Semi-Automated Diagnosis of Pulmonary Hypertension Using PUMA, a Pulmonary Mapping and Analysis Tool

by

Holly Lynn Berty

Bachelors of Science, Chatham University, 2002

Master's in Biology, Chatham University, 2003

Submitted to the Graduate Faculty of

University of Pittsburgh in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

This dissertation was presented

by

Holly Lynn Berty

It was defended on

November 20, 2012

and approved by

Dr. Shyam Visweswaran, Assistant Professor, Department of Biomedical Informatics

Dr. Gregory Cooper, Professor, Department of Biomedical Informatics

Dr. Marc Simon, McGowan Institute for Regenerative Medicine, UPMC

Dissertation Advisor: Dr. Brian Chapman, Assistant Professor, University California, San Diego

Semi-Automated Diagnosis of Pulmonary Hypertension Using PUMA, a Pulmonary Mapping and Analysis tool

Holly Lynn Berty, M.S.

University of Pittsburgh, 2012

Pulmonary Arterial Hypertension (PAH) is a progressive, potentially fatal disease that results in the remodeling of the pulmonary vasculature. Currently the gold standard for diagnosis of pulmonary hypertension is through right heart catheterization, an invasive and costly procedure where pressure measurements are made directly within the affected vessels. Since PAH is associated with the remodeling of the pulmonary arteries, others have proposed quantifying the vessel geometry depicted in computed tomography (CT) images as a non-invasive technique for diagnosis of PAH. The work presented here proposes a similar method of diagnosis by defining and incorporating techniques that are both manual in nature in reference to the segmentation process and automated with the modeling and anatomic measurement quantification steps. Data comprised of both normal and disease cases were gathered and the vessel geometry (specifically the pulmonary trunk, right main pulmonary artery and the left main pulmonary artery) were measured both manually and automatically. A comparison of the automated measurements of the vessel geometry to the manual measurements showed no significant difference between the means of the two groups. A significant difference was found between the cases and the controls leading to the possibility of classifying images based on the vessel geometry. Logistic regression and naïve Bayes models were constructed from the data for discriminating the cases from the controls. Overall, the Naïve Bayes model performed better with a higher sensitivity of 42.9% compared to 19% and a small decrease in specificity of 90.9%

from 96.6%, and the model is able to classify correctly more of the patients with disease. Due to the permanent nature of the disease a type I error is acceptable; we prefer to classify patients that do not have the disease as positives than vice versa. We found that the segmenting of additional branches of the pulmonary vasculature could provide additional information for the improvement of the models presented here. In conclusion, we were able to quantify the vessel geometry depicted in CT images as a non-invasive technique for diagnosing PAH and we have shown that the two classes of measurements are not significantly different.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to acknowledge all of those who have collaborated and helped with this work. First I would like to acknowledge and thank my advisor Dr. Brian Chapman, who has taught me more than I ever could have imagined throughout this process. A true mentor who has helped with every aspect of this work and it could not have been completed without his help and guidance. My committee members: Dr. Marc Simon who gave his guidance and expertise on the disease attributes and for sharing the pressure data with us; Dr. Shyam Visweswaren for his advice and help with the machine learning aspects of this work; Dr. Greg Cooper for his support throughout this process; and Dr. Rodger Day for his guidance in regards to the statistical aspects of this work.

I would like to acknowledge the NLM for providing the funding for the first few years of my work while in the Department of Biomedical Informatics (DBMI) and I would like to thank DBMI for their continued support throughout the duration of this research.

Finally, I would like to acknowledge and thank my family for supporting me and keeping me going on those long nights when I just wanted to hang it up and never see it again. Your faith in me and push gave me the drive I needed to finish what I started so many years ago. Thank you all.

1.0 INTRODUCTION

This chapter summarizes the motivation and the main goals for this study.

1.1 DESCRIPTION AND SIGNIFICANCE OF THE PROBLEM

Pulmonary hypertension (PH) is a common condition that is often overlooked until late in its progression [1]. PH affects thousands of people each year in the United States and a prompt and accurate diagnosis is vital to improving patient outcome. In 2002 in the United States, 15,668 people died and 260,000 were hospitalized for PH [2, 3]. The focus of this research is on a subset of PH specifically, pulmonary arterial hypertension (PAH). Although PAH is a rare disease it is becoming more and more prevalent. A study published in 2011 by Frost et al. stated that the prevalence of category 1 PAH as defined by the world health organization is 10.6 cases/million of adult inhabitants and incidence of 2.0 cases/million of adult inhabitants/year in the United States [4]. In 2010 a study by Bandesch et al. reported using information gathered from registries in France and the United States that the mean age at diagnosis is 50 years with the ratio of female to male of 3.9 [5]. Currently, the median interval from symptom onset to diagnosis is 1.1 years [5]; this has not changed since the 1980's [6]. In 2007 it was reported that 3-year survival has improved from 48% reported in 1991 [7] to 67% in the U.S [8].

Diagnosis of PAH is currently obtained with pressure measurements acquired with right

heart catheterization, an invasive and costly procedure. However, since PAH results in a remodeling of the pulmonary vasculature, it is possible that non-invasive visualization of the pulmonary arteries (via magnetic resonance imaging (MRI) or computed tomography (CT)) may provide sufficient information for diagnosing PAH. It may be possible to identify the presence of disease from the vasculature remodeling prior to symptom onset and improve the time it takes for detection and diagnosis. The work described in this dissertation will establish the technical basis for semi-automated pulmonary vasculature quantification and demonstrate the feasibility of using this semi-automated quantification to diagnose PAH. In this work we use Computed Tomography Pulmonary Angiography (CTPA) examinations, since these examinations are readily available from the UPMC PACS (University of Pittsburgh Medical Center Picture Archiving and Communication Systems). However, in principle, the techniques we have developed here could also be applied to Magnetic Resonance Pulmonary Angiography.

1.2 TWO HYPOTHESES

Hypothesis 1: Accurate models of the pulmonary arterial tree can be generated semi-automatically from CTPA images. (Specific Aims 1 and 2 below)

To address the problem of finding a method of diagnosing PAH in a non-invasive manner, the first step is to establish the technical basis for pulmonary vasculature quantification; this is described in the first hypothesis.

Hypothesis 2: Morphological measurements made from the semi-automatically generated models can accurately differentiate pulmonary hypertension cases from healthy control cases.

(Specific Aim 3 below)

The second hypothesis will demonstrate the feasibility of using the methods derived in the first hypothesis for modeling the pulmonary vasculature for quantification of said vasculature in the diagnosis of PAH.

To test and further define these hypotheses we performed the following specific aims:

1.2.1 Specific Aims for Hypothesis 1

Specific Aim 1: Create a repository of CTPA exams. The exams consisted of cases that are positive for PAH and controls that are negative for PAH.

To determine the ability of modeling the pulmonary vasculature data it is necessary to learn and improve the modeling process. Specific aim 1 is focused solely on gathering this data.

Specific Aim 2: Create and validate PUMA, a **PU**lmonary **M**apping and **A**nalysis tool that semi-automatically generates pulmonary vascular models.

In this aim we take the data identified in aim 1 and use it for generating vascular models for quantification.

1.2.2 Specific Aim for Hypothesis 2

Specific Aim 3: Use PUMA to diagnose PAH. This was done by performing semi-automated measurements of vascular diameters and making comparisons to measurements from known normal cases.

The final aim takes the output (models) generated in the first hypothesis, specifically aim 2 and learns characteristic information of the vasculature along with quantifying the vasculature for the classification of disease.

1.3 GLOSSARY OF TERMS

This section summarizes and defines the key acronyms used throughout this dissertation.

CT or CTPA (Computed Tomography or Computed Tomography Pulmonary Angiography) ~ an imaging modality that uses x-rays for gathering data on the pulmonary vasculature.

LMPA ~ Left Main Pulmonary Artery

MPAP or PAMP (Mean Pulmonary Artery Pressure) ~ Pressure measurement taken during the right heart catheterization procedure used in the diagnosis of disease.

PH (Pulmonary Hypertension) ~ Disease associated with the atrophy of the pulmonary blood vessels.

PAH (Pulmonary Arterial Hypertension) ~ Sub-category of etiologies that are known causes of pulmonary hypertension

PCWP (Pulmonary Capillary Wedge Pressure) ~ Pressure measurement taken during the right heart catheterization procedure used in the diagnosis of disease.

PT ~ Pulmonary Trunk

PVR (Pulmonary Vascular Resistance) ~ is the resulting measure of the TPG divided by the cardiac output. A value greater than 2.5-3 Woods units is indicative of disease.

RMPA ~ Right Main Pulmonary Artery

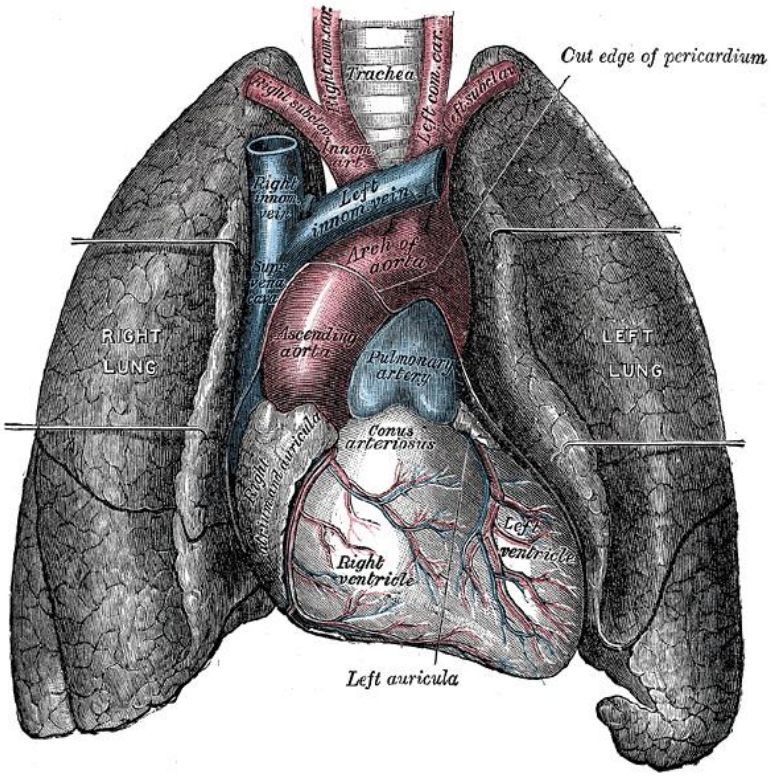
RV (Right Ventricle) ~ the anatomical location in the heart whose health is associated with determining stage of disease.

TPG (Trans-pulmonary Gradient) ~ the difference between the MPAP and the PCWP, a value <15 mmHg indicates the presence of disease.

2.0 BACKGROUND

This chapter is a review of background relevant to both the problem and the proposed solution. An overview is given of PAH, related diagnostic techniques including both imaging and non-imaging tests, and the basis of the algorithmic approaches for the vascular analysis.

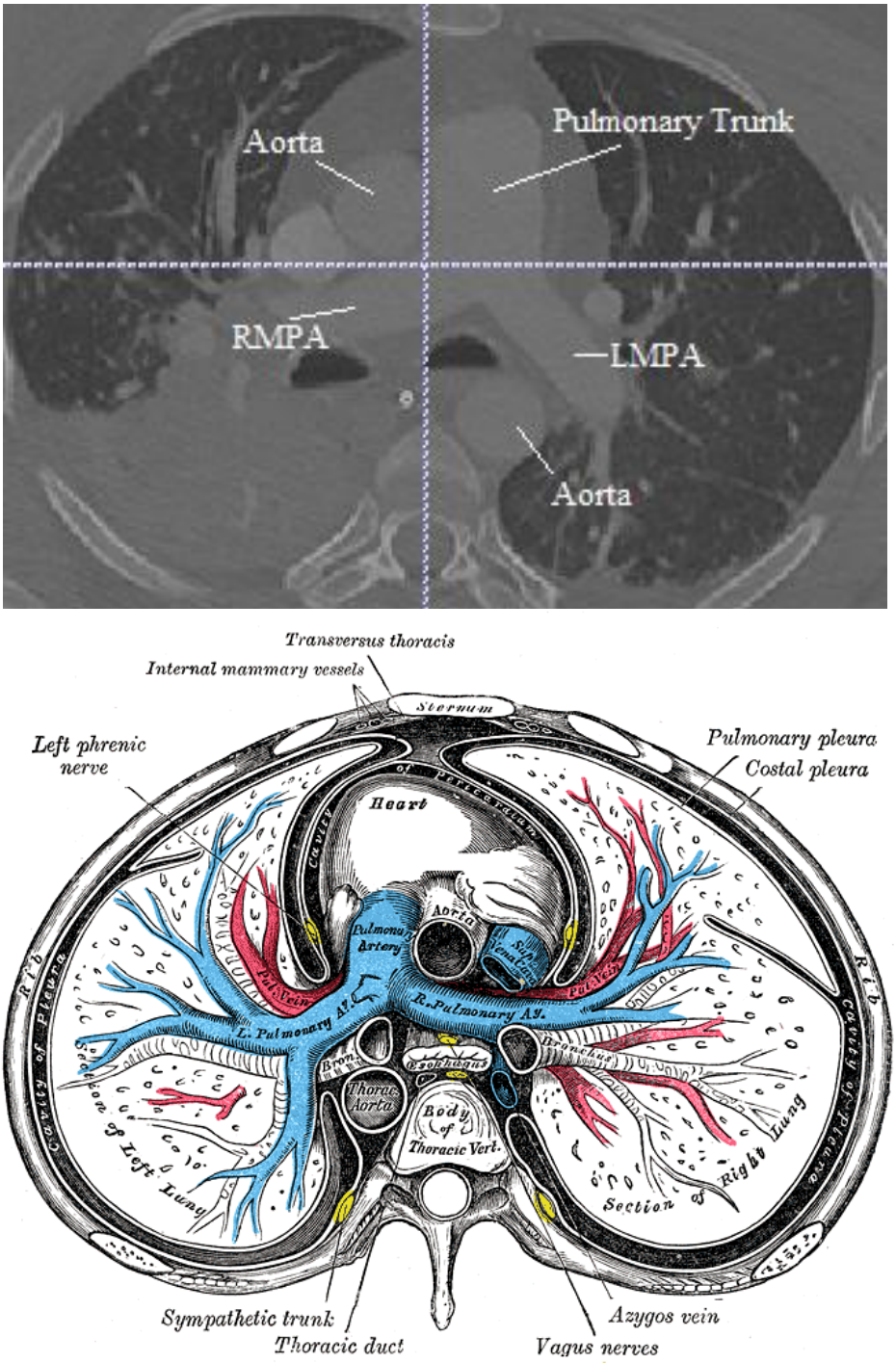
Figure 1. Diagram of the Heart and Lungs as a Reference Guide



**This image is from http://www.bartleby.com/107/138.html and is in the public domain because its copyright has expired.*

2.1 ANATOMY OF THE PULMONARY VASCULATURE

A widely used nomenclature of the pulmonary arteries is as follows. The first order artery is the pulmonary trunk, which then bifurcates into the second order arteries, the left and right main pulmonary arteries (LMPA and RMPA respectively). The RMPA is longer than the left and travels down toward the right lung, passing beneath the aortic arch before bifurcating into the third order arteries or upper and lower trunks also known as lobar arteries. The LMPA is a continuation of the pulmonary trunk and travels over the left main bronchus before splitting into the lobar arteries. The fourth order arteries are the segmental arteries whose branching pattern follows that of the segmental bronchi, with a total of ten segmental arteries on the right and eight on the left. The fifth order arteries are the subsegmental arteries, which arise directly from the segmental arteries. The final order to be discussed is the sixth order which are the arteries that arise directly from the first divisions of the subsegmental arteries [9]. This research is focused on the first and second order arteries only due to the difficulties with the segmentation process that will be described in detail in following sections. Figure 2a is a screenshot of an examination used in this study with the pulmonary vasculature anatomy labeled as used in this research. Figure 2b is a drawing reproduced from “Gray’s Anatomy” showing the pulmonary anatomy.



**This image is from "Gray's Anatomy" and is in the public domain because its copyright has expired.*

Figure 2a) Computed Tomography Exam with Labeled Pulmonary Vasculature, 2b) Drawing of the pulmonary arteries showing the branching of the arteries.

2.2 OVERVIEW OF PULMONARY ARTERIAL HYPERTENSION

Pulmonary hypertension (PH) is a complex disorder with many etiologies which are divided into five main categories. Table 1 outlines these categories in the revised World Health Organization (WHO) classification of PH [10]. According to the current definitions the focus of the research presented here is on pulmonary arterial hypertension (PAH) (WHO Group 1 in Table 1.).

In 1951 Dr. David Dresdale coined the phrases "primary" and "secondary" pulmonary hypertension. These terms have since been replaced with pulmonary arterial hypertension (PAH, WHO Group 1) and PH resulting from other causes (WHO Groups 2-5), respectively (ACCF/AHA 2009).

Pulmonary arterial hypertension (PAH) is defined as abnormal elevation of pressure in the mean pulmonary artery pressure (MPAP) > 25 mmHg at rest or > 30 mmHg during exercise [11] and pulmonary venous pressure (also defined as left atrial or left ventricular end-diastolic pressure) ≤ 15 mm Hg. It is, by definition, an abnormality of the pulmonary arterial vascular bed. In a healthy, normal patient, the pulmonary vascular bed can handle the volume of blood passed through the pulmonary arteries from the right ventricle at a normal resting flow of approximately 5 L/min at rest and up to 30 L/min with exercise. There is low resistance to the blood flow, and when the volume of the blood increases, the vessels dilate in compensation. In patients with PAH, the vascular bed is deteriorated and the ability for the pulmonary vessels to compensate for increased flow or volume is impaired, resulting in an increase in the arterial pressure. In relation to the heart this means that when the pressure is elevated for a prolonged amount of time the right ventricle hypertrophies in order to compensate for the increased resistance. Over time the right ventricle will dilate and eventually fail [1].

The prognosis of PAH is poor; there is a 15% mortality rate for the first year with modern therapies. Median survival without treatment is 2.8 years with 1-, 3-, and 5-year survival rates of 68%, 48%, and 34%, respectively [7]. With continuous prostanoid treatment this has improved to be 87-88%, 63-71%, and 56%, respectively [12, 13]. Similar results have been seen with the oral endothelin receptor antagonist bosentan monotherapy (82-96%, 67-89% at 1 and 2 years) [14]. Characteristics used in predicting a poor prognosis include advanced functional class, poor exercise capacity measured by the 6-minute walk test, high right atrial pressure, significant right ventricular dysfunction, evidence of RV failure, low cardiac index, elevated brain natriuretic peptide, and an underlying diagnosis of scleroderma spectrum of diseases (i.e. diseases that involve the fibrosis or hardening of the skin that can affect all areas of the body including the organs) [10, 15]. RV function in particular is a critical determinant of patient outcomes in PH, causing at least half of all PH deaths, and has recently been recognized as an important avenue for further research [7, 16]. However, current markers of RV failure that have been associated with poor outcomes only recognize end stage disease. Identifying which patients will progress to RV failure and at what time in the course of disease has been difficult.

Table 1. Revised WHO Classification of PH [10].

Main Categories	Sub-Categories	Specific Etiologies
1. Pulmonary Arterial Hypertension (PAH)	1.1. Idiopathic (IPAH) 1.2. Familial (FPAH) 1.3. Associated with (APAH) 1.4. Associated with Significant venous or capillary involvement 1.5. Persistent pulmonary hypertension of the newborn	***1.3.1 APAH causes:*** Connective tissue disorder, Congenital systemic-pulmonary shunts, Portal hypertension, HIV infection, Drugs and toxins, Thyroid disorders, Glycogen storage disease, Gaucher's disease Hereditary hemorrhagic telanglectasia, Hemoglobinopathies, Chronic myeloproliferative disorders, splenectomy ***1.4.1 Causes:*** Pulmonary veno-occlusive disease (PVOD), Pulmonary capillary hemangiomatosis (PCH)
2. Pulmonary Hypertension with Left Heart Disease	2.1 Left-sided atrial or ventricular heart disease 2.2 Left-sided valvular heart disease	
3. Pulmonary Hypertension associated with lung diseases and/or hypoxemia	3.1. Chronic Obstructive Pulmonary Disease 3.2. Interstitial Lung Disease 3.3. Sleep disordered breathing 3.4. Alveolar hypoventilation disorders 3.5. Chronic Exposure to high altitude 3.6. Developmental Abnormalities	
4. Pulmonary Hypertension due to Chronic Thrombotic and/or embolic disease (CTEPH)	4.1. Thromboembolic obstruction of proximal pulmonary arteries 4.2. Thromboembolic obstruction of distal pulmonary arteries 4.3.Nonthrombotic pulmonary embolism	***4.3.1 Causes:*** Tumor, parasites, foreign material
5. Miscellaneous		Sarcoidosis, Histiocytosis X, Lymphangiomatosis, Compression of the pulmonary vessels (adenopathy, tumor, fibrosing meadiastinitis)

2.2.1 Diagnosing PAH

Pressure values obtained from right-sided heart catheterization are the gold standard in PH diagnosis [1, 17-19]. Pressure measurements such as the mean pulmonary artery pressure (MPAP), pulmonary capillary wedge pressure (PCWP), trans-pulmonary gradient (TPG = MPAP - PCWP), and pulmonary vascular resistance (PVR = TPG / cardiac output) are how the disease is defined and diagnosed. A PCWP < 15 mmHg, a PVR value greater than 2.5-3 Woods units and a TPG > 10 mmHg are all indicative of PAH. Heart catheterization is an invasive procedure with known complications [17-19]. According to a study by Hoeper et. al., out of 7,218 right heart catheterization procedures there were 76 (1.1%) adverse events and four (0.055%) fatal events [20]. In regards to the study they report the more frequent complications were related to venous access and arrhythmias [20]. Other common complications include infection, bleeding and pain at the catheter insertion point [21]. Damage to the blood vessels can occur while the catheter is being threaded to the heart. The catheter may cause a hole or scrape the vessel along the way. However, this is a rare event [21]. Other less common complications include arrhythmias, blood clots, low blood pressure, and a buildup of fluid in the pericardium from perforation of the heart, and pulmonary artery rupture [20, 21]. Incidence rates for these complications were not reported in the referenced studies; however, although they are "common" complications they are also rare events.

In addition to the health related complications there is also the issue of expense. Heart catheterization is a costly procedure, reducing the need for this test would lower the expense involved in obtaining the diagnosis.

Studies have been done to identify less invasive procedures for the diagnosis of PH, in particular using computed tomography (CT) of the chest. One such study is by Grubstein et al. [22]. They wanted to "assesses the capability of computed tomography angiography and high resolution CT to diagnose and estimate the severity of pulmonary hypertension as compared with standard means of right heart catheterization, echocardiography and pulmonary function tests" [22]. They measured the diameters of the ascending aorta, main pulmonary trunk and the right and left main arteries. They then compared their findings to their control group and concluded that the "size of the main pulmonary artery on CT angiography has a good predictive value regarding the severity of pulmonary hypertension." It has also been reported by Grubstein et al. that there may be a "correlation between the ratio of the main pulmonary artery ascending aorta diameters and the pressure measurement by right heart catheterization [22]." A key component of this dissertation research is to understand the relationship of the sizes of these vessels and the physical changes they go through when affected by PH. Another group, Engelke et al., generated a review article "High-resolution CT and CT Angiography of Peripheral Pulmonary Vascular Disorders." In their review they reported that "the correlation between pulmonary artery dilation and the degree of pulmonary hypertension at CT angiography is nonlinear" [23]. It also mentions that for adult patients a diameter of 29 mm for the distal main pulmonary artery at its widest point has a positive predictive value of 95%, and if the width exceeds the diameter of the ascending aorta then it also has a positive predictive value of 95% [23]. Ng et al. performed a study to determine if the ratio of the main pulmonary artery diameter to the aortic diameter can be predictive of PH using CT exams [24]. They performed a series of analyses incorporating age as a variable in their analysis. They concluded that if the patient is younger than 50 years of age, there is a strong correlation between the pulmonary artery radius and the mean pulmonary artery

pressure in a heterogeneous study. They conclude that a ratio greater than 1 is indicative of PH [24]. Another group Edwards et al. report that patients with a main pulmonary artery diameter greater than 3.32 cm is indicative of PH [25].

Up to this point all of the referenced studies have performed their analysis using manual measurements of the vasculature by experts. Our approach is a semi-automated method of measuring the necessary vasculature, using manual calculations as a way of checking the validity of our model. Devaraj et al. [26] performed a comparison of the CT measurements and the mean arterial pressure using linear regression, followed by a multivariate regression to establish a pressure index and lastly they ran a logistic regression and ROC analysis to test the diagnostic ability of the CT-echocardiography composite. They concluded that the combination of "CT and echocardiographic markers of PH is more closely related to the mean arterial pressure than either test" alone [26]. In this study we will be performing a similar regression-based analysis using CTPA images and pressure measurements gathered from right heart catheterization tests. The underlying goal of these studies is the same as ours: to find alternative, less invasive yet reliable methods for the diagnosis of PAH.

Another related area of research focuses on the distensibility of the pulmonary artery. The difference in size of the pulmonary artery is measured between systole and diastole during the cardiac cycle. The group Abel et al. in a study titled “Pulmonary artery and right ventricle assessment in pulmonary hypertension: correlation between functional parameters of ECG-gated CT and right-side heart catheterization” reported that “Pulmonary artery distensibility was significantly correlated to mPAP” [27] or mean pulmonary artery pressure.

While the studies listed here are not exhaustive, they provide a representative sample of the studies that have been reported.

For the detection of PH from visualization of the pulmonary arteries, most radiologists concentrate on the dilation of the main pulmonary artery both in absolute terms and in relation to the ascending aorta [23, 26]. Another sign of PH is the tapering of peripheral arteries [23]. It has been suggested that using computer aided detection (CAD) to assist radiologists may reduce the number of false negatives [28].

2.3 IMAGING TECHNIQUES USED IN DIAGNOSING PH

This section provides a review of the imaging tools that are currently available. Although the focus of this dissertation is on the use of CT images all of the modalities discussed can be used in the detection of PH in some capacity and may have the potential for similar studies for the diagnosis of PH. The field of medical imaging is just over 100 years old, however, the concept has been around for centuries [29-33]. The purpose of the field is to allow the clinician to see inside in the patient to diagnose a condition by visualizing the data obtained directly from the patient. The main focus of medical imaging is to improve the quality of the images for radiologists to evaluate. A Czech mathematician, Johann Radon derived a transform for reconstructing cross sectional information from a series of planar projections from around an object, also known as a 3D image [30, 31]. Although these theories have been known for over fifty years, it was not until the 1970's when digital computing was finally powerful enough to create images from the data that cross sectional imaging became prevalent in medicine [31].

2.3.1 X-Ray

Wilhelm Conrad Roentgen accidentally discovered x-rays in 1895 while experimenting with electron beams. While experimenting, he noticed a fluorescent screen in his lab began to glow when the beam was turned on. This was expected to occur when fluorescent materials were exposed to electromagnetic radiation. However, he attempted to block the radiation with cardboard and the glowing screen was not expected. Roentgen experimented by placing a number of different objects between the tube and the screen in an attempt to prevent the screen from glowing. He then placed his hand between the two and the silhouette of the bones in his hand could be seen on the screen. Roentgen immediately recognized the value of using x-ray radiation for imaging and the positive effects it could have on the medical community [30, 31, 33]. Six years after announcing his achievement he received the Nobel prize [30, 31]. X-rays were denoted as such by Roentgen because the type of radiation that was occurring was unknown [29, 32]. X-rays are a form of ionizing radiation and long term exposure can be detrimental to one's health. Health issues can arise if 1) the rays carry enough energy that when they collide with atoms within the patient's body they cause electrons to detach from those atoms resulting in free floating negatively charged electrons and or positively charged ions, and 2) the exposure to the rays is long enough [34]. The heart of an X-ray machine is an electrode pair made up of a cathode and an anode housed in a glass vacuum tube. The cathode contains a filament that is heated by passing current through it. The heat causes negatively charged electrons to be ejected from the filaments surface. The positively charged anode, composed of a flat tungsten disc, pulls the electrons through the tube. There is a large voltage difference between the cathode and anode causing the electrons to fly through the tube with a great deal of force. This process creates x-rays in two ways. The first is the photoelectric effect, which occurs

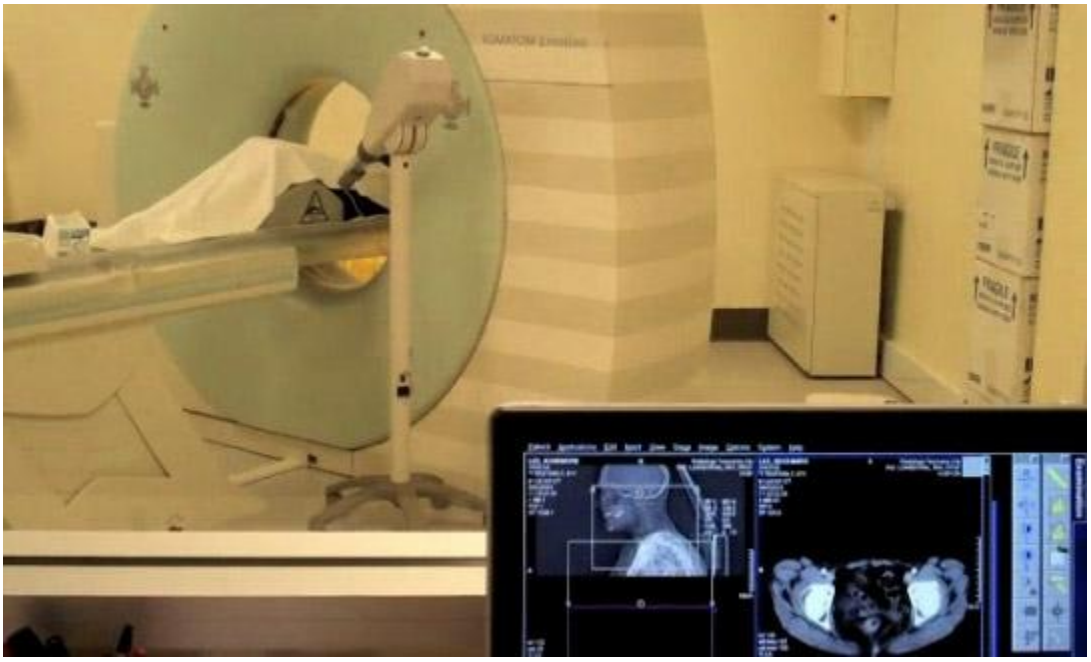
when the electrons collide with the tungsten atoms of the anode with such force that it kicks an electron off the lower orbital of the tungsten atom. To stabilize the tungsten an electron from the higher orbital must drop down to fill the place of the ejected electron. This drop from a high energy level to the low level causes a release in energy in the form of an x-ray photon. The second way is due to Compton's Scattering, which occurs when the tungsten atom's nucleus attracts the moving electron just enough to change its course. The electron slows down as it speeds past the nucleus to change direction. The action of slowing down causes the electron to emit energy in the form of an x-ray photon [34-36]. To combat the heat that these processes generate the machine contains a motor that rotates the anode to prevent it from melting and immerses the anode in an oil bath to diffuse the heat. The entire machine is surrounded by a thick lead shield to prevent the x-rays from escaping in all directions. A small window in the shield allows some of the x-ray photons to escape in a narrow beam. The beam then travels through a series of filters on its way to the patient. A detector placed on the opposite side of the patient records the pattern of x-ray light that passes all the way through the patient's body [34, 36].

2.3.2 Computed Tomography (CT)

Ronald Bracewell and William Oldendorf were the first two pioneers of CT imaging. They gathered their data and in 1955 Bracewell was able to reconstruct a two-dimensional map of a solar image using Fourier transforms. In 1960 Oldendorf lacked the computational tools to interpret the quantity of data he would need to generate the images but was able to demonstrate how to reconstruct a two-dimensional display of images and the beginnings of the CT machine [31]. The main goal of Oldendorf's work was to determine whether internal structures within dense structures could be identified by transmission measurements [37]. It was not until the

1970's when digital computing was finally powerful enough, to create images from the data [30]. The CT scanner began as a crude instrument in 1971 in London by Godfrey Hounsfield [31, 38]. It was installed in Atkinson Morley Hospital[38]. However, three-dimensional imaging was not seen until 1972 when X-ray computed tomography (CT) was independently developed by Godfrey Hounsfield and Alan Cormack. For their discovery they shared the Nobel prize in medicine in 1979. Their primary contribution was the engineering aspect that proved what had previously been theorized on paper by the scientists that came before. These systems where patented in 1975 and began being used shortly thereafter [30]. A CT scan is developed using x-rays to generate tomographic images that "slice" through the body. ("Tomos" is Greek for slice or section.) In a typical CT, the technician first places the patient on the table and performs what is called a "scout view", a 2D image of the patient. The goal is to learn anatomical landmarks and to determine the exact location and range of the CT scans. Once the scout scan is completed, the computer system sends instruction to the table, the x-ray tube and detector. The table travels horizontally at a constant speed while both the x-ray tube and detector remains stationary. See Figure 3, the patient is lying on the table as it moves through the opening in the machine that houses the x-ray tube and detector. The high voltage generator achieves the desired voltage and keeps both the voltage and the current to the x-ray rotation tube at the predetermined level. The x-ray tube produces a fan shaped beam of x-rays directed at the patient and the resulting x-ray photons that exit the patient's body are picked up by the detectors. Occurring simultaneously, the data acquisition system is uniformly sampling the detectors outputs and converting the analog signals to digital signals. The data are then preprocessed and enhanced before viewing and storage [37, 39].

Figure 3. An Example of a patient undergoing a CT exam



*This image is open source licensed under the Creative Commons Attribution 3.0 Unported license

CTA is when CT is used to specifically image the arteries. In conventional angiography, a catheter is threaded through the groin to the vascular structure to be imaged where a contrast media is directly injected. In CTA or CTPA (Computed Tomography Pulmonary Angiography) a catheter is inserted into a vein in the arm where the contrast media is injected. By using a venous injection, the cost and risks of the procedure are dramatically decreased. The contrast media, typically an iodinated material, has a high CT number, allowing the arteries to be well differentiated from any surrounding soft tissue [40].

Interpreting or processing CT scans can become difficult due to the presence of artifacts in the image. A CT artifact is any discrepancy between the reconstructed values in the image and the true attenuation coefficients of the patient, the value quantifying how easily the x-ray beam passes through the tissue. There are four major categories of artifacts: 1) **streaking**, appears as

straight lines going across the image in some direction; 2) **shading**, appears as shadowing near objects of high contrast and can mimic pathology and lead to misdiagnosis; 3) **rings and bands**, appear as rings and bands that are superimposed on the original image structure; 4) **miscellaneous**, covers all of the other artifacts that are not as common. Artifacts can be a result of a failure in the system design, the x-ray tube, the detector or a problem related to the patient [37]. The artifacts of issue in relation to PH are when they appear as streaks through the image or vessel beading, or when there is beading in oblique vessels resulting from partial volume artifacts that appear more severe in peripheral vessels [23]. The presence of artifacts can disrupt the segmentation process. The segmentation algorithm may not be able to differentiate the artifact from the vasculature and mark it as a vessel or it may remove it all together resulting in an inaccurate segmentation and image that will be difficult to use in analysis.

2.3.2.1 Artifacts Common to CT Images

Artifacts Related to System Design

Aliasing is a type of artifact that can occur due to sampling errors. Sampling is the first step when gathering the data to compile a CT image. The detectors sample the continuous wave of x-ray photons to gather signals that represent the current view for that slice. The sampling occurs in groups categorized by time to divide the views into slices. There are rules in place to prevent the production of artifacts, however, aliasing can occur when the data are not sampled at a rate that is twice as high as the highest spatial frequency within the dataset. Let us say Y is the spatial sampling distance that is equal to the detector channel spacing; the difference in frequency absorbed between adjacent detectors. The sampling is occurring at the rate of 1/Y. If

the fraction of the spatial sampling distance is too small, where sampling areas are overlapping then the signals will overlap as well causing the development of aliasing artifacts [37, 41].

Partial volume is another type of artifact that can occur; it happens when the object is partially protruded into the scanning plane. The thicker the slice the more likely partial volume will occur. Due to the way the x-ray beam diverges into the z-direction (the Cartesian coordinate in a three dimensional space). The effect of the partially protruded object is angular dependent. When the CT scanner rotates so that the patient is closer to the detector the x-ray beam profile is wide enough that the patient is within the field of view. However, when it rotates to the opposite side, part of the patient will be out of the beam path creating inconsistencies in the data resulting in shadowing and streak type artifacts [37].

Scatter is another cause of artifacts related to the effects of Compton's scattering. Due to Compton scattering not all of the x-rays that reach the detector are primary photons. Depending on how the CT system is designed a portion of the detected signal can be generated from the scatter. The scattered photons cause the detected signals to deviate from the true measurements of the x-ray intensities and will result in shading, streaks or number shift type artifacts [37].

Noise induced streaks are artifacts that result from the amount of x-ray photons that are generated from the x-ray tube. Excessive photon noise will cause severe streaking in the resulting image [37]. The presence of high attenuation objects is often the culprit in this type of artifact. Often the objects responsible are metal implants such as joint prosthesis, surgical clips and dental fillings [42]. This can also happen when the parameters are not properly selected. The main causes of these types of artifacts are due to the limitations of the CT system itself. For instance, if the patient is larger and not enough x-ray photons can be produced to generate a clear

image of the patient using thin slices. This is the limit of the scanner and can sometimes result in unusable images [37].

Artifacts Related to the X-ray Tube

One cause is ***off-focal radiation***, which is caused by secondary electrons and field emission electrons. When the high speed electron crashes into the target, electrons are emitted from the site of impact. Most of these electrons are high velocity and return to the target at a point outside of the focal point and produce x-ray photons at their point of impact. Therefore, the x-ray source will have a high intensity center spot surrounded by a low intensity halo. This halo area will cause degradation to areas that are of low contrast and shading artifacts will develop that can sometimes render the image useless or mimic pathology leading to a misdiagnosis [37].

Tube-arcing or ***tube split*** will occur if there are impurities in the tube resulting in a temporary short circuit. When this occurs there is a significant increase in current and decrease in voltage. The CT system has a mechanism built in to detect this type of phenomenon, when it is detected the power supply is turned off to prevent further arcing. After a period of time the x-ray tube returns to its normal functioning level. However, the time in between there is a noticeable decrease in the production of the x-ray photons. If this is an isolated event there are methods built into the data acquisition and reconstruction to handle this situation, but if this occurs again the scan will be stopped to prevent unnecessary exposure to the patient and generating a degraded, useless image [37].

Tube rotor wobble is another class of issues that can create artifacts related to the x-ray tube. This is a group of mechanical failures that can result from a lack of rigidity in the gantry, mechanical misalignment, or x-ray tube rotor wobble. All of these have the same effect resulting in streaking artifacts across the images. The cause for this event is when the x-ray beam position

deviates from the position assumed by the reconstruction algorithm. These effects result from the high rate of speed at which the rotor shaft spins. The rotor shaft spins to keep the amount of heat generated by the electron beam down, over time this system wears down and needs to be replaced or artifacts of this nature will be seen [37].

Detector-Induced Artifacts

Offset, gain, nonlinearity and radiation damage all lead to ring artifacts. All of these issues are due to malfunctions with the detector. Another issue deals with primary speed and afterglow. When we expose a patient to the x-ray beam once shut off there are still excited electrons that are passing through and the detector output will not reach zero right away. The amount of time taken for the detector to stop receiving a signal is the afterglow, while the amount of time it takes for the signal to decay with short time constants is the primary speed. The primary speed affects the spatial resolution of the images and the afterglow usually affects image artifacts. Finally there is the uniformity of the detectors response. We assume that the detectors response does not change with where the x-ray photon hits the detector. When this is not the case some of the detector responses are significantly different from that of their neighbors. This can be caused by a change in the reflective material placed between the detectors, mechanical stress, radiation exposure, or age.

Patient-Induced Artifacts

When a patient moves either voluntarily or involuntarily during a scan artifacts can be created. The patient could move in and out of the scanning plane, or they could move within the scanning plane. Another occurrence that causes artifact generation is called beam hardening, which can be a result of the polychromatic x-ray beam spectrum and energy dependent attenuation coefficients. Low energy photons have high energy attenuation coefficients and high

energy photons have low energy attenuation coefficients. When the beam, which is a mix of these two classes of photons, is sent through soft tissue most of the low energy photons are absorbed leaving mostly high energy photons. Due to this phenomenon the issue has been coined "beam hardening". These types of artifacts can be reduced through the use of adequate beam filtration. Another cause of artifacts is when the patient has metal object within their body. These types of artifacts will vary in size and shape significantly based on the type of object the patient has within their body. Finally the last cause for patient induced artifacts is incomplete projection which is simply when part of the image is not able to be reconstructed. This will occur when the patient is not entirely within the scan field of view. This happens more often than not because the opening of the gantry is much larger than the scan field of view. If the patient is not appropriately placed then the image will be truncated. The areas where truncation does occur near the edges there will be bright shading artifacts.

