on the mapping aspect of data and relied more on interactivity to convey the three-dimensionality (Djekidel et al., 2016).

In astrophysics, Courtois et al. took a 3D scatter plot approach to map galaxies in a 3D universe, though using only points without an explicit surface may prove disorienting in depicting the complex topology of the PanIN models (Courtois et al., 2013). Many effective 3D data visualizations combine a few different techniques in order to convey information (Ward et al., 2010). Some approaches include mapping symbols or glyphs that convey information such as magnitude or direction directly onto 3D surfaces, or overlaying colors or contour lines on rubber sheets, which is a data visualization technique that expresses two-dimensional field of values as a height field (Ward et al., 2010).

Since the 3D-ness of the PanIN is an important part of the visualization, the goal of my thesis project was to maximize 3D modeling softwares such as Maxon Cinema 4D to bring out 3D surface details.
**Interactivity in Data Visualization**

While simple static visualizations typically depict relationships between 2 variables, an interactive visualization allows for multiple related facets to be shown simultaneously (Chishtie et al., 2022). These facets as applied to health and science research include spatial, change over time, flow, distribution, magnitude, and correlation (Chishtie et al., 2022). Since an interactive visualization allows for multiple functional aspects of data, it can more easily accommodate for future expansion should more variables need to be incorporated into existing visualization.

Some of the common interaction techniques include navigation, where user can control and alter the camera of view to allow for zooming and rotating, and filtering, where user can specify the amount of data that is displayed (Ward et al., 2010). Allowing users to filter and modify data presented facilitates exploratory analysis. Allowing users to rotate the models enhances the user’s 3D appreciation of the PanIN models compared to static images.

**Thesis Objectives**

The main objectives of this study are to: 1) develop visualizations of the distribution of somatic gene mutations on reconstructed PanIN models, 2) establish a standard for scientists to communicate and report additional data such a characters of the tumor microenvironment, 3) create 4 types of visualizations that can be easily incorporated into presentations and journal publications, or shared over social media, 4) use best practices for user accessibility and usability to promote and engage researchers.

As mentioned above, this project aims to create 4 media outputs. First are static images or figures that can be published in a journal article. Second are video turntables that can easily be shared in presentations or be used as supplemental materials in journal publications. Third is an interactive web application that encourages users to manipulate the 3D model for a more robust and nuanced understanding of tumorigenesis and its contributing factors as related to the tumor microenvironments. Fourth is a promotional image that can either be used as a journal cover or be shared over social media to elicit interest.
Audience

The primary audience includes cancer researchers who are studying pancreatic cancer and looking for novel ways to report data. A secondary audience includes graduate students and trainees who are looking to learn more about tumorigenesis, as well as doctors and pathologists looking for ways to better visualize precancers that are hard to detect in their earlier stages.
MATERIALS AND METHODS

Raw PanIN Volume Data and Other Related Structures

Human pancreatic tissue blocks were harvested in the Pathology Department at the Johns Hopkins Hospital. The tissue blocks were serially sectioned, and every third slide stained and scanned at high resolution. Deep learning was used to label up to 10 distinct tissue types in each slide. The tissue types include normal ductal epithelium, pancreatic precancers, pancreatic cancer, islets of Langerhans, vessels, nerves, acini, stroma, lymph nodes, and fat. Images of labeled tissues were registered to create 3D digital volumes of different structures. Samples with PanINs were selected for DNA sequencing of genetic mutations, and subsequent visualization. Raw volume data were exported from MATLAB as “Standard Triangle Language (STL)” files and were provided by the Wirtz lab; sequencing data and prevalence data of each somatic genetic mutation were provided by the Wood lab.

Sample “TC 92” contains four distinct PanINs (Fig. 1A). Accompanying structures include normal pancreatic ducts, other PanINs that did not contain enough cells to be sequenced, known cancer cells (Fig. 1B). A cross-sectional H&E slide was also included for orientation purposes (Fig. 2). For gene sequencing, each of the PanINs was divided into one of 5 equal regions equal regions in the z plane of the pancreatic tissue block (Fig 3). The number of regions was determined by the Wood lab according to the minimum number of cells required for gene sequencing. Region 1 of PanIN B was excluded in the data mapping and added to the small PanIN file since not enough DNA was harvested for gene sequencing. Refer to Table 1 for gene sequencing and prevalence data for each of the genetic mutations in each region.
Figure 1A. Raw volume data of TC 92 showing 4 distinct PanINs labeled A (orange), B (yellow), C (blue), D (pink).

Figure 1B. Raw volume data of TC 92 showing normal pancreatic ducts (grey), cancer region (red), other small PanIN (green).
Figure 2. H&E-stained cross-sectional slide for TC 92.

Figure 3. PanINs divided into 5 equal regions in the z plane of the tissue block for gene sequencing: Region 1 (orange), 2 (light green), 3 (teal), 4 (blue), 5 (purple).
Table 1. Genetic mutations and their prevalence in each PanIN and region; $P^*$ = prevalence.

<table>
<thead>
<tr>
<th>PanIN</th>
<th>Mutation</th>
<th>P*</th>
<th>Mutation</th>
<th>P*</th>
<th>Mutation</th>
<th>P*</th>
<th>Mutation</th>
<th>P*</th>
<th>Mutation</th>
<th>P*</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KRAS G12D C&gt;T</td>
<td>19%</td>
<td>KRAS G12D C&gt;T</td>
<td>13%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KRAS G12R C&gt;G</td>
<td>13%</td>
<td>KRAS G12R C&gt;G</td>
<td>4%</td>
<td>KRAS Q61H T&gt;G</td>
<td>8%</td>
<td>KRAS G12V C&gt;A</td>
<td>7%</td>
</tr>
<tr>
<td></td>
<td>KRAS G12R C&gt;G</td>
<td>7%</td>
<td>KRAS Q61H T&gt;A</td>
<td>17%</td>
<td>RAS Q61H T&gt;G</td>
<td>5%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>KRAS G12D C&gt;T</td>
<td>13%</td>
<td>KRAS G12D C&gt;T</td>
<td>4%</td>
<td>KRAS G12D C&gt;T</td>
<td>16%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>KRAS G12V C&gt;A</td>
<td>14%</td>
<td>GNAS R844H G&gt;A</td>
<td>4%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GNAS R844H G&gt;A</td>
<td>22%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>KRAS G12V C&gt;A</td>
<td>11%</td>
<td>KRAS G12V C&gt;A</td>
<td>32%</td>
<td>KRAS G12D C&gt;T</td>
<td>5%</td>
<td>KRAS G12D C&gt;T</td>
<td>8%</td>
<td>KRAS G12R C&gt;G</td>
<td>30%</td>
</tr>
<tr>
<td></td>
<td>KRAS G12R C&gt;G</td>
<td>5%</td>
<td>RET R886W C&gt;T</td>
<td>28%</td>
<td>KRAS G12V C&gt;A</td>
<td>14%</td>
<td>KRAS G12V C&gt;A</td>
<td>11%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RET R886W C&gt;T</td>
<td>17%</td>
<td>TP53 R248W G&gt;A</td>
<td>9%</td>
<td>RET R886W C&gt;T</td>
<td>12%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Software

Spatial reconstruction of all structures was done using Maxon Cinema 4D®. Optimization of models, genetic mutation data visualization, UV mapping were done in Pixologic ZBrush®. Lighting of models was done using Cinema 4D®, Redshift® (part of Cinema 4D® package), and Unity. Models were imported into Unity for texture mapping.

