

# A CLOUD-BASED SYSTEM FOR MEASURING RADIATION TREATMENT PLAN SIMILARITY

by

JENNIFER ANDREA

A thesis submitted to the  
School of Computing  
in conformity with the requirements for  
the degree of Master of Science

Queen's University  
Kingston, Ontario, Canada  
September 2016

Copyright © Jennifer Andrea, 2016

# Abstract

**PURPOSE:** Radiation therapy is used to treat cancer using carefully designed plans that maximize the radiation dose delivered to the target and minimize damage to healthy tissue, with the dose administered over multiple occasions. Creating treatment plans is a laborious process and presents an obstacle to more frequent replanning, which remains an unsolved problem. However, in between new plans being created, the patient’s anatomy can change due to multiple factors including reduction in tumor size and loss of weight, which results in poorer patient outcomes. Cloud computing is a newer technology that is slowly being used for medical applications with promising results. The objective of this work was to design and build a system that could analyze a database of previously created treatment plans, which are stored with their associated anatomical information in studies, to find the one with the most similar anatomy to a new patient. The analyses would be performed in parallel on the cloud to decrease the computation time of finding this plan. **METHODS:** The system used SlicerRT<sup>1</sup>, a radiation therapy toolkit for the open-source platform 3D Slicer<sup>2</sup>, for its tools to perform the similarity analysis algorithm. Amazon Web Services was used for the cloud instances on which the analyses were performed, as well as for storage of the radiation therapy studies and messaging between the instances

---

<sup>1</sup>[www.slicerrt.org](http://www.slicerrt.org)

<sup>2</sup>[www.slicer.org](http://www.slicer.org)

and a master local computer. A module was built in SlicerRT to provide the user with an interface to direct the system on the cloud, as well as to perform other related tasks. **RESULTS:** The cloud-based system out-performed previous methods of conducting the similarity analyses in terms of time, as it analyzed 100 studies in approximately 13 minutes, and produced the same similarity values as those methods. It also scaled up to larger numbers of studies to analyze in the database with a small increase in computation time of just over 2 minutes. **CONCLUSION:** This system successfully analyzes a large database of radiation therapy studies and finds the one that is most similar to a new patient, which represents a potential step forward in achieving feasible adaptive radiation therapy replanning.

## Statement of Co-Authorship

The work outlined in this thesis was completed under the mentorship of Dr. Gabor Fichtinger and Csaba Pinter of the Laboratory for Percutaneous Surgery.

An earlier version of this work was presented in a paper at the World Congress on Medical Physics and Biomedical Engineering in June 2015, and in an abstract at the 13th Imaging Network Ontario Symposium in March 2015.

## Acknowledgments

First, I would like to thank Dr. Gabor Fichtinger, whose mentorship throughout my undergraduate and graduate studies I will forever be grateful for. Your unending encouragement and pushing me to always produce the best work I was capable of have helped me to do better than even I sometimes thought possible. The wonderful opportunities I have received through your guidance are ones I will always remember.

I am so appreciative of the help and guidance I have received from Csaba Pinter, research engineer in the Perk Lab, throughout the years. Thank you for all of your help when I hit obstacles, both big and small, and all you have taught me over the years about radiation therapy.

I am grateful to my friends and colleagues in the Perk Lab, some of whom I have known since my undergraduate studies. You have made my experience as an undergraduate and graduate student a wonderful one, and I am so thankful to have worked in a lab that was so supportive and inspiring. I will miss our tea time, sushi lunches, and what was on occasions almost too good company.

Last, but certainly not least, I would like to thank my parents and boyfriend Tim for their unwavering love and support throughout the entirety of my Master's

studies. Thank you for the late night phone calls when I was stressed, the frequent brainstorming sessions when I was stuck or just wanted to make my research work better, and always being there when I needed you most.

My research work was in part supported by an Amazon Web Services Research Grant, and I was personally supported by the Ontario Graduate Scholarship. This work was also funded by Cancer Care Ontario through Applied Cancer Research Unit and Research Chair in Cancer Imaging and OCAIRO grants.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Statement of Co-Authorship</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Contents</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>ix</b>
<b>Glossary</b>	<b>1</b>
<b>Chapter 1: Introduction and background</b>	<b>3</b>
1.1 Radiation therapy treatment planning . . . . .	3
1.1.1 Adaptive radiation therapy . . . . .	6
1.2 Cloud computing . . . . .	13
1.3 Thesis contributions . . . . .	19
<b>Chapter 2: System design and implementation</b>	<b>20</b>
2.1 System overview . . . . .	20
2.2 Software infrastructure . . . . .	21
2.2.1 3D Slicer . . . . .	21
2.2.2 SlicerRT . . . . .	21
2.2.3 Amazon Web Services and Boto . . . . .	22
2.3 System architecture . . . . .	22
2.4 Workflow . . . . .	23
2.4.1 Set cloud parameters . . . . .	23
2.4.2 Upload study to cloud . . . . .	24
2.4.3 Find most similar plan . . . . .	25
2.5 Similarity analysis computation . . . . .	29