The advantages of using CT scans are that they are fast and can scan bone and cartilage. They also improve contrast resolution and decrease structural noise. The disadvantages are that since CT scans also use x-rays they also have ionizing radiation, they have limited tissue definition and application and are expensive [31, 38].

2.3.3 Lung Perfusion Ventilation Scan

A lung perfusion ventilation scan or VQ scan, measures both the oxygen and blood flow in the lungs. This test involves two types of scans. The ventilation scan shows the air flow in the lungs and the perfusion scan shows the blood flow. Both the ventilation and the perfusion scans use radioactive isotopes, a type of nuclear medicine. The isotopes are either inhaled for the

ventilation scan, or injected into a vein in the arm for the perfusion scan. Changes in the flows may be a symptom of pulmonary hypertension[40].

2.3.4 Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging (MRI) uses magnetic fields and radio waves to create images. Tools such as x-rays expose patients to ionizing radiation that can create free radicals or break chemical bonds. MRI employs a method that does not affect the patient's body chemistry but instead records information collected from the movement of the atoms deep in the nuclei of the cells[43]. MRI uses magnetic fields and radio waves to create images. The first human MRI took 5 hours to generate and was performed on July 3, 1977. It took seven years for Drs. Raymond Damadian, Larry Minkoff, and Michael Goldsmith to reach the point of generating practical images. The main obstacle for MRI scanners was having the ability to generate a strong enough magnetic field to create the image. For this reason the maturity of the MRI remained stagnant until engineers were able to build super conducting magnets capable of creating an appropriate magnetic field [30, 31]. Patients are placed on the bore of the scanner and inserted into the magnetic field. To prevent interference from outside, the magnet is housed in large masses of iron surrounded by a Faraday cage. When the patient is exposed to a high magnetic field the hydrogen atoms within the patient have a strong tendency to line up with the direction of the field [30, 43, 44]. The hydrogen atoms are targeted because its nucleus has a single proton and a large magnetic moment. When subjected to the field, the hydrogen protons in the patient will line up in the direction of either the feet or the head. The majority of these protons will cancel each other out, for each one lined up toward the feet, one toward the head will cancel it out. Only a couple of protons out of every million will remain. Once aligned a pulse of low-level

radio waves is directed through the patient. The pulse causes the protons in that area of the body to absorb the energy required to make them spin at a particular frequency, in a particular direction [30, 44]. The specific frequency of resonance is called the Larmor frequency and is calculated based on the particular nuclei being imaged and the strength of the main magnetic field. The goal is to spin more atoms in the opposite direction, which is the higher energy state, than the amount that remain in the low energy state. For this to occur the pulse must be in resonance with the Larmor frequency [44].

The pulses are usually directed through the use of a coil which can be specific to a part of the body. There are three gradient magnets arranged inside the main magnet so that when they are turned on and off very rapidly in a specific manner, they alter the main magnetic field on a very local level. This gives the medical professionals the ability to choose exactly which area of the body to image. An MRI is able to generate a slice from any part of the body in any direction without moving the patient or machine [30, 44].

When the pulse is turned off, the hydrogen protons begin to return to their original alignments within the magnetic field and release their excess stored energy to return to that lower energy level. When this happens, they give off a signal based on their spin densities that the coil now picks up and sends to the computer system. What the system receives is mathematical data that is converted, through the use of Fourier transforms into an image. Image reconstruction uses Fourier transforms to convert the information gathered into image information that represents the two dimensional distribution of the spin densities [44].

The resulting MRI images can be used in the same manner as the CT images for detecting PH. In fact, the methods outlined and described in this dissertation can be directly applied to MR data, CT data was only used because of its availability.

2.3.5 Ultrasound

Ultrasound is unlike any of the other imaging techniques previously discussed. It produces real time images, does not use ionizing radiation, and is fairly inexpensive. It utilizes a piezoelectric quartz crystal to create high frequency acoustic energy in the range of 3 to 10 Megahertz. This energy is then reflected off the surfaces of the organs within the body. The transducer is the same device used to create the signal and is also used to measure the returning echo information. Ultrasound machines typically contain a linear array of transducers and produce an image that is a pie shaped slice of the patient's body [30]. The problem with ultrasound in regards to identifying pulmonary disease is that the images retrieved are not clear in part due to respiration [31]. Ultrasound imaging relies on transmission of sound waves and therefore the signal is degraded when the beam has to travel through air (lungs) as compared to through solid tissue. Conditions of increased lung volumes (such as COPD) exacerbate this.

In 1949 Douglas Howry built the first rudimentary ultrasound machine using Navy surplus SONAR equipment and bomber parts. Employing self-experimentation he focused the machine on his thigh and obtained an image of the tissue. He built a second machine that was more sophisticated than the first that he called a "somascope." It later evolved into B-mode imaging which is a way of obtaining two-dimensional images by recording sound echoes from tissue. By hand he measured the distances from the surface of the body to each organ and using these measurements constructed a compound picture from the cross-sectional images. He continued to refine his work along with other groups working in parallel over the years. When their techniques were merged with digital computers in the 1970's the ultrasound devices of today were created [31].

Some advantages of ultrasound, as mentioned earlier, are that it is inexpensive, functions in real time, and is non-invasive. The drawbacks are that the image is not clear, is very operator dependent, and has difficulty imaging the areas around the lungs [31].

2.3.6 Nuclear Medicine

Nuclear medicine uses radioactive solutions injected into the patient and images how the solution is distributed through the body using a detector array to catch and quantify the amount of radiation being emitted from the body. The main difference between this system and the CT scan is that the source of radiation for the CT is external and the distribution of it is known whereas, for nuclear medicine the source, which is the solution, is internalized and the distribution is not known [30].

In 1952 the first positron scanner was used to image brain tumors in patients. In 1954 the "photoscan" was invented by David Kuhl, he created the first transmission image of the lungs of a naval student. However the connection between transmission and emission was not yet validated. Emission images began with the first SPECT machine by Kuhl in 1968. At the time it was the most inexpensive and convenient imaging technology available. In 1973 Ter-Pogossian and his group created the first positron imaging device and nicknamed it the "lead chicken." In 1983 radioactive tracers pioneered by George Hevesy were first used in PET scans to record and map the functioning of sight. Finally in the mid 1980's PET scanners and the radiopharmaceutical manufacturing reached the point where exploration stopped and simplification and fine tuning has begun [31].

The images generated in nuclear medicine are often hard to distinguish and have poor spatial recognition; they usually have low resolution and a high amount of noise. A benefit of

nuclear medicine is that it allows the physician to capture the physiological activity of the patient and not just the locations of the patient's anatomy. The choice of the radioactive solution will affect the quality of the resulting image. Radioactive iodine is very good for generating images of the thyroid and parathyroid glands. Radioactive tracers can be attached to numerous different substances. The image quality is due to our inability to use high radiation doses because of the detrimental effects it will have on the patients [30].

2.4 MEDICAL IMAGE PROCESSING

Medical imaging systems are tools used by medical professionals to aid in diagnosis and care for a patient. The data gathered by these systems are often in the form of 2D images, composed of data arrays that can be organized using two Cartesian dimensions. This information must be preprocessed prior to reaching the clinician in order to make informed decisions. The process of diagnosing and treating a patient in regards to imaging occurs in three main steps. The data must be first preprocessed, meaning filtered, remove the noise and the features enhanced. Next the information is either given to the clinician where the condition or event can be detected or it is first identified by the machine and then passed on to the clinician. The final step is the quantitative analysis of the detected condition. There are three problems that are inherent to medical image processing. They are filtering, segmentation and registration [30]. Only an overview will be given of the processing techniques; they will not be discussed in detail.

2.4.1 Filtering

This includes the procedures for preprocessing the data gathered by the medical imaging techniques previously described. This happens prior to when detection and analysis are performed. Filtering removes noise, enhances features and reduces any effects that occurred during image acquisition. Generally, a filter defines some neighborhood function. Median filtering for instance, used in this research, takes the 3x3x3 neighborhood of a point and replaces the point with the median value of the neighborhood [45]. Two more examples of filtering are 1) convolution which is a one dimensional operation that applies a filter kernel to an image to smooth it out and 2) Fourier transform which decomposes the image into component sinusoidal spatial functions for computational viewing and manipulation [30].

2.4.2 Segmentation

Segmentation is the process of breaking up the image or 2D array into regions that have cohesive properties. One example of segmentation is thresholding. It is a binary process where for each pixel a decision is made based on its intensity level independent of its neighbors. If the intensity is greater than some value the pixel is classified differently than if its value is below a certain level. For a CT image the intensity is reported in Hounsfield units and are calibrated to correspond to the attenuation of the X-rays measured within the tissue [30].

2.4.3 Registration

Registration is the process of transforming multiple data streams into one coordinate system to create a more powerful diagnostic tool [30, 46]. Multiple images taken from different perspectives, at different times and recorded by different sensors are overlaid on one another to create one master dataset. Registration contains four key steps: 1) feature detection, 2) feature matching, 3) transform model estimation and 4) image resampling and transformation [46].

2.5 EXTRACTING VASCULAR FEATURES

2.5.1 Vascular Segmentation

Studies have been done to find a solution to segmenting the pulmonary vasculature. The closeness of the vessel voxel signals between vessels and the tissue surrounding the lungs make thresholding difficult, as almost inevitably the vascular segmentation "bleeds" into these surrounding structures. Consequently, more involved segmentation techniques are needed. Masutani et al. developed a method for vasculature segmentation using 3D image analysis methods combined with anatomic knowledge [47]. They sequentially segment several anatomic structures using the properties of each structure for the next step in the segmentation and to validate intermediate results [47]. Masutani et al. were able to successfully segment the pulmonary vasculature however, they were not able to separate the pulmonary vessels from the heart and corresponding arteries [47]. Another group Zhou et al. developed a method of

segmentation for the identification of pulmonary embolisms. Zhou's group developed a multi-scale Hessian matrix-based filter that incorporates expectation-maximization (EM) analysis [48]. The EM analysis is used to separate the vasculature from the rest of the anatomy [48]. They then apply a series of filters for vessel enhancement similar to the method presented in this research. The method I used combines a series of mathematical morphological filters for isolating the vasculature and for enhancement of the vessels described in more detail in Section 3.2.1. Zhou et. al. only had 2 cases for development and only segmented the peripheral pulmonary vessels due to the difficulty in segmenting the larger pulmonary vessels from the heart. The research being presented segments the main pulmonary arteries with a few branching levels eliminating all other competing structures. The goals of all the segmentation methods are the same, to isolate the vasculature less any of the surrounding anatomy.

2.5.2 Mathematical Morphology

Mathematical morphology is used often to simplify the image for easier analysis by reducing the noise or generating skeletons (i.e. medial lines). For our purposes we apply these different methods throughout this research from segmenting the image, to cleaning up the image, followed by preprocessing and simplification of the segmented image into the skeleton that represents the vessel paths. Finally, we will use them to model the shape of the vasculature in 3D form. A skeleton in our case is a one voxel thick line that is medially located in all of the vessels of the vasculature.

Mathematical morphology includes opening and closing methods, such as the preprocessing methods used in Section 3.2.3. Opening and closing is also used for extracting information from the segmentation in the generation of the skeleton using the hit-or-miss filter or

parallel thinning algorithms discussed in Section 3.2.4. The two main methods of mathematical morphology are erosion and dilation, demonstrated in Figures 4a-d. Figure 4a is an example of erosion where it is shown that only when the origin of b is completely contained inside a are the pixel values retained, else they are deleted or eroded. We are assuming the origin of b is at the center[49]. Figure 4b is a demonstration of dilation the second key method in mathematical morphology. It is a dual operation to erosion (4c). Dilation (4d) expands the image to include the portions of the structural element that match the image (based on erosion) [49].

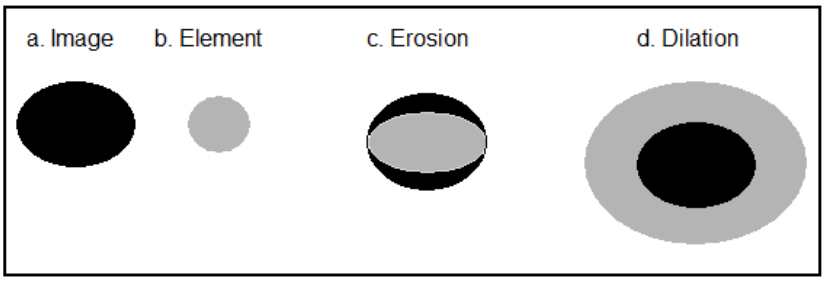


Figure 4a-d: An example of erosion and dilation:(a) is the original image data, (b) is the structural element, (c) is the eroded image and (d) is the dilated image.

Opening and closing are the two secondary classes of methods. Opening is when an image is first eroded and then that resulting image is dilated. Closing is the opposite: dilation followed by erosion; closing typically smooth's out corners and areas that are protruding into the image [49].

2.5.3 Machine Learning

Machine learning is the use of computational methods and algorithms to learn patterns and derive mathematical models from a dataset. .Machine learning is a powerful set of tools because it enables the algorithm being used to predict classifications for outputs for instances

that were not used for learning [50]. Author Tom Mitchell defines Machine Learning as *"the study of computer algorithms that improve automatically through experience.* Applications range from data-mining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests. [51]"

There are two common classes of machine learning methods: supervised and unsupervised learning. Supervised learning is when the training set is labeled with designated classifications. An example is the prediction of whether or not a patient has a specific disease based on a number of variables (e.g., blood pressure, hematocrit level and gender). The training set will have all the patients labeled as either positive for the presence of disease or negative for presence of the disease. This is different from unsupervised learning where there would be no designation of positive or negative for disease [50]. An example is the clustering of patients into similar groups. I used supervised learning methods in this research for addressing hypothesis 2.

2.5.4 Algorithms

There are two main classes of algorithms that can be applied, namely, classification and regression. Classification models group different data points into groups based on their attributes [50]. The underlying goal of all model building techniques is to find the best fitting and biologically reasonable model to describe the relationship between the outcome variable and the predictor variables [52]. In this section I give a brief overview of the four different classifiers used in this research.

Logistic and Polynomial Regression

A regression model fits a mathematical function describing some curve (e.g., logistic or polynomial) that passes as close as possible to all data points. This enables us to predict a target

variable y given some variable values x, even though x may not have been one of the instances used during the training of the data [50]. In logistic regression the outcome variable is binary, in contrast to linear regression where the outcome variable is continuous. In addition, the error distribution is not assumed to follow a normal distribution but instead is assumed to follow a binomial distribution.

CART (Classification and Regression Tree)

CART is a nonparametric method that uses either classification methods if the target variable is categorical or regression trees if the target variable is continuous.. It was first introduced by a group of researchers from Stanford in 1984 that included Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone. They defined three key components for generating a decision tree. First rules are defined for splitting the data based on the value for a particular variable (or feature). Second a rule must be defined for determining when the tree is complete, a stopping rule. Finally, each terminal or leaf node should be assigned a prediction or outcome value. During the decision tree process the data are recursively thinned down into more homogenous groups. The paths from each node to each leaf node are the resulting rules for assigning the outcome values [53]. One benefit of utilizing CART is that although large sample sizes are always preferred, when using CART accurate findings can be learned with smaller sample sizes. Secondly, assumptions such as the data fitting a normal distribution, linear relationships between independent and dependent variables and the data having the same variance are also not necessary for obtaining accurate results [54].

Naïve Bayes

The origin of the Naïve Bayes classifier is debated. However; it is named after Reverend Thomas Bayes who studied how to compute the distribution for the probability parameter of a

binomial distribution during the 18th century. He defines the fundamentals underlying a *general* Bayes classifier in the famous article "An Essay Towards Solving a Problem in the Doctrine of Chances" published in 1763. Bayes' theorem shown below, defines the relationship between a hypothesis *H* and evidence *E* [55, 56].

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} = \frac{P(E|H)P(H)}{\sum_{H'} P(E|H')P(H')}$$

The sum in the denominator in the third installment is over all hypotheses H' that are considered to have non-zero probability$P(H')$. A Naïve Bayes classifier is a specialization of the above general equation. In particular, a Naïve Bayes assumes that each piece of evidence e_i in *E* is independent of each other piece of evidence e_j in E given a hypothesis H orH'. Thus, we have:

$$P(H|E) = \frac{[\prod_i P(e_i|\mathrm{H})]P(H)}{\sum_{H'}[\prod_i (e_i|H)]P(H')}$$

3.0 METHODS

This chapter details the steps taken and methods used for completing the hypotheses described in section 1.2.

3.1 SPECIFIC AIM 1: CREATION OF THE DATA REPOSITORY

We collected 27 de-identified CTPA cases with PH state confirmed via right heart catheterization (22 positive and 5 control cases). The pressure data corresponding to these cases are summarized in Appendix A.

In addition to these 27 cases, we have an additional 94 CTPA cases that have been determined by a radiologist to be negative for PH as well as for any gross abnormality. These 94 cases resulted from a dataset that began with over 500 CTPA cases (de-identified dictated reports and DICOM images) that were ordered to rule out pulmonary embolism (PE). I reviewed the associated radiology reports for those 500 plus cases to ensure that there were no other gross vascular abnormalities, including PH. If cases were rejected based on the contents of the reports, then additional control cases were randomly selected from the database until 100 cases were collected that contained no gross vascular abnormalities. After the initial 100 was identified, one case was eliminated due to an incomplete CT exam, leaving 99 cases that we used for negative

PH cases or control cases. In the end we had a total of 22 positive cases and 99 control cases (5 controls from the initial 27 and 94 identified from the PE dataset).

3.1.1 Power Calculations

Power is defined as the probability of correctly rejecting the null hypothesis when it is false. We have 99 controls and 22 cases for all of the methods that do not involve the use of the pressure data. For methods involving the predictions using the pulmonary pressures we only have 22 cases and 5 controls to make up the dataset. For this power analysis the null hypothesis is that the mean pulmonary pressures of the disease versus control groups are equal. Our desired power is a minimum of 80%. To determine the power of the pressure dataset we chose the mean arterial pressure as the variable for calculation.

In a second power analysis, the null hypothesis is that the mean diameters of the right main pulmonary artery for the disease versus control groups are equal. This analysis is being performed because one of the prediction models focuses on whether or not we can classify patients by disease state. I chose the RMPA values over the pulmonary trunk and the LMPA because of the positioning in the pulmonary tree, the RMPA is deeper than the trunk and the automated measurements are better matched to the manual measurements when compared to the LMPA values.

I used an online power calculator (http://www.statisticalsolutions.net/pss_calc.php.) for determining the sample size needed for the experiments and, the results are provided in Section 4.1.

3.2 SPECIFIC AIM 2: CREATE AND VALIDATE PUMA

Now I will detail the methods and steps taken in the creation of PUMA (**PU**lmonary **M**apping and **A**nalysis tool), the tool we built for semi-automatically generating pulmonary vascular models that will be quantified in specific aim 3 for the diagnosis of disease.

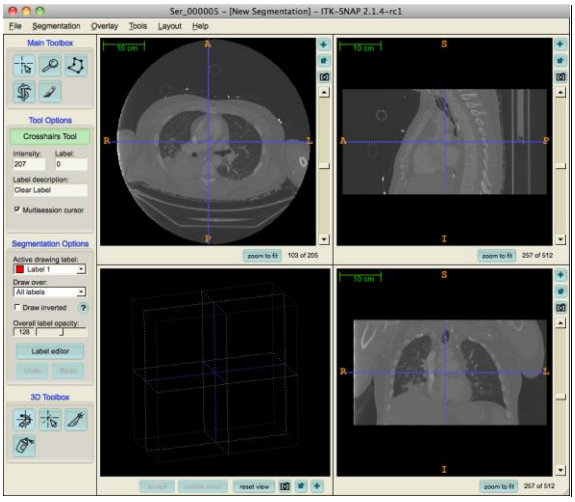
We built PUMA primarily using Python and leveraging a variety of existing tools including:

- Insight Toolkit (ITK), an open source image analysis toolkit developed by the NLM [57] using WrapITK, or more recently SimpleITK, to create Python wrappers.
- Visualization Toolkit (VTK), an open source visualization toolkit developed by GE, the NLM and others [58].
- NetworkX, an open source graph analysis toolkit developed at Los Alamos National Laboratory [59].
- Mayavi, a 3D visualization toolkit written in Python that incorporates VTK http://mayavi.sourceforge.net/index.html

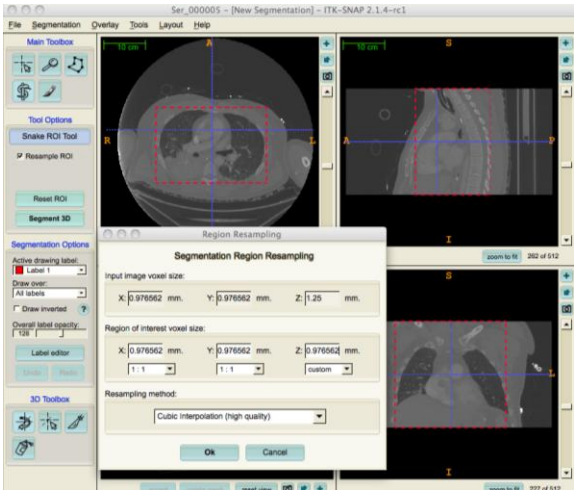
3.2.1 Pulmonary Vascular and Aorta Segmentations

Originally it was proposed that I would incorporate an existing automated algorithm to extract the pulmonary vasculature segmentation. However, due to the dissolution of a joint project I no longer had access to the existing algorithm. Instead I performed segmentations down to the segmental branches using a semi-automated approach on all the exams. Due to incomplete image files an additional three control cases were eliminated from the segmentation dataset. For each of the remaining 118 exams there were two segmentations one for the aorta and one for the

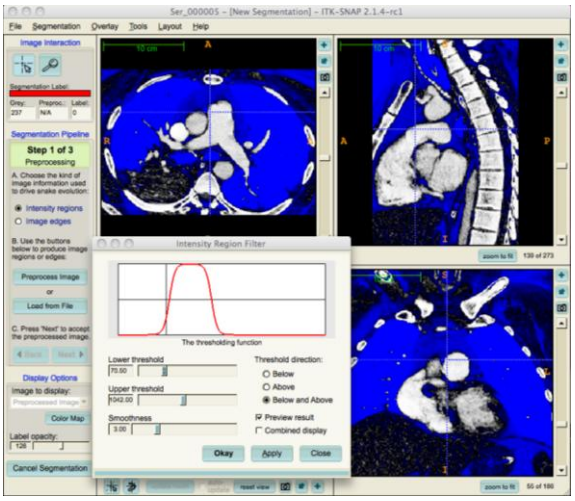
pulmonary vasculature, totaling 236 segmentations. I used ITK-SNAP [60] to generate the segmentations. ITK-SNAP is a software application based on the Insight Tool Kit [57], that is used for either manual or semi-automated segmenting structures in 3D medical images. Semi-automatic segmentation of the images is based on active contour methods[60]. Because the active contour segmentation requires the user to provide seed points and either intensity or gradient mappings, PUMA is now a semi-automated modeling method, although I believe that these semi-automated segmentations can form the basis for shape-based automated segmentation techniques. I segmented the aorta for each case because the measurement of the ascending aorta is used for normalization of the pulmonary vasculature measurements across patients using the same method as described in some of the studies in Section 2.2.1. [22-26]



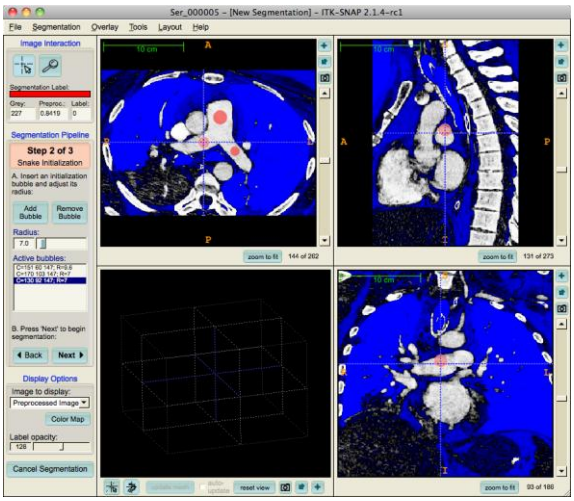
(a)



(b)



(c)



(d)

Figures 5a-d. Screenshots taken during the use of ITK-SNAP for the segmentation process. a) The initial screen once the image has been loaded. b) Setting the sampling region for the segmentation. c) Choosing the intensity values to differentiate between the vasculature to be segmented and the surrounding tissues. d) Seed placement to begin the segmentation.

After the initial segmentations were generated, they were then "cleaned". Cleaning consisted of manually parsing each of the segmentations slice by slice eliminating all areas where bleeding occurred. I define image bleeding as the areas where the image intensities are

very similar to that of the vasculature we are trying to segment so that the segmentation crosses over and begin to include the surrounding tissue; an example is shown in Figure 6a. In the figure the segmentation bleeds from the pulmonary trunk into the aorta. Figure 6b is the same example after it has been cleaned and is considered a completed segmentation. The segmentation parameters used to generate each of the segmentations such as the intensity values shown in 5c, are included in Appendix B.

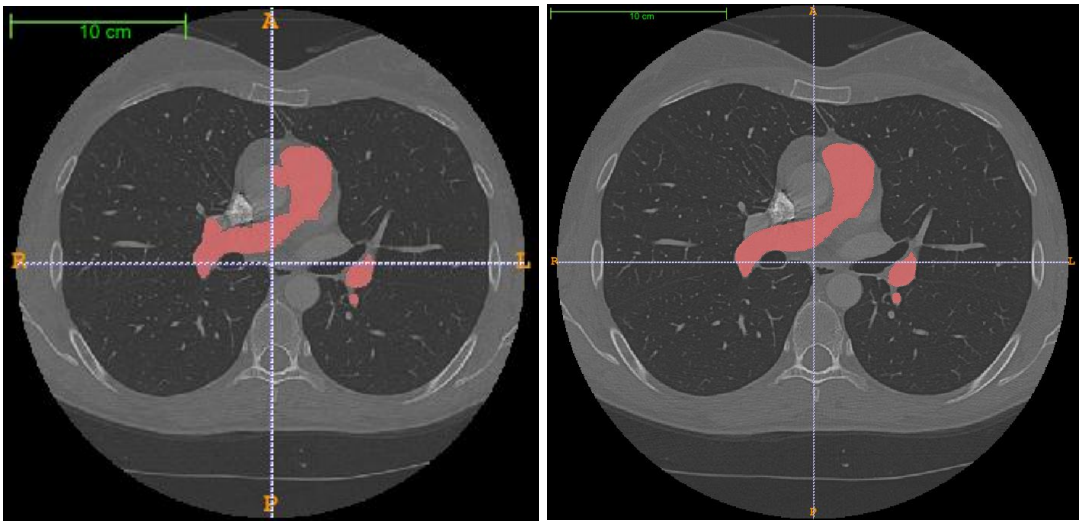


Figure 6a-b. a) Example of segmentation with bleeding b) Example of a cleaned segmentation.

Overall, this entire segmentation process from the development of the initial segmentations to the final clean version took the better part of six months to generate for all the images.

3.2.2 Overview of Modeling Steps

Segmentations are first generated as described in Section 3.2.1. Next, the skeletons are extracted from the segmentation. The skeleton is the backbone of the segmentation comprised of a single-voxel wide collection of the medial most voxels that stretch the length of each vessel

within the segmentation and is described in detail in Section 3.2.4. From the skeleton the graphs are then extracted. The graphs represent the structure of the vasculature and consist of bifurcation and termination nodes connected by vascular segment centerlines. The graphs are described in detail in Section 3.2.5. Even with the tedious cleaning of the segmentations, imperfections still remain and lead to faults in these graphs. To address these faults we explore methods of preprocessing the segmentations described in Section 3.2.3.

3.2.3 Preprocessing of the Segmentations

The remaining segmentation errors primarily consist of small holes in the segmentation (areas within the vascular lumen that the active contour did not "flow" into) and surface irregularities due to partial volume artifacts. These segmentation errors create numerous spurious segments in the skeleton. Before we continue with the modeling of the vasculature we first need to reduce these segmentation errors by preprocessing. We applied four different preprocessing methods and then evaluated the refined segmentation based on the number of edges within the graph. A graph edge corresponds to a single segment of the skeleton. We first calculated the number of edges within the graph generated from the "raw" (not processed) segmentation. We did this on a training set of 27 control cases and ten PH cases.

The preprocessing was performed with a script called PreprocessSegmentation.py developed in house with the goal to reduce the number of edges by smoothing out the segmentation. The script can be found in Appendix C. Smoothing was obtained with combinations of median filtering and morphological closing, where the kernel sizes of these steps were varied. We evaluated the segmentation smoothing and its impact on the generated vascular models. Our outcome measure was the number of edges in the graph of the model. We

used median filtering and morphological closing both alone and in combination and then compared them to the original segmentation with no preprocessing. These findings are summarized in the results Section 4.2.

3.2.4 Vascular Skeleton Generation

At this point we have the finished preprocessed segmentation. The next step is to generate the skeletons (or medial lines). The skeleton is composed of one-voxel thick lines that are at the center of the vessels. To generate the skeleton we used an existing ITK-based parallel thinning algorithm. A segmentation example and the corresponding skeleton are shown in Figure 7.

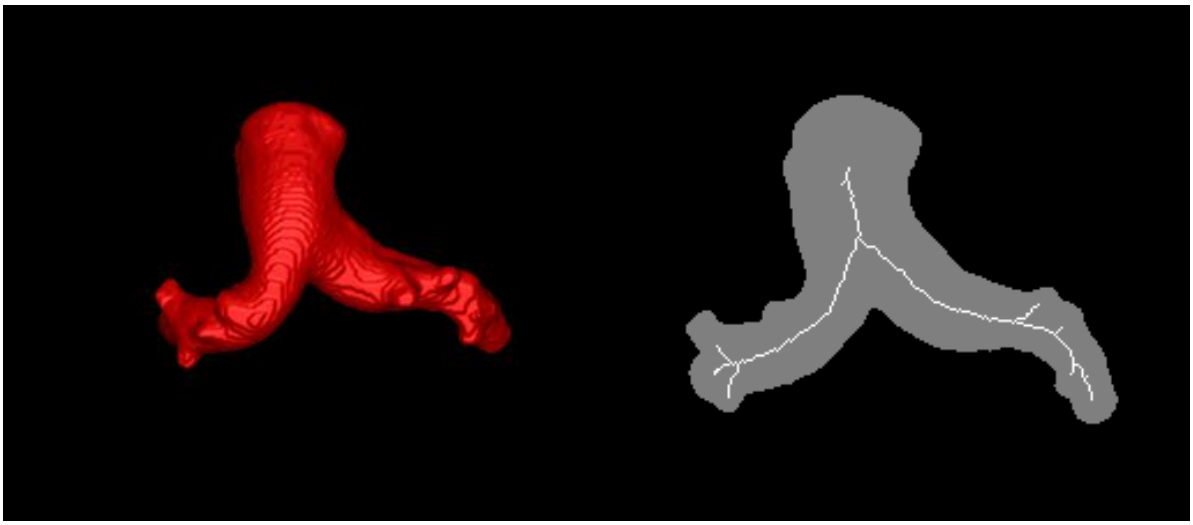


Figure 7). Visualization of the pulmonary vasculature model in ITK-SNAP. (Left) Surface rendering of the segmentation obtained with ITK-Snap. (Right) Illustration of the skeleton achieved from the segmentation using parallel thinning

The parallel thinning algorithm is a mathematical morphological tool we have examined as an approach for extracting a skeleton from the data in a discrete space. The purpose of the

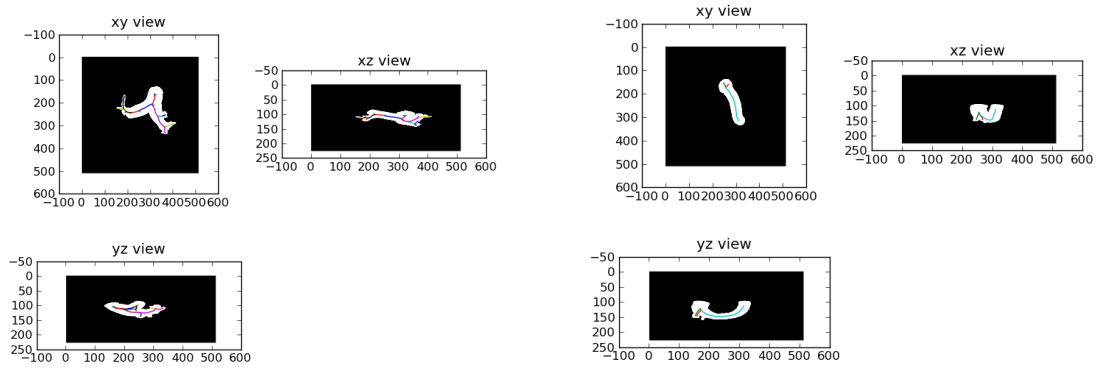
thinning algorithm is to iteratively remove the object's surface until only the centerlines remain. The goal is to preferentially remove large amounts of data such that only the skeleton remains [49, 61, 62]. Erosion, as discussed in Section 2.5.2, must be performed symmetrically from each side to ensure the centerline is preserved. There are two possible approaches to thinning an image: 1) use a kernel-based filter and 2) use decision trees. In kernel-based filters the kernel is the defined structural element that is applied to the image. The method presented here is a decision tree based method for the thinning of a 3D image and was used to generate the skeletons in this research[61] .

A key component of thinning is to be sure that the centerline is at the medial-most position of the vessel within a 3D image. To be sure this is the case, the algorithm performs a series of tests for each pixel to determine if it can be eroded from the image, this process repeats until no more pixels are removed. There is also a topological requirement of the thinning algorithm: the algorithm must preserve the number of connected objects, cavities and holes in the original shape. To ensure this is done the Euler characteristic and the connectivity are preserved to guarantee the invariance of the topology [62]. The Euler characteristic is a value used to describe the shape and structure in a particular space [63]. S is a subset of the data consisting of all ones. The complement of *S* is *s* and it contains all zeros. When *s* is completely surrounded by *S*, there is a cavity [62]. In a 3D image the only difference between a hole and a cavity is that the hole is not completely surrounded by *S*. In the algorithm pixels are deleted based on four characteristics: 1) The pixel is a surface or border pixel; the algorithm checks one of the six directions at a time to make sure the centerline is not shifted to one side of the vessel. 2) The pixel is not at the end of a line. 3) No holes are created when the pixel is removed. 4) The final test, and the most computationally expensive one, looks to see if the current pixel is a simple

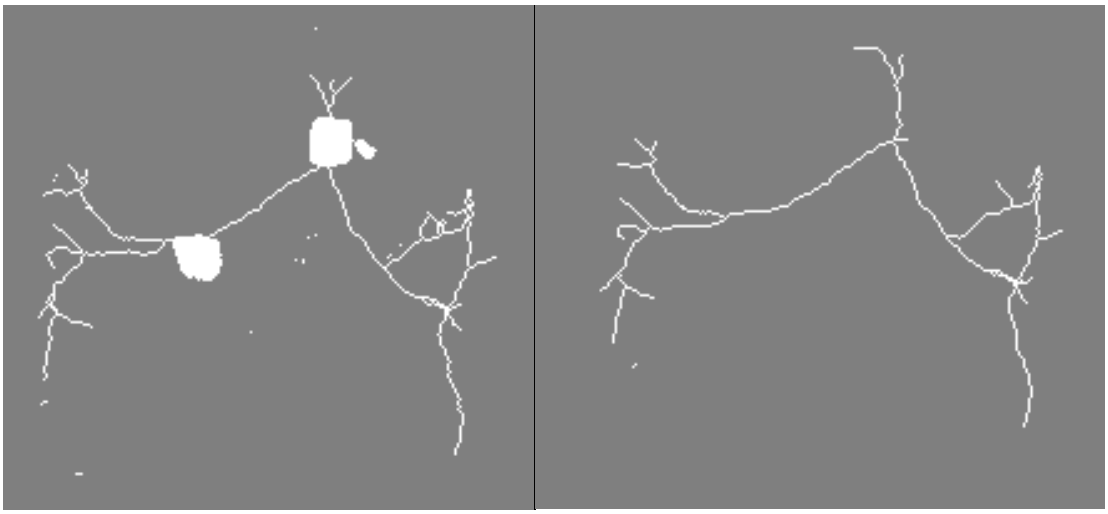
pixel whose deletion will have no effect on the number of connected segments, not interrupting the path [62]. The tests are performed in parallel on all the voxels in the 3D image. The final step is to double check that the connectivity of the skeleton has been preserved[62].

We generate the skeleton from the smoothed segmentation using the BinaryThinning3D executable, a compiled C++ program unmodified from the code acquired from the Insight Journal website that defines the algorithm [61].

Figures 8a and b are examples of the resulting skeletons for the aorta and pulmonary vasculature for one case.



(a) (b)



(c) (d)

Figures 8a-d) Skeleton comparison before and after preprocessing, a) An example skeleton of an aorta b) An example skeleton of the pulmonary vasculature, c) An example of the skeleton of an unprocessed segmentation and d) a skeleton after the preprocessing occurs.

3.2.5 Vascular Graph (Model) Generation (based on NetworkX [59])

The skeleton is simply an image of unordered voxels. In order to make our measurements, we must first transform these unordered voxels into a structure that represents the

underlying anatomy. We did this by creating vascular graphs comprised of three classes of nodes: degree one nodes are either endpoint or terminal voxels. Degree two nodes are centerline voxels and degree three nodes are the bifurcation voxels. This is a multi-step process that we now describe.

First, we created an undirected graph with every voxel in the skeleton being a node in the graph. Edges were added between any nodes coexisting within a 3x3x3 (3-dimensional) neighborhood of voxels. From this, undirected graph bifurcations and endpoints were recognized by the node degree. This is illustrated in Figure 9.

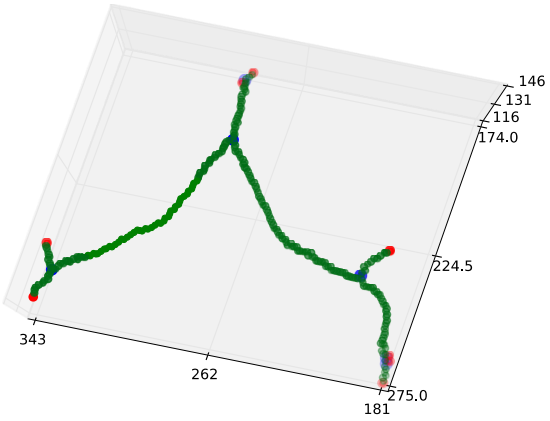


Figure 9) Example of an undirected graph generated from a skeleton image. Here blue nodes are bifurcations, green nodes are centerlines, and red nodes are endpoints.

Second, a directed graph was generated from the undirected graph. A root node was required for the directed graph and we selected the undirected graph node with the highest distance from edge (DFE) value as the root. The DFE was calculated using the city distance transform. Expanding out from the root, all bifurcation and endpoint nodes in the undirected graph were added as nodes in the directed graph with the collection of degree-two nodes (centerlines) forming the directed edge between the nodes. Each node is labeled with the (i, j, k) coordinate of the skeletal image voxel from which the node was obtained. The algorithm

identifies all termination voxels and bifurcations in the undirected graph. The paths between the termination points and the seed node are then traced out using a bidirectional Dijkstra algorithm. The algorithm is used to identify the minimum cost path between the seed node and each endpoint in the undirected graph. The path is split into segments defined by the endpoints and bifurcations along the path. The endpoint and bifurcation nodes are then added as nodes in the directed graph with a directed edge, while the connecting centerline node paths are added as attributes of the corresponding edges

Third, because of imperfections that remain in the segmentation, the resulting directed graph still has edges leading to false endpoint nodes. To further clean the graph we deleted any edge that was shorter than five voxels, because the edges of interest are the three largest edges in the vasculature representing the pulmonary trunk, the right and the left main pulmonary arteries, all of which are larger in length than 5 voxels. This naïve rule was proven inadequate, so we manually deleted extraneous edges as well. We experimented with machine learning techniques to create more comprehensive pruning rules; this is described in the next section. Finally, the edges of the pruned graph were fit with a least-squares cubic spline. The spline fit provides a smooth function that will later be used to capture vascular features from the segmented image. An example of a fitted graph is shown in Figure 10.