Still images for publication use and turntables were exported from Unity. Image editing and retouching were done in Adobe Photoshop®. Turntables were captured in Unity, edited in Adobe After Effects®, and converted to mp4 files using Adobe Media Encoder®. Unity was also used to create the interactive web app. Interface assets were created in Adobe Illustrator®.

Raw Volume Data Reconstruction in Cinema 4D®

Raw volume data were imported into Cinema 4D®. To ensure that the relative size of the exported structures and their spatial relationships with each other accurately reflect the volume data in MATLAB, coordinates of point 1 and point 2 on the body-diagonal of the bounding box were calculated in MATLAB for each of the structures and manually entered (Fig. 4). Though the tissue block was sliced along the z plane of the pancreatic tissue block during slide preparation, the visualization was displayed with the z plane of the tissue block being the “height” of the visualization so orienting H&E slide sits flat at the bottom of the visualization (Fig. 5). In MATLAB, the “height” information was contained in the z-coordinates. However, in modeling software such as Cinema 4D®, coordinates were reinterpreted so y-coordinates display the “height” information. Refer to Table 2 for the reinterpreted coordinates. Difference between the reinterpreted x, y, z coordinates of the two points reflect the width, height, and length of the structure respectively (Table 3). Axis of the model was set to line up with point 1. Coordinates of point 1 determine the position of the structure in the reconstruction. Of note, since the STL file for the normal ducts was too large and too complex, containing 5.845 million active points, to be imported into Cinema 4D®, it was first brought into ZBrush® to be decimated and optimized. Ten percent of decimation was performed on the normal ducts with a final active point count of 584,523. After the reconstruction, each of the structures was exported as new STL files.
Figure 4. Diagram showing point 1 (solid white) and point 2 (solid black with white stroke) on the body-diagonal (blue dotted line) of the cuboidal bounding box.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Point 1 x-coord.</th>
<th>Point 2 y-coord.</th>
<th>Point 1 y-coord.</th>
<th>Point 2 y-coord.</th>
<th>Point 1 z-coord.</th>
<th>Point 2 z-coord.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PanIN A</td>
<td>144.36</td>
<td>18</td>
<td>80.16</td>
<td>172.2</td>
<td>31.32</td>
<td>96.96</td>
</tr>
<tr>
<td>PanIN B</td>
<td>44.76</td>
<td>0</td>
<td>72.6</td>
<td>106.8</td>
<td>39.84</td>
<td>163.8</td>
</tr>
<tr>
<td>PanIN C</td>
<td>50.52</td>
<td>0</td>
<td>143.64</td>
<td>118.44</td>
<td>24.12</td>
<td>221.28</td>
</tr>
<tr>
<td>PanIN D</td>
<td>60.84</td>
<td>0</td>
<td>186.48</td>
<td>222.72</td>
<td>39.84</td>
<td>299.88</td>
</tr>
<tr>
<td>Normal Duct</td>
<td>34.92</td>
<td>0</td>
<td>44.4</td>
<td>240.48</td>
<td>39.84</td>
<td>305.76</td>
</tr>
<tr>
<td>Small PanIN</td>
<td>158.28</td>
<td>15.84</td>
<td>157.56</td>
<td>225.36</td>
<td>39</td>
<td>200.88</td>
</tr>
<tr>
<td>Cancer</td>
<td>108.96</td>
<td>18.48</td>
<td>174.48</td>
<td>229.08</td>
<td>39.84</td>
<td>305.28</td>
</tr>
<tr>
<td>H&amp;E Slide</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>348.12</td>
<td>0</td>
<td>284.16</td>
</tr>
</tbody>
</table>

Table 2. Relevant structures in “TC 92” and their coordinates for point 1 and point 2.
<table>
<thead>
<tr>
<th>Structure</th>
<th>Width (Δx)</th>
<th>Height (Δy)</th>
<th>Length (Δz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PanIN A</td>
<td>27.84</td>
<td>13.32</td>
<td>16.8</td>
</tr>
<tr>
<td>PanIN B</td>
<td>62.04</td>
<td>39.84</td>
<td>91.2</td>
</tr>
<tr>
<td>PanIN C</td>
<td>67.92</td>
<td>24.12</td>
<td>77.64</td>
</tr>
<tr>
<td>PanIN D</td>
<td>161.88</td>
<td>39.84</td>
<td>113.4</td>
</tr>
<tr>
<td>Normal Duct</td>
<td>205.56</td>
<td>39.84</td>
<td>261.36</td>
</tr>
<tr>
<td>Small PanIN</td>
<td>67.08</td>
<td>23.16</td>
<td>43.32</td>
</tr>
<tr>
<td>Cancer</td>
<td>120.12</td>
<td>21.36</td>
<td>130.8</td>
</tr>
<tr>
<td>H&amp;E Slide</td>
<td>348.12</td>
<td>0</td>
<td>284.16</td>
</tr>
</tbody>
</table>

Table 3. Length, width, and height of structures in “TC 92”.

Figure 5. Reconstruction of all structures in “TC 92” sitting on top of the H&E slide and showing accurate relative size and spatial relationships with each other.
**Model Optimization in ZBrush®**

Since the PanINs were divided into one of 5 equal regions to for gene sequencing, five cubes spaced equidistance apart along the y-axis were modeled in Cinema 4D® and exported along with other structures. Reconstructed STL files and cubes were imported into ZBrush®. Boolean operations were done to divide the models into different sections for data mapping.

A combinational approach of glyph or object mapping and an external surface was pursued for the data mapping. Objects representing gene mutations data were mapped onto the surface of the PanINs using the NanoMesh feature of ZBrush®. The objects collectively would create a 3D scatter plot effect. Before the objects could be mapped onto the surfaces of PanINs, the raw volume data needed to be optimized. All sections of the PanINs were processed in the same way to ensure consistency in polygon size, which could affect the base size of the object mapped onto the polygons. Models were optimized for object mapping purposes only. Models with only smoothing applied to the surfaces were used for final visualization to retain as many surface details of the models as possible. Remaining structures including normal ducts, small PanINs, and cancer only have smoothing applied to the surfaces as well. The steps for optimization were outlined below. Figure 6 showed the surfaces before and after the optimization.

1. **Polish by features, 20**
   - to smooth surface staircasing artifacts resulted from stacking slides together

2. **DynaMesh, blur 2, resolution 128**
   - to prepare the models by closing potential holes in the models and to resolve overlapping geometry

3. **Two rounds of decimation, both at 50% of decimation**
   - to decrease the overall number of active points
   - to increase the surface area of the polygons to avoid crowding of mapped objects

4. **DynaMesh, blur 2, resolution 128**
   - decimation by Decimation Master creates topology with mostly triangles; a second round of DynaMesh turns the triangles into quadrangles that are easier
to map object onto

5. Project, distance 0.005, mean 25, pa blur 10
   • since both decimation and DynaMesh cause smooth of the surface details,
     the project function projected surface details of the original models back onto
     optimized models

*Figure 6A. Pre-processed surface details of models showing staircasing artifacts.*

*Figure 6B. Smoothed surface details of models after application of “polish by features”.*
Figure 6C. Polyframe of pre-decimated and pre-optimized surface details showing densely packed polygons with high active point count.