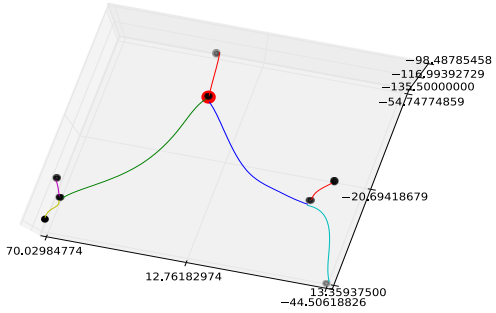
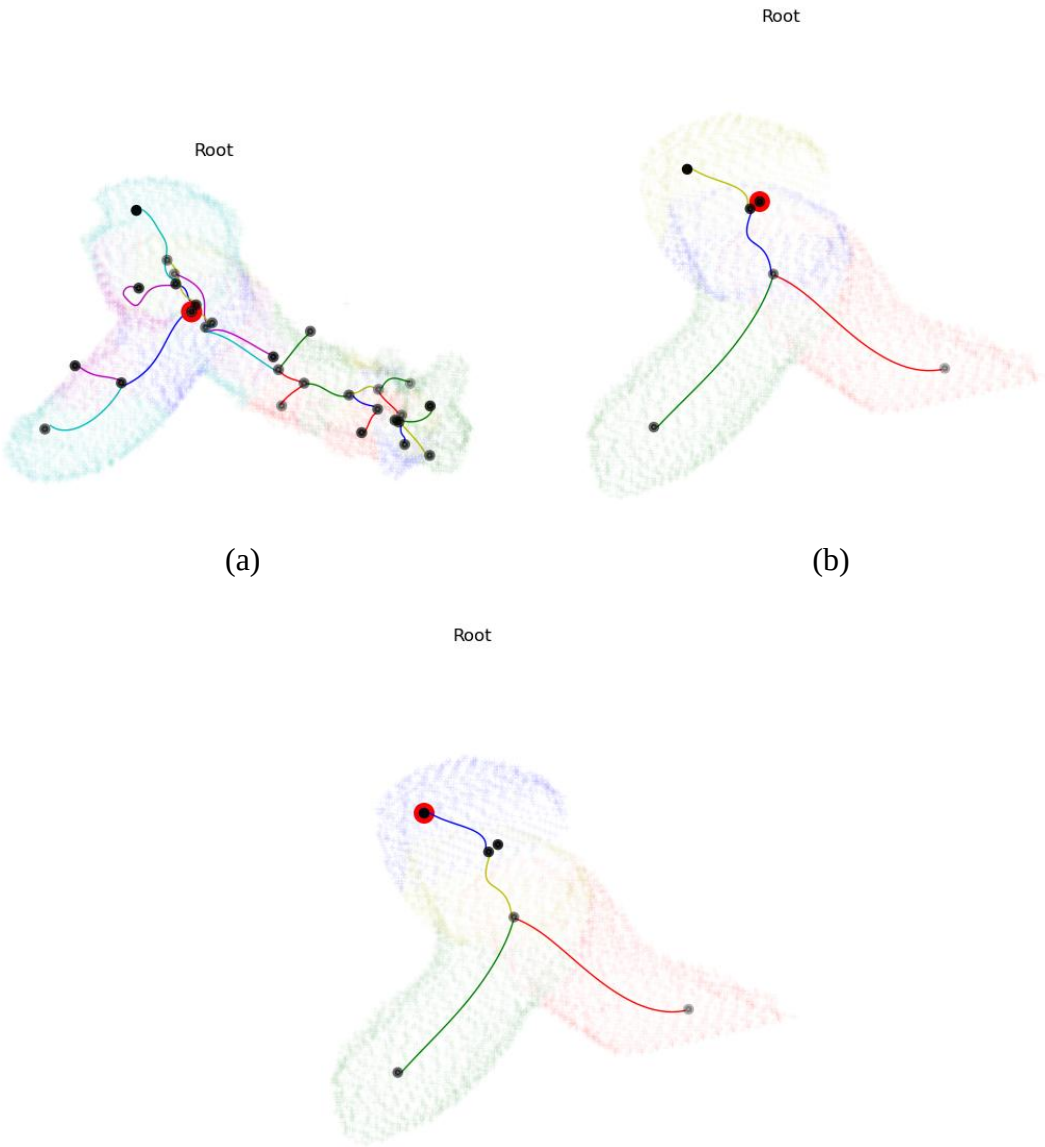


Figure 10) An example of a directed graph that has had the centerlines fit with a least-squared cubic spline. The root of the graph is highlighted in red. Note that in this case the selected root corresponds to the bifurcation of the pulmonary trunk into the left and right pulmonary arteries.

An in-house script was written using NetworkX [59] called GenerateGraph.py for the graph generation and is in Appendix C.

One problem encountered with the graph involved the placement of the root node as seen in Figure 10. Figures 11a-c visually steps through the graph development and refinement process (the root node is highlighted in red). In the original segmentation shown in Figure 11a, the root node is not correctly identified. Figure 11b shows the graph after preprocessing. A script was developed to correct the placement of the root node called rerootGraph.py and is in Appendix C. Figure 11c shows the graph after this script is applied and the root node is now properly placed.



(a) (b)

(c)

Figures 11a-c) Stepwise visual representation of graph development

3.2.6 Vascular Graph Pruning and Rule Development

The basis for all of our analysis is an accurate graph representation of the vasculature. Unfortunately, the graph generation described above results in a graph with a number of errors, most notably a large number of false centerline segments due to imperfections in the surfaces of

the segmentations. Figure 12 below shows an example cartoon demonstrating what happens when there are small divots and bumps in the segmentation causing extra lines to be drawn distorting the skeleton and subsequently the graph.

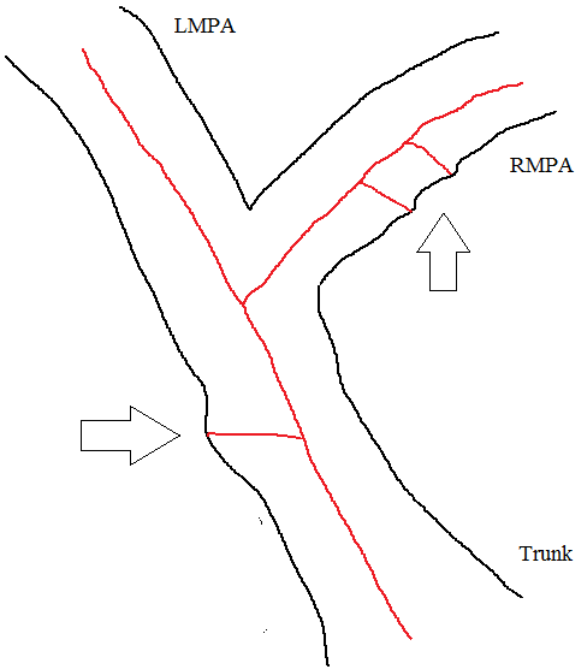


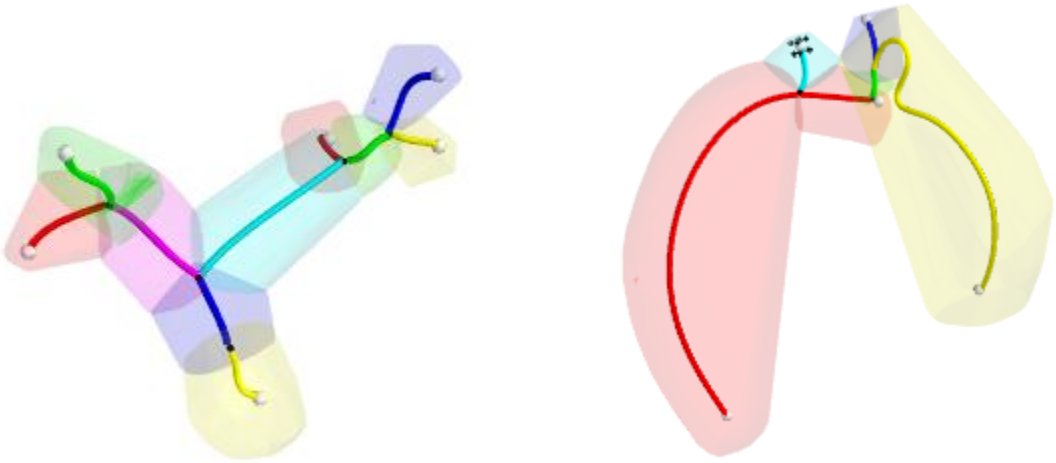
Figure 12. Example of extraneous lines in the skeleton.

The arrows in the figure highlight the addition of unwanted center lines because of inconsistencies in the segmentation.

Thus it is of fundamental importance to be able to delete false segments while preserving true vascular segments. An important element of these algorithms is to be able to delete false nodes/edges. Initially the graph is automatically cleaned by deleting edges with lengths shorter than a specified threshold currently set at five voxels. The appropriate threshold is dependent on the nature of the segmented vasculature. The pulmonary vasculature segmentations are particularly challenging because of the wide range of vascular diameters present and the short segments (with respect to diameter) in the tree.

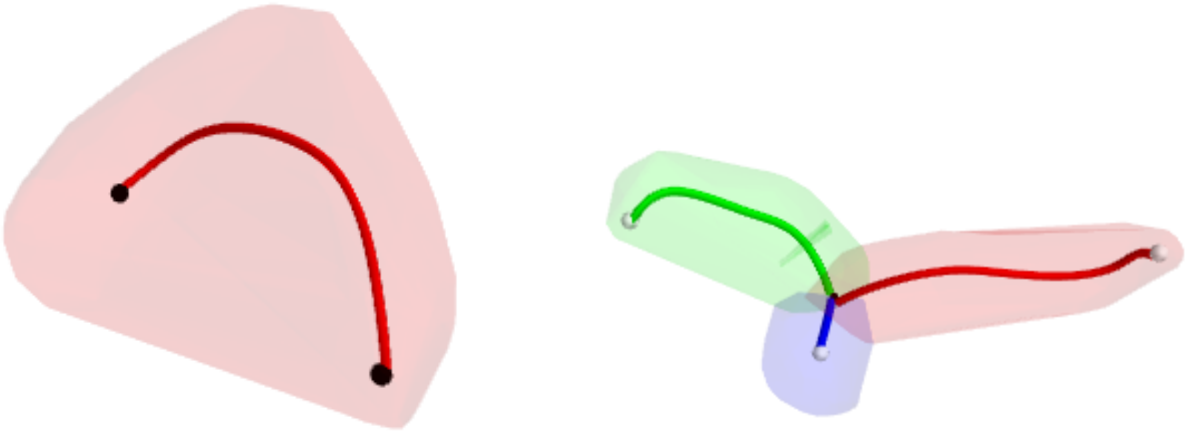
A script called editGraph.py (Appendix C) was developed that allows manual editing of the graphs by giving the user the ability to visualize the graph and remove additional false centerlines. We did this for 15 PH cases and 69 control cases. The program stores the

information of the original graph as well as the edited graph and we are able to identify features by looking at the positive edges that were kept and the negative edges that were removed. During the editing process there were a few cases that had unusual discrepancies in the centerline that needed to be addressed on an individual basis and will be described in detail later. Figures 13a and b are such examples. Figure 13a shows how the pulmonary trunk is separated into two edges when it should all be one. This seems to be an issue with a few of the cases where a segment is separated when it clearly should not be. Figure 13b shows an odd edge in the aorta (highlighted in yellow).



(a) (b)

Figure 13a and b) Edge discrepancies a) Pulmonary vasculature example, b) Aorta example.



(a) (b)

Figures 14a and b. Examples of what the graphs should resemble if no editing is required or after editing has occurred. a) the aorta and b) the pulmonary vasculature.

The complete tables of the edge data from the unedited and edited graphs for this process can be found in Appendix E and the summary results are listed in Section 4.3.

3.2.7 Summary of Steps

This section summarizes in Table 2, the overall steps taken during the creation of PUMA. It also clarifies the steps that were performed manually from those that were done in an automated capacity.

Table 2. Overall Summary of Steps.

Segmentation	Initial Segmentation Generation	A combination of manual input in choosing the threshold values for ITK-SNAP to implement in the automatic generation of the initial segmentation.
	Cleaning	Manually edited the segmentations slice by slice eliminating bleeds.
	Preprocessing	Automatic step using the PreprocessSegmentation.py script
Skeleton	Generate Skeleton	Automatic step using BinaryThinning3D executable
Graph	Generate Initial Graph	Automatic step using GenerateGraph.py script
	Clean Graph	Manually deleted additional unwanted edges using editGraph.py

3.2.8 Prediction Model Development for Pruning Edges

This section describes how the prediction models were generated for classifying correct and incorrect edges for fine tuning and pruning the vascular graphs. The purpose of the model is to be able to eliminate the manual pruning step using editGraph.py described in the previous section.

Gathering Feature Data

We decided to use the open source machine learning software package Weka [64] for generating the Simple Cart decision tree classifier and to generate the logistic regression model for classifying the edges that should remain or be deleted from the graph to eliminate the manual

editing step previously discussed. We first had to generate the distance from edge (DFE) values for each voxel within a given edge. The distance from edge is the value assigned to the distance between a voxel in the segmentation and the nearest voxel not in the segmentation. By calculating the DFE value for each voxel along each of the edges within the skeleton we can calculate different features of the vessels that can be used in the classifier. We calculated seven features for each of the edges within the segmentations. The features are as follows: 1) the *pathlength,* defined as the total number of voxels along a given edge; 2) The value associated with the *shape* of the edge; 3) The *volume* of the edge; 4) the *ratio* of length relative to depth calculated as the minimum DFE value over the maximum DFE value; 5) the difference between the minimum and maximum DFE value or *average mean distance* to the surface of the edge; 6) the minimum DFE value for each edge; and 7) the maximum DFE value for each edge. There are three scripts used for gathering this data. The first is getDFE.py the initial script for calculating the DFE's across the images. The scripts Extract_DFE_Values.py and ExtractShapeVolume.py were used for calculating the specific features. All of these scripts are located in Appendix C. These seven features were gathered for the dataset comprised of 122 images (99 control and 22 disease cases), a total of 1,546 edges each with seven attributes. It is important to note that the edges that were not manually deleted when using editGraph.py (found in Appendix C) were labeled as positive edges to represent the edges that should remain. The edges that were deleted were labeled as negative edges. Due to the size of this dataset it is not included in the appendices. However, it is available upon request.

Prediction Model Development

As stated earlier, we decided to use Weka [64] to generate a Simple CART decision tree for classifying edges. We also used logistic regression to build a classifier for comparison. The

entire dataset containing 1,546 edges was used and 10 fold cross validation was applied for evaluating the model. This model was generated twice: the first with the original data and the second time with normalized data. We normalized each of the attributes using min-max normalization with the exception of the shape and volume. These values were already on a normalized scale from 0 to 1. The normalized values are simply the values divided by the difference of the minimum and the maximum values. The results from this process can be found in results Section 4.4 and the actual output from the model is located in Appendix F.

Weka [64] was also used to generate the logistic regression model again for both the normalized and non-normalized data. The results are summarized in Section 4.4. Due to the length of the output only summary data has been included in this document; however, the entire output for each model is available upon request.

3.3 SPECIFIC AIM 3

Use PUMA to diagnose PH. This will be done by performing a semi-automated measurement of vascular diameters and making comparisons to manual measurements of the same cases.

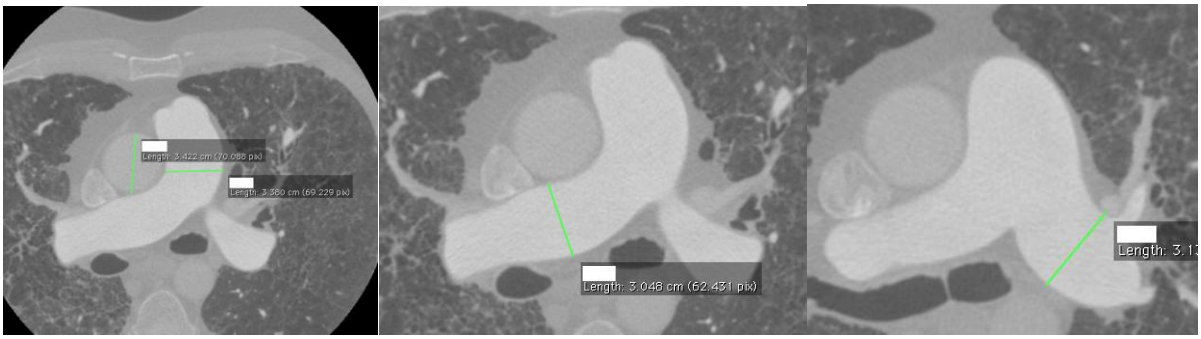
3.3.1 Generating Measurements

We randomly selected 100 normal CT pulmonary angiography (CTPA) exams from our data bank. The exams were determined to be negative with respective to both pulmonary embolism (the indication for the exams) and PH by examining the accompanying dictated radiology reports. From a separate databank of images, we selected 24 CTPA exams where the

patient was diagnosed with PH via right heart catheterization. For each case, human observers manually quantified the vascular geometry. Independent of these measurements, the models of the pulmonary arterial vasculature and the aorta generated in Specific Aim 2 were used to automatically quantify the vascular geometry in the same cases.

3.3.2 Step 1: Make manual measurements of the vasculature and aorta.

In order to determine how well the automated measurements were performing, we first needed to generate a baseline for comparison. We did this by first going though each of the 124 exams (normal cases plus the PH cases) and identifying the slices on which to measure the diameters of the ascending aorta, the pulmonary trunk and the right and left main pulmonary arteries (reviewer 0). Upon review of the images four of the control cases and one disease case were eliminated due to incomplete image files. The remaining 96 control cases and 23 disease cases were used for manual measurement. The slice data is reported in in Appendix G. A Python (www.python.org) script was written in house that was used to isolate these slices for manual quantification. Manual quantification was done by me and three independent reviewers (all graduate students) using OsiriX (http://www.osirix-viewer.com/). For the identified slice, each reviewer used a line ROI (region of interest) tool to measure the diameter of each vessel. Each observer was blind to both the other reviewers' results and the quantification from the vascular models. The actual written instructions provided to the viewer can be found in Appendix H. Figures 15a-c show one of the normal cases where the measurements were made using OsiriX.



Figures 15a-c) Example of manual the measurements made in OsiriX. a) Aorta and pulmonary trunk, b) right main pulmonary artery and c) left main pulmonary artery.

Statistics defined in the following sections were calculated for each reviewer's measurements for comparisons between reviewers to determine agreement as well as for comparisons between disease and control cases. Then the average of the reviewer's measurements for each case was taken for comparison to the automated measurements. For instance, each case has four measurements associated with each vessel. The average of those four measurements was taken for each vessel to give one value to use for comparison. These average values can be found in Appendix I. The original reviewer measurements can be found in Appendix J.

3.3.3 Step 2: Making automated measurements for comparison.

A Python script called AutomatedMeasurements4.py is in Appendix C. It was generated in house for identifying the segments of interest and calculating the necessary measurements. We first identified the median or mean location of the aorta centerlines. The ascending aorta is defined as the centerline points with x values less than the median (mean). Then from those centerlines satisfying the criteria we selected the midpoint measurement as simply the point closest to the median z value for these points.

The pulmonary trunk segment is defined as the edge between the root and the bifurcation node in the direction of the greatest number of descendants. The left and right main pulmonary artery segments were identified as the edges that followed directly from the root-identified bifurcation. A local coordinate system was defined at each point along the spline-fitted centerline. This local coordinate system was then used to define orthogonal planes at each point; the surface points (points with a distance-from-edge value of zero) for that vascular segment were mapped to orthogonal planes corresponding to each point on the fitted centerline. We estimated the vessel radius at each point along the fitted centerline as the average distance between the centerline point and the surface points mapped to the corresponding orthogonal plane. We computed the radius at the midpoint of each vessel centerline. A full table of these resulting automated measurements along with the average of the manual measurements can be found in Appendix I.

In order to make automated measurements from an ordered graph, we must first match the graph to the anatomy of interest. Although originally motivated by the need to recognize the pulmonary trunk, our heuristic of setting the root node to be the node with the maximum distance from edge was not sufficiently accurate, since, as illustrated with Figure 9, the root node was placed at the bifurcation rather than in the trunk (other incorrect locations were also observed).

The identification of the root node in the pulmonary trunk in the segmentation is difficult particularly in creating a method that is robust in the presence of segmentation imperfections. We explored a median (mean) x measure as well as the root mean square (RMS) algorithm. We found the best performance using a heuristic that assumes the terminus of the pulmonary trunk lies near the center in the left-right direction, near the back in the posterior-anterior direction, and near the top in the inferior-superior direction. Finding the degree-one node with the minimum

root mean square distance from the coordinate (DIM_X/2, 0, max (Z)) correctly identifies the root node location in the 118 cases examined. This heuristic works because our segmentation is not extending deeply into the vascular tree. The re-rooted graph from Figure 10 is shown in Figure 16. The pulmonary trunk is defined as the edge between the root node and its child bifurcation. The left and right pulmonary arteries are the edges exiting this child bifurcation. The rerootGraph.py script discussed above can be seen in Appendix C.

With the graph matched to anatomy, we went back to the original segmentation and matched each surface voxel to the nearest edge. Finally, the spline-fit to each edge was used to define a plane at each point along the centerline that was orthogonal to the local direction of the centerline. The mapped surface points for each edge were then mapped to one of these orthogonal planes. These steps are illustrated in Figure 16. The average distance between centerline and surface points in the orthogonal plane then define a local radius measure.

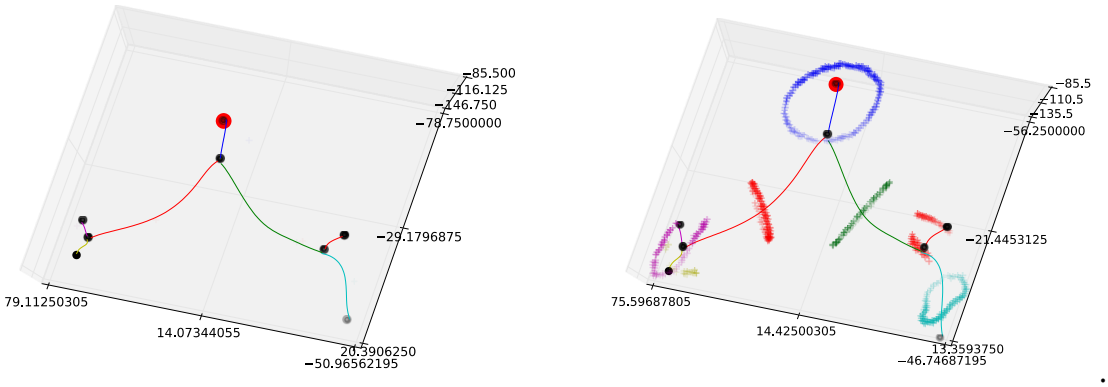


Figure 16) (Left) Graph from Figure 9 with surface points matched to each edge. (Right) Surface points mapped to the plane orthogonal to the midpoint of the centerline.

When generating the automated measurements, out of the 118 cases nine were eliminated due to the inability to generate the automated measurement. The remaining 109 (21 disease cases and 88 control cases) were used for comparisons. The nine cases that were eliminated returned the same error relating to the plane points and the measurements were unable to be obtained. The

exact cause of the error is unclear. At each point along the edge there is a plane of points that expands in the x, y and z directions. I believe there may be a gap in these points due to an imperfection in the segmentation making a measurement impossible because the points cannot be traversed to the surface of the vessel. An error in these points could be caused by an error in the graph generation from a faulty segmentation. Further scrutiny will be required to address this flaw in the program.

3.3.4 Statistics for Comparison between Measurements

This section describes the methods for the analysis of the measurement data. The following sections include comparisons within the manual measurements only, the automated measurements only, and then a comparison between the two classes of measurements as well. t-tests were used for determining statistical significance between the classes of measurements and the FDA's method of equivalence analysis was applied to test the agreement between reviewers.

3.3.5 Determining Agreement between Reviewers of Manual Measurements

A series of comparisons were performed to determine if the measurements generated by the different reviewers met the assumption of no significant difference between the reviewer's measurement values. If there is a large variation in the measurements across reviewers then the comparison to the automated measurements will not be informative. We chose to use the FDA's method of equivalence analysis that is used when comparing an original drug to the generic version. In this situation the null hypothesis (H_0) is the means between the two groups are significantly different, the alternative hypothesis (H_A) is the means are not significantly different.

Six comparisons were generated reviewer 0 vs. 1, 0 vs. 2, 0 vs. 3, 1 vs. 2, 1 vs. 3, and 2 vs. 3. The means ($\bar{X}$), were calculated for each group.

The following are the specific steps taken for the generation of this data as reported and used by the FDA [65].

H_0: $\mu_T / \mu_R \leq \theta_1$ or $\mu_T / \mu_R \geq \theta_2$ versus $H_A : \theta_1 < \mu_T / \mu_R < \theta_2$, where $\theta_1 = 0.80$ and $\theta_2 = 1.25$.

Where μ_T = mean of the first comparison group, and μ_R = mean of the second comparison group. Typically, the H_0 is rejected with a type I error $\alpha = 0.05$ (two 1-sided tests), if the 90% confidence interval for the ratio of means between the two groups (μ_T / μ_R) is contained within the interval $[\theta_1, \theta_2]$. Rejection of the null hypothesis H_0 supports the conclusion of equivalence of the two products. In summary, the ratio of the means, 90%, 95% and 99% confidence interval was calculated for each comparison and then compared to the specified interval.

These findings are found in the results Section 4.5.1.

3.3.6 Manual Measurements of the Negative Cases Compared to the Disease Cases

We wanted to be sure that the measurements of the control cases were significantly different from the disease cases. First, using the manual measurements only, a comparison was performed between the control cases and the disease cases. This comparison was done for the aorta, pulmonary trunk, right main pulmonary artery and the left main pulmonary artery. The error mean difference and the unpaired t-test were used as measures of significance. The online software package GraphPad was used to perform these calculations (http://www.graphpad.com/quickcalcs/ttest1/?Format=C). The specific calculations are detailed in Appendix D and the findings are outlined in the results Section 4.5.2.

The same calculations were used for the comparison within the automated measurements as well, for the aorta, pulmonary trunk, right main pulmonary artery and the left main pulmonary artery. The findings for the automated comparisons for the three major pulmonary vessels and the aorta can be found in Section 4.5.3.

3.3.7 Comparison between the Manual Measurements and the Automated Measurements

We wanted to determine if there was a statistically significant difference between the manual and the automated measurements. We calculated the mean ($\bar{X}$), variance (S^2) and standard deviation (S) for each group for each vessel. We then applied the paired t-test using the same online software package as in the comparison between disease and controls (GraphPad, (http://www.graphpad.com/quickcalcs/ttest1/?Format=C). .

When determining statistical significance, if the calculated t value was less than the critical t-value, there was no difference found between the means of the automated and manual measurements. If the calculated t value was greater than the critical -value, then the two means were significantly different, and the null was rejected [66]. In addition to the t-test further evaluation was performed using Equivalence Analysis as described in Section 3.3.5. In our null hypothesis we are making the assumption that there is a difference between the manual and the automated measurements. These results as well as the paired t-test values can be found in Section 4.5.4.

3.3.8 Predicting the Presence of Pulmonary Hypertension

The next step was to determine if we can accurately classify patients based on the vasculature measurements as having pulmonary hypertension or not. To answer this question we built a logistic regression model using Weka. The model was given all four of the automated measurement variables (the Aorta, PT, RMPA and LMPA) as predictors and the label of disease versus control as the target to be predicted for each of the 109 images. 10-fold cross validation was used for evaluating the model. We then repeated the experiment using normalized data. We normalized each of the four measurements using min-max normalization. The normalized values are the values divided by the difference of the minimum and the maximum values. For comparison and to rule out any uncertainty surrounding the automated measurements, the same models were generated using the manual measurements.

Table 3). Ranges of Measurements

	Automated		Manual	
Vessel/Class	Minimum	Maximum	Minimum	Maximum
PT Disease	1.29	3.98	2.81	4.23
RMPA Disease	1.83	3.04	1.70	3.63
LMPA Disease	1.82	3.09	1.97	3.53
PT Control	1.08	4.46	1.93	3.90
RMPA Control	1.1	`3.67	1.48	3.36
LMPA Control	1.28	3.68	1.32	3.18

An ROC (Receiver Operating Characteristic) analysis was performed. This includes calculations of the sensitivity or true positive rate and specificity or true negative rate of the model. Sensitivity is calculated as the number of true positives (TP) over the sum of the TP and the false negatives (FN). Specificity is calculated as the true negatives (TN) over the sum of the false positives (FP) and the TN. The TP's are the patients that have PH and were predicted as

such. The TN's are the patients that do not have PH and are predicted as such. The FP's are those predicted to have PH but in reality do not and finally, the FN's are those that have the disease and are classified as negative for the presence of disease. The classification accuracy of the model was assessed with the area under the ROC curve (AUC).

In addition to the logistic regression model a Naïve Bayes model was also generated for comparison. The data and methods are the same only the classifying algorithm was changed. The findings for both sets of models (one set for the automated measurement data and one set for the manually measured data) are detailed in the results Section 4.5.5 and the output is shown in Appendix K.

3.3.9 Generating a Polynomial Regression Model for Predicting Pressure

This section is focused on the relationship between the measurements and the pressure values. Can we predict the arterial pressure based on the diameter measurements gathered? We model this relationship by generating a polynomial regression model in SPSS [67]. Polynomial regression determines the polynomial equation to predict a response of dependent variables (Y) based on a predictor (X) the independent variable. The order of the polynomial determines the number of possible inflections on the curvi-linear fitted line. The results from this model are summarized in Section 4.5.6 and can be seen in their entirety in Appendix L.

There are four measurements that were included 1) aorta, used for normalizing across patients, 2) pulmonary trunk, 3) right main pulmonary artery and 4) the left main pulmonary artery. Additional variables gathered are the mean pulmonary arterial pressure (MPAP), vascular pulmonary resistance (VPR), trans-pulmonary gradient (TPG), PA (pulmonary artery) systolic and PA diastolic pressures. The MPAP was chosen as the dependent variable because it is

measured directly through right heart catheterization where the majority of the other pressures are calculated based on other values. The measurement data that are included is dependent on the pressure. As the pressure increases the vessel dilates to accommodate; the longer the pressure persists the weaker the vessel becomes until it atrophies and begins to deteriorate. For these reasons, MPAP was named the dependent variable. The model was generated for both the pressure values in combination with the manual measurements and then the automated measurements.

4.0 RESULTS

This section summarizes the data gathered and reports the findings in response to the methods previously detailed.

4.1 POWER CALCULATIONS

I used an online power calculator (http://www.statistical_solutions.net/pss_calc.php) to determine the power of the pressure dataset for either accepting or rejecting the null hypothesis with 95% confidence in reference to the pressure data (27 cases). We are testing the null hypothesis that the mean pulmonary arterial pressure (MPAP) in the presence of pulmonary hypertension (PH) is equal to the MPAP in the absence of PH. We chose a one sided test because the only acceptable alternative hypothesis is that the mean pulmonary arterial pressure (MPAP) in the presence of pulmonary hypertension (PH) is greater than the MPAP in the absence of PH. The mean of the MPAP for the disease cases is 46.5; the mean for the normal cases is 18 and the standard deviation of the group is 17.63. According to the online calculator the power for the pressure dataset of 27 cases is 1.0.

I repeated these steps a second time calculating the power based on the automated right main pulmonary artery measurements. In this case we are testing the null hypothesis that the mean diameter of the right main pulmonary artery (RMPA) in the presence of PH is equal to the

mean diameter of the RMPA in the absence of PH. Again I used a one sided test because the only viable alternative hypothesis is that the mean diameter of the RMPA in the presence of PH is greater than the mean diameter of the RMPA in the absence of PH. The mean of the diameter of the RMPA for the disease cases is 2.46; the mean for the normal cases is 2.21 and the standard deviation of the group is 0.4. According to the online calculator the power is also 1.0.

4.2 PREPROCESSING OF THE SEGMENTATIONS

We found that each of the preprocessing methods reduced the number of edges in the original segmentation by about 50%. Increasing the kernel size had no effect on edge reduction. However, using both median filtering and morphological closing resulted in a reduction in the number of edges by half compared to using either method alone. I chose to use the method that utilized both smoothing methods in combination with an increase in the kernel size giving the lowest average number of edges for either segmentation (Column 6).

Tables 4 and 5 below are the summarized preprocessing results. The full tables of edge values for each case in the training set can be found in Appendix E.

Table 4). Preprocessing of the Aorta Segmentation Data

	No Preprocessing	**median only**	**close only**	**median & close**	**median & close with a kernel size of (2,2,2)**
Total Number of Edges	1510	373	303	228	214
Average Number of Edges per case	40.81	10.08	8.19	6.16	5.78

Table 5). Preprocessing of the Pulmonary Vascular Segmentation Data					
	No Preprocessing	**median only**	**close only**	**median & close**	**median & close with a kernel size of (2,2,2)**
Total Number of Edges	1927	678	671	757	626
Average Number of Edges per case	52.08	18.32	18.14	20.46	16.92

4.3 EDGE DATA

In summary, out of the 15 disease cases, 12 (80%) of the aorta segmentations and eight or 53.3% of the pulmonary vasculature segmentations required editing. For the 70 control cases 34 aortas (48.57%) required editing and 20 (28.57%) of the pulmonary vasculature segmentations required editing. Complete tables of the edge data can be found in Appendix E.

4.4 PRUNING PREDICTION MODEL FINDINGS

Appendix F shows the summary statistics from the output of the two models that were built for classifying the edges into positives (the edges that should remain) and the negatives (the edges that should be removed). When looking at the results there is not much difference between the two types of models built whether it is the Simple CART decision tree or the logistic regression model. Normalization did not affect the model results.

Looking at the summary statistics in Table 6 the Simple CART model has a sensitivity or true positive rate of 82.4%. The specificity is 96.3%. The logistic regression model gave a sensitivity of 78.7% and a specificity of 87%. In comparing the two classifiers the Cart model

appears to perform slightly better with an ROC area of 89.3% versus the logistic regression at 84.8%.The actual output generated by Weka for the logistic regression model is also located in Appendix F. Due to the size of the output generated by Weka the entire output is available upon request but only the summary of the models can be found in Appendix F.

Table 6 lists the summary statistics for these models. For reference, a perfect model would give an ROC area of 1.0.

Table 6. Summary of model statistics.

Models	Sensitivity	Specificity	AUC	Mean Absolute Error
CART	82.4%	96.3%	89.3 %	0.0899
Logistic Regression	78.7%	87	84.8	0.1551

4.5 MANUAL MEASUREMENTS VS AUTOMATED MEASUREMENTS

4.5.1 Determining Agreement between Reviewers

We first analyze the variability between the manual measurements among the different reviewers.

Tables 7a-b) Summary Statistics of the manual measurements for user agreement comparisons (n=119 for each reviewer)

Table 7a.

Vessel	Aorta				Pulmonary Trunk			
Reviewer	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
Mean	3.194	3.284	3.114	3.021	2.826	2.864	2.832	2.733
Variance	0.343	0.401	0.342	0.383	0.374	0.420	0.402	0.394
Standard Deviation	0.586	0.633	0.585	0.619	0.612	0.648	0.634	0.628
SEM	0.054	0.058	0.054	0.057	0.056	0.059	0.058	0.058

Table 7b.

Vessel	RMPA				LMPA			
Reviewer	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
Mean	2.284	2.279	2.223	2.197	2.249	2.213	2.209	2.175
Variance	0.331	0.351	0.346	0.333	0.326	0.297	0.319	0.340
Standard Deviation	0.575	0.592	0.588	0.577	0.571	0.545	0.565	0.583
SEM	0.053	0.054	0.054	0.053	0.052	0.05	0.052	0.053

Table 8a) Aorta

Comparison	0 and 1		0 and 2		0 and 3	
Ratio of Means	0.9726		1.0257		1.0573	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9342	1.0126	0.9857	1.0673	1.0142	1.1023
95% CI	0.927	1.0206	0.9782	1.0756	1.0061	1.1113
99% CI	0.9129	1.0364	0.9635	1.092	0.9904	1.1291

Table 8b). Aorta

Comparison	1 and 2		1 and 3		2 and 3	
Ratio of Means	1.0546		1.08717		1.0308	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	1.0125	1.0984	1.0418	1.1344	0.9884	1.0752
95% CI	1.0045	1.1071	1.0333	1.1438	0.9804	1.084
99% CI	0.989	1.1244	1.0167	1.1626	0.9649	1.1016

Table 9a). Pulmonary Trunk

Comparison	0 and 1		0 and 2		0 and 3	
Ratio of Means	0.9867		0.9979		1.034	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9411	1.0347	0.9519	1.0461	0.9858	1.0848
95% CI	0.9325	1.0443	0.9433	1.0557	0.9768	1.0949
99% CI	0.9159	1.0633	0.9266	1.0749	0.9592	1.115

Table 9b). Pulmonary Trunk

Comparison	1 and 2		1 and 3		2 and 3	
Ratio of Means	1.0113		1.0479		1.0362	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9637	1.0612	0.998	1.1004	0.9871	1.0878
95% CI	0.9548	1.0712	0.9886	1.1108	0.9779	1.0981
99% CI	0.9374	1.091	0.9704	1.1317	0.96	1.1186

Table 10a). RMPA

Comparison	0 and 1		0 and 2		0 and 3	
Ratio of Means	1.0022		1.0274		1.0396	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9488	1.0587	0.9722	1.086	0.9839	1.0986
95% CI	0.9388	1.07	0.9619	1.0877	0.9375	1.1104
99% CI	0.9195	1.0923	0.9419	1.1211	0.9534	1.134

Table 10b). RMPA

Comparison	1 and 2		1 and 3		2 and 3	
Ratio of Means	1.0118		1.0373		0.9754	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9221	1.0317	0.9809	1.097	0.9563	1.0706
95% CI	0.9122	1.043	0.9703	1.109	0.9459	1.0823
99% CI	0.8928	1.0655	0.9499	1.1329	0.9258	1.1059

Table 11a). LMPA

Comparison	0 and 1		0 and 2		0 and 3	
Ratio of Means	1.0163		1.0181		1.034	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9632	1.0722	0.964	1.0752	0.9779	1.0936
95% CI	0.9533	1.0833	0.9539	1.0867	0.9674	1.1055
99% CI	0.934	1.1056	0.9343	1.1095	0.9471	1.1294

Table 11b). LMPA

Comparison	1 and 2		1 and 3		2 and 3	
Ratio of Means	0.9982		1.0175		1.0156	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	0.9459	1.0533	0.963	1.0753	0.9603	1.0743
95% CI	0.9361	1.0643	0.9529	1.0869	0.9499	1.0861
99% CI	0.917	1.0862	0.9332	1.1101	0.9299	1.1096

For each comparison all of the ratio values along with the calculated confidence interval values fall within the designated threshold of 0.80 (θ_1) and 1.25 (θ_2) therefore we reject the null and find that the measurements are effectively equivalent. We will now take the average across reviewers for each vessel for each case for comparison to the automated values.

4.5.2 Summary Statistics from Manual Measurements Only: a Comparison between Case and Control

For comparison purposes we need to test whether the manual measurements between the disease cases and the control cases are overall significantly different. Since agreement has been determined in the previous section, the average across reviewers for each vessel for each case was taken.

A comparison of the average manual measurements was performed between the control cases (n=96) and the disease cases (n=23) for each the Aorta, Pulmonary Trunk (PT), Right Main Pulmonary Artery (RMPA) and the Left Main Pulmonary Artery (LMPA). These findings are detailed in Tables 12 and 13. When looking at the comparisons of the aortas, the difference in means (control versus disease) is less than the error mean difference (EMD) and therefore, the controls are not significantly different from the cases. In addition, the p-value is greater than 0.05 also implying no significant difference. The results are as expected, the aorta was used for normalization and is not affected by pulmonary hypertension, and there should not be a difference between cases versus controls. On the other hand we expected to find a difference between the cases and controls when looking at the comparisons between the different pulmonary vessels. In each of these comparisons (the pulmonary trunk, RMPA and LPMA) the difference in means is greater than the EMD and therefore, the controls are significantly different from the cases. We found when comparing the t values to the t critical value ($t_{.05,22} = 1.717$) for these vessels the t critical is much less than the calculated t therefore, the null of equal means is rejected and the control cases are significantly different from the disease cases. For the aorta we accept the null of equal means.