Figure 6D. Polyframe of surface details after decimation, optimization, and projection showing more regularly packed polygons and a lower active point count.
Data Mapping in ZBrush®

Next, data points would be mapped onto the volume surface using NanoMesh. Before objects could be mapped, the polygons in each of the regions were split into 3 polygroups because in each of the regions, up to 3 genetic mutations could present at the same location (individual cells can have multiple mutations). To ensure the objects would be distributed evenly and have the same density across all regions, all regions would have their polygons split into 3 polygroups regardless the number of genetic mutations were present. Splitting the polygons into 3 polygroups also limited the mapping of the objects to the polygroups and minimized overlapping of objects. Next, NanoMesh were applied on each of the polygroups according to the genetic mutation distribution data. If only one genetic mutation was found in a region, only one polygroup would be mapped. Each polygroup contained one instance of NanoMesh. The steps were outlined below. Refer to Figure 7 for the splitting of polygons and mapping of objects.

1. Polygroups > GroupVisible, coverage 0.3333, cluster 0. GroupVisible functionality randomly assigns polygons into 3 different polygroups with the first polygroup have an “n” number of polygons, the second polygroup having a “2n” number of polygons, and the third “3n” number of polygons.

2. Groups split: divide the polygroups into individual subtools.

3. Check total number of polygons in each subtool, merge the 2 subtools with “n” and “3n” number of polygons; merge down to combine the 2 subtools.

4. Polygroups > GroupVisible, coverage 1, cluster 0. Merging of subtools retains the different subtools as different polygroups. GroupVisible with coverage of 1 makes all the visible polygroups into 1 single polygroup.

5. Polygroups > GroupVisible, coverage 0.5, cluster 0. This randomly assigns polygons into 2 different polygroups with roughly the same number of polygons in each polygroup; merge this subtool with the first subtool to end up with a single subtool with polygons randomly and equally distributed into 3 different polygroups.

6. Select ZModler brush, click on the face of a polygon > insert NanoMesh > Polygroup all

8. Nanomesh setting, size 6, ZOffset 0.15, random distribution 0.5

9. If the region contains more than one gene mutations, repeat steps 1-3 for the other polygroup(s).

**Figure 7A.** Surface showing before polygons have been split into 3 different polygroups.

**Figure 7B.** Surface showing after polygons have been split into 3 different polygroups.
Object Size to Represent Prevalence Information

Objects were proportionally sized to reflect prevalence of each of the genetic mutations in each region. A width, length, and height of 1 reflect 10% prevalence, while 0.5 reflects 5% prevalence and 2 reflects 20% prevalence. ZOffset of objects was adjusted to ensure smaller objects could be seen on top of large objects. Random distribution or density of objects remained the same.
**Model Organization**

After all objects were mapped, objects representing the same genetic mutations were separated from their placement polygons and grouped together. The steps are outlined below.

1. **Geometry > Convert BPR to geometry**
   - This function converts instances of NanoMesh objects into actual geometry that can be separated from their placement polygons. An instance of objects would now be in their own polygroup.

2. Mask the objects one polygroup at a time and apply “split masked points” to split the polygroup into its own subtool.

3. Group subtools that represent the same genetic mutation

4. Placement polygons were deleted. High resolution models with only smoothing applied to the surface were used as base models.

**UV mapping in ZBrush®**

In order for the mapped objects to collectively show more three-dimensionality of the PanIN and to optimize performance for the web application, texture mapping with baked shadows were applied to the objects. UV mapping were thus created to accommodate for texture mapping. All subtools containing mapped objects had UV maps created.

Each subtool was unwrapped using UV Master. Unwrapped subtools were flattened to check for overlapping geometry. UV maps were created from objects with the lowest level of subdivision. The objects were then divided once, or added one higher level of subdivision, to round out the objects. Delete the lower subdivision. Inflate the objects by a factor of 3 since adding higher subdivision levels shrink the objects slightly. Figure 8 showed one of the UV maps created.

UV mapping was also done for H&E slide so the slide image could be added onto the model as texture.
Figure 8A. UV map for mapped objects representing KRAS G12V (no prevalence).

Figure 8B. Texture map for mapped objects representing KRAS G12V (no prevalence).
Exporting Models from ZBrush®

All models were exported from ZBrush® via 3D print hub as Wavefront OBJ files. All structures and models needed to be exported were put into the same tool and were perfectly registered with each other to ensure all exported models were of the correct size, had the correct position information and UV mapping. The subtools were sized individually and the size ratio was updated every time upon export. Subtools were exported one at a time to minimize incorrect sizing of exported subtools. Refer to Table 4 for list of models exported.

<table>
<thead>
<tr>
<th>Models with smoothed surface details</th>
<th>UV Mapping</th>
<th>Final format imported into Unity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PanIN A</td>
<td>No</td>
<td>OBJ</td>
</tr>
<tr>
<td>PanIN B</td>
<td>No</td>
<td>OBJ</td>
</tr>
<tr>
<td>PanIN C</td>
<td>No</td>
<td>OBJ</td>
</tr>
<tr>
<td>PanIN D</td>
<td>No</td>
<td>OBJ</td>
</tr>
<tr>
<td>Normal pancreatic ducts</td>
<td>No</td>
<td>OBJ</td>
</tr>
<tr>
<td>Small PanINs</td>
<td>No</td>
<td>OBJ</td>
</tr>
<tr>
<td>Cancer</td>
<td>No</td>
<td>OBJ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mapped representations of genetic mutations (no prevalence)</th>
<th>UV Mapping</th>
<th>Final format imported into Unity</th>
</tr>
</thead>
<tbody>
<tr>
<td>KRAS G12D</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>KRAS G12V</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>KRAS G12R</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>KRAS Q61H</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>GNAS R844H</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>RET R886W</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>TP53 R248W</td>
<td>Yes</td>
<td>FBX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mapped representations of genetic mutations, (objects sized to prevalence of genetic mutations)</th>
<th>UV Mapping</th>
<th>Final format imported into Unity</th>
</tr>
</thead>
<tbody>
<tr>
<td>KRAS G12D</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>KRAS G12V</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>KRAS G12R</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>KRAS Q61H</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>GNAS R844H</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>RET R886W</td>
<td>Yes</td>
<td>FBX</td>
</tr>
<tr>
<td>TP53 R248W</td>
<td>Yes</td>
<td>FBX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other</th>
<th>UV Mapping</th>
<th>Final format imported into Unity</th>
</tr>
</thead>
<tbody>
<tr>
<td>H&amp;E slide</td>
<td>Yes</td>
<td>FBX</td>
</tr>
</tbody>
</table>

*Table 4. Exported models for “TC 92”.*
Lighting and Texture Mapping in Cinema 4D®

Models of mapped objects were brought back into Cinema 4D® for texture mapping. The goal was to use shadows as texture on mapped objects to accentuate the overall three-dimensionality of the PanINs lesions. The H&E slide and PanIN models were also imported to serve as basis for shadowing casting. Multiple lights at low intensity were set up around the models to mimic global illumination and to prevent casting of harsh shadows. Generic white material was applied to mapped objects. Redshift® was used as the renderer for instant feedback on lighting. Different objects representing different genetic mutations were baked separately. Baked texture maps were saved as Tagged Image File Format (TIFF). Models with baked texture maps were exported as FBX files since only FBX files retain the information needed for texture map application.