Table 12. Average manual measurements for comparison between control and disease cases for the Pulmonary Trunk and the RMPA.

Vessel	**Pulmonary Trunk**		**RMPA**	
Label	**Control**	**Disease**	**Control**	**Disease**
Mean	2.83	3.31	2.18	2.52
Standard Deviation	0.4313	0.451	0.378	0.438
N	96	23	96	23
95% Confidence Interval	-0.687 to -0.287		-0.524 to -0.165	
Difference in Means	0.484		0.34	
Standard Error of the Mean	0.044	0.094	0.039	0.091
Error Mean Difference	0.101		0.09	
t-value	4.8169		3.8079	
p-value	< 0.0001		< 0.0002	

Table 13. Average manual measurements for comparison between control and disease cases for the LMPA and the Aorta.

Vessel	**LMPA**		**Aorta**	
Label	**Control**	**Disease**	**Control**	**Disease**
Mean	2.13	2.57	3.15	3.16
Standard Deviation	0.3478	0.327	0.455	0.427
N	96	23	96	23
95% Confidence Interval	-.6056 to -0.2893		-0.2149 to 0.1986	
Difference in Means	0.44		0.01	
Standard Error of the Mean	0.036	0.068	0.0464	0.0891
Error Mean Difference	0.08		0.104	
t-value	5.604		0.781	
p-value	< 0.0001		0.9379	

4.5.3 Summary Statistics from Automated Measurements Only: a Comparison between Case and Control

Tables 14 and 15 contain the summary statistics for the automated measurement comparisons.

For each comparison, with the exception of the aorta, all the calculated t-values are greater than the critical $t_{.05,20}$ =1.725. Therefore the null hypotheses that the means in each of the three main pulmonary vessel comparisons (Pulmonary Trunk, RMPA and LPMA) are equal are rejected. The means are significantly different, which was expected as with the findings in the comparisons within the manual measurements. Similarly, according to the p value of 0.1844 there was no difference found for the values of the Aorta and again as in the manual measurements the null of equal means is accepted.

Table 14). Comparisons between the Disease Cases (n=21) and the Control Cases (n=88) within the Automated Measurements for the Pulmonary Trunk and RMPA.

Vessel	**Pulmonary Trunk**		**RMPA**	
Label	**Control**	**Disease**	**Control**	**Disease**
Mean	2.74	3.12	2.21	2.46
Standard Deviation	0.513	0.568	0.436	0.38
N	88	21	88	21
95% Confidence Interval	-.06330 to -0.1288		-0.4561 to -0.0457	
Difference in Means	0.3809		0.2509	
Standard Error of the Mean	0.055	0.124	0.047	0.083
Error Mean Difference	0.127		0.104	
t-value	2.9951		2.4241	
p-value	0.0034		0.017	

Table 15). Comparisons between the Disease Cases (n=21) and the Control Cases (n=88) within the Automated Measurements for the LMPA and Aorta.

Vessel	**LMPA**		**Aorta**	
Label	**Control**	**Disease**	**Control**	**Disease**
Mean	2.2	2.53	3.14	3.32
Standard Deviation	0.383	0.381	0.576	0.475
N	88	21	88	21
95% Confidence Interval	-0.5112 to -0.1430		-0.4503 to 0.0877	
Difference in Means	0.3271		0.18	
Standard Error of the Mean	0.041	0.083	0.061	0.104
Error Mean Difference	0.093		0.136	
t-value	3.5227		1.3359	
p-value	0.0006		0.1844	

4.5.4 Automated versus Manual Measurement Comparison

Now that we have discussed the results for the comparisons for each class of measurements individually, we will now discuss the comparison of the automated to the manual measurements. According to the following Tables 16 and 17, the manual measurements were found to be significantly different in reference to the pulmonary trunk and the RMPA. However, when looking at the values in Table 17 for the LMPA and the aorta there was no significant difference found between the measurements. There are a number of reasons to cause these findings and are described at length in the discussion.

Table 16). Paired t-test results for the comparison between the automated and manual measurements (n=109) for the pulmonary trunk and the RMPA.

Vessel	**Pulmonary Trunk**		**RMPA**	
Label	**Manual**	**Automated**	**Manual**	**Automated**
Mean	2.93	2.81	2.179	2.025
Standard Deviation	0.464	0.543	0.378	0.742
N	109		109	
95% Confidence Interval	0.037 to 0.198		0.0085 to 0.2988	
Difference in Means	0.12		0.1536	
Standard Error of the Mean	0.0444	0.052	0.0385	0.0757
Error Mean Difference	0.041		0.073	
t-value	2.892		2.102	
p-value	0.0046		0.0382	

Table 17). Paired t-test results for the comparison between the automated and manual measurements (n=109) for the LMPA and the Aorta.

Vessel	**LMPA**		**Aorta**	
Label	**Manual**	**Automated**	**Manual**	**Automated**
Mean	2.215	2.263	3.151	3.175
Standard Deviation	0.373	0.402	0.445	0.561
N	109		109	
95% Confidence Interval	-0.1021 to 0.0066		-0.0973 to 0.050	
Difference in Means	0.0477		0.024	
Standard Error of the Mean	0.0357	0.0385	0.0426	0.561
Error Mean Difference	0.027		0.037	
t-value	1.74		0.6354	
p-value	0.0847		0.5265	

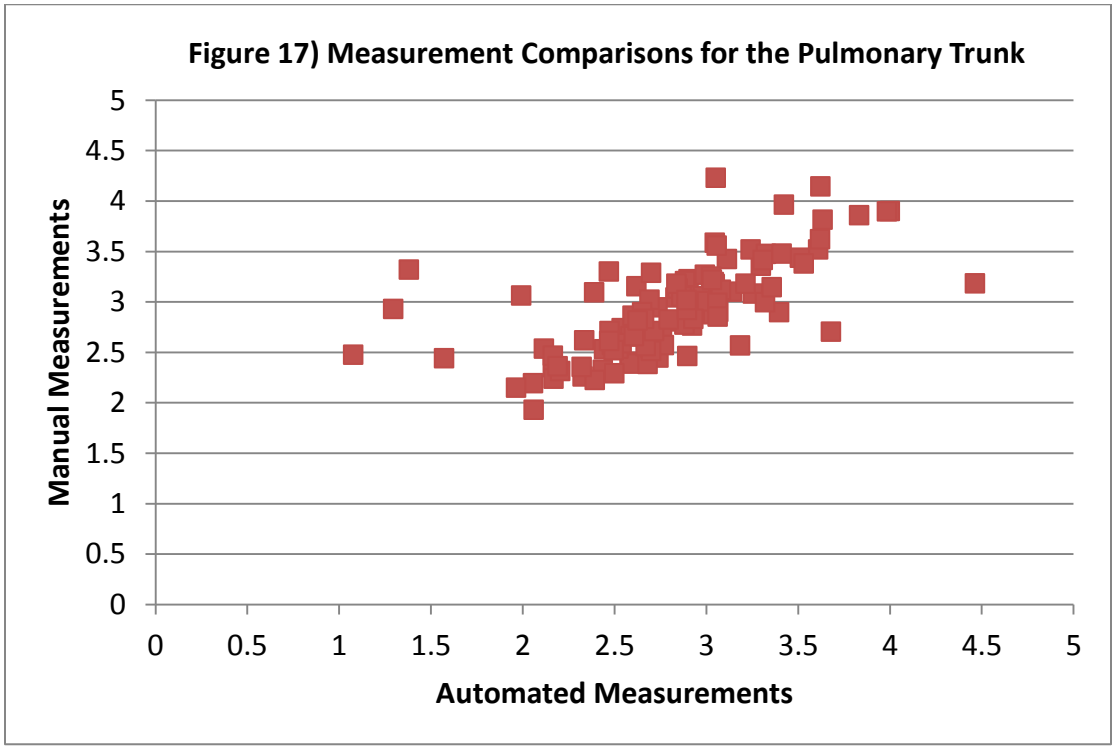
The following Table 18 summarizes the results from the equivalence analysis. Each of the ratio values fall within the specified threshold of 0.80 and 1.25. Therefore, we reject the null hypothesis of the two groups having significantly different means and instead accept the

alternative that the two groups are not significantly different. These findings further support that our automated measurements are comparable to the manual measurements.

Table 18). Equivalence analysis results for the comparison between the automated and manual measurements (n=109) for each of the vessels.

	Pulmonary Trunk		**RMPA**		**LMPA**		**Aorta**	
Ratio of Means	1.0427		1.0765		0.9823		0.9906	
Confidence Intervals	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
90% CI	1.002	1.085	1.011	1.149	0.945	1.021	0.956	1.027
95% CI	0.9949	1.0933	0.999	1.165	0.938	1.029	0.949	1.034
99% CI	0.9802	1.1101	0.976	1.195	0.924	1.044	0.937	1.048

The following four plots Figures 17-20 are scatterplots of the comparison between the automated and manual measurements for each vessel. The automated values are along the x-axis and the manual measurements are plotted along the y-axis.



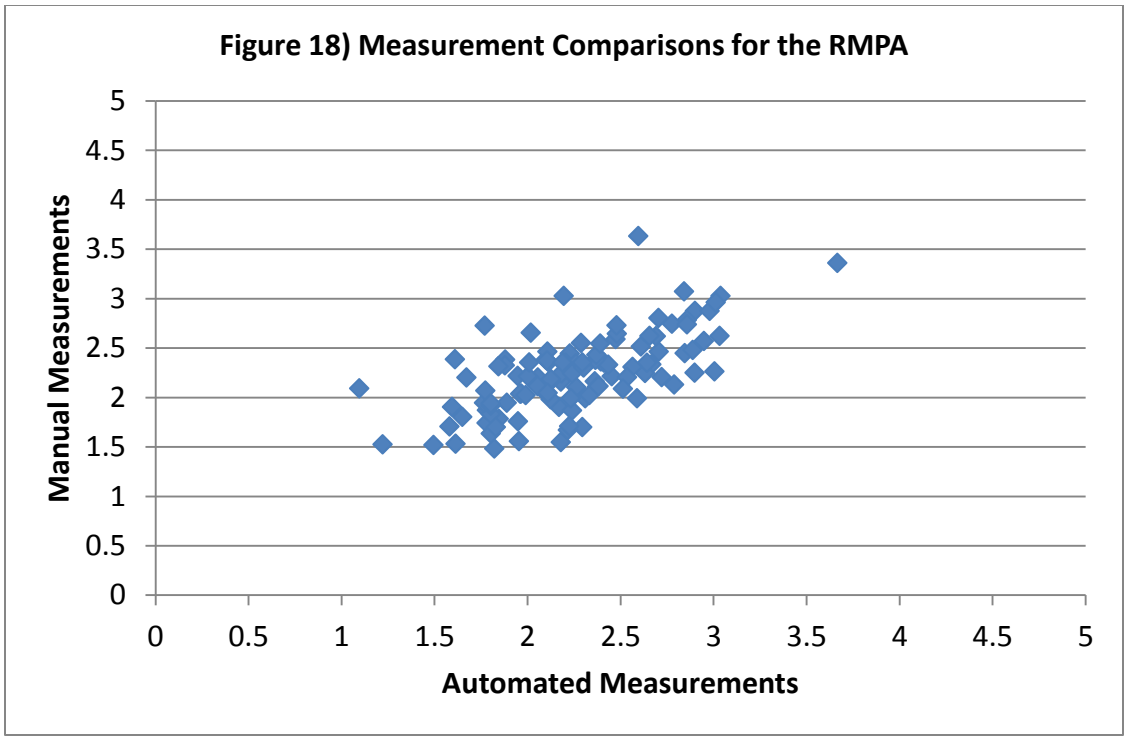


Figure 18) Measurement Comparisons for the RMPA
Manual Measurements
Automated Measurements
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

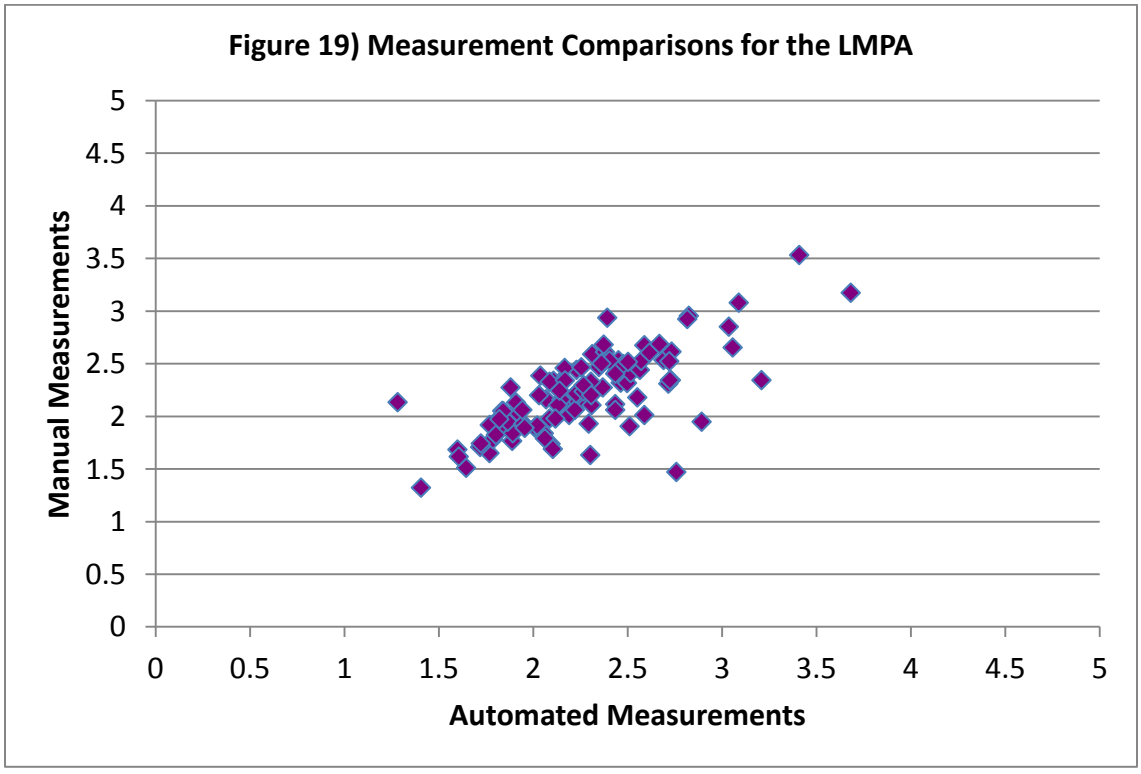
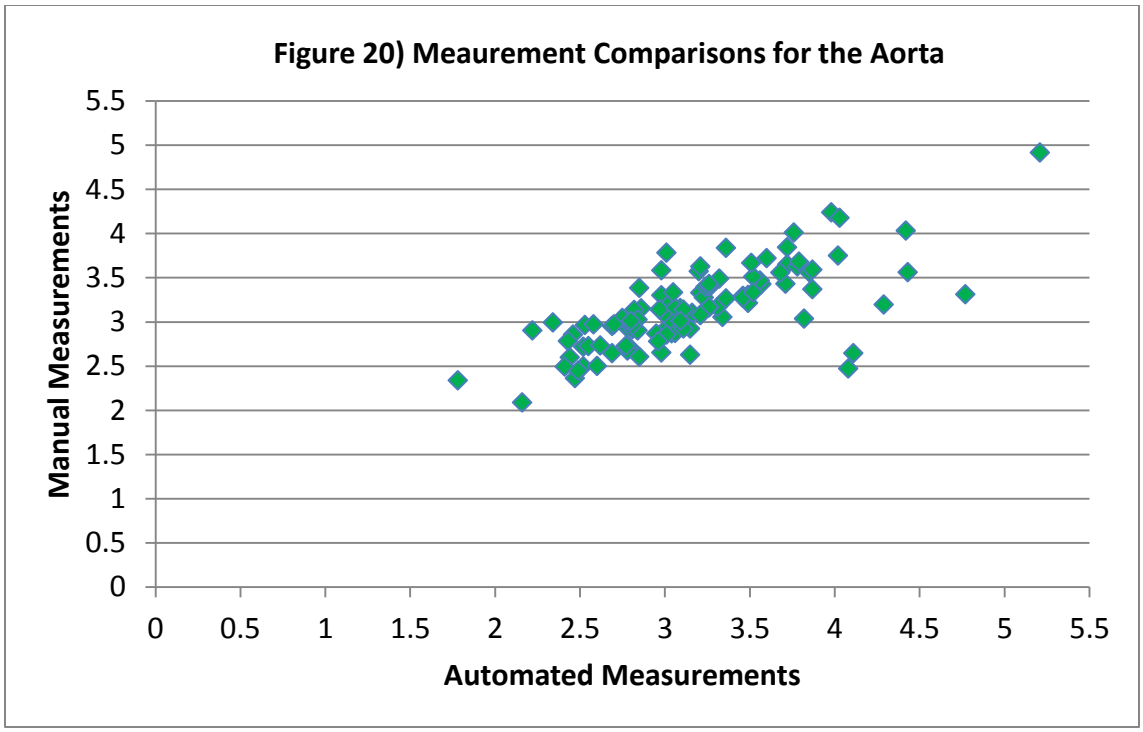


Figure 19) Measurement Comparisons for the LMPA
Manual Measurements
Automated Measurements
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5



4.5.5 Models for Predicting Disease State with Automated versus Manual Measurements

The first set of models generated used the automated measurement data and they include a logistic regression model followed by a naïve Bayes that according to Weka literature follows a Gaussian distribution and the default classification cutoff threshold value of 0.5 was used. The complete output for both of these models using both the original and the normalized data for both measurement types can be found in Appendix K. According to the output generated by Weka, there was no difference between the use of the normalized data or the original data for either the automated or manual measurement datasets. The models developed using the automated measurements are discussed first. The logistic regression model performed slightly better than the Naïve Bayes model For the logistic regression model, out of the 88 control cases it classified

85 of them correctly; unfortunately, it only classified 4 out of 21 of the disease cases correctly; giving a specificity of 96.6% but a sensitivity of 19%. The naïve Bayes model performed better: it classified 80 of the controls correctly and 9 of the disease cases correctly, giving a specificity of 90.9% and a sensitivity of 42.9%. The sensitivity and specificity values are shown in the output, but were also calculated based on the information shown in the confusion matrix.

The findings for the models built using the manually measured data appear to be quite similar. This was expected because as we have previously shown in Section 4.5.4, the automated measurements are not significantly different from the manual measurements.

Table 19). Summary of ROC Characteristics

	Logistic Regression				Naïve Bayes			
	Manual		Automated		Manual		Automated	
	Original	Normal	Original	Normal	Original	Normal	Original	Normal
Accuracy	83.4%	83.7%	79.82%	79.82%	81.2%	84%	77.98%	77.98%
SN	28.6%	28.6%	19%	19%	47.6%	52.4%	42.9%	42.9%
SP	94.3%	93.2%	96.6%	96.6%	92%	85.2%	90.9%	90.9%
Mean Absolute Error	0.2281	0.2286	0.2811	0.2812	0.2096	0.2047	0.2687	0.2696
ROC Area	83.7%	83.7%	69.4%	69.5%	81.2%	84%	68.1%	68.2%

4.5.6 Polynomial Regression Model for Predicting Pressure

Table 20 gives the R (column 2) and adjusted R^2 (column 3) statistics. These values summarize the goodness of the polynomial fit to the observations. The closer the values are to 1 the better the fit, with a value of 1 indicating a perfect fit. The adjusted R^2 is similar to R^2 except it accounts for the number of predictors in the model allowing adjusted R^2 (column 4) statistics from models with a different number of predictors to be compared, where R^2 values cannot. In reference to the table, each of the pressure variables (PA-systolic, PA-diastolic, TPG, and VPR) showed adjusted R^2 and R^2 values close to 1. The aorta measurement variable gave the poorest

values which were expected since the aorta should not be affected by PAH, it is used for normalization across patients. The standard error of the estimate is a measure of accuracy of the predictions; the smaller the value, the more accurate. In this case the measurements appear to have the lower error rate and seem to be better predictors when it comes to the pressure.

In addition to the R statistics, analysis of variance was also used to test the hypothesis that the polynomial fit is a better fit than the mean. The total variance, the variance of the predictor fitted to just the mean, is partitioned into variance explained by the polynomial regression model and residual variance (the difference from the fitted line to the observations). An *F*- test then compares the variances to determine if they are significantly different. The F statistic shows the ratio of the variances (column 6 in Table 20), and the *p*- value (column 7) the probability that the polynomial fit is no better than fitting to the mean. If the *p*- value is significant then polynomial fit is better than the mean [68].

The complete output from SPSS for the model is found in Appendix L.

Table 20). Output from the polynomial regression model for both sets of measurements

Variable	**R**	**R^2**	**Adjusted R^2**	**Std. Error of the Estimate**	**F-value**	**p-value**
Automated Measures						
Aorta	.412	.170	.062	.385	1.572	0.223
PT	.619	.383	.302	.497	4.756	0.10
RMPA	.712	.507	.443	.312	7.895	0.001
LMPA	.758	.575	.519	.309	10.362	0.000
Manual Measures						
Aorta	.485	.236	.136	.352	2.364	.097
PT	.753	.568	.511	.380	10.066	.000
RMPA	.656	.431	.356	.392	5.801	.004
LMPA	.778	.605	.553	.276	11.734	.000

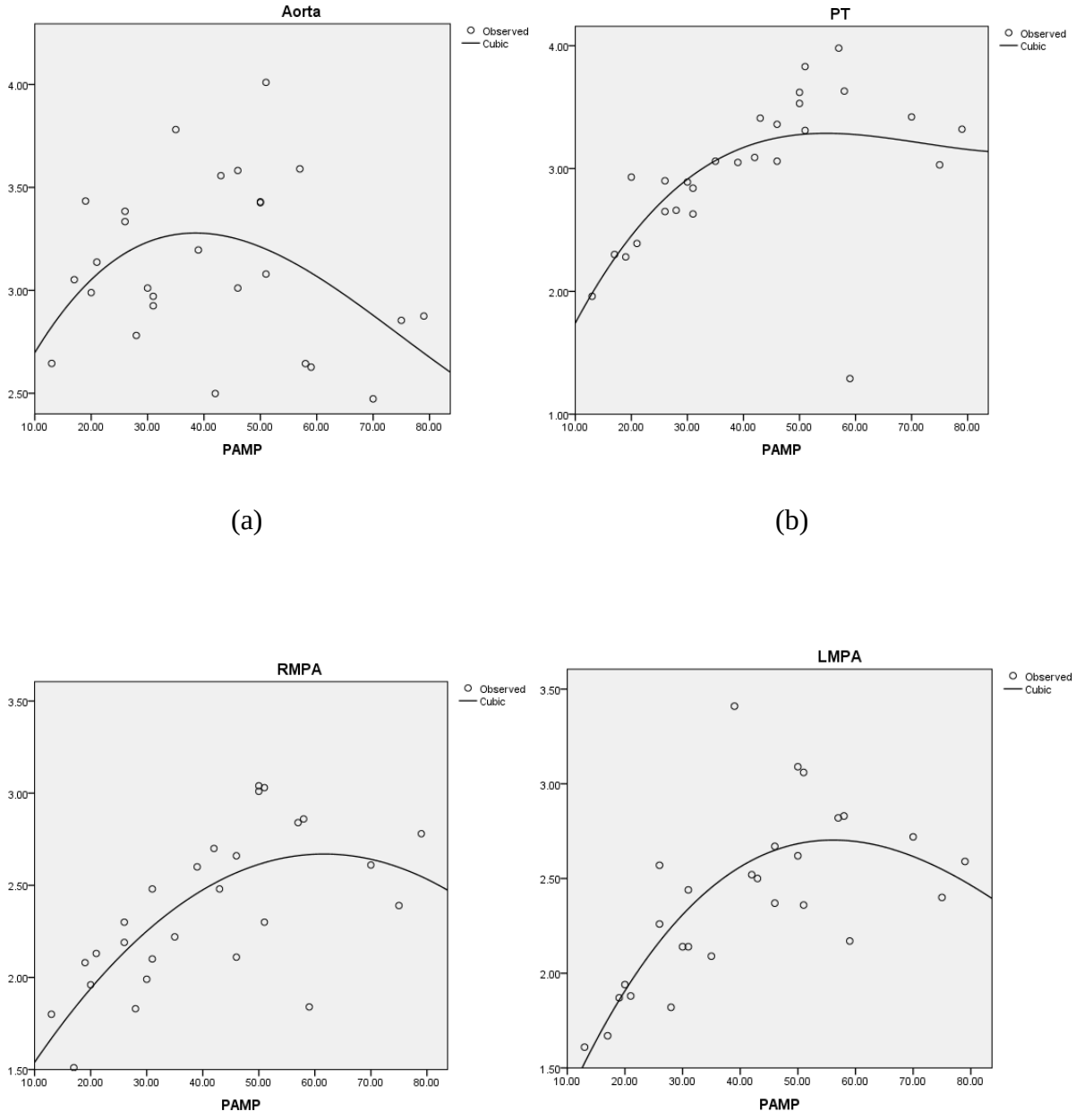
Table 21. Output from the polynomial regression model for the pressure measurements.

Variable	R	R^2	Adjusted R^2	Std. Error of the Estimate	F-value	p-value
TPG	.973	.946	.939	4.702	135.303	0.000
VPR	.933	.871	.854	2.679	51.631	0.000
PA_Systolic	.981	.962	.958	6.107	196.412	0.000
PA_Diastolic	.920	.846	.826	5.317	42.057	0.000

The results in Table 20 are comparable across the different sources of measurements. The results for the PT, RMPA and LMPA are all larger or closer to 1 than the aorta values.Since the aorta is not affected by disease we would expect these values to be close to zero. The manual aorta measurements did not give R^2 values as close to zero as the automated aorta giving values of 0.236 and 0.170 respectively. The corresponding p-values were 0.223 amd 0.97 both greater than 0.05 and therefore the aorta measurement is not a predictor of pulmonary artery mean pressure. This will be disussed at length in the discussion in section 5.2.1. It has been stated that disgnosis of pulmonary hypertension is done via these specific pressure values. The results shown in Table 21 were as expected, the pressure values seem to give the best fit for the model with the R^2 values closest to 1 and p-values less than 0.05.

Figure 21 shows the model plots for each variable as generated in SPSS.

Figure 21. Model plots by measurement variable.

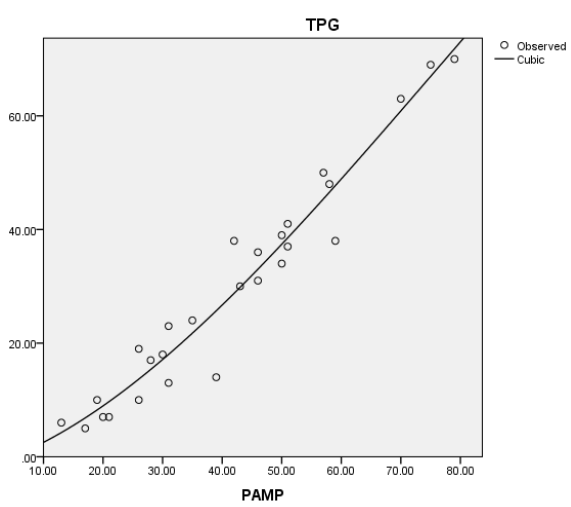


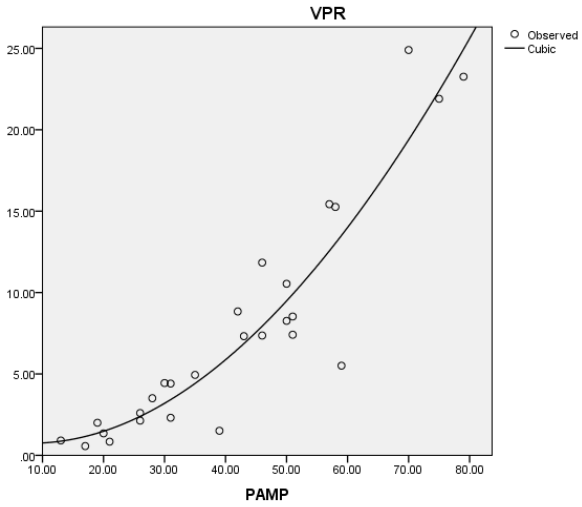
(a) (b)

(c) (d)

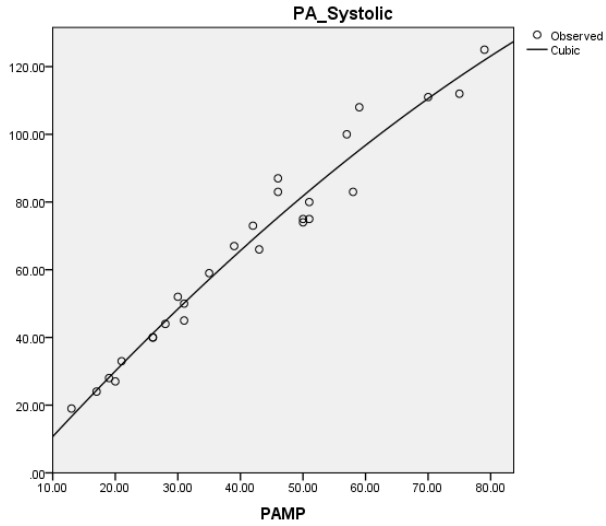
* For each plot the x axis is the pulmonary artery mean pressure (PAMP) measured by catheterization in mmHg units. For plots a-d, the y-axes are the different automated vessel diameter measurements in cm.

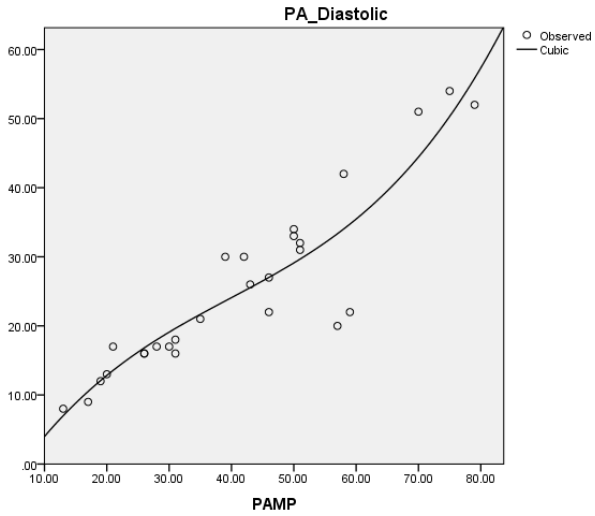
Figure 22. Model plots by pressure variables.





(a) (b)





(c) (d)

* For each plot the x axis is the pulmonary artery mean pressure (PAMP) measured by catheterization in mmHg units. Plot a plots the trans-pulmonary gradient (TPG) also in mmHg units along the y axis and plot b shows the vascular pulmonary resistance (VPR) in Woods units.

Plots c and d have along the y axis the systolic and diastolic pressures by catheterization in mmHg units.

Referring to the plots specific to the measurements plots a-d in Figure 21, there appears to be a functional relationship between the pulmonary artery mean pressure and the RMPA and LMPA measurements in the 2.5 cm region. A functional relationship means that when the vessels are at a particular size there is a particular range of pressures associated with them.

The plots are generated using the automated data; the manual measurements would have shown a stronger relationship according to the values in Table 19. The aorta measurements as expected are scattered and no direct relationship can be seen. For the PT measurements in plot b, there are a few outliers that may be skewing the data and will be discussed in Section 5.2.1. The question we are asking is can we predict the pulmonary artery mean pressure from the vessel diameters? Since we can as previously reported diagnose disease state and now we have identified a functional relationship between the vessel diameters and the pressure, the future step would be to use the combination of the vessel diameters to classify pressure based on the polynomial regression model.

5.0 DISCUSSION

5.1 HYPOTHESIS 1

The primary goal of this research was to be able to accurately model and obtain vasculature measurements of the three primary arteries of the pulmonary vasculature anatomy that in turn could be used for the disease classification and pressure prediction of PAH. If such classification and prediction could be done well, it has the potential to aid in the early diagnosis and characterization of PAH without the need for invasive catheterization. The gold standard for diagnosis of PAH is through right heart catheterization (RHC), an invasive and costly procedure, where pressure measurements are made directly within the affected vessels. During RHC a catheter is passed in to the right side of the heart through the ventricle to the pulmonary trunk to monitor the blood flow as it is being pumped to the lungs as well as cardiac output, the hearts function. The fact that PAH is associated with the remodeling of the pulmonary arteries, raises the possibility of quantifying the vessel geometry depicted in CT images as an alternative, non-invasive technique for diagnosing PAH.

5.1.1 Specific Aim 1: Create a repository of CTPA exams. The exams will consist of cases that are both positive for PAH and are negative for PAH

Specific aim 1 is self-explanatory, however, the problem encountered during data gathering is not. As shown there was plenty of data to perform all of the tasks outlined in this research with exception of the pressure data. Although the pressure data does not come into play until the end of the second hypothesis, I would like to discuss it now while on the subject of the data. Due to the low prevalence of PAH, it takes years to gather cases. We ended up with 22 PAH cases that were positive for disease with accompanying CT images. The problem lies with the acquisition of the control cases because the majority of people that undergo the right heart catheterization procedure do so because elevated pressures are expected, resulting in a low number of normal cases partnered with CT exams. Due to the risk of the procedure, it is rarely done unless warranted based on an underlying disease involving the heart or the lungs and in most cases a normal pressure is a rare event. Although the normal cases with corresponding pressure are limited, it did not affect the outcomes presented in this work. According to the power calculation results shown in Section 4.1 both the pressure dataset and the measurement dataset gave a power of 1.0.

5.1.2 Specific Aim 2: Create and validate PUMA, a PUlmonary Mapping and Analysis tool that semi-automatically generates pulmonary vascular models.

To achieve the end result of a semi-automated model, numerous steps had to be taken. One of the more difficult steps was in the development of the segmentations. The vasculature was segmented as described in the methods in Section 3.2.1. The main problem with this process

was choosing an intensity value that would reduce the amount of “bleeding” into the surrounding anatomy. The intensity values of the pulmonary vasculature are almost the same as the surrounding tissues, making it extremely difficult to segment out the entire vessel of interest and not include portions of the heart or the venous system, a problem encountered with the studies discussed in Section 2.2.1. Due to this complexity, we were not able to fully automate this process as we had originally proposed. Instead we took on the painstakingly tedious job of manually segmenting the pulmonary vasculature as described in Section 3.2.1. Having a successful automated segmentation method for separating the pulmonary vasculature is a popular problem and unfortunately like other researchers we had limited success in finding a solution due to the complexity of the vasculature and the low variability in the image intensities of the various tissues in the thoracic cavity.

The next step was to extract the skeletons from the segmentations. Here we encountered our next set of problems. The resulting segmentation had holes and errors within the segmentation that developed during the skeleton generation process. Three rounds of cleaning and inspection reduced these imperfections but did not eliminate them entirely. To try and eliminate these imperfections we applied a variety of preprocessing methods to help smooth out the segmentation and get a more accurate depiction of the vasculature’s backbone as detailed in Section 3.2.3.

We found that a combined method of median filtering and morphological closing reduced the number of the unwanted edges the most. Although the preprocessing greatly reduced these unwanted edges, additional methods were needed. After the graphs were generated, we manually deleted the extra unwanted edges that remained after preprocessing, as described in Section

3.2.6. We then used this data to develop a classifier that separates the edges into those that should remain (positive) and those that should be deleted (negative).

According to the results shown in Table 6 in Section 4.4, the ROC area for the CART model is 89.3% and the logistic regression model gave 84.8%. An online calculator developed at Vassar College (http://www.vassarstats.net/roc_comp.html) was used to test whether these ROC areas are statistically significantly different. According to the calculator these areas are statistically different giving p-values less than 0.05 at 0.04.

Seven features proved to be strong enough for classifying edges. The features used in the development of the pruning model were reasonably discriminatory. However, future automated pruning algorithms could consider the anatomic locations (depth in the vascular tree) of the edges. Had we incorporated this feature, it would have been much easier to identify the vessels we wanted to keep simply by counting down the branches and referring to the three dimensional coordinate locations.

A possible problem with the pruning data may be the editing of the graph edges. Due to human error it is possible that edges remained in the dataset that should have been eliminated that were missed, resulting in edges predicted to be negative edges that were labeled as positive , and vice versa. An additional problem that led to the retention of unwanted edges occurred during the manual editing process; some edges that should have been eliminated were not visualized and led to retaining some unwanted edges. Specifically, there were a number of times that an edge with the path length of zero was listed and kept in the edited graph because it was not visualized to be removed during the manual editing process. Why these zero path length edges are listed is not clear. This is an area for further exploration. A more detailed analysis of the individual edges, perhaps with the addition of more descriptive features, such as proximity to

the larger vessels, as well as branching depths, would help construct a more accurate predictive model. This is an area of the research that needs further exploration to find a less tedious method of pruning the unwanted edges.

Future exploration would be to take these classifiers and apply them to the original un-cleaned graphs to see how well they can classify the wanted from unwanted edges, and how successful they are in pruning the graphs.

At the completion of hypothesis 1, although the model of the vasculature is imperfect, it is a semi-automated model that depicts the three major vessels of interest allowing us to proceed with the second hypothesis and final aim.

5.2 HYPOTHESIS 2

5.2.1 Specific Aim 3: Use PUMA to diagnose PAH. This will be done by performing a semi-automated measurement of vascular diameters and making comparisons to measurements from known normal cases.

The main goal of this research is to answer the question can we predict the presence of pulmonary hypertension based on CT images of the pulmonary vasculature.

When looking at the measurements, specifically Table 3 in Section 3.3.8, we would expect the overall minimum and the overall maximum measurement values to be greater for the disease cases than the normal cases. This is in fact what is shown. The effect of pulmonary hypertension on the vessels is similar to the effects of stretching a steel spring. As the spring stretches past its plastic deformation limit, it does not return to its original size. This is what is

happening in the vessels. The increase in the pressure distends the vessels and prolonged exposure reduces the chances that the vessel will begin to lose its elasticity and start to atrophy. An area of future research related to this was touched upon in the background. It would be interesting to look at the difference in the diameter of the vessels across the cardiac cycle. One measurement would be taken during systole and one during diastole. The difference between these two measurements would be taken and used for study. Just like the balloon example the difference is expected to be greater for control patients or patients with mild hypertension compared to those with more advanced disease. One possible avenue for future work would be to add more vessel measurements to the model. If we were to proceed deeper into the vasculature and obtain measurements at deeper branching depths and include those values in the development of the prediction model, it may improve the accuracy of the model. In particular, due to the nature of the disease the deeper vessels will show the effects of disease if pulmonary hypertension is present making it easier to classify disease from controls. Signs of pulmonary hypertension are first seen in the most distal vessels or deepest branches and travels up the tree until it begins to affect the elasticity of the pulmonary trunk [1]. Obtaining information from deeper branching depths would be expected to also improve the model when predicting pressure based on the measurements because it would give a more complete picture of the effects of the remodeling and the severity of the disease. Incorporating this additional information into the regression model would allow the comparison of more severe changes in the vasculature sizes to be compared to the pressure changes, it seems plausible that there would be a larger difference between the normal cases and the disease cases. This would not only improve the classification of the model but it would open up the possibility of classifying the degree of severity of the disease by tracking the progression as we travel along the branches of the pulmonary tree. Of

course, the problem with doing this task would be the segmenting out of those additional levels. I have shown that it can be done manually, but not easily or quickly.

When comparing disease and control cases for each of the vessels, we expected to see that they are significantly different with the exception of the aorta comparisons. This was the case when analyzing the manual measurement data: there is a difference between the control and the disease cases shown in Tables 12 and 13 in Section 4.5.2. The same conclusions are made when looking at the automated measurements in Tables 14 and 15 in Section 4.5.3. Each of the vessels, excluding the aorta, showed a significant difference between the disease and control cases. The p-values are all less than 0.05. For both the manual and automated measurements the aorta showed no significant difference between disease cases and control cases. When comparing the manual measurements to the automated measurements there are variations in the measurements. In Table 16 in Section 4.5.4, we report that there is no significant difference between the two types of measurements in regards to the pulmonary trunk and the right main pulmonary artery. In Table 17 we show the findings for the comparison for left main pulmonary artery and the aorta; here there is a significant difference found with respect to the p-values, both are greater than 0.05. The equivalence analysis performed for these comparisons reported in Table 18 in the same section show that the measurements are comparable and fall within the designated threshold across all confidence intervals. The causes for possible differences between the measurements may have to do with the location of where the measurement is being made within the vessel. Future work to address this problem may be to have the automated measurement made at the same location in the vessel that corresponds to the slice where the manual measurement is made. This would eliminate the possible variable of measurement location affecting the results. Figures 17-20 are scatterplots of the manual measurements on the

y-axis and the automated measurements along the x-axis. These plots show the variation in values. Specifically Figures 19 and 20, the outliers are easily identified where the automated measurements are much greater than the manual measurements. Consistent anatomic location for measurement may be able to reduce this variation and help reduce the measurement error.