Setting Up the Visualization in Unity

Models were imported into Unity. Since the visualizations were 3D and web-based with simple lightings, 3D template was chosen as the project template. FBX files were scaled up by a factor of 100 since FBX files were sized down upon being imported into Unity. Fourteen different materials were set up for the 14 mapped object models (7 different genetic mutations with and without prevalence information reflected in object size). The 7 different genetic mutations were represented by 7 different colors. Different levels of emission were assigned to increase and decrease brightness of a color. All colors were chosen from a colorblind-friendly palette and were simulated in Adobe Photoshop® to ensure distinctness of colors chosen. Texture maps were added to “albedo” under “main maps” and “color” under “emission” to add some dimensionality to the mapped objects. Texture map with the H&E slide image was added to the H&E slide model.
Figure 9A. Colors representing different genetic mutations as seen with normal vision.

Figure 9B. Colors representing different genetic mutations as seen with protanopia-type color blindness.
Figure 9C. Colors representing different genetic mutations as seen with deuteranopia-type color blindness.

Figure 10. Material settings showing how texture maps are added to various channels.
Figure 11A. 3D visualization with texture maps, showing more three-dimensionality.

Figure 11B. 3D visualization without texture maps.
Still Images and Turntables

Screenshots showing the desired combinations of models and genetic mutations were saved as Portable Game Notation (PNG) files. A “turntable” script was added to the main camera to allow for rotation of the camera around a “target” point. Refer to Appendix 1. for the turntable script. Unity recorder was used to record an 18-second-long turntable animation at 30 frames per second (FPS) and were saved as MP4 files. Editing of the turntables was done in Adobe After Effects®, and converted to mp4 files using Adobe Media Encoder®.

Interactive Web-Based Application Development

Interactions in the web application was first mocked up in Adobe XD®. This helped determine the flow of interactivities that would take place in the web application and provided guidelines for how the C# scripts were to be written. Some considerations included what should be the default state of the visualization when user first access the web application, and when toggling between the prevalence and no prevalence visualizations, what other models would be affected.

The main interactions were built in Unity and coded in C#, a programming language. Scripts attached to game objects in Unity made different types of interactions possible. The main camera acted as the main hub for scripts for organizational purposes. Either the script would be automatically run and be applied to the game object it is attached to, or a toggle button would execute the script by calling the relevant function in the relevant script attached to the main camera. Refer to Table 5. for C# scripts written and descriptions of their interactions.

Since some of the interactions required a mouse over action, such as hovering mouse over a PanIN region to display different prevalence information, mesh colliders hugging the topology of the PanINs were needed to define the region where collision would take place. Meshes of PanINs that had been segmented according to the gene sequencing region they were in were imported into Cinema 4D®. Their axis repositioned to align with the center of the mesh. Previously, all models had their axis set to align with the bottom left corner of the mesh (point 1 in Fig. 4). Not having axis set to the center interfered with positioning of the
Meshes were exported out of Cinema 4D® as FBX files so axis information could be preserved. Empty game objects were created to hold both the mesh colliders and the HoverPanel script so the HoverPanel script can detect the region of collision. Meshes were then added to mesh colliders to define the shape of the colliders.

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Appendix</th>
<th>Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrbitalCamera*</td>
<td>B</td>
<td>Rotate, zoom the camera of view around a “target”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zoom out to maximum distance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset to default rotation</td>
</tr>
<tr>
<td>SetTarget*</td>
<td>C</td>
<td>Double clicking on a region of the PanIN zooms in to that region</td>
</tr>
<tr>
<td>Turntable*</td>
<td>A</td>
<td>Rotates the camera of view around a “target” from a set default camera view to create a turntable; click again to disable the turntable</td>
</tr>
<tr>
<td>ToggleItem*</td>
<td>D</td>
<td>Toggle different gene mutation meshes on and off</td>
</tr>
<tr>
<td>ToggleOtherStructure*</td>
<td>E</td>
<td>Toggle other structures on and off, including normal ducts, small PanINs, cancer, H&amp;E slide</td>
</tr>
<tr>
<td>SwitchMesh*</td>
<td>F</td>
<td>Toggle between prevalence and no prevalence mode</td>
</tr>
<tr>
<td>HoverPanel</td>
<td>G</td>
<td>Hovering mouse over a region of the PanIN displays the gene mutations present in that region and their prevalence</td>
</tr>
<tr>
<td>FollowLabel</td>
<td>H</td>
<td>A camera-facing label that follows a tagged region of the PanIN</td>
</tr>
</tbody>
</table>

*Table 5. C# scripts for Unity; *attached to the main camera.
User Interface (UI)

To create a user interface for the web application, a “canvas” that housed all the UI elements was created. The render mode of the canvas was set to overlay on top of the screen space. All toggles and buttons that existed in the interface were made children under the canvas. A style guide defining the font, weight, colors was created in Adobe Illustrator®. Icons representing zoom out, reset to default rotation and position, and turntable were created. A light grey color was used as background to push the colors of the visualizations forward. A muted palette of greys was chosen for the buttons and panels to not compete with the representational colors chosen to indicate the different genetic mutations. UI assets created in Adobe Illustrator® were set as backgrounds for the buttons and panels.

To create labels that followed each of the regions to display prevalence information, additional canvases were created. These canvases were sized to the size of the labels and made children of the regions they would follow. A “FollowLabel” script was added to each of the labels to make the display of the labels face the camera regardless of the size and rotation of the models. Refer to Fig. 12 for a visualization of these labels. These labels could be toggled on and off, but toggling for these labels was only active when only one genetic mutation was displayed on the models.

Figure 12. 3D visualization with front-facing labels attached to different regions of the PanIN.
Promotional Image

Lastly, models of the genetic mutations and PanINs were brought into Cinema 4D®. Camera, materials, and lightings were set up using Redshift®. Different camera angles and lightings were explored with the aim to general curiosity and interest in the viewers. The final image was exported as series of multipass images. Final image was composed in Adobe Photoshop®.
RESULTS

3D Data Visualization

A 3D scatter plot approach was used to visualize the 3D distribution of genetic mutations in PanINs. Seven genetic mutations were visualized, including KRAS G12D, KRAS G12V, KRAS G12R, KRAS Q61H, GNAS R844H, RET R886W, TP53 R248W. Since 4 of the 7 genetic mutations were in the KRAS gene, a range of blue hues were assigned to this group. TP53 was assigned a high-contrast yellow color for to highlight these mutations which typically occur in high-grade PanINs and PDACs. Table 6 shows the different colors that were used for the different structures. Figures 12 – 23 show still images that can be used in a journal publication. Genetic mutations are displayed individually in Figures 16 – 22. This allows for easier comparisons between different genetic mutations in their distribution across PanINs.

<table>
<thead>
<tr>
<th>Genetic Mutations</th>
<th>Main Color (MC)</th>
<th>MC Hex Code</th>
<th>Emission Color (EC)</th>
<th>EC Hex Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>KRAS G12D</td>
<td>#001552</td>
<td></td>
<td>#18192E</td>
<td></td>
</tr>
<tr>
<td>KRAS G12V</td>
<td>#2D8FFF</td>
<td></td>
<td>#004AFA</td>
<td></td>
</tr>
<tr>
<td>KRAS G12R</td>
<td>#BFFFFC</td>
<td></td>
<td>#001F33</td>
<td></td>
</tr>
<tr>
<td>KRAS Q61H</td>
<td>#180093</td>
<td></td>
<td>#101026</td>
<td></td>
</tr>
<tr>
<td>GNAS R844H</td>
<td>#7BDD50</td>
<td></td>
<td>#031313</td>
<td></td>
</tr>
<tr>
<td>RET R886W</td>
<td>#FF6F25</td>
<td></td>
<td>#5C1414</td>
<td></td>
</tr>
<tr>
<td>TP53 R248W</td>
<td>#FFBA00</td>
<td></td>
<td>#FFB000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Structures</th>
<th>Main Color (MC)</th>
<th>MC Hex Code</th>
<th>Emission Color (EC)</th>
<th>EC Hex Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PanINs</td>
<td>#434343</td>
<td></td>
<td>#060606</td>
<td></td>
</tr>
<tr>
<td>Small PanINs</td>
<td>#434343</td>
<td></td>
<td>#060606</td>
<td></td>
</tr>
<tr>
<td>Normal ducts</td>
<td>#4B3326</td>
<td></td>
<td>#131313</td>
<td></td>
</tr>
<tr>
<td>Cancer</td>
<td>#80610D</td>
<td></td>
<td>#131313</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Color guide for genetic mutations and other structures.
Figure 13. 3D visualization of PanINs, genetic mutations (no prevalence), normal ducts, and cancer H&E slide included for orientation.

Figure 14. 3D visualization of PanINs, and genetic mutations (no prevalence) H&E slide included for orientation.
**Figure 15.** 3D visualization of PanINs, genetic mutations (no prevalence), and normal ducts (light brown).

**Figure 16.** 3D visualization of PanINs (grey), normal ducts (light brown), and cancer (yellow).
Figure 17. 3D visualization PanINs (grey) with KRAS G12D (no prevalence).

Figure 18. 3D visualization of PanINs (grey) with KRAS G12V (no prevalence).
Figure 19. 3D visualization of PanINs (grey) with KRAS G12R (no prevalence).

Figure 20. 3D visualization of PanINs (grey) with KRAS Q61H (no prevalence).
**Figure 21.** 3D visualization of PanINs (grey) with GNAS R844H (no prevalence).

**Figure 22.** 3D visualization of PanINs (grey) with RET R886W (no prevalence).
Figure 23. 3D visualization of PanINs (grey) with TP53 R248W (no prevalence).
**Figure 24A.** 3D visualization of genetic mutations without object size reflecting prevalence of each genetic mutation.

**Figure 24B.** 3D visualization of genetic mutations with object size reflecting prevalence of each genetic mutation.
Turntables

Turntables supplemented the still images by offering viewers more angles of view of the 3D visualization. Twelve-second turntables corresponding to each of Figures 12 – 23 except Figure 15 were recorded. A summarizing turntable was created displaying all the structures stripping away to reveal the PanINs and the genetic mutations, before showing all the genetic mutations one by one. Select screenshots from one of the turntables are shown below in Figure 24.

Interactive Web-Based Application

Refer to Table 5 for a summary of all the interactions in the web application. Figures 25 – 28 displayed a few screenshots from the application. Figures 29 – 30 displayed the style sheet for the visualization.
Figure 25. Select screenshots from the summarizing turntable showing PanINs and distribution of genetic mutations.
Figure 26. Web application showing the default state of the visualization when the application is first accessed.
Figure 27. Web application showing different structures being toggled on and off.
Figure 28. Hovering over a region displayed the region's relevant genetic mutation information in the bottom panel.

KRAS G12D (32%); RET R886W (28%); TP53 R248W (9%)
Figure 29. 3D visualization with prevalence mode being toggled on.
Main colours

D9D9D9  C4C4C4  FFFFFF  000000

Secondary colours

EF7C51  FDEC50  96DA7E  767676  666666

DCFEEF  5E99EA  2A2EAF  2A376B

Font

Helvetica Neue Medium
Helvetica Neue Bold

Figure 30. Style guide showing colors used for the web application.
Figure 31. Style guide for the buttons, scroll bars, icons used in the web application.
Promotional Image

Figure 31 shows the promotional image being mocked up as a journal cover.

Figure 32. Mock journal cover with the 3D visualization.
Access to Assets

Images, turntables, and demo videos of the visualization can be partially viewed on https://t-wang.studio/.

Access to assets created may be granted by contacting the author at tingi.tina.wang@gmail.com or through the Department of Art as Applied to Medicine at Johns Hopkins University School of Medicine, http://medicalart.johnshopkins.edu/.
DISCUSSION

3D Visualization

A 3D scatter plot approach was used to visualize the 3D distribution of genetic mutations in PanINs. Objects were uniformly mapped across surfaces of PanINs. Different colors were assigned to objects to represent different genetic mutations. Prevalence of each genetic mutation was represented by size. Four outputs were created, including stills that could be used in journal publications, turntable videos, an online interactive platform, as well as a promotional image that could be shared over social media.

Various object sizes, densities, and coverage were explored to find a best solution to convey the 3D distribution of genetic mutations while retaining the dimensionality of PanINs. Having smaller objects at a higher density better captured the surface details and dimensionality of PanINs, but the smaller size made the different colors harder to discern compared to larger-sized objects. Larger-sized objects made it easier to discern between colors but at lower densities could obscure the dimensionality of PanINs. Larger-sized objects at the desired densities to convey the dimensionality of PanINs were, however, prone to overlapping with each other. This is due to the fact that objects were modelled in ZBrush®, which does not contain simulations or algorithms to automatically detect, minimize or eliminate overlapping geometry. Earlier attempts showed overlapping objects with marbling effects. This could be falsely interpreted as cells harboring multiple genetic mutations at once. However, due to limitations in genetic sequencing, it was not known what populations of cells harbored multiple genetic mutations. The overlapping of objects was improved by having objects mapped onto models with larger polygons and whose polygons were split into different polygroups. Having different polygroups in some ways limited where objects could be mapped, but it did not completely resolve overlapping objects.

Currently, prevalence of genetic mutations is represented by object size. One reason prevalence could not be represented by a ratio of different densities was because populations of cells harboring multiple genetic mutations is not known. Having single-colored objects represented at different densities gave the false impression that genetic mutations were
mutually exclusive and that populations contained more than one genetic mutation at once. In the future, should the populations of cells containing multiple genetic mutations be known, and objects could exhibit a marbling of different colors, there was still the potential problem those objects would be hard to distinguish among the tightly packed objects. On smaller screens, single objects with multiple colors would be even harder to discern, decreasing the user experience for the viewers. If prevalence could be represented by densities, this would free up object size as a parameter to be used to represent another facet such as gene expression.

Another issue with using object size to represent prevalence was that when there was a huge difference in the prevalence values, one set of objects, because of their smaller size, would need to be positioned on top of their larger counterparts. This created an issue when modelling in ZBrush® since objects could not be uniformly offset. Some objects would always appear to float instead of sitting on top of the other objects. These objects needed to be manually repositioned. Again, when displaying genetic mutations with high prevalence, the representational objects were inevitably much larger in size. This resulted in objects clipping into each other if same densities were to be preserved.

Spheres were naturally chosen as the default shape of the mapped objects. However, other geometries such cubes and cones were explored as an additional parameter to represent a facet of the information. However, different geometries were hard to discern given the size of the objects. Because of how objects were mapped onto surfaces, having objects all uniformly point in a global direction was an additional challenge. One advantage to using spheres was that spheres looked identical when viewed from different angles. Such was not the case for cubes and cones. Having cones and cubes all point in different direction could detract or even obscure the visualization. Additionally, there is no a way to standardize the size and volumes of different geometries in ZBrush®. The size of an object is determined by its bounding boxes in ZBrush®. A sphere with a bounding box of 1x1x1 was not of the same volume as a cube with the same bounding box. Another approach that was explored was mapping of 2D objects directly onto the PanIN surfaces. One method is to simplify the topology of the PanINs and divide the polygons into differently colored polygroups. However, this approach simplified the surface details of PanIN that would be used in the final visualization. Due to complexity of the topology, once simplified, it was not possible to project all original details back onto the processed
models. Another approach tested was to wrap a dot texture around the models. Again, due to complexity of the surfaces, creating UV maps for PanINs and ensuring a seamless mapping of the texture would be a time-consuming process.