We investigated disease classification using machine learning. Two different methods of classification, logistic regression and naïve Bayes were applied. Due to the variation between the two types of measurement as previously discussed we developed models for both the manual and automated measurements. We found that for both classifiers (using normalized data) the manual measurements classified more cases correctly with an ROC area of 83.7% for the logistic regression model and 84% for the naïve Bayes. . The automated measurements resulted in a ROC area for the logistic regression model of 69.5% and 68.2% for the naïve Bayes model. These results are encouraging. If improvements are made to this process such as the segmenting of additional branches, this may further improve classification. It is very possible that some of the disease cases are in the beginning stages and segmenting out the higher order vessels alone may not show enough difference from the normal cases to diagnose disease. The model may only be classifying the more severe cases, since we do not have a severity of disease attribute and there is no way of knowing for sure if this is the case; only future research will resolve this issue.

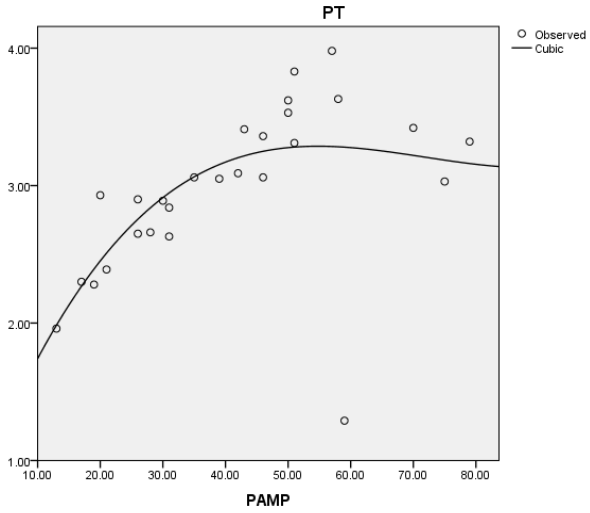
Trying to predict a relationship between pressure and size proved difficult. It is plausible that the inclusion of gender, weight and height data would have improved the relationship by helping to define the baseline size of the pulmonary vessels; however, this data was not available to us. The polynomial regression model indicated that pressure was a good classifier of disease. However, the measurements were not nearly as strong and this was the case for both types of measurements (manual and automated). An important limitation is that we only had five normal

cases with pressures and measurements and 27 disease cases with pressures. Depending on the severity of disease, it is possible that the three vessels we measured were not yet affected enough by disease to reveal the disease reliably. When looking at the output from Table 23 in Section 4.5.6 for the measurements, as the vessels decrease in size with depth in the pulmonary branches the R^2 values increase.

Section 2.2.1 described a study by Devaraj et. al.. They manually measured the segmental and subsegmental arteries, which are branches deeper than those presented here. Then using linear regression they assessed the relationship of the measurements with the mean pulmonary artery pressure. Their main focus was determining the ratio of the main pulmonary artery and the ascending aorta because it is a strong indicator of the presence of pulmonary hypertension in patients with pulmonary fibrosis. We did not specifically look at this ratio but it would be an area for future exploration with this dataset. Our primary focus was to measure the change in vessel diameter based on branching depth and the difference between normal and disease cases based on vessel diameters at the different branching levels. The majority of their findings are not comparable directly because of the stated differences, however, they did report that the correlation between the segmental artery and the mean pulmonary artery pressure was found to have an $R^2 = 0.19$, p-value = 0.001 in 56% of their patients. They found no correlation for the subsegmental arteries [26]. According to our results shown in Table 23, we found a relationship between the mean pulmonary artery pressure and the PT, RMPA, and LMPA with the exception of the automated PT measurement. These arteries are higher in the pulmonary tree and would be less affected by disease than the deeper branches. The exception of the PT value may have to do with the potential for extreme outliers because of the method of measurement. Looking at the following Figure 23, the same plot found in Section 4.5.6, we can see that at around 60 mmHG

there are two extreme PT measurements: one very small around 1.5 cm and one very large around 4.0 cm. Since these measurements are automated it is possible that there are errors in the anatomic location of the measurements in some of the cases resulting in erratic measurements. This is an area for additional research in the future as discussed earlier.

Figure 23 Plot of PT versus mean pulmonary artery pressure.



Another study mentioned in Section 2.2.1 was Grubstein et al.. This group measured the same vessels as we did and reported a correlation between the pulmonary arteries and pressure but not as significant a correlation as were the pressures measured by echocardiography.

6.0 CONCLUSION

In conclusion, PAH is a progressive and potentially fatal disease. PAH causes the remodeling of the pulmonary vasculature and because of this we were able to quantify the vessel geometry depicted in CT images as a non-invasive technique for diagnosing PAH. The research presented in this work provides the foundation for future work of the fully automated measurements of the major vessels for diagnosis of disease. We have shown that using measurements of the blood vessels are strong indicators for presence of disease, and we are one step closer to eliminating the need for the invasive and costly right heart catheterization procedure. Even with the limitations of this research we were able to classify just under 75% of the patients correctly when using the automated measured data as to whether or not the disease is present. If we build upon this work we should include travelling deeper into the vasculature and gathering data on deeper branches. For the disease cases, the deeper branches may better reveal the effects of the disease and therefore, be better indicators for and improve the classification of PAH cases. Of course, gathering additional data, specifically pressures from normal cases, would be optimal. Although we have shown there is a relationship between the vessel size and the mean pulmonary artery pressure, the increase in branching depth along with more data would vastly improve this relationship due to the nature of the disease. The deeper vessels will show the effects of the disease before the higher level branches that were segmented in this research. It is entirely possible that incorporating this information would give us the ability to predict the

pressure accurately within a specified range, thus removing the necessity for RHC. This work lays a useful foundation for future work, but improvements need to be made to the generation of the automated measurements, specifically consistency needs to be maintained in regards to anatomic location to make the values comparable to manual measurements to make this a fully automated process.

APPENDIX A

Table 22). Pressure data for the 27 specified cases

Image	Catheter Date	PA Systolic Pressure	PA Diastolic Pressure	PAMP	TPG
1	2/25/2005	73	30	42	38
3	5/16/2005	125	52	79	70
5	5/26/2005	45	18	31	23
6	6/1/2005	83	42	58	48
8	6/27/2005	40	16	26	19
9	7/7/2005	33	17	21	7
10	7/11/2005	111	51	70	63
11	8/11/2005	27	13	20	7
13	9/22/2005	100	20	57	50
15	10/11/2005	28	12	19	10
19	11/17/2005	75	33	50	39
20	12/15/2005	74	34	50	34
23	1/10/2006	40	16	26	10
24	3/2/2006	24	9	17	5
25	3/22/2006	112	54	75	69
27	12/27/2006	75	31	51	37
28	5/2/2007	59	21	35	24
29	5/17/2007	19	8	13	6
32	1/15/2008	44	17	28	17
33	10/1/2009	80	32	51	41
34	10/16/2009	83	27	46	36
35	11/17/2009	66	26	43	30
36	1/26/2010	108	22	59	38
37	5/4/2010	52	17	30	18
38	5/7/2010	67	30	39	14
40	6/23/2010	50	16	31	13
41	6/25/2010	87	22	46	31

*PAMP ~ Pulmonary Artery Mean Pressure, TPG ~ Trans-pulmonary Gradient

Table 23). Additional Pressure Data collected for the 27 cases.

Image	PVR	Days between RHC and CT	CT Date	Case Label
1	8.84	-2	2/23/2005	PH
3	23.26	+3	5/19/2005	PH
5	4.41	+12	6/8/2005	PH
6	15.26	-32	4/30/2005	PH
8	2.6	+2	6/29/2005	PH
9	0.84	0	7/7/2005	Normal
10	24.9	+3	7/14/2005	PH
11	1.35	-8	8/3/2005	Normal
13	15.43	0	9/22/2005	PH
15	2	-12	9/29/2005	Normal
19	10.54	+8	11/25/2005	PH
20	8.26	+1	12/16/2005	PH
23	2.13	+30	2/9/2006	PH
24	0.57	0	3/2/2006	Normal
25	21.9	-1	3/21/2006	PH
27	8.53	+1	12/28/2006	PH
28	4.94	+1	5/3/2007	PH
29	0.91	+1	5/18/2007	Normal
32	3.51	0	1/15/2008	PH
33	7.41	+1	10/2/2009	PH
34	11.84	-1	10/15/2009	PH
35	7.32	0	11/17/2009	PH
36	5.51	0	1/26/2010	PH
37	4.44	0	5/4/2010	PH
38	1.51	0	5/7/2010	PH
40	2.31	0	6/23/2010	PH
41	7.36	0	6/25/2010	PH

*PVR~ Pulmonary Vascular Resistance, RHC ~ Right Heart Catheterization

APPENDIX B

This Section lists the parameters used to generate each of the segmentations for both the pulmonary vasculature and the aorta described in Section 3.2.1. The ROI value or region of interest is the value associated with the placement of the red box as demonstrated in Figure 4b. The lower and upper are the bounds for the intensity value used for differentiating between the vasculature to be segmented out and the surrounding tissues. The curvature value refers to the smoothness of the expansion of the bubbles that are placed as seed points to begin the segmentation. We found the closer to 1 the value the more smooth, which proved to result in cleaner segmentations.

Table 24. Segmentation Parameters

Image	ROI	Pulmonary Vasculature			Aorta		
		Lower	Upper	Curvature	Lower	Upper	Curvature
3	0.814453	-161.37	1364.73	1	70.87	468.98	1
9	0.582031	69.34	577.08	1	69.34	577.08	1
34	0.53125	101.11	859.26	1	58.99	859.26	1
39	0.564453	-43.99	937.87	1	-83.27	1252.07	1
46	0.523438	46.1	715.14	1	46.10	715.14	1
57	0.650391	120.38	966.61	0.92	819.20	505.03	0.98
105	0.714844	186.12	577.47	1	55.67	512.25	1
109	0.796875	20.01	668.23	0.97	60.83	587.79	0.97
111	0.568359	148	601	0.96	146.15	435.82	0.98
119	0.976562	128.97	444.61	1	160.54	318.35	1
127	0.857422	60.69	718.66	1	60.69	796.07	1
136	0.78125	54.55	558.53	1	121.75	457.73	1
142	0.537109	252.94	1115.42	0.95	6.52	417.23	0.98
146	0.486328	321.31	903	1	94.18	903	1
148	0.859375	48.17	664.09	0.96	23.70	556.01	0.97

Image	ROI	Pulmonary Vasculature			Aorta		
		Lower	Upper	Curvature	Lower	Upper	Curvature
156	0.585938	175.75	793.18	0.92	11.10	875.5	0.96
169	0.734375	17.55	551.32	1	89.20	658.07	1
184	0.708984	111.51	967.21	1	111.51	967.21	1
196	0.511719	269.2	816.66	1	184.97	732.44	1
201	0.703125	89.9	533.28	1	9.16	775.18	1
202	0.613281	56.13	765.42	1	91.59	800.89	1
207	0.625	95.89	547.49	1	50.62	660.32	1
209	0.703125	-12.39	546.13	1	-12.39	865.29	1
218	0.576172	133.84	833.35	1	92.69	874.5	1
225	0.662109	224.62	561.59	1	71.45	500.32	1
228	0.78125	78.93	1380.18	0.99	43.76	817.48	0.98
231	0.511719	100.73	431	1	11.00	1153	1
239	0.626953	30.08	730.41	1	30.08	730.41	1
253	0.703125	-13.42	918.63	1	-225.25	1172.83	1
259	0.802734	101.93	327.11	1	69.76	455.79	1
278	0.523438	164.46	717.33	0.99	206.99	674.8	0.98
282	0.585938	-25.23	635.74	0.95	52.54	557.98	0.97
288	0.78125	290.26	427.25	0.98	85.28	358.86	0.99
294	0.613281	94.64	850.51	1	-31.34	1228.45	1
322	0.724609	27.68	973.38	1	64.05	718.77	1
328	0.714844	155.11	535.31	0.99	171.67	297.95	0.98
353	0.703125	87.43	491.46	0.98	53.78	188.44	0.99
367	0.78125	-128.85	341.26	1	51.96	811.37	1
369	0.630859	173.68	523.61	1	138.68	978.51	1
378	0.558594	167.39	1615.27	1	-2.94	1189.42	1
381	0.601562	101.74	1074.75	1	61.20	628.79	1
392	0.625	64.75	622.68	1	64.51	478.83	1
393	0.619141	64.51	622.68	1	64.51	478.83	1
410	0.564453	301.77	1009.72	1	78.20	1046.98	1
441	0.617188	112.6	663.08	1	112.6	663.08	1
453	0.503906	19.63	989.68	0.96	146.15	567	0.98
455	0.533203	216.82	845.65	1	216.82	845.65	1
460	0.533203	166.24	576.1	1	43.28	699.06	1
462	0.535156	193.39	753.49	0.92	153.39	553.46	0.98
471	0.912109	187.14	737.92	1	125.94	309.54	1
473	0.78125	17.85	725.53	1	55.09	837.27	1
482	0.12891	91.8	1093.73	1	-68.51	652.88	1
499	0.585938	-21.73	35.39	0.96	176.83	653.37	0.96
502	0.533203	79.36	1151.05	1	117.63	768.3	1

Image	ROI	Pulmonary Vasculature			Aorta		
		Lower	Upper	Curvature	Lower	Upper	Curvature
518	0.703125	235.17	513.2	1	26.65	443.69	1
522	0.6875	210.35	1019.59	1	-87.79	295.54	1
527	0.708984	86.69	536.61	1	-25.79	911.54	1
542	0.654297	81.96	644.32	0.91	38.55	363.14	0.98
547	0.712891	300.91	1209.54	1	82.84	700.7	1
550	0.847656	126.87	400.45	1	-9.92	1084.39	1
555	0.556641	155.34	889.45	0.95	155.34	673.54	0.96
564	0.642578	191.08	585.31	0.92	-6.03	322.49	0.96
569	0.578125	293.96	962.8	1	43.14	461.17	1
572	0.599609	284.89	951.14	0.96	76.68	618.01	0.98
588	0.701172	33.72	677.58	1	67.61	642.03	1
593	0.654297	17.9	734.3	1	-24.24	355.03	1
611	0.650391	178.58	896.17	1	-20.75	497.51	1
612	0.566406	67.58	737.25	1	-44.04	774.46	1
624	0.603516	189.06	708.14	0.91	-90.45	268.92	0.96
626	0.652344	97.45	922.24	0.96	14.97	757.28	0.99
628	0.607422	175.27	1139.65	1	7.55	552.63	1
629	0.748047	68.35	925.02	1	31.11	403.57	1
637	0.625	58.51	656.44	1	25.3	1121.5	1
650	0.642578	96.42	353.75	1	59.66	464.03	1
671	0.529297	67.77	546.59	1	27.87	626.39	1
672	0.632812	169.17	371.39	1	-100.48	843.25	1
679	0.517578	418.54	1087.15	1	84.23	794.63	1
680	0.642578	114.27	821.59	1	-3.62	978.77	1
686	0.966797	99.62	333.53	0.98	118.49	660.69	0.99
690	0.78125	64.17	805.58	1	-9.97	768.51	1
816	0.78125	109	659	0.91	67.00	532	0.98
818	0.507812	284.85	928.11	0.92	124.03	606.48	0.98
820	0.78125	41.7	1152.69	1	80.01	846.21	1
821	0.564453	396.94	926.48	0.92	152.00	926	0.96
825	0.632812	201	609	0.9	37.54	323.65	0.98
829	0.976562	-3.06	184.31	1	104.01	425.22	1
838	0.78125	50.99	527.23	0.92	50.00	527	0.99
878	0.828125	-31.14	263.32	0.96	99.73	328.76	0.98
896	0.660156	179.9	736.73	0.96	60.68	657.2	0.98
916	0.548828	167.08	710.13	0.92	125.28	668.35	0.96
919	0.683594	181.94	524.32	0.91	105.86	600.41	0.98
RV01	0.632812	220.35	566.21	0.98	143.49	912.08	0.97
RV03	0.587891	151.92	764.52	0.92	229.07	730.58	0.96

Image	ROI	Pulmonary Vasculature			Aorta		
		Lower	Upper	Curvature	Lower	Upper	Curvature
RV05	0.578125	200.42	438.17	0.97	200.42	540.08	0.96
RV06	0.625	90.83	873.23	0.92	59.54	560.27	0.96
RV08	0.621094	162.68	484.09	1	122.51	684.97	0.99
RV09	0.658203	121.03	337.83	1	229.43	410.1	1
RV10	0.695312	192.94	810.86	0.95	192.94	772	0.9
RV11	0.777344	80.89	552.94	1	80.89	485.51	1
RV13	0.541016	242.3	890.93	1	204.14	738.31	1
RV15	0.591797	151.79	1226.31	1	82.46	394.42	1
RV19	0.488281	276.9	756.44	0.9	92.46	387.56	0.9
RV20	0.703125	184.41	481.93	0.9	110.03	333.17	0.92
RV23	0.525391	164.23	566.27	1	3.42	1410.56	1
RV24	0.488281	210.18	511.14	1	172.56	661.62	1
RV25	0.619141	208.71	856.66	0.92	208.00	856	0.92
RV27	0.617188	156.7	437	0.92	611.90	53.42	0.92
RV28	0.541016	150.54	368.21	1	107.00	803.56	1
RV29	0.501953	428.45	1183.49	1	145.31	711.59	1
RV32	0.578125	190.17	867.7	1	99.28	641.86	1
RV33	0.333984	136.38	459.27	1	182.51	874.41	1
RV34	0.384766	217.37	821.2	0.92	217.00	670.24	0.96
RV35	0.609375	258.84	636.5	1	371.93	560.91	1
RV36	0.330078	247.44	617.51	1	136.42	1838.74	1
RV37	0.488281	475.95	1108.81	1	249.93	701.97	1
RV38	0.753906	161.68	578.72	0.96	161.00	578	0.96
RV40	0.726562	235.96	844.75	0.92	45.71	502.3	0.92
RV41	0.662109	316.7	1139.52	1	171.49	849.11	1

APPENDIX C

This Section contains all of the PUMA scripts referenced throughout this work.

PreprocessSegmentation.py (referenced in Section 3.2.3)

```
#!/usr/bin/env python
"""
This package selects a random  set of points from an image and writes them to
disk either in an XML scheme that is compatiable with the MITK mps data structure
or as a cPickle file wiwth my CritPoint structure.

Points are selected from regions that satisfy simple threshold criteria specified
by the user. Optional morphological operators can be applied to modify the
regions prior to sampling
"""
import itk
import sys
import os

from optparse import OptionParser

def getParser():
    """create an OptionParser instance and create parsing rules"""
    try:
        parser = OptionParser()
        parser.add_option("-k", "--kernel", dest='kernel',nargs=3,
                          default=(1,1,1),type='int')
        parser.add_option("-f", "--file", dest='fname',default='')
        parser.add_option("-m", "--median",action='store_true', dest='median',default=False)
        parser.add_option("-c", "--close",action='store_true', dest='close',default=False)
        return parser
    except Exception,  error:
        print "failed to generate parser",  error
```

```
def dilateMask(fname, median, closing, kernel):
    """dilate the binary mask passed in with the positional argument img.
    Prior to dilation an itk.BinaryMedianImageFilter is used with a (1,1,1)
    kerenl

    Dilation is applied using the itk.BinaryDilateImageFilter"""
    # If user didn't provide a kernel size, use the default value
    reader = itk.ImageFileReader.IUC3.New()
    reader.SetFileName(fname)
    fmod = ""
    # get image range

    img = reader.GetOutput()
    img.Update()
    rlow,rhigh = itk.range(img)
    if( median ):
        print "running median"
        fmod += "_median_True"
        median = itk.BinaryMedianImageFilter.IUC3IUC3.New()

          median.SetInput(img)
          median.SetRadius([1,1,1])
          median.SetForegroundValue(rhigh)
          median.SetInput(img)
          img = median.GetOutput()
    else:
        fmod += "_median_False"
    img.Update()
    if( closing ):
        print "running closing"
        fmod += "_closing_True_kernel_%d_%d_%d"%(kernel[0],kernel[1],kernel[2])
        closing = itk.BinaryMorphologicalClosingImageFilter.IUC3IUC3SE3.New()
        closing.SetForegroundValue(rhigh)
        kd = closing.GetKernel()
        kd.SetRadius(kernel)
        closing.SetKernel(kd)
        closing.SetInput(img)
        img = closing.GetOutput()
    else:
        fmod += "_closing_False"
    img.Update()
    # this is a really, really slow step
    writer = itk.ImageFileWriter.IUC3.New()
    tmp = fname.split(".nii.gz")
    writer.SetFileName(tmp[0]+fmod+".nii.gz")
    writer.SetInput(img)#closing.GetOutput())
    writer.Update()
```

```
def main():
    parser = getParser()
    (options, args) = parser.parse_args()
    print options
    dilateMask(options.fname, options.median, options.close,options.kernel)


if __name__=='__main__':
    main()
```

GenerateGraph.py (referenced in Section 3.2.5)

```
#!/usr/bin/env python
from vasctrees.SkeletonGraph import SkeletonGraph
import imageTools.ITKUtils.io as io
import sys
import scipy.ndimage as ndi
import numpy as np

simg,descr = io.readImage( sys.argv[1],
                           returnITK=False,
                          imgMode='uchar',
                          returnDescriptors = True)
try:
    simg
except:
    sys.exit("read of skeleton image failed")

sg = SkeletonGraph(img=simg,
                   spacing=descr['scale'],
                   origin=descr['origin'],
                   orientation=descr['orientation'],
                   label = sys.argv[1])

print descr
print sg.spacing, sg.origin, sg.orientation

oimg = io.readImage( sys.argv[2],returnITK=False,imgMode='uchar')
try:
    oimg
except:
    sys.exit("read of mask image failed")
```

```
sg.setOriginalImage(oimg)
sg.getGraphsFromSkeleton(verbose=False)
sg.setLargestGraphToCurrentGraph()
sg.graphs.keys()
sg.findEndpointsBifurcations()
root = sg.selectSeedFromDFE()
#sg.viewGraph()
#raw_input('continue')
sg.setRoot(root,key="mp_graphs")
sg.traceEndpoints(key='mp_graphs')
ogkey = sg.getLargestOrderedGraphKey()
sg.deleteDegree2Nodes(ogkey)
sg.prunePaths(ogkey)
sg.deleteDegree2Nodes(ogkey)
sg.fitEdges(key=ogkey)
print "Define orthogonal planes"
sg.defineOrthogonalPlanes(ogkey)
# Now get surface points of original image to map to the centerlines
dfe = ndi.distance_transform_cdt(oimg)
points_toMap = np.array(np.nonzero(np.where(dfe==1,1,0))[::-1]).transpose().astype(np.int32)
print "mapVoxelsToGraph"
sg.mapVoxelsToGraph(points_toMap,ogkey)
print "assignMappedPointsToPlanes"
sg.assignMappedPointsToPlanes(ogkey)
sg.saveCompressedGraphs(sys.argv[3])
```

rerootGraph.py (referenced in Section 3.2.5)

```
#!/usr/bin/env python
import networkx as nx
import gzip
import cPickle
import sys
import numpy as np
import vasctrees.viewGraph as viewGraph
def readGraphs():
    try:
        fo = gzip.open(sys.argv[1],"rb")
        data = cPickle.load(fo)
        fo.close()
    except:
        fo = file(sys.argv[1],"rb")
        data = cPickle.load(fo)
        fo.close()
    return data
def rerootGraph(graph,newRoot):
    og = graph.copy()
    oldRoot = og.graph['root']
    path = nx.shortest_path(og,oldRoot,newRoot)
    og.graph["root"] = newRoot
    for i in xrange(len(path)-1):
        d = og[path[i]][path[i+1]]
        og.remove_edge(path[i],path[i+1])
        og.add_edge(path[i+1],path[i],attr_dict=d)
# I wonder if I need to reorder the edge attributes?
    return og
```

```
def main():
    data = readGraphs()
    num = int(sys.argv[2])
    label = sys.argv[3]
    graph = data['orderedGraphs'][(num,label)]
    endp = [n for n in graph.nodes() if graph.degree(n)==1]
    endpa = np.array(endp)
    medianx = np.median(endpa[:,0])
    print "computing based on graph median of",medianx
    endp.sort(key=lambda n: abs(n[0]-medianx))
    newRoot = endp[0]
    ng_medianx = rerootGraph(graph,newRoot)
    ng_medianx.graph["root description"] = "root based on graph median x"
    viewGraph.viewGraph2(ng_medianx, root=ng_medianx.graph['root'])
    data['orderedGraphs'][(num,label+"_reroot_medianx")] = ng_medianx
    meanx = np.mean(endpa[:,0])
    print "computing based on graph mean of",meanx
    endp.sort(key=lambda n: abs(n[0]-meanx))
    newRoot = endp[0]
    ng_meanx = rerootGraph(graph,newRoot)
    ng_meanx.graph["root description"] = "root based on graph mean x"
    viewGraph.viewGraph2(ng_meanx, root=ng_meanx.graph['root'])
    data['orderedGraphs'][(num,label+"_reroot_meanx")] = ng_meanx

    fo = gzip.open(sys.argv[1],"wb")
    cPickle.dump(data,fo)
    fo.close()
```

editGraph.py (referenced in Section 3.2.6)

```
#!/usr/bin/env python
"""A simple class for viewing and editing the directed graphs generated with a
SkeletonGraph object. Relies on mayavi/vtk for the visuzalization. My understanding of
the mayavi interface is quiet limited. Program will occassionally freeze for no
apparent reason
The program is run as a script with three arguments:
skeleton-graph-file object-number graph-key
skeleton-graph-file: a pickle file generated with a SkeletonGraph save
object-number The object number from which the graphs were generated.
(this refers to the sequence of labeled-objects used to generated the graph
and is the integer in the tuple-key used in SkeletonGraph object)
graph-key the string label associated with the graph.
If the graph is modified, a new graph is added to the SkeletonGraph object with
a graph-key equal to the specified graph-key concatenated with '_edited'"""
import cPickle
import numpy as np
import sys
from optparse import OptionParser
from vasctrees.SkeletonGraph import SkeletonGraph
from mayavi import mlab

class GraphViewer(object):
    def __init__(self,fname, objectnum = -1, keyname = ''):
        self.fname = fname
        self.efile = file(fname+".error","w")
        self.stderr = sys.stderr
        f = file(fname)
        self.data = cPickle.load(f)
        f.close()
        if( objectnum == -1 or keyname == ''):
            self.key = getOrderedGraphKeys(self.data['orderedGraphs'])
        else:
            self.key = (objectnum,keyname)
        self.sg = SkeletonGraph()
        self.sg.orderedGraphs = self.data['orderedGraphs']
        self.sg.roots = self.data['roots']
        if(not self.sg.orderedGraphs[self.key].graph.has_key("root")):
            self.sg.orderedGraphs[self.key].graph["root"] = self.sg.roots[self.key]

        #self.picker = self.figure.scene.get()['picker']
        #self.figure.scene.picker.pointpicker.add_observer('EndPickEvent',self.picker_callback)
        self._setGraphData()
        # Decrease the tolerance, so that we can more easily select a precise
        # point.
        self.editnum = 1
        # create a new key and copy of graph
        newKey = (self.key[0],"%s_edited"%(self.key[1],))
        self.sg.orderedGraphs[newKey] = self.sg.orderedGraphs[self.key].copy()
        self.key = newKey
        self.deleteNode = None
```

```
    def saveModifiedData(self):
        print "saving modified data"
        self.data['orderedGraphs'] = self.sg.orderedGraphs
        f = file(self.fname,"wb")
        cPickle.dump(self.data,f)
    def _setGraphData(self):
        self.og = self.sg.orderedGraphs[self.key]
        self.edges = self.og.edges(data=True)
        self.nodes = self.og.nodes(data=True)

    def _clearGraphDrawables(self):
        self.npts = 0
        self.node_points = 0
        self.surfaces = 0
        self.lines = 0
        self.crds = 0

    def drawGraph(self):
        self.figure = mlab.figure(self.editnum, fgcolor=(0, 0, 0), bgcolor=(1, 1, 1))
        self.gcf = mlab.gcf()
        #mlab.clf()
        self.picker = self.figure.on_mouse_pick(self.picker_callback)
        self.picker.tolerance = 0.01
        self.figure.scene.disable_render = True
        #self.gcf.scene.picker.pointpicker.add_observer("EndPickEvent",self.picker_callback)
        node_color = []
        self.crds = {}

        # create drawables for bifurcations and endpoints
        self.narray = np.zeros((len(self.nodes),3),dtype=np.float32)
        s = np.zeros((len(self.nodes),))
        for i in range(self.narray.shape[0]):
            n = self.nodes[i]
            self.narray[i,:] =n[1]['wcrd']
            dg = self.og.degree(n[0])
            if( dg == 1 ):
                wc = n[1]['wcrd']
                txt = "(%d,%d,%d)"%(n[0][0],n[0][1],n[0][2])

                #self.crds[n[0]] = mlab.text3d(wc[0],wc[1],wc[2],
                #                              "this is a text", color=(0,0,0), scale=2.0)
                s[i] = 1.0
            else:
                s[i] = 0.5
        self.npts = mlab.points3d(self.narray[:,0],self.narray[:,1],self.narray[:,2], s,colormap="gray", scale_factor=4.0)
        self.node_points = self.npts.glyph.glyph_source.glyph_source.output.points.to_array()
```

```
        self.surfaces = {}
        self.lines = {}
        colors = ((1,0,0),(0,1,0),(0,0,1),(1,1,0),(1,0,1),(0,1,1))
        counter = 0
        for e in self.edges:
            try:
                clr = colors[counter%len(colors)]
                counter += 1
                mp = e[2]["mappedPoints"]#np.concatenate((mp,e[2]["mappedPoints"]),axis=0)
                sp = e[2]['d0']
                x = mp[::4,0]
                y = mp[::4,1]
                z = mp[::4,2]
                #Visualize the points
                pts = mlab.points3d(x, y, z, scale_mode='none', scale_factor=0.2, opacity=0.0)

                # Create and visualize the mesh
                # redirect sys.stderr for this step
                sys.stderr = self.efile
                mesh = mlab.pipeline.delaunay3d(pts)
                self.surfaces[(e[0],e[1])] = mlab.pipeline.surface(mesh,color=clr,opacity=0.1)
                sys.stderr = self.stderr
                pts = 0
                self.lines[(e[0],e[1])] = mlab.plot3d(sp[0],sp[1],sp[2],color=clr, tube_radius=1.0)
            except KeyError:
                pass
        #mlab.view(47, 57, 8.2, (0.1, 0.15, 0.14))
        self.figure.scene.disable_render = False
        print "ready for editing"
        mlab.show()
```

```
    def picker_callback(self, pckr):
        # currently this callback gets disconnected when I clear the figure
        # so I am compensating by creating a new figure to draw on with each call
        print "in picker_callback3"
        delta = 2
        surfaceKeys = self.surfaces.keys()
        for sk in surfaceKeys:
            try:
                if( pckr.actor in self.surfaces[sk].actor.actors ):
                    print "we picked a surface"
                    print sk
                    newNode = sk[1]
                    if( self.sg.orderedGraphs[self.key].degree(newNode) == 1 ):
                        if( self.deleteNode == newNode):
                            #mlab.clf(self.editnum)
                            self.sg.pruneSpecifiedDegreeOneNode(self.key,self.deleteNode)
                            self.sg.remapVoxelsToGraph(self.key)
                            self.editnum += 1
                            self.saveModifiedData()
                            self._setGraphData()
                            self._clearGraphDrawables()
                            self.drawGraph()
                            self.deleteNode = None

                        else:
                            self.deleteNode = newNode
                            wcrd = self.sg.orderedGraphs[self.key].node[newNode]['wcrd']
                            outline = mlab.outline(line_width=3)
                            outline.outline_mode = 'cornered'
                            outline.bounds = (wcrd[0]-delta, wcrd[0]+delta,
                                              wcrd[1]-delta, wcrd[1]+delta,
                                              wcrd[2]-delta, wcrd[2]+delta)
                    else:
                        print "This is not a degree 1 node. Degree = %s"%self.sg.orderedGraphs[self.key].degree(newNode)
            except KeyError:
                pass
                # Getting strange key error on redrawing figure after editing.
                # I wonder if this is a qued click from the previous window
```

```
def getOrderedGraphKeys(ogs):
    keys = ogs.keys()
    txt = "Select number of desired key:\n"
    for i in range(len(keys)):
        txt += """%d\t\t%s\n"""%(i,keys[i])
    while(True):
        try:
            keyNum = input(txt)
            if( 0 <= keyNum and keyNum < len(keys) ):
                return keys[keyNum]
        except:
            pass
    return None
def getParser():
    try:
        parser = OptionParser()
        parser.add_option("-f","--file",dest='fname',
                          help='name or directory for fixedImage')
        parser.add_option("-o","--object_number",dest='objNum',type='int',default='-1')
        parser.add_option('-l','--label',dest='label',default='')

        return parser
    except Exception, error:
        print "failed in getParser", error
        sys.exit(0)

if __name__ == '__main__':
    parser = getParser()
    (options, args) = parser.parse_args()
    gv = GraphViewer(options.fname, objectnum = options.objNum, keyname=options.label)
    gv.drawGraph()
```

getDFEModified.py (referenced in Section 3.2.8.1)

```
#!/usr/bin/env python
import sys
import numpy as np
import scipy
import cPickle
from optparse import OptionParser
from vasctrees.SkeletonGraph import SkeletonGraph
import SimpleITK as sitk
import scipy.ndimage as ndi
import gzip

class GraphEdgeGrabber(object):
    def __init__(self,fname, dfe):
        self.fname = fname
        self.dfe = dfe
        f = file(fname)

#         try:
        fo = gzip.open(fname,"rb")
        self.data = cPickle.load(fo)
        fo.close()
#         except:
#             fo = file(fname,"rb")
#             data = cPickle.load(fo)
#             fo.close()
#         return data
#         self.data = cPickle.load(f)
#         f.close()
        #if( objectnum == -1 or keyname == ''):
            #    self.key = getOrderedGraphKeys(self.data['orderedGraphs'])
            #else:
                #    self.key = (objectnum,keyname)
        self.sg = SkeletonGraph()
        self.sg.orderedGraphs = self.data['orderedGraphs']
        self.sg.roots = self.data['roots']
        #if(not self.sg.orderedGraphs[self.key].graph.has_key("root")):
            #    self.sg.orderedGraphs[self.key].graph["root"] = self.sg.roots[self.key]

        #newKey = (self.key[0],"%s_with_dfe"%(self.key[1],))
        #self.sg.orderedGraphs[newKey] = self.sg.orderedGraphs[self.key].copy()
        #self.key = newKey
        #self.deleteNode = None

        #self._setGraphData()
```

```
    def saveModifiedData(self):
        print "saving modified data"
        self.data['orderedGraphs'] = self.sg.orderedGraphs
        f = file(self.fname,"wb")
        cPickle.dump(self.data,f)
    def _setGraphData(self):
        print self.sg.orderedGraphs.keys()
        for key in self.sg.orderedGraphs.keys():
            self.key = key
            print "processing graph",key
            self.og = self.sg.orderedGraphs[self.key]
            self.edges = self.og.edges(data=True)
            self.nodes = self.og.nodes(data=True)

            for e in self.edges:
                print "processing edge %s %s"%(e[0],e[1])
                try:
                    e[2]['dfe_values'] = [self.dfe[p[::-1]] for p in e[2]['path']]
                except KeyError:
                    pass
            #mlab.view(47, 57, 8.2, (0.1, 0.15, 0.14))


def getOrderedGraphKeys(ogs):
    keys = ogs.keys()
    txt = "Select number of desired key:\n"
    for i in range(len(keys)):
        txt += """%d\t\t%s\n"""%(i,keys[i])
    while(True):
        try:
            keyNum = input(txt)
            if( 0 <= keyNum and keyNum < len(keys) ):
                return keys[keyNum]
        except:
            pass
    return None

def getParser():
    try:
        parser = OptionParser()
        parser.add_option("-i","--img",dest='iname', help='name or directory for image')
        parser.add_option("-g","--graph",dest='gname', help='filename for graph')
        #parser.add_option("-o","--object_number",dest='objNum',type='int',default='-1')
        #parser.add_option('-l','--label',dest='label',default='')

        return parser
    except Exception, error:
        print "failed in getParser", error
        sys.exit(0)
```

```
def main():
    parser = getParser()
    (options, args) = parser.parse_args()
    img = sitk.ReadImage(options.iname)
    print "computing distance from edge map"
    dfe = ndi.morphology.distance_transform_edt(sitk.GetArrayFromImage(img))
    gv = GraphEdgeGrabber(options.gname, dfe)
    gv._setGraphData()
    gv.saveModifiedData()


if __name__ == '__main__':
    main()
```

grabVolumes.py (referenced in Section 3.2.8.1)

```
#!/usr/bin/env python
"""A simple class for grabbing the volume data associated with each
edge in the specified graph
The program is run as a script with three arguments:
skeleton-graph-file object-number graph-key
skeleton-graph-file: a pickle file generated with a SkeletonGraph save
object-number The object number from which the graphs were generated.
(this refers to the sequence of labeled-objects used to generated the graph
and is the integer in the tuple-key used in SkeletonGraph object)
graph-key the string label associated with the graph.
If the graph is modified, a new graph is added to the SkeletonGraph object with
a graph-key equal to the specified graph-key concatenated with '_volume'"""
import cPickle
import numpy as np
import sys
from optparse import OptionParser
from vasctrees.SkeletonGraph import SkeletonGraph
import imageTools.ITKUtils.io as io
import scipy.ndimage as ndi

class VolumeGrabber(object):
    def __init__(self,fname, objectnum = -1, keyname = '', img='',lvalue=1):
        self.fname = fname
        self.lvalue = lvalue
        f = file(fname)
        self.data = cPickle.load(f)
        f.close()
        self.imgfile = img
        if( objectnum == -1 or keyname == ''):
            self.key = getOrderedGraphKeys(self.data['orderedGraphs'])
        else:
            self.key = (objectnum,keyname)
        self.sg = SkeletonGraph()
        self.sg.orderedGraphs = self.data['orderedGraphs']
        self.sg.roots = self.data['roots']
        if(not self.sg.orderedGraphs[self.key].graph.has_key("root")):
            self.sg.orderedGraphs[self.key].graph["root"] = self.sg.roots[self.key]

        self._setGraphData()
        # create a new key and copy of graph
        newKey = (self.key[0],"%s_volume"%(self.key[1],))
        self.sg.orderedGraphs[newKey] = self.sg.orderedGraphs[self.key].copy()
        self.key = newKey
```

```
    def readImage(self):
        self.simg,self.descr = io.readImage( self.imgfile,
                         returnITK=False,
                        imgMode='uchar',
                        returnDescriptors = True)
        self.dfe = ndi.distance_transform_cdt(self.simg)
        cg = self.sg.orderedGraphs[self.key]
        self.sg.spacing=cg.graph["spacing"]
        self.sg.origin=cg.graph["origin"]
        self.points_toMap = np.array(np.nonzero(np.where(self.simg==self.lvalue,1,0))[::-1]).transpose().astype(np.int32)

    def mapVolumePoints(self,key="volumePoints"):
        #print "in grabVolumes",self.points_toMap
        self.sg.mapVoxelsToGraph(self.points_toMap,self.key,mp_key=key, verbose=True)
```

```
    def computeVolumeMeasures(self, key="volumePoints"):
        cg = self.sg.orderedGraphs[self.key]
        edges = cg.edges(data=True)
        for e in edges:
            try:
                mask = np.zeros(cg.graph["imgSize"],np.uint8)
                worldVolume = e[2][key]
 # now we need to transform the worldVolume coordinates back to image coordinates
                ivi = ((worldVolume-cg.graph["origin"])/cg.graph["spacing"]).astype(np.int32)
                mask[ivi[:,2],ivi[:,1],ivi[:,0]] = 1
                surface = mask - ndi.binary_erosion(mask)
 # now we've got img indicies we can grab the surface based on dfe values
                sinds =  np.nonzero(surface)
                dfeVals = self.dfe[sinds]
                edgeSurface = np.nonzero(dfeVals==1)
                surface_to_volume = float(sinds[0].size)/float(ivi[:,0].size)
                exterior_to_surface = float(edgeSurface[0].size)/float(sinds[0].size)
                e[2]["surface2volume"] = surface_to_volume
                e[2]["exterior2surface"] = exterior_to_surface
            except ValueError:
                e[2]['surface2volume'] = None
                e[2]['exterior2surface'] = None

    def saveModifiedData(self):
        print "saving modified data"
        self.data['orderedGraphs'] = self.sg.orderedGraphs
        f = file(self.fname,"wb")
        cPickle.dump(self.data,f)
    def _setGraphData(self):
        self.og = self.sg.orderedGraphs[self.key]
        self.edges = self.og.edges(data=True)
        self.nodes = self.og.nodes(data=True)
```

```
def getOrderedGraphKeys(ogs):
    keys = ogs.keys()
    txt = "Select number of desired key:\n"
    for i in range(len(keys)):
        txt += """%d\t\t%s\n"""%(i,keys[i])
    while(True):
        try:
            keyNum = input(txt)
            if( 0 <= keyNum and keyNum < len(keys) ):
                return keys[keyNum]
        except:
            pass
    return None
```

```
def getParser():
    try:
        parser = OptionParser()
        parser.add_option("-f","--file",dest='fname',
                          help='name or directory for fixedImage')
        parser.add_option("-o","--object_number",dest='objNum',type='int',default='-1')
        parser.add_option("-v","--value",dest="lvalue",type='int',default='1')
        parser.add_option("-i","--img",dest="img",default="",help="name or directory of segmented image")
        parser.add_option('-l','--label',dest='label',default='')

        return parser
    except Exception, error:
        print "failed in getParser", error
        sys.exit(0)

if __name__ == '__main__':
    parser = getParser()
    (options, args) = parser.parse_args()
    gv = VolumeGrabber(options.fname, objectnum = options.objNum,
                       keyname=options.label,img=options.img,
                       lvalue=options.lvalue)
    gv.readImage()
    gv.mapVolumePoints()
    gv.computeVolumeMeasures()
    gv.saveModifiedData()
```