Regarding shading, currently 7 colors were used to represent the 7 different genetic mutations. One concern regarding this method was if additional genetic mutations needed to visualized, what other colors could be used while ensuring the visualization remained colorblind friendly. In addition to having a colorblind friendly palette, patterns and textures such as stripes had been suggested as possible ways to make visualization more colorblind friendly. However, such stripe patterns to the visualization created a shuttering artifact when the visualization was moved in a 3D space.

Structures such as pancreatic ducts, other small PanINs, and an H&E slide were added to the visualization to help viewers orient the visualization. CODA had also identified several other structures in “TC 92”, including blood vessels, islets of Langerhans, nerves, acini, stroma, lymph nodes, and fat. Though not currently in the visualization, these structures could be added to the visualization to give the viewers options to examine structures they are interested in.

**Future Expansion of the Visualization**

With CODA still in its early stages, the only two parameters depicted were the identity and location of each genetic mutation. However, other parameters can affect cancer progression and can be of diagnostic value include RNA and protein expression. These other factors include microRNA (miRNA) expression, telomere shortening, KRAS gene copy number, and inflammation. MiRNAs can negatively regulate gene expression, and their aberrant expressions have been implicated in PDAC (Lee et al., 2021). Telomere shortening has been observed in 90% of low-grade PanINs (Lee et al., 2021). Changes to KRAS gene copy number can lead to a worse prognosis (Thompson et al., 2020). Inflammatory cells in TME tend to harbor an immunosuppressive phenotype (Padoan et al., 2019). How can the visualization be expanded to accommodate for all these new information?

If prevalence could be represented spatially, perhaps object size could be used
to represent RNA expression. Alternatively, different modes would need to be added, with object size showing different parameters in different modes. The downside of this approach is that only limited number of parameters can be viewed at once. If all these data are to be incorporated into the interactive application, what would the flow of information look like? How can the different modes be organized in such a way to prevent a clutter of buttons in the interface?

Moreover, RNA expression can change over time and can differ in different regions of tumor (Thompson et al., 2020). Perhaps a short animation or a scroll bar allowing the user to scroll through the different time points could be used to visualize selected features over time. Similarly, a short animation or a scroll bar could help visualize how genetic mutations accumulate over time. Though broadly speaking, some genetic mutations develop before the others; for example, \textit{KRAS} mutations typically happen in low-grade PanINs versus \textit{TP53} happening in high-grade PanINs. It would be interesting to see if the various \textit{KRAS} mutations observed happen simultaneously and independently of each other, or if certain regions of pancreatic ducts with certain morphology were more prone to acquiring \textit{KRAS} mutations, which in turn drive \textit{KRAS} mutations in other regions. If two regions each acquired their own genetic mutations at the same time, one could then ask what were some similarities shared by these two regions. This could be represented by a change-over-time scatter plot with different colored spheres representing genetic mutations of different origins.

Regarding the resolution of gene sequencing, though the regions were divided according the minimum number of cells, the current visualization showed as if the genetic mutations were equally spaced throughout the region. It does not provide information regarding if one part of the region contained a higher density of cells possessing the mutation. Should more information be known in the future, distribution of mapped objects could be changed to more accurately reflect actual distribution of the genetic mutation, as well as having a heat map mode to show areas with high densities of genetic mutations.

\textbf{Interactive Application Interface and User Experience}

For the interface, a light grey shade was used as a background to enhance the contrast
if the colorized visualizations. A muted palette of greys was chosen for the buttons and panels to not compete with the representational colors indicative of the different genetic mutations. An eye icon was added to buttons to help indicate whether a structure was visible. The eye icon was chosen over a color change to keep the text readable and interface as clean as possible. The scroll bar indicating whether a genetic mutation had been toggled on to visually separated the genetic mutations from the other togglable structures. The scroll from left to right also gives viewers a simple visual cue whether a genetic mutation had been toggled on. The scroll bars were colorized to match the colors of the mapped objects.

**Visualization as a Platform**

The visualization process depicted so far involves multiple programs, and manually setting up the mapped objects, materials, lighting, and interactions. If this type of visualization is to be become one of the standard ways for reporting and communicating 3D data, a more streamlined workflow is needed. A program would need to be designed starting from how models of PanINs and other structures could be uploaded, genetic mutation information and other relevant data be entered, to how the program will calculate how objects are mapped onto PanINs. Outputs generated include the interactive platform with the ability export still images and turntable videos.
CONCLUSION

PDAC is one of the deadliest forms of cancers in the United States and is often diagnosed at advanced stages with poor prognosis. There has yet to be reliable screening approach to catch the disease in its early stage. The first step to develop such an approach is to understand the precancerous lesions in the pancreas. CODA was established to examine human pancreatic lesions in a 3D space. The application of gene sequencing to tissue blocks reconstructed in 3D with CODA allows for true 3D genetics. This approach was applied to different regions of PanINs. Seven genetic mutations were identified across 4 distinct PanINs in sample “TC 92”. The challenge faced was how to visualize the resultant data. That is how to show the complex 3D distribution of genetic mutation so that it could be understood by cancer researchers.

This thesis project took a 3D scatter plot approach to visualize the 3D distribution of genetic mutations. It also used object size to represent prevalence of each genetic mutation within the region. The 3D models preserved as much of the structures of PanINs as possible while different colors allowed all genetic mutations to be displayed at once. Four outputs were created, including still images, turntable videos, an interactive platform, and a promotional image.

The visualization represented a first step in creating new dialogues amongst cancer researchers, with the possibility of expanding in several different directions as new data become available. It encourages examining contributing factors to tumorigenesis in humans in three-dimensions as compared to a 2D space. This can lead to a more robust and nuanced understanding of the disease progression that, in the future, may result in better outcomes for patients of PDAC and other pancreatic cancers.
REFERENCES


Appendix A. Turntable script

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Turntable : MonoBehaviour
{
    public float speed;
    public GameObject objectSpin;
    public bool spin = false;
    public bool spinStart = false;
    public OrbitalCamera orbitalCamera;

    void Awake()
    {
        orbitalCamera = GameObject.Find("Main Camera").GetComponent<OrbitalCamera>();
    }

    void Update()
    {
        if(spin)
        {
            objectSpin.transform.Rotate(0, speed*Time.deltaTime, 0);
        } else
        {
            orbitalCamera.SetHome();
            spinStart = false;
        } else
        {
            orbitalCamera.SetHome();
            spinStart = false;
        } else
        {
            orbitalCamera.SetHome();
            spinStart = false;
        }
    }

    void Turn()
    {
        spin = !spin;
    }

    public void Turn()
    {
        spin = !spin;
    }

    if(spin)
    {
        spinStart = true;
    }
}
```
**Appendix B. Orbital camera script**

```csharp
using UnityEngine;
using System.Collections;

public class OrbitalCamera : MonoBehaviour
{
    public Transform target;
    public Vector3 newTarget;
    public float distanceSpeed = 1f;
    public Vector3 homeTarget;

    public Vector3 homePos;
    public Quaternion homeRot; //rotation
    public bool goHome = false;

    public GameObject objectSpin;

    public float distance = 2.0f;
    public float newDistance;
    public bool changeDist = false;

    public float xSpeed = 20.0f;
    public float ySpeed = 20.0f;
    public float yMinLimit = -90f;
    public float yMaxLimit = 90f;
    public float distanceMin = 10f;
    public float distanceMax = 10f;
    public float smoothTime = 2f;

    float rotationYAxis = 0.0f;
    float rotationXAxis = 0.0f;
    public float xRotation;
    public float yRotation;

    float velocityX = 0.0f;
    float velocityY = 0.0f;

    public bool zoomIN = false;

    void Awake()
    {
        objectSpin = GameObject.Find("tc 92");
    }

    // Use this for initialization
    void Start()
    {
        Vector3 angles = transform.eulerAngles;
        rotationYAxis = angles.y;
    }

    // Update is called once per frame
    void Update()
    {\n    }
}
```
rotationXAxis = angles.x;
// Make the rigid body not change rotation
if (GetComponent<Rigidbody>())
{
    GetComponent<Rigidbody>().freezeRotation = true;
}
distance = distanceMax;
newDistance = distance;
homeTarget = target.position;
newTarget = target.position;

homePos = transform.position;
homeRot = Quaternion.Euler(rotationXAxis, rotationYAxis, 0);
}

void LateUpdate()
{
    if (goHome)
    {
        target.position = homeTarget;
        newDistance = distanceMax;

        objectSpin.transform.rotation = Quaternion.Euler(0,0,0);
    }
    if (newDistance != distance && changeDist == true)
    {
        distance = Mathf.Lerp(newDistance, distance, distanceSpeed * Time.deltaTime);

        if (Mathf.Abs(newDistance - distance) < .5 )
        {
            changeDist = false;
        }
    }
}

if (target)
{
    if (goHome)
    {
        Quaternion rotation = homeRot;

        Vector3 negDistance = new Vector3(0.0f, 0.0f, -distance);
        Vector3 position = rotation * negDistance + target.position;

        transform.rotation = rotation;
        transform.position = position;

        rotationYAxis = homeRot.eulerAngles.y;
        rotationXAxis = homeRot.eulerAngles.x;
        goHome = false;
    }
else
{
    if (Input.GetMouseButton(0))
    {
        velocityX += xSpeed * Input.GetAxis("Mouse X") * distance * 0.02f;
        velocityY += ySpeed * Input.GetAxis("Mouse Y") * 0.02f;
    }
    rotationYAxis += velocityX;
    rotationXAxis -= velocityY;
    rotationXAxis = ClampAngle(rotationXAxis, yMinLimit, yMaxLimit);

    Quaternion rotation = Quaternion.Euler(rotationXAxis, rotationYAxis, 0);
    distance = Mathf.Clamp(distance - Input.GetAxis("Mouse ScrollWheel") * 5,
                           distanceMin, distanceMax);

    Vector3 negDistance = new Vector3(0.0f, 0.0f, -distance);
    Vector3 position = rotation * negDistance + target.position;

    transform.rotation = rotation;
    transform.position = position;
    velocityX = Mathf.Lerp(velocityX, 0, Time.deltaTime * smoothTime);
    velocityY = Mathf.Lerp(velocityY, 0, Time.deltaTime * smoothTime);
}

}

public static float ClampAngle(float angle, float min, float max)
{
    if (angle < -360F)
        angle += 360F;
    if (angle > 360F)
        angle -= 360F;
    return Mathf.Clamp(angle, min, max);
}

public void ZoomOut()
{
    target.position = homeTarget;
    changeDist = true;

    if(zoomIN)
    {
        distanceMin = 500;
        distanceMax = 1000;

        zoomIN = false;
    }

    newDistance = distanceMax;
Appendix C. Set target script

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SetTarget : MonoBehaviour
{
    public OrbitalCamera orbitalCamera;
    public bool oneClick = false;
    public float timerForDoubleClick = 0.5f;
    public float delay = 0.3f;
    public float zoomDistance;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            // Set target position
            target.position = homeTarget;
            changeDist = true;

            if (zoomIN)
            {
                distanceMin = 500;
                distanceMax = 1000;
                zoomIN = false;
            }

            newDistance = distanceMax;
            goHome = true;
        }
    }
}

Appendix C. Set target script

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SetTarget : MonoBehaviour
{
    public OrbitalCamera orbitalCamera;
    public bool oneClick = false;
    public float timerForDoubleClick = 0.5f;
    public float delay = 0.3f;
    public float zoomDistance;

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            // Set target position
            target.position = homeTarget;
            changeDist = true;

            if (zoomIN)
            {
                distanceMin = 500;
                distanceMax = 1000;
                zoomIN = false;
            }

            newDistance = distanceMax;
            goHome = true;
        }
    }
}
if (oneClick == false)
{
    //Single click, not sure if double click yet
    oneClick = true;
    timerForDoubleClick = Time.time;
}
else
{
    //Double click
    oneClick = false;
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;
    if (Physics.Raycast(ray, out hit))
    {
        orbitalCamera.target.position = hit.transform.position;
        orbitalCamera.newDistance = zoomDistance;
        orbitalCamera.distanceMax = zoomDistance;
        orbitalCamera.distanceMin = zoomDistance;

        orbitalCamera.changeDist = true;
        orbitalCamera.zoomIN = true;
    }
}
if (oneClick == true)
{
    if((Time.time - timerForDoubleClick) > delay)
    {
        //Single click
        oneClick = false;
    }
}
}

Appendix D. Toggle item script
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ToggleItems : MonoBehaviour
{
    public bool g12d = true;
    public GameObject g12dGene;
    public GameObject g12dGenePrev;
}
public void Toggle12d ()
{
    if (g12d == false)
    {
        g12dGene.SetActive(true);
        g12dGenePrev.SetActive(true);
        g12d = true;
    }
    else if (g12d == true)
    {
        g12dGene.SetActive(false);
        g12dGenePrev.SetActive(false);
        g12d = false;
    }
}

public bool g12v = true;
public GameObject g12vGene;
public GameObject g12vGenePrev;

public void Toggle12v ()
{
    if (g12v == false)
    {
        g12vGene.SetActive(true);
        g12vGenePrev.SetActive(true);
        g12v = true;
    }
    else if (g12v == true)
    {
        g12vGene.SetActive(false);
        g12vGenePrev.SetActive(false);
        g12v = false;
    }
}

public bool g12r = true;
public GameObject g12rGene;
public GameObject g12rGenePrev;

public void Toggle12r ()
{
    if (g12r == false)
    {
        g12rGene.SetActive(true);
        g12rGenePrev.SetActive(true);
        g12r = true;
    }
else if (g12r == true)
{
    g12rGene.SetActive(false);
    g12rGenePrev.SetActive(false);
    g12r = false;
}

public bool q61h = true;
public GameObject q61hGene;
public GameObject q61hGenePrev;

public void Toggleq61h()
{
    if (q61h == false)
    {
        q61hGene.SetActive(true);
        q61hGenePrev.SetActive(true);
        q61h = true;
    }
    else if (q61h == true)
    {
        q61hGene.SetActive(false);
        q61hGenePrev.SetActive(false);
        q61h = false;
    }
}

public bool gnas = true;
public GameObject gnasGene;
public GameObject gnasGenePrev;

public void Togglegnas()
{
    if (gnas == false)
    {
        gnasGene.SetActive(true);
        gnasGenePrev.SetActive(true);
        gnas = true;
    }
    else if (gnas == true)
    {