Extract_DFE_Values.py (referenced in Section 3.2.8.1)

```
import sys
import cPickle

class ExtractDFEValues(object):
    """This class is defined for extracting scalar values from the DFE data"""
    def __init__(self, data):
        self.og=data['orderedGraphs'][(1,'mp_graphs')]
        self.edges=self.og.edges(data=True)
        print "There are",len(self.edges),"edges in the unedited graph"

    def EditedGraph(self, data):
        self.og2=data['orderedGraphs'][(1,'mp_graphs_edited')]
        self.edges2=self.og2.edges(data=True)
        print "There are",len(self.edges2),"edges in the edited graph"


    def CalculatePathLength(self):
        """CALCULATES THE NUMMBER OF DFE VALUES FOR EACH EDGE AND APPENDS THE VALUE TO THE EDGE IN A DICTIONARY."""
        count=0
        for edge in self.edges:
                count +=1
                dfe=edge[2].get('dfe_values')
            #print "dfe",dfe
                pathLength=len(dfe)
                print "For edge",count ,"the path length is", pathLength

    def CalculatePathLengthEdited(self):
            count=0
        for edge in self.edges2:
                count +=1
                dfe=edge[2].get('dfe_values')
                #print "dfe",dfe
            pathLength=len(dfe)
                print "For edge",count,"the path length is", pathLength
```

```
    def MinMaxDiff(self):
        """CALCULATES THE DIFFERENCE BETWEEN THE MINIMUM DFE VALUE AND THE MAXIMUM DFE VALUE."""
        count=0
        for edge in self.edges:
                count +=1

                dfe=edge[2].get('dfe_values')
            if len(dfe)=='0':
            'break'
            elif len(dfe)>0:
                minDFEvalue=min(dfe)
                maxDFEvalue=max(dfe)
                    Diff=maxDFEvalue-minDFEvalue
                    print "For edge",count,"the difference in the min/max dfe values is", Diff
            print "For edge",count,"the min value is", minDFEvalue
            print "For edge", count, "the max value is", maxDFEvalue
                else:
            print"Error in DFE data"

    def MinMaxDiffEdited(self):
        count=0
            for edge in self.edges2:
                count +=1
                dfe=edge[2].get('dfe_values')
                if len(dfe)=='0':
            'break'
            elif len(dfe)>0:
            minDFEvalue=min(dfe)
                    maxDFEvalue=max(dfe)
                    Diff=maxDFEvalue-minDFEvalue
                    print "For edge",count, "the difference in the  min/max dfe values is", Diff
                    print "For edge", count, "the min value is", minDFEvalue
            print "For edge", count, "the max value is", maxDFEvalue
            else:
            print "Error in DFE data"
```

```
def main():
    fo=open(sys.argv[1])
    data=cPickle.load(fo)
    q1= raw_input("Is there an edited graph? Please enter y or n ")
    if q1=='n':
        dfe=ExtractDFEValues(data)
        PathLength=dfe.CalculatePathLength()
        MinMaxDiff=dfe.MinMaxDiff()
    elif q1=='y':
        dfe=ExtractDFEValues(data)
        EG=dfe.EditedGraph(data)
        PathLength=dfe.CalculatePathLength()
        MinMaxDiff=dfe.MinMaxDiff()
        PathLengthEdited=dfe.CalculatePathLengthEdited()
        MinMaxDiffEdited=dfe.MinMaxDiffEdited()
    else:
        print "ERROR INVALID ENTRY"
        print "COMPLETE"

#if __name__ == "__main__" :
main()
```

ExtractShapeVolume.py (referenced in Section 3.2.8.1)

```
#!/usr/bin/env python
import sys
import cPickle

class ExtractShapeVolume(object):
    """This class is defined for extracting the shape and volume values from the DFE data"""
    def __init__(self, data):
        self.og=data['orderedGraphs'][(1,'mp_graphs_volume')]
        self.edges=self.og.edges(data=True)
        print "There are",len(self.edges),"edges in the unedited graph"

    def EditedGraph(self, data):
        self.og2=data['orderedGraphs'][(1,'mp_graphs_edited_volume')]
        self.edges2=self.og2.edges(data=True)
        print "There are",len(self.edges2),"edges in the edited graph"

    def ExtractShape(self):
        count=0
        for edge in self.edges:
            count +=1
            shape=edge[2].get('exterior2surface')
            print "For edge",count ,"the shape is", shape

    def ExtractShapeEdited(self):
        count=0
        for edge in self.edges2:
            count +=1
            shape=edge[2].get('exterior2surface')
            print "For edge",count ,"the shape is", shape

    def ExtractVolume(self):
        count=0
        for edge in self.edges:
            count +=1
            volume=edge[2].get('surface2volume')
            print "For edge",count ,"the volume is", volume


    def ExtractVolumeEdited(self):
        count=0
        for edge in self.edges2:
            count +=1
            volume=edge[2].get('surface2volume')
            print "For edge",count ,"the volume is", volume
```

```
def main():
    fo=open(sys.argv[1])
    data=cPickle.load(fo)
    q1= raw_input("Is there an edited graph? Please enter y or n")
    if q1=='n':
        dfe=ExtractShapeVolume(data)
        Shape=dfe.ExtractShape()
        Volume=dfe.ExtractVolume()
    elif q1=='y':
        dfe=ExtractShapeVolume(data)
        EG=dfe.EditedGraph(data)
        PathLength=dfe.ExtractShape()
        MinMaxDiff=dfe.ExtractVolume()
        PathLengthEdited=dfe.ExtractShapeEdited()
        MinMaxDiffEdited=dfe.ExtractVolumeEdited()
    else:
        print "ERROR INVALID ENTRY"
    print "COMPLETE"

#if __name__ == "__main__" :
main()
```

AutomatedMeasurements4.py (referenced in Section 3.4.2)

```
#!/usr/bin/env python
import cPickle
import sys
import matplotlib.pyplot as plt
#from viewSelectedEdge import GraphViewer
import numpy as np
import gzip
"""
The aorta often is just a single segment. So we would a separate set of rules for that
In the case of multiple segments with valid x values, pick the segment with the minimum z value.
I'm not sure that we always have the same orientation in z, but I'm going to start with
a simple assumption that the images are inferior to superior ordering
The minimum meadian z value
"""
def readGraphs(fname):
    try:

        fo2 = gzip.open(fname,"rb")
        data2 = cPickle.load(fo2)
        fo2.close()
    except Exception, error:
        print "error reading zip format",error
        fo2 = file(fname,"rb")
        data2 = cPickle.load(fo2)
        fo2.close()
    return data2

def getAllDescendents(g,n,descendents):
    """for graph g get all descendents of node n """
    successors = g.successors(n)
    if( not successors ):
        return descendents
    else:
        for s in successors:
            descendents.add(s)
            descendents = getAllDescendents(g,s,descendents)
        return descendents
def countDescendents(g,n):
    d = set([])
    d = getAllDescendents(g,n,d)
    return len(d)
def getMidPointIndex(edge):
    """
    Return the index of the midpoint on the fitted edge.
    Because the edges have been fit with a smoothing spline, the points are ordered
    and are ordered, so the midpoint can be found simply from ordering."""
    pp = edge[2]['planePoints']
    midpoint = len(pp.keys())/2
    return midpoint
```

```
def getDiameter(edge, mpoint):
    """
    Compute the diameter of an edge at measurepoint mpoint.
    The radius is estimated from the average dispersion of the
    mapped planePoints from the fitted centerline points.
    """
    mpps = edge[2]['planePoints'][mpoint]
    d0 = edge[2]['d0']
    mp=np.array([d0[0][mpoint],d0[1][mpoint],d0[2][mpoint]])
    mpp=np.array(mpps).transpose()
    radius=np.mean(np.sqrt(np.array( (mpp[0,:]-mp[0])**2+
                                     (mpp[1,:]-mp[1])**2+
                                     (mpp[2,:]-mp[2])**2 )))
    diameter=radius*2/ 10.0 # convert to diameter and report as cm
    return diameter

def measureDiameter(edge):
    """measure the diameter at the midpoint of the edge"""
    mp = getMidPointIndex(edge)
    return getDiameter(edge,mp)

class AutomatedMeasurements(object):
    """This Class is defined for making the automated measurements of the  midpoint of
    the ascending aorta as well as the midpoints of the pulmonary trunk, and the right
    and left main pulmonary arteries"""
    def __init__(self, pulmonaryData):
        #print aortaData['orderedGraphs'].keys()
        #num,label = input("enter number and label for key")
        num=1; label="test"
        #self.AortaOG=aortaData['orderedGraphs'][(num,label)]
        #self.AortaEdgeData=self.AortaOG.edges(data=True)
        #self.AED=self.AortaEdgeData[2]
        #self.AE=self.AED[2]# dictionary with keys ['mappedPoints', 'wpath',
        #'planePoints', 'p', 'path', 'd2', 'd0', 'd1']
        print pulmonaryData['orderedGraphs'].keys()
        #num,label = input("enter number and label for key")
        label = sys.argv[3]
        self.PVOG=pulmonaryData['orderedGraphs'][(num,label)]
        self.PVEdgeData=self.PVOG.edges(data=True)
        self.PVED=self.PVEdgeData[2]
        self.archLoc=[]
        self.RevisedarchLoc=[]
```

```
    def IdentifyPulmonaryArteries(self):
        """ """
        lrDesc = self.PVOG.successors(self.pulmonaryTrunk[1])
        # determine which value has the minimum x value and
        # which has the maximum x value
        # sort the descendents by the x index
        lrDesc.sort(key=lambda n: n[0])
        rightTerm = lrDesc[0]
        leftTerm = lrDesc[-1]
        rightEdgeData = self.PVOG.get_edge_data(self.pulmonaryTrunk[1],
                                                rightTerm)
        leftEdgeData = self.PVOG.get_edge_data(self.pulmonaryTrunk[1],
                                               leftTerm)
        self.rightPulmonaryArtery = (self.pulmonaryTrunk[1],
                                     rightTerm,
                                     rightEdgeData)
        self.leftPulmonaryArtery = (self.pulmonaryTrunk[1],
                                    leftTerm,
                                    leftEdgeData)

def main():
    if( len(sys.argv) < 1 ):
        # no files specified so use fixed files for debugging
        fname1 = "RV19_Aorta_skeleton_graphs.pckle"
        fname2 = "RV19_PV_skeleton_graphs.pckle"
    else:
        fname1 = sys.argv[1]
    print "Reading PV graph file"
    data2 = readGraphs(fname1)
    print "Graph files successfully read in"
    print "Identifing midpoint of ascending aorta"
    am = AutomatedMeasurements(data2)
    am.IdentifyPETrunkEdge()
    am.IdentifyPulmonaryArteries()
    print "Calculating PV Measurements"
    print "Trunk Diameter is",measureDiameter(am.pulmonaryTrunk)
    print "Right Diameter is",measureDiameter(am.rightPulmonaryArtery)
    print "Left Diameter is",measureDiameter(am.leftPulmonaryArtery)

if __name__ == '__main__':
    main()
```

APPENDIX D

The specific steps for the calculation of the statistical measures for comparison.

- Calculate the mean $\bar{X}$, variance (S^2) and standard deviation(S) for the two groups (with and without PH).

$$\bar{X} = (\sum_{i=1}^{n})/n$$

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})}{n}$$

$$S = \sqrt{s^2}$$

- Calculate the difference in means.
- Calculate the Pooled Standard Deviation: This will provide insight into how much variance exists between the groups.

$$Pstdev. = \sqrt{\left[\frac{[(n_h-1)*S_n^2]+[(n_{wh}-1)*S_c^2]}{n_h+n_{wh}-2}\right]}$$

$$n_h = number\ of\ cases\ with\ hypertension,$$

$$n_{wh} = number\ of\ cases\ without\ hypertension$$

- Calculate the Error in the Standard Deviation of the Mean Difference:

$$SDM\ (Standard\ Deviation\ of\ the\ mean\ difference)$$

$$= Pstdev * \sqrt{\left(\frac{1}{n_h} + \frac{1}{n_{wh}}\right)}\ Error\ mean\ Difference = SDM * t_{95}$$

t_{95} is the t value associated with 95% confidence with degrees of freedom of $n_h + n_{wh} - 2$.

- Calculate the t-statistic and compare it to the critical value and determine whether to accept or reject the null of equal means.

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(s_1^2 / n_1 + s_2^2 / n_2)}}$$

Determining statistical significance: if the difference in means is greater than the Error mean difference then the two means are statistically different at the 95% confidence level. This means that we are 95% confident that the difference in means is not due to random error.

APPENDIX E

This section contains the complete tables of the resulting edge counts for the different preprocessing methods, one for the pulmonary vasculature and one for the aorta as discussed in Section 3.2.3 and 4.2. The Tables 25 and 26 contain the edge counts that resulted from the use of the editgraph.py script as discussed in Section 3.2.6. on both the pulmonary vasculature and the aorta for each case. Table 27 reports the values for the 69 control cases in the training set, and Table 28 contains the edge totals for the 15 disease cases.

Table 25. Pulmonary Vasculature Preprocessing Edge Data

Image	Without Preprocessing	median only	close only	median & close	median & close with a kernel size of (2,2,2)
57	13	9	9	11	10
109	97	13	13	15	9
111	21	16	16	16	16
142	81	29	30	33	30
148	7	5	5	7	7
156	23	23	24	17	15
228	92	13	15	24	11
278	25	17	15	15	21
282	17	11	9	9	9
288	28	20	14	27	17
353	9	7	5	10	7
453	25	23	23	39	21
462	29	25	27	25	21
499	86	13	19	19	9
542	75	51	48	50	47
555	272	31	31	34	29
564	19	16	18	19	16

572	9	7	7	7	7
624	114	18	13	13	15
626	38	7	7	9	7
818	14	13	13	15	11
821	18	19	17	18	18
825	13	11	13	11	11
838	56	12	13	11	11
896	12	7	7	12	9
916	193	30	30	36	27
919	126	20	19	19	19
RV03	27	15	13	23	14
RV06	126	12	12	20	10
RV10	13	13	13	13	13
RV19	69	71	67	67	67
RV20	9	9	11	11	7
RV25	13	13	13	11	11
RV27	25	23	26	29	16
RV34	86	9	11	19	13
RV38	31	31	29	29	29
RV40	16	16	16	14	16

Table 26. Aorta Preprocessing Edge Data

Image	Without Preprocessing	median only	close only	median & close	median & close with a kernel size of (2,2,2)
57	3	3	4	3	3
109	1	2	1	2	2
111	67	20	14	9	9
142	16	11	14	6	6
148	2	ERROR	1	1	1
156	13	8	13	8	9
228	7	7	7	7	7
278	7	7	7	7	7
282	150	4	3	4	2
288	40	6	10	5	5
353	12	7	9	7	7
453	19	7	9	7	7
462	4	5	4	5	6
499	6	6	5	5	1
542	100	11	11	10	10

555	78	12	10	7	11
564	4	2	6	2	2
572	5	2	5	2	1
624	387	13	18	9	10
626	3	1	3	1	1
818	62	15	19	13	9
821	16	10	12	12	10
825	111	9	14	13	9
838	7	7	8	7	5
896	2	2	2	2	2
916	16	122	14	7	5
919	8	7	8	7	7
RV03	3	2	3	2	2
RV06	77	9	14	4	4
RV10	5	5	5	3	1
RV19	9	7	8	7	7
RV20	7	7	5	7	5
RV25	7	7	7	7	7
RV27	234	8	9	8	12
RV34	8	8	9	8	8
RV38	5	5	5	5	5
RV40	9	9	7	9	9

Table 27. The Edge counts for the control cases in the training dataset

	Aorta		**Pulmonary Vasculature**	
Image	**Unedited**	**Edited**	**Unedited**	**Edited**
3	3	NA	7	NA
9	3	2	3	NA
39	2	NA	23	NA
46	9	1	35	20
57	4	2	10	NA
105	1	NA	10	8
109	2	NA	5	NA
111	2	NA	12	NA
119	2	NA	5	NA
142	9	3	5	4
148	1	NA	5	NA
156	9	2	3	NA
201	2	NA	7	NA

202	2	NA	3	NA
207	3	2	11	9
209	1	NA	8	NA
223	7	2	8	NA
228	7	2	11	NA
231	4	2	5	4
239	3	NA	5	NA
253	4	2	13	NA
278	5	1	11	9
282	1	NA	3	NA
322	2	1	9	NA
328	3	NA	5	NA
353	7	2	4	3
369	8	3	14	12
378	4	2	19	17
392	2	NA	15	NA
410	3	2	22	NA
453	5	2	11	NA
455	2	NA	40	37
462	1	NA	14	NA
471	4	2	17	15
473	4	2	9	NA
482	2	NA	15	NA
499	2	NA	11	NA
518	5	2	11	NA
522	2	NA	11	NA
542	2	NA	8	NA
547	3	NA	7	NA
564	3	NA	15	13
569	6	3	10	NA
572	2	1	7	NA
588	1	NA	11	NA
593	8	2	22	NA
612	4	2	12	NA
626	1	NA	9	NA
628	3	NA	14	13
629	7	2	11	NA
650	9	1	19	11
679	4	2	19	15

680	1	NA	23	NA
686	3	NA	9	NA
690	2	NA	5	NA
818	8	1	13	NA
820	5	2	6	5
829	2	NA	9	NA
838	5	2	10	9
849	9	2	19	NA
878	7	2	11	NA
896	2	NA	9	NA
900	2	NA	9	NA
916	5	2	7	NA
919	2	NA	7	NA
RV09	3	NA	9	NA
RV11	2	NA	5	4
RV15	2	NA	12	9
RV24	3	2	13	NA
RV29	2	NA	10	9

Table 28. The Edge counts for the disease cases in the training dataset

	Aorta		**Pulmonary Vasculature**	
Image	**Unedited**	**Edited**	**Unedited**	**Edited**
1	5	3	8	7
3	5	2	9	7
6	3	NA	12	9
10	3	NA	11	9
19	6	2	67	NA
20	9	2	7	NA
23	4	3	20	19
25	9	2	11	NA
27	5	2	7	NA
32	5	2	15	NA
34	8	2	13	9
35	3	NA	11	NA
38	7	3	19	17
40	7	2	9	8
41	9	3	21	NA

*Unedited refers to the edge counts after preprocessing with additional edges that needed to be manually removed. Edited contains the edge counts after the use of editGraph.py.

APPENDIX F

This section contains the limited output for the classification models for pruning edges referred to in 3.8.2 and Section 4.4. Due to length the entire output is available upon request.

The output from Weka for the Simple CART model

```
=== Run information ===

Scheme:weka.classifiers.trees.SimpleCart -S 1 -M 2.0 -N 5 -C 1.0
Relation:     FeatureDataSetNormalizedCompleteRev
Instances:    1750
Attributes:   8
              PathLength
              MinMaxDiff
              Min
              Max
              Ratio to depth
              Shape
              Volume
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

CART Decision Tree

Volume=(0.263247598)|(0.338448776)|(0.155548493)|(0.228946966)|(0.20561028

1)|(0.211827489)|(0.401991288)|(0.262370062)|(0.374797407)|(0.307465205)|(

0.287486218)|(0.432587859)|(0.21388567)|(0.191460935)|(0.160623093)|(0……

Number of Leaf Nodes: 8

Size of the Tree: 15

Time taken to build model: 1851.4 seconds

=== Stratified cross-validation ===
=== Summary ===
```

```
Correctly Classified Instances        1628               93.0286 %
Incorrectly Classified Instances       122                6.9714 %
Kappa statistic                          0.8013
Mean absolute error                      0.0899
Root mean squared error                  0.2676
Relative absolute error                 25.1294 %
Root relative squared error             63.2978 %
Total Number of Instances             1750

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC
Area  Class
                 0.824     0.037      0.87      0.824     0.846      0.893
positve
                 0.963     0.176      0.947     0.963     0.955      0.893
negative
Weighted Avg.    0.93      0.144      0.929     0.93      0.93       0.893

=== Confusion Matrix ===

    a    b   <-- classified as
  336   72 |    a = positve
   50 1292 |    b = negative
```

The output from Weka for the Logistic Regression model

Time taken to build model: 1013.41 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	1489	85.0857 %
Incorrectly Classified Instances	261	14.9143 %
Kappa statistic	0.6117	
Mean absolute error	0.1551	
Root mean squared error	0.3734	
Relative absolute error	43.3628 %	
Root relative squared error	88.2984 %	
Total Number of Instances	1750	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.787	0.13	0.648	0.787	0.711	0.848	positve
	0.87	0.213	0.931	0.87	0.899	0.848	negative
Weighted Avg.	0.851	0.194	0.865	0.851	0.856	0.848	

=== Confusion Matrix ===

```
  a    b   <-- classified as
 321   87 |   a = positve
 174 1168 |   b = negative
```

```
=== Run information ===

Scheme:weka.classifiers.trees.SimpleCart -S 1 -M 2.0 -N 5 -C 1.0
Relation:     FeatureDataSet
Instances:    1546
Attributes:   8
              Pathlength
              MinMaxDiff
              Min
              Max
              Ratio to depth
              Shape
              Volume
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

CART Decision Tree
: positive(1342.0/204.0)

Number of Leaf Nodes: 1

Size of the Tree: 1

Time taken to build model: 1452.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1342               86.8047 %
Incorrectly Classified Instances       204               13.1953 %
Kappa statistic                          0
Mean absolute error                      0.2291
Root mean squared error                  0.3384
Relative absolute error                 99.8305 %
Root relative squared error             99.9999 %
Total Number of Instances             1546

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.868     1         0.929      0.494    positive
                 0         0          0         0         0          0.494    negative
Weighted Avg.    0.868     0.868      0.754     0.868     0.807      0.494
```

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1342    0 |    a = positive
  204    0 |    b = negative
```

```
=== Run information ===

Scheme:weka.classifiers.trees.SimpleCart -S 1 -M 2.0 -N 5 -C 1.0
Relation:     FeatureDataSetNormalized
Instances:    1546
Attributes:   8
              PathLength
              MinMaxDiff
              Min
              Max
              Ratio to depth
              Shape
              Volume
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

CART Decision Tree
: positive(1342.0/204.0)

Number of Leaf Nodes: 1

Size of the Tree: 1

Time taken to build model: 1228.66 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1342               86.8047 %
Incorrectly Classified Instances       204               13.1953 %
Kappa statistic                          0
Mean absolute error                      0.2291
Root mean squared error                  0.3384
Relative absolute error                 99.8305 %
Root relative squared error             99.9999 %
Total Number of Instances             1546

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.868     1         0.929      0.494    positive
                 0         0          0         0         0          0.494    negative
Weighted Avg.    0.868     0.868      0.754     0.868     0.807      0.494

=== Confusion Matrix ===

    a    b   <-- classified as
 1342    0 |    a = positive
  204    0 |    b = negative
```

The output from Weka for the Simple Cart model using the normalized dataset

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         572               65.3714 %
Incorrectly Classified Instances       303               34.6286 %
Kappa statistic                         -0.0568
Mean absolute error                      0.3522
Root mean squared error                  0.5677
Relative absolute error                 98.3873 %
Root relative squared error            134.25   %
Total Number of Instances              875

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.137     0.189      0.181     0.137     0.156      0.43     negative
                 0.811     0.863      0.756     0.811     0.782      0.43     positive
Weighted Avg.    0.654     0.706      0.622     0.654     0.636      0.43

=== Confusion Matrix ===

   a   b   <-- classified as
  28 176 |   a = negative
 127 544 |   b = positive
```

APPENDIX G

The following table contains the slice data, the levels at which the manual measurements were made for each image along with the series from which the measurements were made.

Table 29. Slice Data

Cases	Aorta	PT	RMPA	LMPA	Image Series
3	131	131	127	138	4
9	129	129	130	137	6
34	104	104	105	109	6
39	142	142	140	147	5
46	101	101	95	110	5
57	141	141	139	143	5
105	178	178	180	198	4
109	138	138	134	150	5
111	98	98	96	107	5
119	29	29	28	37	6
127	105	105	106	97	4
136	92	92	94	81	3
142	111	111	112	123	5
146	109	109	118	135	5
148	77	77	86	77	3
156	105	105	110	121	5
169	121	121	120	129	5
184	99	99	98	105	6
196	104	104	102	118	5
201	128	128	131	144	5
202	129	129	131	138	5
207	149	149	153	161	5
209	65	65	62	64	5
218	115	115	112	124	5
223	37	37	38	52	4
225	37	37	43	52	5

228	94	94	89	85	3
231	122	122	123	132	5
239	74	74	82	89	5
253	111	111	112	107	6
259	57	57	58	61	104
278	102	102	112	120	5
282	109	109	106	115	5
288	96	96	90	85	3
294	93	93	94	98	5
322	122	122	125	134	5
328	119	119	118	123	5
353	98	98	95	114	4
367	94	94	93	84	3
369	106	106	111	119	5
378	104	104	103	116	5
392	133	133	139	149	4
393	93	93	92	102	5
410	120	120	118	137	5
441	117	117	120	133	5
453	124	124	127	131	5
455	111	111	117	132	5
460	125	125	120	148	5
462	130	130	129	134	5
471	254	254	238	241	4
473	92	92	95	87	3
482	158	158	154	169	5
499	102	102	105	93	4
502	96	96	99	106	1
518	76	76	80	93	5
522	132	132	133	144	5
527	29	29	27	31	6
542	133	133	129	136	7
547	136	136	139	146	5
550	83	83	84	79	3
555	125	125	123	131	5
564	64	64	63	68	4
569	125	125	127	146	5
572	281	281	260	290	5
588	122	122	117	125	6

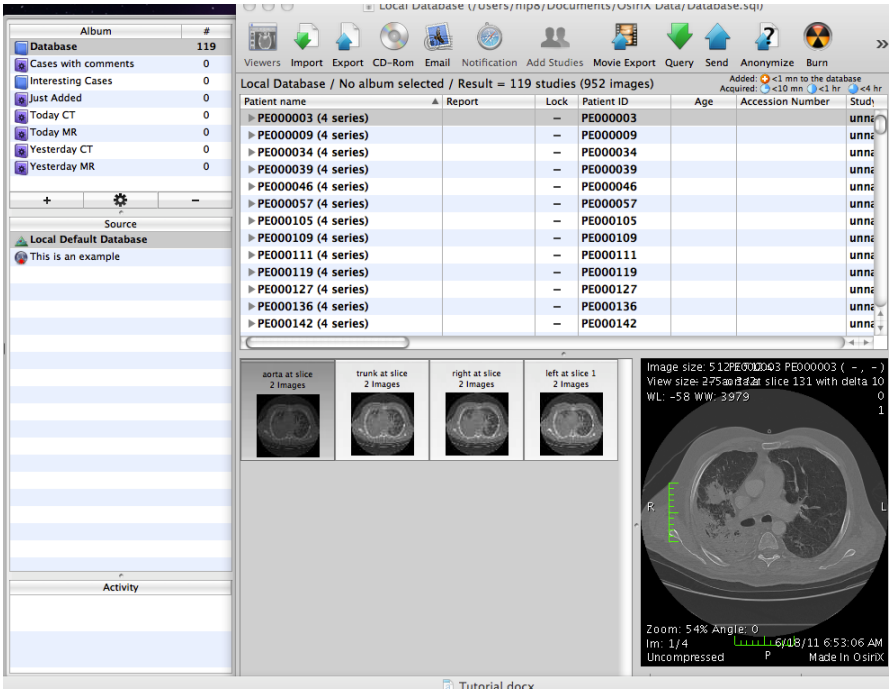
593	103	103	108	120	5
611	119	119	117	127	5
612	110	110	109	118	5
624	127	127	125	140	5
626	109	109	116	113	5
628	116	116	117	130	5
629	118	118	129	130	5
637	126	126	125	134	5
650	122	122	125	121	5
671	105	105	112	125	5
672	90	90	84	81	3
679	174	174	168	193	5
680	114	114	116	129	5
686	66	66	73	68	2
690	70	70	75	60	3
816	105	105	107	99	2
818	140	140	139	145	5
820	79	79	85	74	3
821	83	83	87	101	5
825	125	125	126	135	5
829	119	119	120	98	3
838	84	84	93	87	3
878	121	121	124	130	5
896	127	127	122	135	5
916	117	117	113	124	5
919	153	153	155	164	5
RV09	116	116	115	124	N/A
RV11	128	128	126	139	N/A
RV15	79	79	78	70	N/A
RV24	208	208	212	227	N/A
RV29	118	118	127	135	N/A

APPENDIX H

Familiarizing Yourself with Osirix

- These are your mouse controls: 

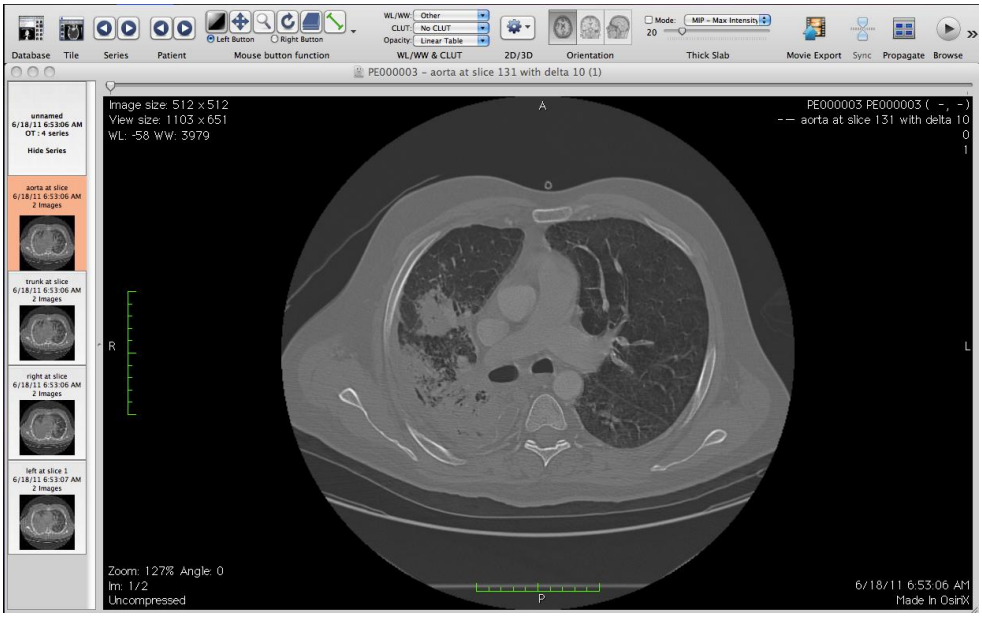
- The only two buttons you will be using will be the magnifying glass if you choose to zoom in and out (when going from left to right it is the fourth button over) and the measuring tool the last button (green line).
- To zoom in and out you select the magnifying glass and then click on the image and hold the button down as you zoom in and out
- To make a measurement you select the measurement tool and then click the spot where you want the line to begin and draw over and click where it will end. If you want to erase a measurement just click on the measurement and hit delete, to move just again select the measurement and drag to where you would like to have it.
- If the images have not already been imported, you must select the import tab at the top of the frame and go to the folder where the data is stored and it will import the folder. When starting in Osirix it will look like this after the data is imported:



Then just double click on the first series and a new window will open where you can begin making your measurements as described below.

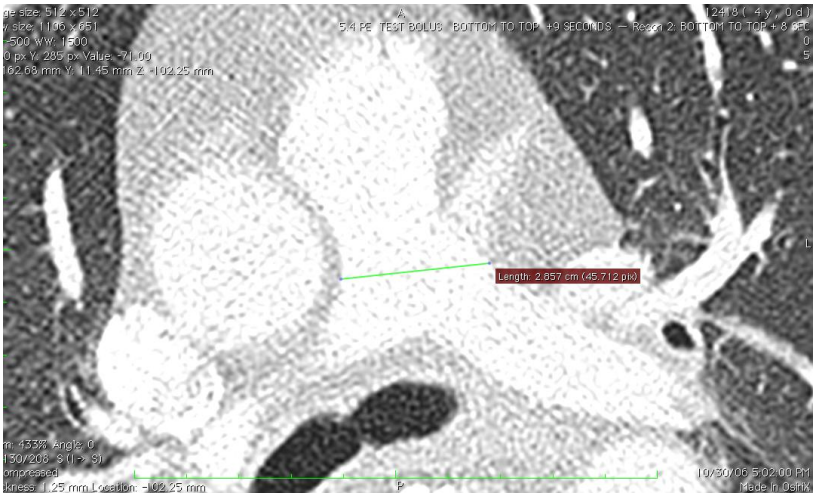
Making the Requested Measurements

Step1) Select an image. Each image will contain 4 series, shown in the left-hand column.



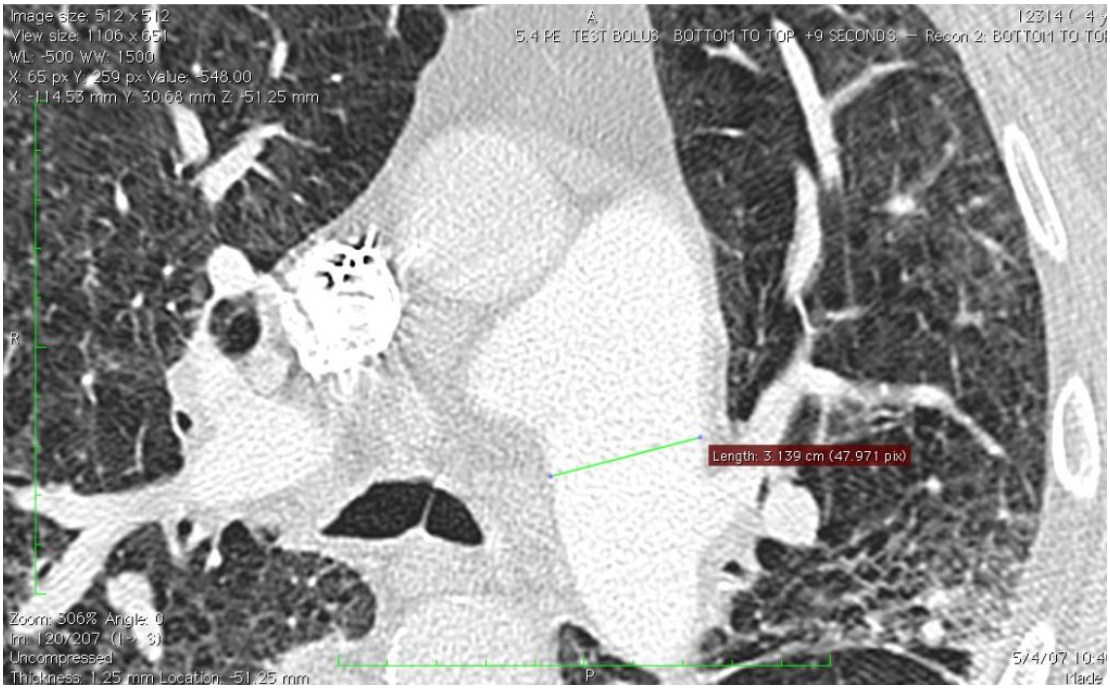
Step 2) The first series is the aorta. It is marked as such. In this slice you make a diameter measurement of the aorta and a diameter measurement of the pulmonary trunk at the midpoint as in the example below.

Step 3) The second series is labeled as the trunk. On this slice you are to make a measurement at the bifurcation point between the pulmonary trunk and where the split occurs between the right and left pulmonary arteries. As shown below.



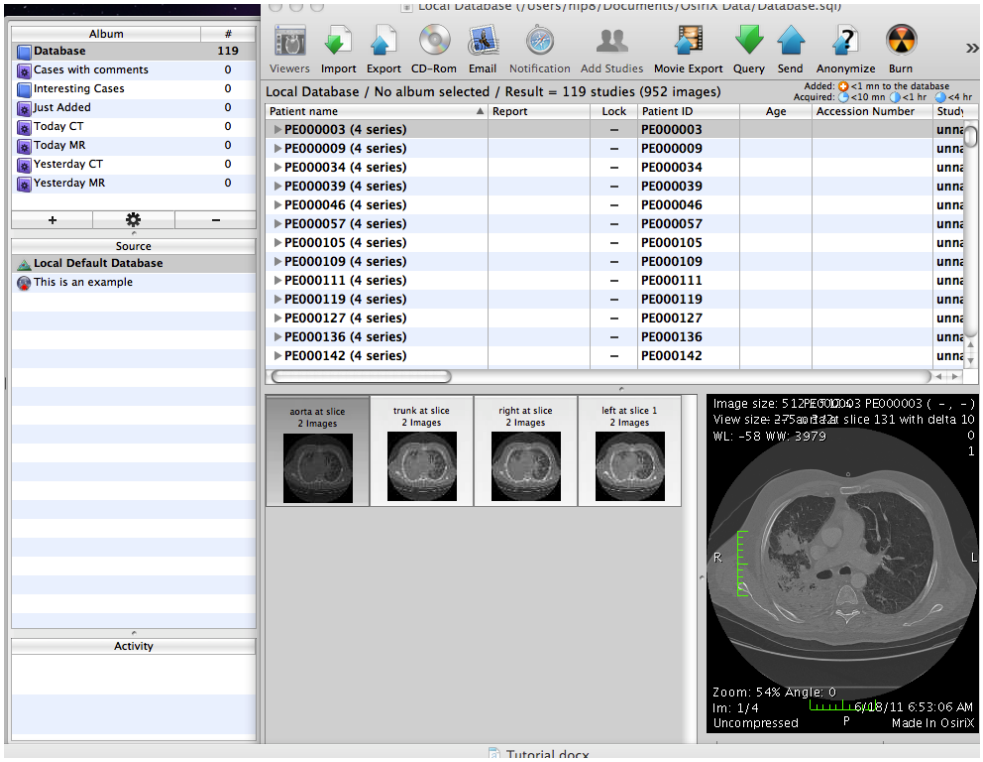
Step 4) Make a measurements of the right main pulmonary artery.

Step 5) Make a measurement of the left main pulmonary artery.



Once finished with a case just close out the window that popped up, not out of Osirix completely, click on this red x button above the left hand column that shows the 4 slices and below the database button. This will put you back to the initial screen and you can select the next

case. Once finished do not close Osirix I will take over from there. Thank you!



APPENDIX I

This is a table containing the averages of the manual measurements partnered with the corresponding automated measurements. The third column under each major vessel is the difference between the two measurements.