        gnasGene.SetActive(false);
        gnasGenePrev.SetActive(false);
        gnas = false;
    }
}
public bool ret = true;
public GameObject retGene;
public GameObject retGenePrev;

public void Toggleret ()
{
    if (ret == false)
    {
        retGene.SetActive(true);
        retGenePrev.SetActive(true);
        ret = true;
    }
    else if (ret == true)
    {
        retGene.SetActive(false);
        retGenePrev.SetActive(false);
        ret = false;
    }
}

public bool tp53 = true;
public GameObject tp53Gene;
public GameObject tp53GenePrev;

public void Toggletp53 ()
{
    if (tp53 == false)
    {
        tp53Gene.SetActive(true);
        tp53GenePrev.SetActive(true);
        tp53 = true;
    }
    else if (tp53 == true)
    {
        tp53Gene.SetActive(false);
        tp53GenePrev.SetActive(false);
        tp53 = false;
    }
}

Appendix E. Toggle other structure script
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class ToggleOtherStructures : MonoBehaviour
{
    public bool smallP = true;
    public GameObject smallPanin;

    public void ToggleSmallPanin ()
    {
        if (smallP == false)
        {
            smallPanin.SetActive(true);
            smallP = true;
        }
        else if (smallP == true)
        {
            smallPanin.SetActive(false);
            smallP = false;
        }
    }

    public bool normalD = true;
    public GameObject normalDucts;

    public void ToggleNormalDucts ()
    {
        if (normalD == false)
        {
            normalDucts.SetActive(true);
            normalD = true;
        }
        else if (normalD == true)
        {
            normalDucts.SetActive(false);
            normalD = false;
        }
    }

    public bool he = true;
    public GameObject heSlide;

    public void ToggleSlide ()
    {
        if (he == false)
        {
            heSlide.SetActive(true);
            he = true;
        }
        else if (he == true)
Appendix F. Switch mesh script

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SwitchMesh : MonoBehaviour
{
    public GameObject noPrev;
    public GameObject prev;
    public bool switchPrev = false;

    // Start is called before the first frame update
    void Start()
    {
        prev.SetActive(false);
    }

    public void SwitchtoPrev()
    {
        if (switchPrev == false)
        {
            prev.SetActive(true);
            noPrev.SetActive(false);
        }
    }
}
switchPrev = true;
}
else if (switchPrev == true)
{
    prev.SetActive(false);
    noPrev.SetActive(true);
    switchPrev = false;
}
}

Appendix G. Hover panel script
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
public class HoverPanel : MonoBehaviour
{
    public ToggleItems ToggleItems;
    public TextMeshProUGUI hoverText;
    public GameObject prevalencePanel;

    [TextArea]
    public string prevalenceText;
    [TextArea]
    public string prevalenceText2;
    [TextArea]
    public string prevalenceText3;
    [TextArea]
    public string prevalenceText4;
    [TextArea]
    public string prevalenceText5;
    [TextArea]
    public string prevalenceText6;
    [TextArea]
    public string prevalenceText7;

    [TextArea]
    string prevalenceTextUse;
    [TextArea]
    string prevalenceTextUse2;
    [TextArea]
string prevalenceTextUse3;
[TextArea]
string prevalenceTextUse4;
[TextArea]
string prevalenceTextUse5;
[TextArea]
string prevalenceTextUse6;
[TextArea]
string prevalenceTextUse7;

public ToggleItems toggleItems;

void Awake()
{
    toggleItems = GameObject.Find("Main Camera").GetComponent<ToggleItems>();
    //collect the panel
    prevalencePanel = GameObject.Find("Prevalence Panel");
    hoverText = GameObject.Find("Prevalence Text").GetComponent<TextMeshProUGUI>();
}

void OnMouseEnter()
{
    if(toggleItems.g12d)
    {
        prevalenceTextUse = prevalenceText;
    }
    else
    {
        prevalenceTextUse = null;
    }

    if(toggleItems.g12v)
    {
        prevalenceTextUse2 = prevalenceText2;
    }
    else
    {
        prevalenceTextUse2 = null;
    }

    if(toggleItems.g12r)
    {
        prevalenceTextUse3 = prevalenceText3;
    }
    else
    {
        prevalenceTextUse3 = null;
    }
}
if(toggleItems.q61h)
{
    prevalenceTextUse4 = prevalenceText4;
} else
{
    prevalenceTextUse4 = null;
}

if(toggleItems.gnas)
{
    prevalenceTextUse5 = prevalenceText5;
} else
{
    prevalenceTextUse5 = null;
}

if(toggleItems.ret)
{
    prevalenceTextUse6 = prevalenceText6;
} else
{
    prevalenceTextUse6 = null;
}

if(toggleItems.tp53)
{
    prevalenceTextUse7 = prevalenceText7;
} else
{
    prevalenceTextUse7 = null;
}

prevalencePanel.SetActive(true);

    hoverText.text = prevalenceTextUse + prevalenceTextUse2 + prevalenceTextUse3 + prevalenceTextUse4 + prevalenceTextUse5 + prevalenceTextUse6 + prevalenceTextUse7;
    }

void OnMouseExit()
{
    hoverText.text = null;
}
}
Appendix H. Follow label script

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FollowLabel : MonoBehaviour
{
    public float fixedSize = 0.0005f;
    public Camera cam;
    public float distance;
    public float size;

    // Update is called once per frame
    void Update()
    {
        distance = (cam.transform.position - transform.position).magnitude;
        size = distance * fixedSize * cam.fieldOfView;
        transform.localScale = Vector3.one * size;
        transform.forward = transform.position - cam.transform.position;
    }
}
Ting I Wang, was born in Taiwan. She and her family immigrated to Canada where she spent many years learning English the hard way while exploring nature with the new puppy she could finally have. Growing up, she had always anticipated a cookie-cutter life of learning piano as a little girl and then becoming a doctor later in life. She drew more than she played piano. When she graduated from University of British Columbia with a Bachelor of Science in Microbiology and Immunology, she was more intrigued with the complexity of biological sciences than “becoming a doctor”.

Ting had worked in a variety of places, most having nothing to do sciences. During her undergraduate years, Ting worked at the Child and Family Research Institute (CFRI) as a student research assistant studying respiratory viruses. After graduation, she dove head-first into the tourism industry. It was her meandering curiosity in the sciences and always loving to draw that somehow lead her into medical illustration.

A somewhat unfamiliar idea, her colleague at the CFRI had introduced her to the field of medical illustration. She bought her first digital tablet, learned how to paint digitally, but medicine and illustration still seemed a “Frankenstein” concept with two oddly stitched-together disciplines. Later when she worked as a tour guide with the Rocky Mountaineer, entertaining and educating guests seeking interesting recreational activities, she found herself drawn to stories that pertained to nature, biology, and ecology. With a solid background in the sciences, painting these scientific subjects for her audience with words lead to the funniest comments. That was her first glimpse into how art could cross paths with science. Ting matriculated to the Medical and Biological Illustration Graduate Program at Johns Hopkins University.

Her experiences at Hopkins have opened her eyes to the endless opportunities in the field. Her efforts lead to her being named the recipient to the Frank Netter Scholarship. She also received Award of Merit for her “Anatomy of Inguinal Canal” piece at the annual Association of Medical Illustrators conference, and a Vesalius Trust Research Grant for her thesis endeavors. She aims to keep drawing and exploring, seeking the best visual solutions to all complex scientific problems she encounters.