Table 30. Comparison table of manual and automated measurements

	Aorta			PT			RMPA	LMPA				
Cases	**Man.**	**Auto.**	**Diff.**	**Man.**	**Auto.**	**Diff.**	**Man.**	**Auto.**	**Diff.**	**Man.**	**Auto.**	**Diff.**
3	2.90	2.22	0.68	3.01	3.25	0.24	2.33	2.67	0.33	2.43	2.44	0.02
9	2.85	2.46	0.39	2.79	2.84	0.04	2.08	2.08	0.00	1.89	2.01	0.12
34	2.96	2.53	0.43	2.28	2.58	0.30	1.74	1.78	0.04	1.84	2.06	0.22
39	3.65	3.72	0.07	3.16	2.47	0.69	3.03	2.20	0.83	2.59	2.31	0.28
46	3.30	3.49	0.19	3.26	3.61	0.35	2.87	2.90	0.03	2.85	3.04	0.19
57	3.31	4.77	1.46	2.39	2.74	0.35	1.99	2.59	0.60	2.01	2.59	0.58
105	2.50	2.52	0.02	2.67	2.56	0.11	1.80	1.65	0.15	1.65	1.77	0.12
111	2.34	1.78	0.56	2.18	2.44	0.25	2.20	2.06	0.14	1.91	2.51	0.61
119	2.60	22.44	0.15	3.07	3.05	0.02	2.16	2.18	0.02	1.88	1.81	0.06
127	2.68	2.78	0.10	2.64	2.54	0.10	1.99	2.31	0.32	1.90	1.96	0.06
136	2.49	2.41	0.08	2.33	2.90	0.57	2.10	2.26	0.16	2.12	2.44	0.32
142	2.95	2.69	0.26	2.23	2.16	0.07	2.45	2.23	0.22	2.22	2.23	0.01
146	3.05	3.34	0.29	2.08	2.17	0.09	2.38	1.88	0.50	2.28	2.37	0.09
148	3.15	2.86	0.30	3.13	3.04	0.08	2.16	2.36	0.20	2.61	2.38	0.23
156	2.09	2.16	0.07	1.78	2.06	0.28	1.53	1.22	0.30	1.32	1.41	0.08
169	2.87	3.04	0.17	2.86	3.01	0.15	2.24	2.18	0.06	2.33	2.15	0.19
184	3.49	3.32	0.17	2.61	2.76	0.15	2.11	2.37	0.27	2.33	2.31	0.02
196	3.33	3.21	0.12	2.67	2.47	0.20	2.21	2.72	0.52	2.32	2.46	0.15
201	3.84	3.36	0.48	3.45	3.62	0.17	2.35	2.41	0.06	2.01	2.19	0.18
202	2.72	2.52	0.20	2.24	2.59	0.34	1.70	1.58	0.12	1.77	1.89	0.12
207	2.93	2.78	0.15	2.67	3.68	1.01	2.25	2.63	0.39	2.31	2.72	0.41
218	3.37	3.87	0.50	2.40	1.08	1.33	2.09	1.10	1.00	2.13	1.28	0.85

225	2.78	2.43	0.35	3.24	3.30	0.06	2.13	2.79	0.66	2.52	2.37	0.15
228	3.28	3.23	0.05	3.12	2.90	0.22	2.02	2.33	0.31	1.95	2.89	0.94
231	3.17	3.31	0.14	2.51	2.12	0.39	1.70	2.29	0.60	1.93	2.29	0.37
239	3.04	2.75	0.29	3.12	2.88	0.23	1.55	2.18	0.63	2.21	2.30	0.09
253	3.56	4.43	0.87	3.42	3.05	0.37	2.11	2.38	0.27	2.15	2.17	0.02
259	2.73	2.55	0.18	2.84	3.03	0.19	2.39	1.61	0.78	2.05	1.84	0.21
278	2.93	3.01	0.08	3.07	2.62	0.45	2.46	2.71	0.24	2.53	2.69	0.16
282	3.30	2.98	0.32	2.96	2.96	0.01	2.80	2.70	0.10	2.34	2.73	0.39
294	4.91	5.21	0.30	2.52	3.18	0.66	2.36	2.39	0.03	2.15	2.25	0.10
322	2.73	2.62	0.11	2.53	2.77	0.24	2.78	2.86	0.07	2.18	2.21	0.03
328	3.16	3.09	0.07	3.27	3.11	0.16	2.21	2.45	0.24	2.31	2.12	0.19
353	3.14	2.97	0.17	2.58	2.63	0.06	1.90	2.17	0.27	1.71	1.72	0.01
367	2.97	2.58	0.39	2.74	2.92	0.18	2.35	2.64	0.29	2.21	2.24	0.03
369	3.57	3.20	0.37	2.16	2.50	0.34	1.78	1.85	0.06	1.74	2.09	0.35
378	3.63	3.78	0.15	2.97	2.83	0.14	2.35	2.21	0.14	2.32	2.50	0.18
392	3.16	3.26	0.10	2.92	2.74	0.18	1.48	1.82	0.34	2.00	1.89	0.11
393	2.50	2.60	0.10	2.25	2.68	0.43	1.76	1.95	0.19	1.84	1.89	0.05
410	3.07	3.01	0.06	2.49	2.70	0.21	2.31	2.56	0.26	2.16	2.07	0.09
441	3.56	3.85	0.29	2.48	2.44	0.04	2.33	2.43	0.10	2.94	2.39	0.54
453	2.72	2.79	0.07	2.45	2.70	0.25	2.20	2.54	0.33	1.98	2.09	0.11
455	2.99	2.34	0.65	2.52	2.52	0.00	2.38	2.10	0.28	2.06	2.22	0.16
460	3.29	3.46	0.17	2.96	3.14	0.18	2.57	2.95	0.38	2.41	2.52	0.11
462	3.19	3.02	0.17	2.60	2.59	0.01	2.55	2.29	0.26	2.11	2.31	0.20
471	3.40	3.24	0.16	3.25	1.38	1.87	2.65	2.02	0.64	2.34	2.73	0.38
473	2.92	2.81	0.11	3.12	4.46	1.35	2.26	3.01	0.74	2.34	3.21	0.87
482	3.72	3.60	0.12	2.88	3.04	0.16	2.35	2.26	0.08	2.18	2.55	0.37
499	3.01	3.12	0.11	2.35	1.57	0.78	2.72	1.77	0.95	1.47	2.76	1.29
502	2.68	2.78	0.10	2.49	2.67	0.18	1.87	1.78	0.09	1.68	1.60	0.08
518	3.16	3.26	0.10	2.97	1.99	0.98	2.09	2.51	0.42	2.16	2.15	0.01
522	3.75	4.02	0.27	2.17	2.21	0.04	1.83	1.80	0.03	2.04	1.91	0.13
527	3.13	2.82	0.31	2.48	2.49	0.00	2.36	2.11	0.24	2.13	1.91	0.22
542	3.41	3.27	0.14	2.87	3.40	0.53	2.38	2.36	0.01	2.06	2.43	0.37
547	3.21	3.49	0.28	2.96	2.69	0.27	1.94	1.77	0.18	2.11	2.13	0.03
550	2.92	3.15	0.23	3.10	2.84	0.26	2.32	1.88	0.44	2.39	2.04	0.35
555	3.07	3.01	0.06	2.10	2.33	0.22	1.98	2.13	0.15	1.63	2.30	0.67
564	2.61	2.85	0.24	2.40	2.16	0.23	1.52	1.49	0.02	1.51	1.65	0.14
569	2.90	2.84	0.06	2.21	2.19	0.02	1.90	1.60	0.31	1.78	1.79	0.01
572	2.36	2.47	0.11	2.08	2.39	0.31	1.92	1.80	0.11	1.95	1.80	0.15
588	4.03	4.42	0.39	3.20	2.99	0.21	2.62	2.69	0.07	2.44	2.23	0.21

593	3.40	3.26	0.14	3.67	4.00	0.33	3.36	3.67	0.31	3.18	3.68	0.51
611	3.43	3.57	0.14	2.76	2.88	0.12	2.04	1.97	0.07	2.20	2.03	0.17
612	2.86	2.95	0.09	2.66	2.72	0.06	1.67	2.22	0.55	1.69	2.10	0.41
624	3.47	3.56	0.09	2.99	3.00	0.01	2.22	1.95	0.27	2.53	2.45	0.08
626	3.51	3.52	0.01	2.53	2.54	0.01	1.87	2.24	0.37	2.22	2.23	0.01
628	3.63	3.21	0.42	2.82	2.60	0.22	2.16	2.12	0.04	2.24	2.27	0.03
629	4.24	3.98	0.26	3.15	3.03	0.12	2.45	2.85	0.40	2.30	2.27	0.03
637	3.27	3.46	0.19	3.14	2.70	0.44	1.56	1.96	0.40	1.95	1.87	0.08
650	3.14	2.97	0.17	3.40	3.06	0.34	2.46	2.11	0.36	2.46	2.17	0.29
671	3.10	3.16	0.06	2.29	2.32	0.03	1.94	1.89	0.06	1.92	1.77	0.15
672	3.08	3.21	0.13	3.05	2.88	0.16	2.24	2.24	0.00	2.44	2.57	0.12
679	3.16	3.06	0.10	3.36	3.24	0.12	2.48	2.89	0.41	2.50	2.48	0.02
680	3.84	3.72	0.12	2.65	2.61	0.05	2.87	2.98	0.11	2.47	2.35	0.12
690	3.27	3.36	0.09	3.04	3.21	0.18	2.43	2.37	0.06	1.93	1.95	0.02
816	3.02	2.84	0.18	3.07	3.08	0.01	2.20	1.67	0.53	1.79	2.06	0.27
820	2.96	2.81	0.15	2.80	2.80	0.01	2.07	1.77	0.30	2.05	1.84	0.21
821	3.33	3.52	0.19	2.47	2.50	0.03	2.21	2.00	0.21	1.91	2.02	0.11
825	3.68	3.79	0.11	2.48	2.47	0.01	2.35	2.01	0.34	1.89	1.96	0.06
829	3.67	3.51	0.16	3.29	3.51	0.22	2.25	2.90	0.65	2.61	2.73	0.12
838	2.73	2.77	0.04	3.05	2.87	0.18	1.99	2.23	0.24	1.98	2.12	0.14
878	3.04	3.82	0.78	2.86	3.07	0.20	2.11	2.05	0.06	2.20	2.31	0.11
896	4.18	4.03	0.15	3.32	3.33	0.00	2.59	2.47	0.12	2.34	2.11	0.23
916	2.45	2.49	0.04	1.99	2.06	0.06	1.53	1.61	0.08	1.74	1.72	0.02
919	2.65	2.98	0.33	2.53	2.34	0.20	1.94	1.80	0.14	1.82	1.80	0.02
RV09	3.14	3.11	.03	3.07	2.39	0.68	2.19	2.13	0.06	2.27	1.88	0.39
RV11	2.99	3.05	.06	2.82	2.93	0.11	2.04	1.96	0.07	2.06	1.94	0.12
RV29	2.65	2.69	0.04	1.99	1.96	0.03	1.64	1.80	0.17	1.62	1.61	0.01
RV03	2.99	3.06	0.07	2.89	3.32	0.43	2.75	2.78	0.03	2.68	2.59	0.09
RV05	3.18	3.11	0.07	3.18	2.84	0.34	2.37	2.10	0.27	2.30	2.14	0.17
RV06	3.82	4.11	0.30	3.59	3.63	0.05	2.74	2.86	0.12	2.96	2.83	0.13
RV08	2.90	3.05	0.15	2.66	2.65	0.01	2.35	2.19	0.16	2.53	2.57	0.05
RV10	3.96	4.08	0.12	3.67	3.42	0.25	2.51	2.61	0.10	2.52	2.72	0.20
RV13	3.90	3.87	0.03	3.62	3.98	0.36	3.07	2.84	0.23	2.92	2.82	0.11
RV19	3.38	3.26	0.12	3.33	3.53	0.20	3.03	3.04	0.01	3.08	3.09	0.01
RV20	4.15	4.19	0.04	3.77	3.62	0.15	2.96	3.01	0.05	2.61	2.62	0.01
RV23	2.92	2.85	0.07	2.86	2.90	0.03	2.30	2.30	0.00	2.46	2.26	0.21
RV25	3.22	3.01	0.21	3.16	3.03	0.13	2.54	2.39	0.15	2.53	2.40	0.12
RV27	3.86	3.76	0.10	3.61	3.83	0.22	2.62	3.03	0.41	2.65	3.06	0.40
RV28	2.99	3.01	0.02	2.80	3.06	0.26	1.71	2.22	0.52	2.33	2.09	0.24

RV32	2.83	2.96	0.13	2.49	2.66	0.17	1.70	1.83	0.13	1.97	1.82	0.15
RV33	3.41	3.21	0.20	3.33	3.31	0.02	2.35	2.30	0.06	2.50	2.36	0.14
RV34	2.85	2.80	0.05	2.62	3.06	0.44	2.05	2.11	0.06	2.68	2.37	0.30
RV35	3.48	3.68	0.20	3.38	3.41	0.03	2.64	2.48	0.16	2.52	2.50	0.01
RV36	2.93	3.15	0.22	2.69	1.29	1.40	2.31	1.84	0.47	2.34	2.17	0.17
RV37	3.02	3.09	0.07	2.92	2.89	0.03	2.02	1.99	0.03	2.24	2.14	0.10
RV38	4.23	4.29	0.06	3.91	3.05	0.85	3.63	2.60	1.04	3.53	3.41	0.12
RV40	2.81	2.7	0.11	2.52	2.63	0.11	2.73	2.48	0.25	2.41	2.44	0.03
RV41	3.14	2.98	0.16	3.06	3.36	0.30	2.62	2.66	0.03	2.69	2.67	0.02

APPENDIX J

This section contains the tables of the four reviewer's original measurements and summary data.

Table 31. Reviewer 0's measurements for the Control Cases

	Aorta	PT	RMPA	LMPA
3	3.007	3.027	2.352	2.453
9	2.907	2.794	2.046	1.941
34	2.95	2.250	1.742	1.835
39	3.656	3.329	2.826	2.541
46	3.332	3.795	3.097	2.818
57	3.389	2.522	2.207	1.840
105	2.563	2.663	1.793	1.731
109	3.401	2.899	2.614	2.662
111	2.478	2.178	2.226	2.038
119	2.865	3.183	2.253	1.880
127	2.674	2.338	2.034	1.977
136	2.791	2.204	2.160	2.204
142	3.066	2.266	2.390	2.179
146	3.118	2.104	2.327	2.237
148	3.263	3.037	2.174	3.263
156	2.072	2.143	1.533	1.412
169	2.976	2.858	2.259	2.440
184	3.548	2.598	2.127	2.372
196	3.453	2.898	2.260	2.253
201	3.927	3.418	2.622	2.038
202	2.735	2.181	1.964	1.910
207	2.928	2.621	2.529	2.428
209	3.264	4.040	2.907	3.023
218	3.402	2.456	2.122	2.246

223	4.106	2.997	2.483	2.687
225	2.878	3.241	2.573	2.502
228	3.321	3.161	2.098	2.086
231	3.183	2.636	1.731	1.977
239	3.073	3.098	1.740	2.240
253	3.678	3.413	2.159	2.127
259	2.812	2.910	2.165	1.977
278	3.047	3.073	2.574	2.646
282	3.392	3.152	3.066	2.111
288	2.565	2.170	2.092	1.817
294	4.971	2.559	2.327	2.271
322	2.761	2.577	2.763	2.103
328	3.176	3.369	2.320	2.338
353	3.112	2.444	1.732	1.444
367	3.012	2.830	2.272	2.305
369	3.778	2.234	1.819	1.867
378	3.619	3.036	2.246	2.362
392	3.303	2.855	1.505	1.926
393	2.622	2.233	1.763	1.954
410	3.071	2.526	2.399	2.307
441	3.604	2.555	2.491	2.622
453	2.778	2.476	2.214	1.909
455	3.078	2.677	2.278	2.294
460	3.336	3.320	2.878	2.591
462	3.049	2.499	2.585	1.957
471	3.398	3.034	2.532	2.481
473	2.84	3.009	2.258	2.317
482	3.77	2.741	2.415	2.079
499	3.172	2.390	2.728	1.613
502	2.791	2.452	1.953	1.694
518	3.264	3.104	2.047	2.126
522	3.758	2.169	1.881	2.112
527	3.078	2.383	2.232	2.095
542	3.477	2.622	2.321	1.912
547	3.105	2.989	2.098	2.167
550	2.972	2.876	2.183	2.561
555	3.04	2.125	1.986	1.381
564	2.705	2.467	1.538	1.565
569	2.843	2.153	1.893	1.861

572	2.335	2.185	1.922	1.974
588	3.864	3.023	2.686	2.664
593	3.308	3.662	3.430	3.139
611	3.58	2.783	2.120	2.148
612	2.864	2.960	1.585	1.753
624	3.485	2.804	2.240	2.563
626	3.407	2.517	1.718	2.231
628	3.678	2.912	2.253	2.381
629	4.136	3.139	2.644	2.366
637	3.297	3.318	1.666	2.077
650	2.889	3.393	2.249	2.396
671	3.201	2.492	2.088	1.981
672	3.08	2.815	2.208	2.555
679	3.214	3.397	2.598	2.676
680	3.94	2.723	3.083	2.431
686	3.424	2.952	2.702	2.714
690	3.309	3.321	2.537	1.918
816	3.162	2.936	2.130	1.730
818	2.501	2.031	1.346	1.405
820	3.173	2.840	1.859	2.091
821	3.325	2.551	2.094	1.910
825	3.613	2.221	2.272	1.983
829	3.694	3.340	2.411	2.667
838	2.814	2.771	1.946	2.153
878	3.17	2.894	2.089	2.086
896	4.3	3.354	2.638	2.394
916	2.471	1.952	1.518	1.655
919	2.849	2.415	1.902	1.845
RV09	3.012	2.840	2.156	2.180
RV11	2.969	2.724	1.978	1.839
RV15	3.472	2.126	2.112	1.978
RV24	2.989	2.221	1.530	1.659
RV29	2.61	2.124	1.690	1.678

Table 32). Reviewer 0's measurements for the Disease Cases

Cases	Aorta	PT	RMPA	LMPA
RV01DW	2.589	2.951	2.768	2.735
RV03CH	3.001	2.978	2.778	2.774
RV05TH	3.048	3.143	2.387	2.193
RV06AP	3.646	3.609	2.822	3.051
RV08JR	3.312	2.633	2.32	2.652
RV10JB	2.5	3.749	2.67	2.537
RV12DC	2.579	3.147	2.549	2.294
RV13CB	3.706	3.669	3.041	2.959
RV19RD	3.368	3.308	3.09	3.107
RV20CT	3.305	3.709	2.895	2.557
RV23RB	3.415	2.997	2.086	2.479
RV25KN	2.929	3.035	2.609	2.488
RV27TH	4.087	3.819	2.768	2.928
RV28TB	3.892	2.747	2.086	2.479
RV32ST	2.837	2.477	1.742	2.036
RV33TM	3.128	3.302	2.396	2.452
RV34VS	3.077	2.727	2.065	2.751
RV35RH	3.614	3.301	2.539	2.57
RV36LH	2.686	2.676	2.31	2.232
RV37MW	3.101	3.037	2.172	2.32
RV38JM	3.207	3.739	3.813	3.532
RV40RS	2.939	2.514	2.743	2.329
RV41CB	3.658	2.906	2.807	2.844

Table 33) Reviewer 1's measurements for the Control Cases

	Aorta	PT	RMPA	LMPA
3	2.8540	3.0830	2.4240	2.5000
9	2.9410	2.9180	2.1900	1.8620
34	3.0300	2.2850	1.7530	1.9920
39	3.6750	3.2150	3.0110	2.6060
46	3.1590	3.6370	2.7780	2.8430
57	3.5540	2.2510	1.8300	1.9760
105	2.4380	2.4820	1.7400	1.7460
109	3.3530	2.8430	2.7470	2.5910
111	2.5460	2.1340	2.1120	1.8150
119	2.7390	2.9050	2.0980	1.8750
127	2.7170	3.0430	1.9570	2.1000
136	2.7550	2.3260	2.0600	2.1570
142	2.7880	2.2910	2.5050	2.1160
146	3.1930	2.1510	2.5650	2.4490
148	3.5170	3.1770	2.2720	2.4600
156	2.2600	1.7060	1.4650	1.4110
169	2.8880	3.2310	2.2760	2.1970
184	3.5950	2.7390	2.2420	2.2700
196	3.5020	2.4950	2.4110	2.4480
201	4.0110	3.5560	2.3930	2.0350
202	2.7490	2.2750	1.7050	1.7850
207	2.8950	2.7170	2.2210	2.1320
209	3.9590	3.4310	2.7640	2.7640
218	3.5360	2.4570	2.1220	2.0830
223	3.9840	2.7510	2.2180	2.6870
225	2.9160	3.2620	2.0280	2.5190
228	3.3390	3.0010	1.9210	1.9280
231	3.2600	2.5610	1.7880	1.9670
239	3.2400	3.0530	1.4790	2.2320
253	3.7350	3.6370	2.1680	2.3250
259	2.6250	3.0420	2.1010	2.0570
278	2.9440	3.2890	2.7900	2.1390
282	3.4100	2.9310	2.9000	2.5140
288	2.7710	2.5090	2.3100	1.5640
294	5.2250	2.5610	2.4800	2.1390
322	2.8490	2.3960	2.9560	2.3780

328	3.4000	3.2730	2.3710	2.4100
353	3.5960	2.5990	2.0300	1.9170
367	3.2000	3.0110	2.4730	2.2150
369	3.6500	2.2150	1.6860	1.8360
378	4.0380	2.9450	2.4620	2.4080
392	3.2090	3.2730	1.5560	1.9610
393	2.6370	2.2890	1.7420	1.8100
410	3.0210	2.3980	2.5110	1.9190
441	3.6920	2.5730	2.3430	2.8780
453	2.8330	2.3290	2.1970	1.9680
455	2.9650	2.5230	2.4400	1.8960
460	3.5120	3.2280	2.8510	2.5420
462	3.2830	2.5480	2.5870	1.9790
471	3.5870	3.0850	2.7990	2.3850
473	2.9600	3.0130	2.4380	2.3300
482	3.8210	2.4730	2.3480	2.1110
499	3.1610	2.2510	2.7840	1.4700
502	2.8670	2.5000	1.9330	1.7670
518	3.2190	3.2460	1.9160	2.2730
522	4.1720	2.0390	1.7710	2.1150
527	3.4680	2.5620	2.3950	2.1990
542	3.3780	3.0900	2.4750	2.1470
547	3.2560	3.0460	1.9690	2.1420
550	3.0700	3.1020	2.3850	2.3670
555	3.0230	2.0770	2.0090	1.6290
564	2.5770	2.3380	1.7870	1.6440
569	2.9640	2.2810	1.8760	1.9330
572	2.5110	2.0010	1.9610	1.8720
588	4.3970	3.2520	2.6860	2.5210
593	3.8110	3.5730	3.5070	3.1570
611	3.6680	2.7940	2.1810	2.3190
612	2.9860	2.6800	1.6930	1.5760
624	3.7020	3.3750	2.2310	2.4430
626	3.6720	2.5400	1.9740	2.2930
628	3.7790	2.8820	2.3300	2.3880
629	4.5820	3.2310	2.4360	2.4970
637	3.4430	3.4200	1.4660	2.0960
650	3.3140	3.5970	2.5850	2.7290
671	3.4340	2.3170	2.0140	1.8400

672	3.1610	3.2320	2.2640	2.3940
679	3.5730	3.4970	2.5540	2.2860
680	3.9720	2.5290	2.8320	2.4160
686	4.2510	3.1180	3.0130	2.2680
690	3.6300	3.4410	2.4580	2.2160
816	3.2650	3.5720	2.2270	2.0060
818	2.4770	1.9110	1.1790	1.4990
820	3.4590	2.9600	1.5110	2.1140
821	3.3810	2.5310	2.2080	1.9210
825	3.7190	2.4370	2.5200	1.9060
829	3.7370	3.4150	2.2420	2.4770
838	2.5200	3.3840	2.1000	2.0870
878	3.0170	2.7630	2.1680	2.2870
896	4.3790	3.1690	2.5650	2.3540
916	2.5020	1.9440	1.5500	1.7710
919	2.8420	2.6150	1.9480	1.6870
RV09	3.2870	3.5750	2.2070	2.3710
RV11	3.1790	2.8220	2.1920	1.9430
RV15	3.4940	2.1880	2.1360	1.7130
RV24	3.0720	2.2450	1.4620	1.5820
RV29	2.5960	2.0140	1.6730	1.5230

Table 34) Reviewer 1's measurements for the Disease Cases

	Aorta	PT	RMPA	LMPA
RV01DW	2.462	3.1	2.753	2.364
RV03CH	2.985	2.92	2.816	2.633
RV05TH	2.961	3.183	2.379	2.213
RV06AP	3.692	3.625	2.757	2.994
RV08JR	3.469	2.736	2.419	2.413
RV10JB	2.516	3.738	2.727	2.568
RV12DC	3.179	2.822	2.192	1.943
RV13CB	3.623	3.753	3.296	3.001
RV19RD	3.48	3.159	2.976	3.073
RV20CT	3.601	4.096	3.017	2.504
RV23RB	3.501	3.1	2.397	2.614
RV25KN	2.996	3.313	2.625	2.585
RV27TH	3.962	3.501	2.439	2.151
RV28TB	3.75	2.828	1.489	2.247
RV32ST	2.832	2.466	1.728	1.942
RV33TM	3.155	3.364	2.449	2.586
RV34VS	3.121	2.672	1.914	2.609
RV35RH	3.693	3.414	2.599	2.474
RV36LH	2.648	2.694	2.372	2.312
RV37MW	3.095	2.898	2.005	2.228
RV38JM	3.287	3.872	3.691	3.409
RV40RS	2.897	2.634	2.43	2.46
RV41CB	3.599	3.282	2.694	2.519

Table 35). Reviewer 2's measurements for the Control Cases

	Aorta	PT	RMPA	LMPA
3	3.079	2.932	2.37	2.319
9	2.95	2.639	2.139	1.863
34	2.889	2.405	1.885	1.969
39	3.756	3.139	3.099	2.695
46	3.8	3.053	2.902	2.998
57	3.391	2.322	2.077	1.871
105	2.566	3.029	1.824	1.532
109	3.407	3.031	2.532	2.641
111	2.2	2.09	2.125	1.866
119	2.837	3.181	2.242	1.954
127	2.693	3.122	2.041	1.699
136	2.222	2.421	2.08	2.009
142	3.029	2.264	2.485	2.315
146	2.93	2.098	2.346	2.277
148	2.965	3.053	2.063	2.477
156	2.109	1.677	1.545	1.32
169	3.058	3.086	2.27	2.219
184	3.521	2.609	2.021	2.323
196	3.253	2.844	1.808	2.294
201	3.74	3.343	2.099	2.087
202	2.811	2.281	1.533	1.815
207	3.145	2.58	2.262	2.32
209	3.269	3.904	3.193	3.144
218	3.291	2.234	2.153	2.21
223	3.666	2.637	2.25	2.592
225	2.819	3.225	2.034	2.5
228	3.308	3.112	2.048	1.801
231	3.146	2.492	1.708	1.793
239	3.029	3.222	1.51	2.177
253	3.362	3.069	2.127	2.152
259	2.867	3.005	3.427	2.245
278	3.038	3.261	2.114	2.519
282	3.197	2.96	2.984	2.457
288	2.618	2.235	1.958	1.421
294	4.657	2.472	2.547	2.075
322	2.712	2.36	2.795	2.156

328	3.078	3.215	2.077	2.306
353	2.744	2.47	1.868	1.636
367	3.041	2.942	2.349	2.219
369	3.552	2.159	1.849	1.661
378	3.503	3.201	2.483	2.352
392	3.234	3.055	1.572	2.053
393	2.175	2.084	1.711	1.894
410	3.158	2.495	2.271	2.309
441	3.518	2.529	2.253	3.188
453	2.749	2.463	2.21	2.144
455	3.012	2.504	2.475	2.244
460	3.051	2.615	2.114	1.875
462	3.393	2.972	2.506	2.545
471	3.198	3.639	2.624	2.382
473	2.91	3.512	2.017	2.356
482	3.745	3.606	2.274	2.381
499	2.859	2.335	2.703	1.428
502	2.463	2.751	1.809	1.702
518	3.039	3.007	2.044	2.152
522	3.631	2.116	1.796	2.073
527	3.033	2.243	2.432	2.259
542	3.358	3.032	2.438	2.064
547	3.153	2.865	1.851	2.083
550	2.891	3.558	2.281	2.236
555	3.084	2.096	1.932	1.77
564	2.871	2.374	1.284	1.487
569	2.906	2.354	1.882	1.758
572	2.348	2.178	1.938	1.997
588	3.97	3.413	2.487	2.531
593	3.335	3.573	3.373	3.206
611	3.067	2.706	1.819	2.15
612	2.652	2.876	1.618	1.662
624	3.101	3.24	2.294	2.739
626	3.395	2.466	1.914	2.181
628	3.648	2.84	2.182	2.184
629	3.992	3.037	2.58	2.191
637	3.239	3.169	1.419	1.923
650	3.674	3.311	2.736	2.614
671	2.542	2.305	1.87	2.014

672	2.841	3.355	2.24	2.477
679	2.864	3.364	2.331	2.655
680	3.69	2.577	2.752	2.653
686	3.267	2.994	2.677	2.315
690	3.221	2.803	2.224	1.98
816	3.084	2.946	2.488	1.685
818	2.39	1.883	1.109	1.401
820	2.749	2.888	1.493	1.98
821	3.343	2.542	2.204	1.841
825	3.555	2.439	2.284	1.848
829	3.644	3.292	2.076	2.494
838	2.71	3.24	1.815	1.8
878	3.087	2.735	2.132	2.173
896	3.95	3.477	2.456	2.352
916	2.466	1.973	1.528	1.682
919	2.494	2.288	1.988	1.859
RV09	3.147	3.364	2.084	2.335
RV11	2.989	2.706	2.165	1.819
RV15	3.454	2.117	2.103	1.817
RV24	2.995	2.149	1.494	1.626
RV29	2.603	1.99	1.597	1.639

Table 36). Reviewer 2's measurements for the Disease Cases

	Aorta	PT	RMPA	LMPA
RV01DW	2.585	2.948	2.733	2.412
RV03CH	2.829	2.786	2.577	2.666
RV05TH	2.819	3.066	2.343	2.334
RV06AP	3.598	3.52	2.54	2.916
RV08JR	3.228	2.643	2.289	2.69
RV10JB	2.576	3.653	2.508	2.488
RV12DC	2.534	3.007	2.323	2.192
RV13CB	3.492	3.559	2.946	2.71
RV19RD	3.435	3.48	2.988	3.004
RV20CT	3.546	3.676	2.766	2.617
RV23RB	3.42	2.879	2.301	2.575
RV25KN	2.84	3.17	2.519	2.427
RV27TH	3.885	3.615	2.892	2.607
RV28TB	3.724	2.839	1.635	2.378
RV32ST	2.825	2.501	1.685	1.993
RV33TM	3.111	3.315	2.411	2.47
RV34VS	3.029	2.674	1.947	2.614
RV35RH	3.549	3.312	2.632	2.48
RV36LH	2.597	2.765	2.355	2.325
RV37MW	3.072	2.903	2.054	2.196
RV38JM	3.25	3.702	3.567	3.409
RV40RS	2.954	2.4	2.7	2.329
RV41CB	3.531	2.839	2.61	2.563

Table 37). Reviewer 3's measurements for the Control Cases

	Aorta	PT	RMPA	LMPA
3	2.67	3.009	2.189	2.435
9	2.616	2.826	1.947	1.912
34	2.985	2.183	1.591	1.561
39	3.507	2.942	3.166	2.524
46	2.924	2.542	2.715	2.742
57	2.91	2.46	1.849	2.365
105	2.418	2.49	1.856	1.592
109	3.156	3.336	2.573	2.573
111	2.131	2.335	2.335	1.906
119	1.945	3.004	2.045	1.793
127	2.651	2.041	1.929	1.812
136	2.206	2.352	2.116	2.09
142	2.925	2.105	2.4	2.274
146	2.969	1.954	2.288	2.138
148	2.868	3.234	2.137	2.25
156	1.91	1.589	1.559	1.145
169	2.552	2.255	2.171	2.48
184	3.282	2.506	2.035	2.353
196	3.105	2.451	2.342	2.278
201	3.67	3.502	2.291	1.888
202	2.574	2.242	1.617	1.553
207	2.751	2.774	1.969	2.36
209	2.71	4.016	2.845	2.934
218	3.254	2.465	1.969	1.996
223	3.643	2.738	2.329	2.542
225	2.514	3.212	1.878	2.564
228	3.133	3.207	2.021	1.983
231	3.078	2.336	1.566	1.98
239	2.835	3.095	1.46	2.208
253	3.463	3.567	1.995	1.995
259	2.598	2.393	1.855	1.926
278	2.687	2.67	2.37	2.832
282	3.199	2.785	2.262	2.289
288	2.313	2.446	2.016	1.54
294	4.806	2.49	2.092	2.103
322	2.615	2.778	2.622	2.102
328	2.968	3.217	2.089	2.182

353	3.102	2.803	1.982	1.838
367	2.639	2.193	2.301	2.114
369	3.302	2.036	1.783	1.595
378	3.346	2.71	2.206	2.146
392	2.901	2.492	1.293	2.051
393	2.57	2.41	1.808	1.701
410	3.027	2.534	2.044	2.109
441	3.432	2.263	2.238	3.055
453	2.511	2.517	2.198	1.894
455	2.913	2.364	2.334	1.815
460	3.271	2.666	2.443	2.636
462	3.039	2.372	2.519	1.946
471	3.399	3.241	2.662	2.129
473	2.96	2.94	2.336	2.371
482	3.543	2.683	2.343	2.142
499	2.842	2.422	2.682	1.37
502	2.58	2.245	1.792	1.572
518	3.137	2.533	2.349	2.079
522	3.441	2.35	1.886	1.874
527	2.96	2.746	2.369	1.971
542	3.41	2.718	2.279	2.122
547	3.338	2.943	1.857	2.032
550	2.762	2.872	2.445	2.387
555	3.113	2.116	1.994	1.751
564	2.274	2.403	1.459	1.346
569	2.9	2.056	1.967	1.576
572	2.257	1.969	1.839	1.97
588	3.892	3.108	2.617	2.046
593	3.141	3.88	3.125	3.199
611	3.39	2.774	2.036	2.181
612	2.957	2.118	1.779	1.774
624	3.58	2.552	2.108	2.385
626	3.564	2.598	1.865	2.155
628	3.408	2.65	1.866	1.996
629	4.238	3.197	2.125	2.132
637	3.1	2.671	1.67	1.704
650	2.681	3.281	2.282	2.096
671	3.221	2.05	1.804	1.834
672	3.224	2.7782	2.257	2.34

679	2.994	3.172	2.436	2.365
680	3.774	2.785	2.829	2.368
686	3.607	2.89	2.047	1.839
690	2.904	2.581	2.502	1.608
816	2.587	2.832	1.963	1.738
818	2.677	1.87	1.125	1.152
820	2.467	2.52	3.413	2.006
821	3.285	2.255	2.344	1.971
825	3.829	2.819	2.333	1.83
829	3.59	3.109	2.266	2.814
838	2.864	2.798	2.1	1.876
878	2.872	3.062	2.07	2.258
896	4.077	3.289	2.7	2.254
916	2.345	2.105	1.533	1.857
919	2.426	2.821	1.921	1.902
RV09	3.101	2.504	2.293	2.211
RV11	2.818	3.04	1.805	2.645
RV15	3.314	2.062	1.982	1.964
RV24	3.15	2.058	1.567	1.803
RV29	2.772	1.835	1.58	1.623

Table 38). Reviewer 3's measurements for the Disease Cases

	Aorta	PT	RMPA	LMPA
RV01DW	2.363	3.087	2.565	2.586
RV03CH	2.683	2.874	2.812	2.633
RV05TH	2.873	3.324	2.387	2.475
RV06AP	3.638	3.591	2.836	2.863
RV08JR	3.327	2.631	2.382	2.35
RV10JB	2.299	3.542	2.15	2.496
RV12DC	2.262	3.289	2.21	2.328
RV13CB	3.538	3.498	3.007	3.029
RV19RD	3.422	3.38	3.048	3.137
RV20CT	3.267	3.615	3.169	2.742
RV23RB	3.199	2.481	2.421	2.19
RV25KN	2.651	3.114	2.419	2.616
RV27TH	4.121	3.5	2.384	2.93
RV28TB	3.756	2.798	1.62	2.206
RV32ST	2.631	2.502	1.648	1.919
RV33TM	2.92	3.325	2.15	2.504
RV34VS	2.816	2.421	2.26	2.745
RV35RH	3.372	3.488	2.805	2.544
RV36LH	2.576	2.632	2.219	2.497
RV37MW	2.775	2.85	1.855	2.233
RV38JM	3.038	4.312	3.454	3.777
RV40RS	3.093	2.536	3.042	2.505
RV41CB	3.541	3.219	2.372	2.823

APPENDIX K

This section contains the output from the disease classification model in its entirety.

LOGISTIC REGRESSION MODEL OUTPUT WITH ORIGINAL DATA

```
=== Run information ===

Scheme:weka.classifiers.functions.Logistic -R 1.0E-8 -M -1
Relation:     DiseaseStateData
Instances:    109
Attributes:   5
              Aorta
              PT
              RMPA
              LMPA
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
               Class
Variable     disease
====================
Aorta         0.3382
PT            1.0368
RMPA         -1.0815
LMPA          2.0157
Intercept    -7.7743
```

```
Odds Ratios...
                Class
Variable      disease
=====================
Aorta          1.4024
PT             2.8201
RMPA           0.3391
LMPA           7.5058


Time taken to build model: 0.05 seconds


=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               81.6514 %
Incorrectly Classified Instances        20               18.3486 %
Kappa statistic                          0.2096
Mean absolute error                      0.2811
Root mean squared error                  0.3865
Relative absolute error                 89.2418 %
Root relative squared error             97.9455 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.19      0.034      0.571     0.19      0.286      0.694    disease
                 0.966     0.81       0.833     0.966     0.895      0.694    negative
Weighted Avg.    0.817     0.66       0.783     0.817     0.777      0.694

=== Confusion Matrix ===

  a  b   <-- classified as
  4 17 |  a = disease
  3 85 |  b = negative
```

NAÏVE BAYES MODEL OUTPUT WITH ORIGINAL DATA

```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     DiseaseStateData
Instances:    109
Attributes:   5
              Aorta
              PT
              RMPA
              LMPA
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                 Class
Attribute       disease negative
                  (0.2)    (0.8)
==================================
Aorta
  mean           3.3621   3.1394
  std. dev.      0.4918   0.5718
  weight sum         21       88
  precision      0.0445   0.0445

PT
  mean           3.1197    2.737
  std. dev.      0.5546    0.511
  weight sum         21       88
  precision      0.0417   0.0417

RMPA
  mean           2.4585   2.2096
  std. dev.      0.3695   0.4356
  weight sum         21       88
  precision      0.0325   0.0325
```

```
LMPA
  mean              2.528   2.1993
  std. dev.        0.3731   0.3793
  weight sum           21       88
  precision         0.032    0.032



Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               81.6514 %
Incorrectly Classified Instances        20               18.3486 %
Kappa statistic                          0.3641
Mean absolute error                      0.2687
Root mean squared error                  0.4032
Relative absolute error                 85.3031 %
Root relative squared error            102.1649 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.429     0.091       0.529     0.429     0.474      0.681    disease
                 0.909     0.571       0.87      0.909     0.889      0.681    negative
Weighted Avg.    0.817     0.479       0.804     0.817     0.809      0.681

=== Confusion Matrix ===

  a  b   <-- classified as
  9 12 |  a = disease
  8 80 |  b = negative
```

NAÏVE BAYES MODEL OUTPUT WITH NORMALIZED DATA

```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     DiseaseStateDataNormalized
Instances:    109
Attributes:   5
              Aorta
              PT
              RMPA
              LMPA
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                 Class
Attribute       disease negative
                  (0.2)    (0.8)
==================================
Aorta
  mean            0.9792   0.9155
  std. dev.        0.145   0.1666
  weight sum          21       88
  precision       0.0125   0.0125

PT
  mean            0.9225   0.8094
  std. dev.       0.1641   0.1508
  weight sum          21       88
  precision       0.0093   0.0093

RMPA
  mean             0.957   0.8598
  std. dev.       0.1448   0.1687
  weight sum          21       88
  precision       0.0093   0.0093
```

```
LMPA
  mean              1.053    0.917
  std. dev.        0.1545   0.1587
  weight sum           21       88
  precision        0.0093   0.0093



Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               81.6514 %
Incorrectly Classified Instances        20               18.3486 %
Kappa statistic                          0.3641
Mean absolute error                      0.2696
Root mean squared error                  0.4034
Relative absolute error                 85.5859 %
Root relative squared error            102.2249 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.429     0.091       0.529     0.429     0.474      0.682    disease
                 0.909     0.571       0.87      0.909     0.889      0.682    negative
Weighted Avg.    0.817     0.479       0.804     0.817     0.809      0.682

=== Confusion Matrix ===

  a  b   <-- classified as
  9 12 |  a = disease
  8 80 |  b = negative
```

LOGISTIC REGRESSION MODEL OUTPUT WITH NORMALIZED DATA

```
=== Run information ===

Scheme:weka.classifiers.functions.Logistic -R 1.0E-8 -M -1
Relation:     DiseaseStateDataNormalized
Instances:    109
Attributes:   5
              Aorta
              PT
              RMPA
              LMPA
              Label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                 Class
Variable      disease
=====================
Aorta           1.169
PT             3.5116
RMPA          -2.7519
LMPA           4.7923
Intercept     -7.7694


Odds Ratios...
                 Class
Variable      disease
=====================
Aorta          3.2187
PT            33.5009
RMPA           0.0638
LMPA         120.5751


Time taken to build model: 0 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               81.6514 %
Incorrectly Classified Instances        20               18.3486 %
Kappa statistic                          0.2096
Mean absolute error                      0.2812
Root mean squared error                  0.3865
Relative absolute error                 89.2467 %
Root relative squared error             97.9354 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.19      0.034      0.571     0.19      0.286      0.695    disease
                 0.966     0.81       0.833     0.966     0.895      0.695    negative
Weighted Avg.    0.817     0.66       0.783     0.817     0.777      0.695

=== Confusion Matrix ===

  a  b   <-- classified as
  4 17 |  a = disease
  3 85 |  b = negative
```

LOGISTIC REGRESSION MODEL NORMALIZED MANUAL DATA

```
=== Run information ===

Scheme:weka.classifiers.functions.Logistic -R 1.0E-8 -M -1
Relation:     manualDiseaseState
Instances:    109
Attributes:   5
              PT
              RMPA
              LMPA
              Aorta
              label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                Class
Variable     negative
=====================
PT            -1.4245
RMPA           1.1376
LMPA          -4.6901
Aorta          1.2724
Intercept     10.1207


Odds Ratios...
                Class
Variable     negative
=====================
PT             0.2406
RMPA           3.1192
LMPA           0.0092
Aorta          3.5693


Time taken to build model: 0.06 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          89               81.6514 %
Incorrectly Classified Instances        20               18.3486 %
Kappa statistic                          0.2796
Mean absolute error                      0.2281
Root mean squared error                  0.357
Relative absolute error                 72.4011 %
Root relative squared error             90.4699 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.943     0.714      0.847     0.943     0.892      0.837    negative
                 0.286     0.057      0.545     0.286     0.375      0.837    disease
Weighted Avg.    0.817     0.588      0.789     0.817     0.793      0.837

=== Confusion Matrix ===

  a  b   <-- classified as
 83  5 |  a = negative
 15  6 |  b = disease
```

LOGISTIC REGRESSION MODEL NORMALIZED DATA MANUAL DATASE

```
=== Run information ===

Scheme:weka.classifiers.functions.Logistic -R 1.0E-8 -M -1
Relation:     manualDiseaseStateNormalized
Instances:    109
Attributes:   5
              PT
              RMPA
              LMPA
              AORTA
              label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                           Class
Variable                negative
================================
PT                       -2.7695
RMPA                      2.5581
LMPA                     55.4021
AORTA                  -184.7612
Intercept                12.9752


Odds Ratios...
                           Class
Variable                negative
================================
PT                        0.0627
RMPA                     12.9106
LMPA         1.15032084200469E24
AORTA                          0


Time taken to build model: 0.01 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          88               80.7339 %
Incorrectly Classified Instances        21               19.2661 %
Kappa statistic                          0.2599
Mean absolute error                      0.2286
Root mean squared error                  0.3547
Relative absolute error                 72.554  %
Root relative squared error             89.8829 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.932     0.714      0.845     0.932     0.886      0.837    negative
                 0.286     0.068      0.5       0.286     0.364      0.837    disease
Weighted Avg.    0.807     0.59       0.779     0.807     0.786      0.837

=== Confusion Matrix ===

  a  b   <-- classified as
 82  6 |  a = negative
 15  6 |  b = disease
```

NAÏVE BAYES MODEL WITH ORIGINAL DATA ON MANUAL DATASET

```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     manualDiseaseState
Instances:    109
Attributes:   5
              PT
              RMPA
              LMPA
              Aorta
              label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                Class
Attribute      negative  disease
                  (0.8)    (0.2)
==================================
PT
  mean           2.8315   3.3332
  std. dev.      0.4061   0.4609
  weight sum         88       21
  precision      0.0277   0.0277

RMPA
  mean            2.179   2.5254
  std. dev.      0.3547   0.4412
  weight sum         88       21
  precision      0.0312   0.0312

LMPA
  mean           2.1255   2.5927
  std. dev.      0.3245   0.3184
  weight sum         88       21
  precision      0.0335   0.0335
```

```
Aorta
  mean            3.1464   3.1694
  std. dev.       0.4527      0.4
  weight sum          88       21
  precision       0.0274   0.0274


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          91               83.4862 %
Incorrectly Classified Instances        18               16.5138 %
Kappa statistic                          0.4277
Mean absolute error                      0.2096
Root mean squared error                  0.3638
Relative absolute error                 66.518  %
Root relative squared error             92.1807 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.92      0.524      0.88      0.92      0.9        0.812    negative
                 0.476     0.08       0.588     0.476     0.526      0.812    disease
Weighted Avg.    0.835     0.438      0.824     0.835     0.828      0.812

=== Confusion Matrix ===

  a  b   <-- classified as
 81  7 |  a = negative
 11 10 |  b = disease
```

NAÏVE BAYES MODEL WITH NORMALIZED DATA AND MANUAL DATASET

```
=== Run information ===

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     manualDiseaseStateNormalized
Instances:    109
Attributes:   5
              PT
              RMPA
              LMPA
              AORTA
              label
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                Class
Attribute      negative  disease
                  (0.8)    (0.2)
==================================
PT
  mean             1.231   1.4479
  std. dev.        0.176   0.1995
  weight sum          88       21
  precision       0.0156   0.0156

RMPA
  mean            1.0145   1.1737
  std. dev.       0.1675   0.2082
  weight sum          88       21
  precision       0.0196   0.0196

LMPA
  mean            0.9611   1.1712
  std. dev.       0.1459   0.1464
  weight sum          88       21
  precision       0.0192   0.0192
```

```
AORTA
  mean            0.3403   0.4152
  std. dev.       0.0516   0.0513
  weight sum          88       21
  precision       0.0044   0.0044


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          86               78.8991 %
Incorrectly Classified Instances        23               21.1009 %
Kappa statistic                          0.3567
Mean absolute error                      0.2047
Root mean squared error                  0.3781
Relative absolute error                 64.9651 %
Root relative squared error             95.7936 %
Total Number of Instances              109

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.852     0.476       0.882     0.852     0.867      0.84     negative
                 0.524     0.148       0.458     0.524     0.489      0.84     disease
Weighted Avg.    0.789     0.413       0.801     0.789     0.794      0.84

=== Confusion Matrix ===

  a  b   <-- classified as
 75 13 |  a = negative
 10 11 |  b = disease
```

APPENDIX L

The output from the polynomial regression model for predicting pressure.

Curve Fit

Notes

	Output Created		24-OCT-2012 20:17:55
	Comments		
	Input	Data	C:\Users\dave\Documents\My Dropbox\Data\Prediction_Models\Pressure Data.sav
		Active Dataset	DataSet2
		Filter	<none>
		Weight	<none>
		Split File	<none>
		N of Rows in Working Data File	27
	Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
		Cases Used	Cases with a missing value in any variable are not used in the analysis.
	Syntax		CURVEFIT /VARIABLES= Aorta PT RMPA LMPA TPG VPR PA_Systolic PA_Diastolic WITH PAMP /CONSTANT /MODEL=CUBIC /PRINT ANOVA /PLOT NONE /ID=Disease_State.
	Resources	Processor Time	00:00:00.06
		Elapsed Time	00:00:00.09
	Use	From	First observation
		To	Last observation
	Predict	From	First Observation following the use period
		To	Last observation
		Amount of Output	PRINT = DEFAULT
		Saving New Variables	NEWVAR = NONE
	Time Series Settings (TSET)	Maximum Number of Lags in Autocorrelation or Partial Autocorrelation Plots	MXAUTO = 16
		Maximum Number of Lags Per Cross-Correlation Plots	MXCROSS = 7
		Maximum Number of New Variables Generated Per Procedure	MXNEWVAR = 60

Maximum Number of New Cases Per Procedure	MXPREDICT = 1000
Treatment of User-Missing Values	MISSING = EXCLUDE
Confidence Interval Percentage Value	CIN = 95
Tolerance for Entering Variables in Regression Equations	TOLER = .0001
Maximum Iterative Parameter Change	CNVERGE = .001
Method of Calculating Std. Errors for Autocorrelations	ACFSE = IND
Length of Seasonal Period	Unspecified
Variable Whose Values Label Observations in Plots	Unspecified
Equations Include	CONSTANT

Model Description

Model Name		MOD_1
Dependent Variable	1	Aorta
	2	PT
	3	RMPA
	4	LMPA
	5	TPG
	6	VPR
	7	PA_Systolic
	8	PA_Diastolic
Equation	1	Cubic
Independent Variable		PAMP
Constant		Included
Variable Whose Values Label Observations in Plots		Disease_State
Tolerance for Entering Terms in Equations		.0001

Case Processing Summary

	N
Total Cases	27
Excluded Cases[a]	0
Forecasted Cases	0
Newly Created Cases	0

a. Cases with a missing value in any variable are excluded from the analysis.

Variable Processing Summary

		Variables								
		Dependent								Independent
		Aorta	PT	RMPA	LMPA	TPG	VPR	PA_Systolic	PA_Diastolic	PAMP
Number of Positive Values		27	27	27	27	27	27	27	27	27
Number of Zeros		0	0	0	0	0	0	0	0	0
Number of Negative Values		0	0	0	0	0	0	0	0	0
Number of Missing Values	User	0	0	0	0	0	0	0	0	0
	System	0	0	0	0	0	0	0	0	0

Aorta/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.412	.170	.062	.385

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.700	3	.233	1.572	.223
Residual	3.416	23	.149		
Total	4.117	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	T	Sig.
	B	Std. Error	Beta		
PAMP	.067	.077	3.020	.868	.394
PAMP ** 2	-.001	.002	-4.815	-.642	.527
PAMP ** 3	5.228E-006	.000	1.689	.400	.693
(Constant)	2.140	.966		2.214	.037

PT/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.619	.383	.302	.497

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	3.524	3	1.175	4.756	.010
Residual	5.681	23	.247		
Total	9.204	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.117	.099	3.547	1.182	.249
PAMP ** 2	-.002	.002	-4.772	-.738	.468
PAMP ** 3	8.066E-006	.000	1.743	.479	.637
(Constant)	.732	1.246		.587	.563

RMPA/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.712	.507	.443	.312

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	2.307	3	.769	7.895	.001
Residual	2.240	23	.097		
Total	4.547	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.054	.062	2.304	.859	.399
PAMP ** 2	.000	.001	-1.801	-.312	.758
PAMP ** 3	2.720E-007	.000	.084	.026	.980
(Constant)	1.049	.783		1.340	.193

LMPA/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.758	.575	.519	.309

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	2.970	3	.990	10.362	.000
Residual	2.198	23	.096		
Total	5.168	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.086	.062	3.461	1.389	.178
PAMP ** 2	-.001	.001	-3.803	-.709	.486
PAMP ** 3	3.219E-006	.000	.928	.307	.761
(Constant)	.578	.775		.745	.464

TPG/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.973	.946	.939	4.702

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	8974.234	3	2991.411	135.303	.000
Residual	508.507	23	22.109		
Total	9482.741	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.323	.940	.304	.344	.734
PAMP ** 2	.012	022	1.016	.533	.599
PAMP ** 3	5.247E-005	.000	-.353	-.329	.745
(Constant)	-1.826	11.790		-.155	.878

VPR/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.933	.871	.854	2.679

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	1111.886	3	370.629	51.631	.000
Residual	165.103	23	7.178		
Total	1276.989	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	-.083	.536	-.212	.155	.878
PAMP ** 2	.005	.013	1.233	.417	.681
PAMP ** 3	5.154E-006	.000	-.095	-.057	.955
(Constant	1.067	6.718		.159	.875

PA_Systolic/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.981	.962	.958	6.107

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	21979.926	3	7326.642	196.419	.000
Residual	857.925	23	37.301		
Total	22837.852	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	2.081	1.222	1.261	1.703	.102
PAMP ** 2	-.005	.029	-.260	-.163	.872
PAMP ** 3	7.021E-006	.000	-.030	-.034	.973
(Constant)	-9.607	15.315		-.627	.537

PA_Diastolic/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.920	.846	.826	5.317

The independent variable is PAMP.

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	3566.390	3	1188.797	42.057	.000
Residual	650.129	23	28.266		
Total	4216.519	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	1.507	1.063	2.126	1.417	.170
PAMP ** 2	-.026	.025	-3.320	-1.027	.315
PAMP ** 3	.000	.000	2.186	1.201	.242
(Constant)	-8.713	13.332		-.654	.520

Output from the polynomial regression model using the manual measurements with the pressure data.

```
* Curve Estimation.
TSET NEWVAR=NONE.
CURVEFIT
  /VARIABLES=PA_Systolic PA_Diastolic TPG PVR PT RMPA LMPA Aorta WITH
PAMP
  /CONSTANT
  /MODEL=CUBIC
  /PRINT ANOVA
  /PLOT FIT
  /ID=label.
```

Curve Fit

Notes

Output Created		01-NOV-2012 16:36:34
Comments		
Input	Active Dataset	DataSet0
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	27
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Cases with a missing value in any variable are not used in the analysis.
Syntax		CURVEFIT /VARIABLES=PA_Systolic PA_Diastolic TPG PVR PT RMPA LMPA Aorta WITH PAMP /CONSTANT /MODEL=CUBIC /PRINT ANOVA /PLOT FIT /ID=label.
Resources	Processor Time	00:00:03.84
	Elapsed Time	00:00:04.04
Use	From	First observation
	To	Last observation
Predict	From	First Observation following the use period
	To	Last observation
Time Series Settings (TSET)	Amount of Output	PRINT = DEFAULT
	Saving New Variables	NEWVAR = NONE
	Maximum Number of Lags in Autocorrelation or Partial Autocorrelation Plots	MXAUTO = 16
	Maximum Number of Lags Per Cross-Correlation Plots	MXCROSS = 7
	Maximum Number of New Variables Generated Per Procedure	MXNEWVAR = 60
	Maximum Number of New Cases Per Procedure	MXPREDICT = 1000
	Treatment of User-Missing Values	MISSING = EXCLUDE
	Confidence Interval Percentage Value	CIN = 95
	Tolerance for Entering Variables in Regression Equations	TOLER = .0001
	Maximum Iterative Parameter Change	CNVERGE = .001

	Method of Calculating Std. Errors for Autocorrelations	ACFSE = IND
	Length of Seasonal Period	Unspecified
	Variable Whose Values Label Observations in Plots	Unspecified
	Equations Include	CONSTANT

[DataSet0]

Model Description

Model Name		MOD_2
Dependent Variable	1	PA_Systolic
	2	PA_Diastolic
	3	TPG
	4	PVR
	5	PT
	6	RMPA
	7	LMPA
	8	Aorta
Equation	1	Cubic
Independent Variable		PAMP
Constant		Included
Variable Whose Values Label Observations in Plots		label
Tolerance for Entering Terms in Equations		.0001

Case Processing Summary

	N
Total Cases	27
Excluded Cases[a]	0
Forecasted Cases	0
Newly Created Cases	0

a. Cases with a missing value in any variable are excluded from the analysis.

Variable Processing Summary

		Variables								
		Dependent								Independent
		PA_Systolic	PA_Diastolic	TPG	PVR	PT	RMPA	LMPA	Aorta	PAMP
Number of Positive Values		27	27	27	27	27	27	27	27	27
Number of Zeros		0	0	0	0	0	0	0	0	0
Number of Negative Values		0	0	0	0	0	0	0	0	0
Number of Missing Values	User-Missing	0	0	0	0	0	0	0	0	0
	System-Missing	0	0	0	0	0	0	0	0	0

PA_Systolic/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.981	.962	.958	6.107

The independent variable is PAMP.

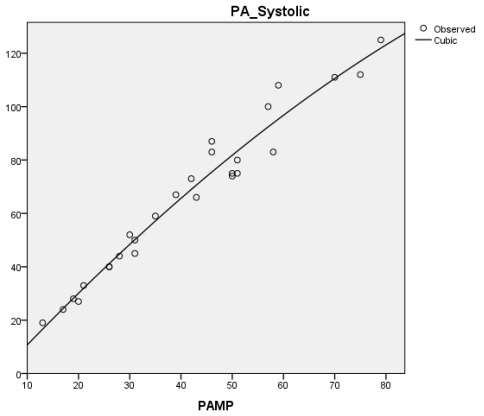
ANOVA

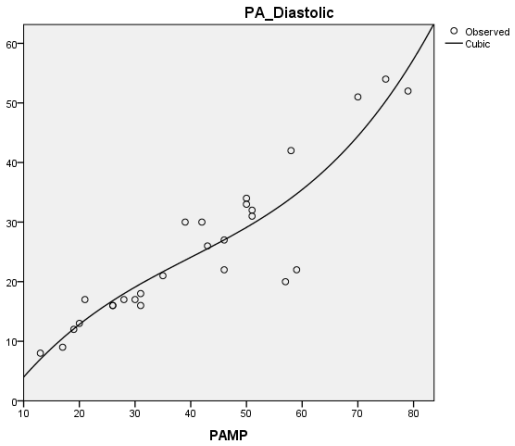
	Sum of Squares	Df	Mean Square	F	Sig.
Regression	21979.926	3	7326.642	196.419	.000
Residual	857.925	23	37.301		
Total	22837.852	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	2.081	1.222	1.261	1.703	.102
PAMP ** 2	-.005	.029	-.260	-.163	.872
PAMP ** 3	-7.021E-006	.000	-.030	-.034	.973
(Constant)	-9.607	15.315		-.627	.537





PA_Diastolic/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.920	.846	.826	5.317

The independent variable is PAMP.

ANOVA

	Sum of Squares	Df	Mean Square	F	Sig.
Regression	3566.390	3	1188.797	42.057	.000
Residual	650.129	23	28.266		
Total	4216.519	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	1.507	1.063	2.126	1.417	.170
PAMP ** 2	-.026	.025	-3.320	-1.027	.315
PAMP ** 3	.000	.000	2.186	1.201	.242
(Constant)	-8.713	13.332		-.654	.520

TPG/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.973	.946	.939	4.702

The independent variable is PAMP.

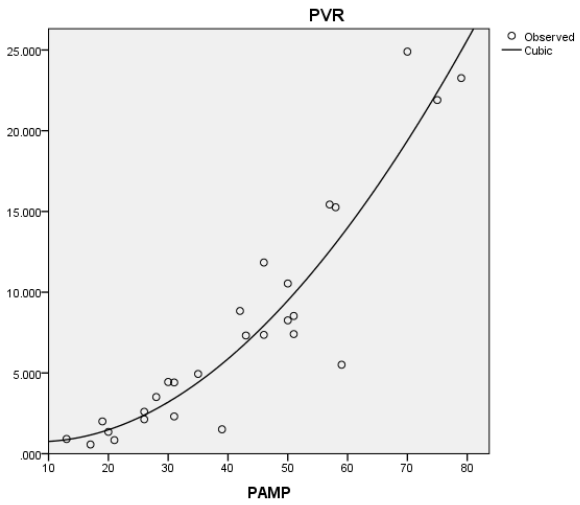
ANOVA

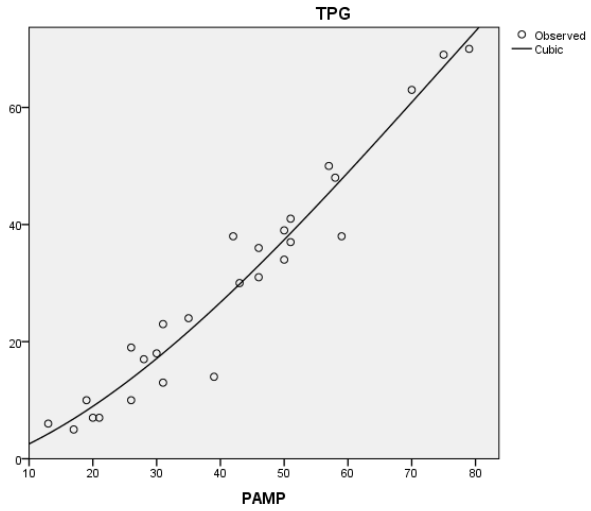
	Sum of Squares	Df	Mean Square	F	Sig.
Regression	8974.234	3	2991.411	135.303	.000
Residual	508.507	23	22.109		
Total	9482.741	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients		Sig.
	B	Std. Error	Beta		
PAMP	.323	.940	.304	.344	.734
PAMP ** 2	.012	.022	1.016	.533	.599
PAMP ** 3	-5.247E-005	.000	-.353	.329	.745
(Constant)	-1.826	11.790		.155	.878





PVR/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.933	.871	.854	2.679

The independent variable is PAMP.

ANOVA

	Sum of Squares	Df	Mean Square	F	Sig.
Regression	1111.886	3	370.629	51.631	.000
Residual	165.103	23	7.178		
Total	1276.989	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	-.083	.536	-.212	-.155	.878
PAMP ** 2	.005	.013	1.233	.417	.681
PAMP ** 3	-5.154E-006	.000	-.095	-.057	.955
(Constant)	1.067	6.718		.159	.875

PT/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.753	.568	.511	.380

The independent variable is PAMP.

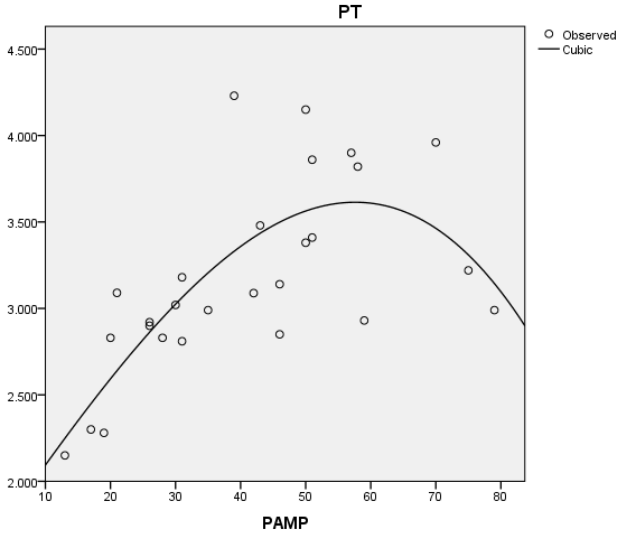
ANOVA

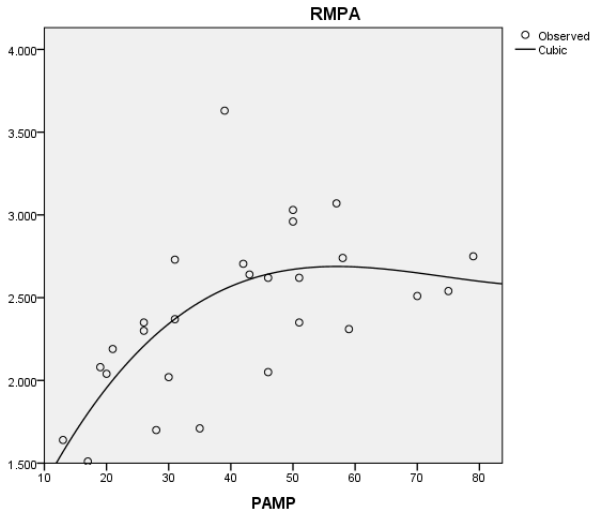
	Sum of Squares	Df	Mean Square	F	Sig.
Regression	4.363	3	1.454	10.066	.000
Residual	3.323	23	.144		
Total	7.686	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.054	.076	1.793	.714	.482
PAMP ** 2	-2.644E-005	.002	-.080	-.015	.988
PAMP ** 3	-5.148E-006	.000	-1.217	-.400	.693
(Constant)	1.559	.953		1.636	.116





RMPA/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.656	.431	.356	.392

The independent variable is PAMP.

ANOVA

	Sum of Squares	Df	Mean Square	F	Sig.
Regression	2.669	3	.890	5.801	.004
Residual	3.527	23	.153		
Total	6.196	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.095	.078	3.483	.208	.239
PAMP ** 2	-.001	.002	-4.523	.728	.474
PAMP ** 3	6.077E-006	.000	1.601	458	.651
(Constant)	.553	.982		563	.579

LMPA/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.778	.605	.553	.276

The independent variable is PAMP.

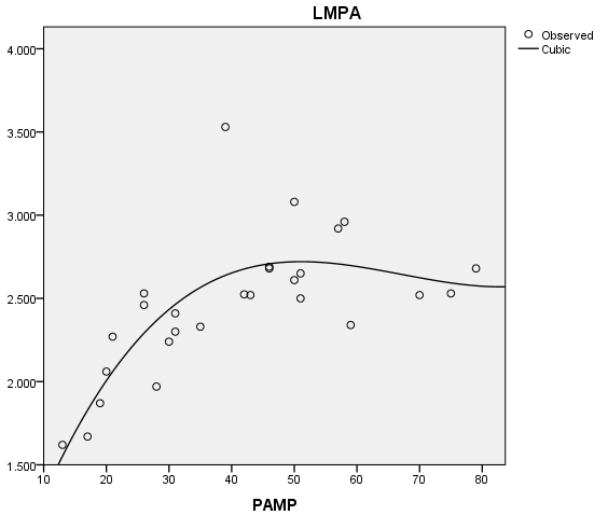
ANOVA

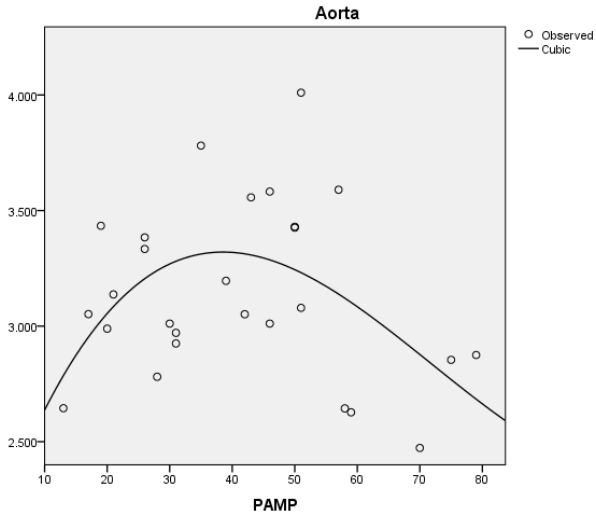
	Sum of Squares	Df	Mean Square	F	Sig.
Regression	2.688	3	.896	11.734	.000
Residual	1.756	23	.076		
Total	4.444	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.119	.055	5.182	2.158	.042
PAMP ** 2	-.002	.001	-7.465	-1.443	.163
PAMP ** 3	9.382E-006	.000	2.918	1.002	.327
(Constant)	.300	.693		.433	.669





Aorta/Cubic

Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.485	.236	.136	.352

The independent variable is PAMP.

ANOVA

	Sum of Squares	Df	Mean Square	F	Sig.
Regression	.877	3	.292	2.364	.097
Residual	2.845	23	.124		
Total	3.722	26			

The independent variable is PAMP.

Coefficients

	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
PAMP	.079	.070	3.774	1.130	.270
PAMP ** 2	-.001	.002	-6.098	-.847	.405
PAMP ** 3	6.547E-006	.000	2.225	.549	.588
(Constant	1.976	.882		2.241	.035

BIBLIOGRAPHY

1. Smeltzer, S.C. and B.G. Bare, *Brunner & Suddarth's Textbook of Medical Surgical Nursing*. 9 ed2000, Philadeliphia: Lippincott Williams and Wilkins. 1973.
2. Hyduk A., et al., *Pulmonary Hypertension Surveillance --- United States, 1980-2002.* CDC Surveillance Summaries, 2005. **54**(SS05): p. 1-28.
3. CDC, *Pulmonary Hypertension Fact Sheet*, D.o.H.a.H. Services, Editor 2010, Centers for Disease Control and Prevention.
4. Frost, A.E., et al., *The changing picture of patients with pulmonary arterial hypertension in the United States: how REVEAL differs from historic and non-US Contemporary Registries.* Chest, 2011. **139**(1): p. 128-37.
5. Badesch, D.B., et al., *Pulmonary arterial hypertension: baseline characteristics from the REVEAL Registry.* Chest, 2010. **137**(2): p. 376-87.
6. Rich, S., et al., *Primary pulmonary hypertension. A national prospective study.* Annals of internal medicine, 1987. **107**(2): p. 216-23.
7. D'Alonzo, G.E., et al., *Survival in patients with primary pulmonary hypertension. Results from a national prospective registry.* Annals of internal medicine, 1991. **115**(5): p. 343-9.
8. Thenappan, T., et al., *A USA-based registry for pulmonary arterial hypertension: 1982-2006.* The European respiratory journal : official journal of the European Society for Clinical Respiratory Physiology, 2007. **30**(6): p. 1103-10.
9. Rubin, G. and N. Rofsky, *CT and MR angiography: comprehensive vascular assesment*, ed. K.B. Lisa McAllister2009, China: Lippencott Williams & Wilkins a Wolters Kluwer buisness. 604-605.
10. McLaughlin, V.V., et al., *ACCF/AHA 2009 expert consensus document on pulmonary hypertension a report of the American College of Cardiology Foundation Task Force on Expert Consensus Documents and the American Heart Association developed in collaboration with the American College of Chest Physicians; American Thoracic Society, Inc.; and the Pulmonary Hypertension Association.* Journal of the American College of Cardiology, 2009. **53**(17): p. 1573-619.
11. Hegewald, M.J., B. Markewitz, and C.G. Elliott, *Pulmonary hypertension: clinical manifestations, classification and diagnosis.* International journal of clinical practice. Supplement, 2007(156): p. 5-14.
12. McLaughlin, V.V., A. Shillington, and S. Rich, *Survival in primary pulmonary hypertension: the impact of epoprostenol therapy.* Circulation, 2002. **106**(12): p. 1477-82.

13. Barst, R.J., et al., *Long-term outcome in pulmonary arterial hypertension patients treated with subcutaneous treprostinil.* The European respiratory journal : official journal of the European Society for Clinical Respiratory Physiology, 2006. **28**(6): p. 1195-203.
14. Badesch, D.B., et al., *Medical therapy for pulmonary arterial hypertension: updated ACCP evidence-based clinical practice guidelines.* Chest, 2007. **131**(6): p. 1917-28.
15. Hinchcliff, M. and J. Varga, *Systemic sclerosis/scleroderma: a treatable multisystem disease.* American family physician, 2008. **78**(8): p. 961-8.
16. Voelkel, N.F., et al., *Right ventricular function and failure: report of a National Heart, Lung, and Blood Institute working group on cellular and molecular mechanisms of right heart failure.* Circulation, 2006. **114**(17): p. 1883-91.
17. Devaraj, A., et al., *Detection of pulmonary hypertension with multidetector CT and echocardiography alone and in combination.* Radiology. **254**(2): p. 609-16.
18. McGoon, M., et al., *Screening, early detection, and diagnosis of pulmonary arterial hypertension: ACCP evidence-based clinical practice guidelines.* Chest, 2004. **126**(1 Suppl): p. 14S-34S.
19. Tan, R.T., et al., *Utility of CT scan evaluation for predicting pulmonary hypertension in patients with parenchymal lung disease. Medical College of Wisconsin Lung Transplant Group.* Chest, 1998. **113**(5): p. 1250-6.
20. Hoeper, M.M., et al., *Complications of right heart catheterization procedures in patients with pulmonary hypertension in experienced centers.* J Am Coll Cardiol, 2006. **48**(12): p. 2546-52.
21. NHLBI. *Cardiac Catheterization*. What are the risks of cardiac catheterization? 2010 [cited 2010 november 19, 2010]; Available from: http://www.nhlbi.nih.gov/health/dci/Diseases/cath/cath_risk.html.
22. Grubstein, A., et al., *Computed tomography angiography in pulmonary hypertension.* Isr Med Assoc J, 2008. **10**(2): p. 117-20.
23. Engelke, C., et al., *High-resolution CT and CT angiography of peripheral pulmonary vascular disorders.* Radiographics, 2002. **22**(4): p. 739-64.
24. Ng, C.S., A.U. Wells, and S.P. Padley, *A CT sign of chronic pulmonary arterial hypertension: the ratio of main pulmonary artery to aortic diameter.* J Thorac Imaging, 1999. **14**(4): p. 270-8.
25. Edwards, P.D., R.K. Bull, and R. Coulden, *CT measurement of main pulmonary artery diameter.* Br J Radiol, 1998. **71**(850): p. 1018-20.
26. Devaraj, A., et al., *Detection of Pulmonary Hypertension with Multidetector CT and Echocardiography Alone and in Combination.* Rasiology, 2010. **254**(2): p. 609-616.
27. Abel, E., et al., *Pulmonary artery and right ventricle assessment in pulmonary hypertension: correlation between functional parameters of ECG-gated CT and right-side heart catheterization.* Acta radiologica, 2012. **53**(7): p. 720-7.
28. Chan, H.P., et al., *Computer-aided diagnosis of lung cancer and pulmonary embolism in computed tomography-a review.* Acad Radiol, 2008. **15**(5): p. 535-55.
29. Anonymous. *The Columbia Encyclopedia*. X ray 2008 [cited 2009 October 16th]; 6th:[Available from: http://www.encyclopedia.com/topic/X_ray.aspx.
30. Yoo TS, Stetten GD., and Lorensen B., *Insight into Images: Principles and Practice for Segmentation, Registration and Image Analysis*.

, ed. T.S. Yoo2004, Wellesey, MA: AK Peters, Ltd. 3-45.

31. Kevles, B.H., *Naked to the Bone Medical Imaging in the Twentieth Century*1997, Reading, Massachusettes: Helix Books & Addison Wesley. 17-76.
32. NASA. *X-rays*. Electromagnetic Spectrum 2007 [cited 2009 November 2]; Available from: http://science.hq.nasa.gov/kids/imagers/ems/xrays.html.
33. Harris, T. *How X-rays Work*. How Stuff Works 2009 [cited 2009 October 26,2009]; Available from: http://health.howstuffworks.com/xray.htm.
34. Hugh Young and R. Freedman, *University Physics with Modern Physics*. Tenth ed, ed. Lisa Weber and J. Lake2000, San Fransico: Addison-Wesley Longman Inc. 1232-1238, 1254-1263, 1402-1404.
35. Boone, J., *Handbook of Medical Imaging*. Physics and Psychophysics, ed. Jacob Beutel, Harold Kundel, and R.V. Metter. Vol. 1. 2000, Bellingham, Washington: SPIE The International Society of Optical Engineering. 1-77.
36. Raymond Serway and J. Faughn, *College Physics*, ed. E. Ahrens1999, Fort Worth: Harcourt College Publishers.
37. Hsieh, J., *Computed TomographyL Principles, Design, Artifacts and Recent Advances*. Vol. 1-12, 147-233. 2003, Belligham, Washington: SPIE Press.
38. Goodenough, D., *Handbook of Medical Imaging*. Physics and Psychophysics, ed. Jacob Beutel, Harold Kundel, and R.V. Metter. Vol. 1. 2000, Bellingham, Washington: SPIE The International Society of Optical Engineering. 511-552.
39. FDA. *What is Computed Tomography*. Radiation-Emitting Products 2009 November 30, 2009 10/30/2009]; Available from: http://www.fda.gov/Radiation-EmittingProducts/RadiationEmittingProductsandProcedures/MedicalImaging/MedicalX-Rays/ucm115318.htm.
40. NIH. *National Heart and Lung Blood Institute Diseases and Conditions Index*. Pulmonary Embolism 2010 [cited 2010 August 2010]; Available from: http://www.nhlbi.nih.gov/health/dci/Diseases/pe/pe_diagnosis.html.
41. Gonzalez, R. and R. Woods, *Digital Image Processing*1992, Reading, Massachusetts: Addison-Wesley Publishing Company.
42. Yazdi, M. and L. Beaulieu, *Artifacts in Spiral X-ray CT Scanners: Problems and Solutions.* International Journal of Biological and Life Sciences, 2008. **4**(3): p. 135-139.
43. Keith, N., *A Magnet With a View: Exploring new frontiers in MRI* Berkley Science Review, 2009: p. 19-23.
44. Pickens, D., *Handbook of Medical Imaging*. Physics and Psychophysics, ed. Jacob Beutel, Harold Kundel, and R.V. Metter. Vol. 1. 2000, Bellingham, Washington: SPIE The International Society of Optical Engineering. 375-458.
45. E. Arias-Castro and D.L. Donoho, *Does median filtering truly preserve edges better than linear filtering?* Annals of Statistics, 2009. **37**(3): p. 1172-1206.
46. Zitova, B. and J. Flusser, *Image registration methods: a survey*. Image and Vision Computing, 2003(21): p. 977-1000.
47. Masutani, Y., H. MacMahon, and K. Doi, *Automated segmentation and visualization of the pulmonary vascular tree in spiral CT angiography: an anatomy-oriented approach based on three-dimensional image analysis.* J Comput Assist Tomogr, 2001. **25**(4): p. 587-97.
48. Zhou, C., et al., *Automatic multiscale enhancement and segmentation of pulmonary vessels in CT pulmonary angiography images for CAD applications.* Med Phys, 2007. **34**(12): p. 4567-77.

49. Dougherty, E. and R. Lotufo, *Hands-on Morphological Image Processing*. Tutorial Texts in Optical Engineering, ed. A. WeeksJr.2003, Bellingham, Washington: SPIE Press. 1-44, 79-90.
50. Marsland, S., *Machine Learning An algorithmic Perspective*. machine learning & pattern recognition series2009, Boca Raton, FL: Chaoman & Hall/CRC, Taylor & Francis Group, LLC. 383.
51. Mitchell, T., *Machine Learning*, 1997, McGraw Hill.
52. David W. Hosmer, S.L., *Applied Logistic Regression Second Edition*, 2000, John Wiley & Sons, Inc.
53. Robert Nisbet, J.E.I., Gary Miner, *Handbook of Statistical Analysis and Data Mining Applications*, 2009, Elsiever Inc.
54. Mandy C. Phelps, E.C.M., *Classification and Regression Trees as Alternatives to Regression.* Proceedings of the 4th Annual GRASP Symposium, 2008.
55. Neopolitan, R.E., *Learning Bayesian Networks*. Artificail Intelligence, ed. P. Hall2004, Upper Saddle River, NJ: Pearson Prentice Hall.
56. Anonymous *Naive Bayes Classifier*. Summary description of Naive Bayes.
57. Luis Ibanez, et al., *Insight Tool Kit*, 2005, Kitware Inc.
58. Kitware. *Visualization Toolkit*. 2008-09; Available from: http://www.vtk.org/.
59. Aric A. Hagberg, Daniel A. Shult, and P.J. Swart, *Exploring network structure, dynamics, and function using NetworkX.* Proceedings of the 7th Python in Science Canference (SciPy 2008), 2008: p. 11-15.
60. *itk-SNAP*. 2012; Available from: http://www.itksnap.org/pmwiki/pmwiki.php?n=Main.HomePage.
61. Homann, H., *Implementaion of a 3D thinning algorithm.* The Insight Journal, 2007.
62. Lee, T. and R. Kashyap, *Building Skeleton Models via 3-D Medial Surface/Axis Thinning Algorithms.* CVGIP: Graphical Models and Image Processing, 1994. **56**(6): p. 462-478.
63. Early, E., *On the Euler Characteristic.* MIT Undergraduate journal of Mathematics: p. 37-47.
64. Mark Hall, E.F., Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, *The WEKA Data Mining Software: An Update.* SIGKDD Explorations, 2009. **11**(1).
65. FDA, *Draft Guidance on Metronidazole*, 2010.
66. Rosner, B., *Fundamentals of Bostatistics*. 6th ed2006, United States of America: Thompson Brooks/Cole.
67. Corporation, I., *IBM SPSS Statistics for Windows*, 2012, IBM Corp.: Armonk, NY.
68. Analyse-it Software, L., *Analyse-it for Microsoft Excel (version 2.20)*, 2009.