2.6	User interface . . . . .	31
<b>Chapter 3:</b>	<b>Results, validation and discussion</b>	<b>33</b>
3.1	Testing methodology . . . . .	33
3.1.1	Test data . . . . .	33
3.1.2	Testing hardware . . . . .	34
3.1.3	Computation time . . . . .	34
3.1.4	Accuracy . . . . .	35
3.2	Results and discussion . . . . .	36
3.2.1	Results for computation time . . . . .	36
3.2.2	Results for accuracy . . . . .	43
3.2.3	Earlier iterations . . . . .	44
<b>Chapter 4:</b>	<b>Conclusion</b>	<b>46</b>
4.1	Summary . . . . .	46
4.2	Future Work . . . . .	47
	<b>Bibliography</b>	<b>50</b>



# List of Tables

3.1	Results of accuracy testing . . . . .	44
-----	---------------------------------------	----

# List of Figures

1.1	Contours and radiation dose for a head and neck case in SlicerRT . .	4
1.2	Diagram showing the clinical target volume (CTV) which contains the tumor, the planning target volume (PTV) which adds a margin of error to the CTV, and the nearby healthy organs at risk (OAR) . . . . .	5
1.3	Dose volume histogram for a prostate case in SlicerRT . . . . .	6
1.4	Overview of the different cloud service models and example use cases (wikipedia.org) . . . . .	15
2.1	Overview of the system architecture . . . . .	23
2.2	“Set cloud parameters” section of the module prior to the parameters being set . . . . .	24
2.3	“Upload study to cloud” section of the module with prostate cancer CT and contours loaded and parameters set . . . . .	25
2.4	“Find most similar plan” section of the module at the end of the computation . . . . .	28
2.5	Workflow of the similarity analysis computation . . . . .	30
2.6	“Find most similar plan” section prior to parameters being set . . . .	32
3.1	Average computation times for different analysis methods . . . . .	37
3.2	Average computation time using high cloud instance level . . . . .	38

3.3	Variability of similarity analysis computation times when using high cloud instance level . . . . .	39
3.4	Average computation times for different analysis methods . . . . .	39
3.5	Average computation time using medium instance level . . . . .	40
3.6	Average number of comparisons per instance for medium instance level	40
3.7	Average computation time using low instance level . . . . .	41
3.8	Average number of comparisons per instance for low instance level . .	41
3.9	Cloud overhead time for different numbers of instances . . . . .	41
3.10	Overhead measured in percentage of computation time for different number of instances . . . . .	42
3.11	Average time required to start up cloud instances . . . . .	43

## Glossary

**AMI** Amazon Machine Image.

**ART** Adaptive Radiation Therapy.

**AWS** Amazon Web Services.

**CBCT** Cone Beam Computed Tomography.

**cl-PTV** confidence limited Planning Target Volume.

**CPU** Central Processing Unit.

**CT** Computed Tomography.

**CTV** Clinical Target Volume.

**DVH** Dose Volume Histogram.

**EC2** Elastic Compute Cloud.

**GPU** Graphics Processing Unit.

**GUI** Graphical User Interface.

**IaaS** Infrastructure as a Service.

**IGRT** Image-Guided Radiation Therapy.

**IMRT** Intensity-Modulated Radiation Therapy.

**LAN** Local Area Network.

**MLC** Multileaf Collimator.

**OAR** Organs at Risk.

**OVH** Overlap Volume Histogram.

**PaaS** Platform as a Service.

**PACS** Picture Archiving and Communication System.

**PTV** Planning Target Volume.

**RAM** Random Access Memory.

**RT** Radiation Therapy.

**S3** Simple Storage Service.

**SaaS** Software as a Service.

**sim-CT** Computed Tomography Simulator.

**SQL** Structured Query Language.

**SQS** Simple Queue Service.

**VM** Virtual Machine.

**VMAT** Volumetric Arc Therapy.

# Chapter 1

## Introduction and background

### 1.1 Radiation therapy treatment planning

Radiation therapy (RT) is a type of cancer treatment in which ionizing radiation is used to kill cancerous cells by damaging their DNA. Radiation therapy is used to treat cancer directly by irradiating the tumour, used after surgery to prevent cancer recurrence, and used to reduce tumor size and its associated painful side effects in palliative treatment. Intensity-modulated radiation therapy (IMRT) is a form of radiation therapy in which the radiation dose is delivered via external beams which have varied intensity across the beam. This is done by using a multileaf collimator (MLC) to shape the radiation beam and control the intensity of small segments of the radiation beam, called beamlets, as determined by the treatment plan. This allows for higher doses of radiation to be delivered to the targets and allows for improved sparing of healthy tissue. IMRT is used most commonly to treat prostate, central nervous system, and head and neck cancer (as seen in Figure 1.1), but has also been used to treat other types of cancer including breast, thyroid, and lung. IMRT treatment is directed by carefully designed plans in order to maximize the

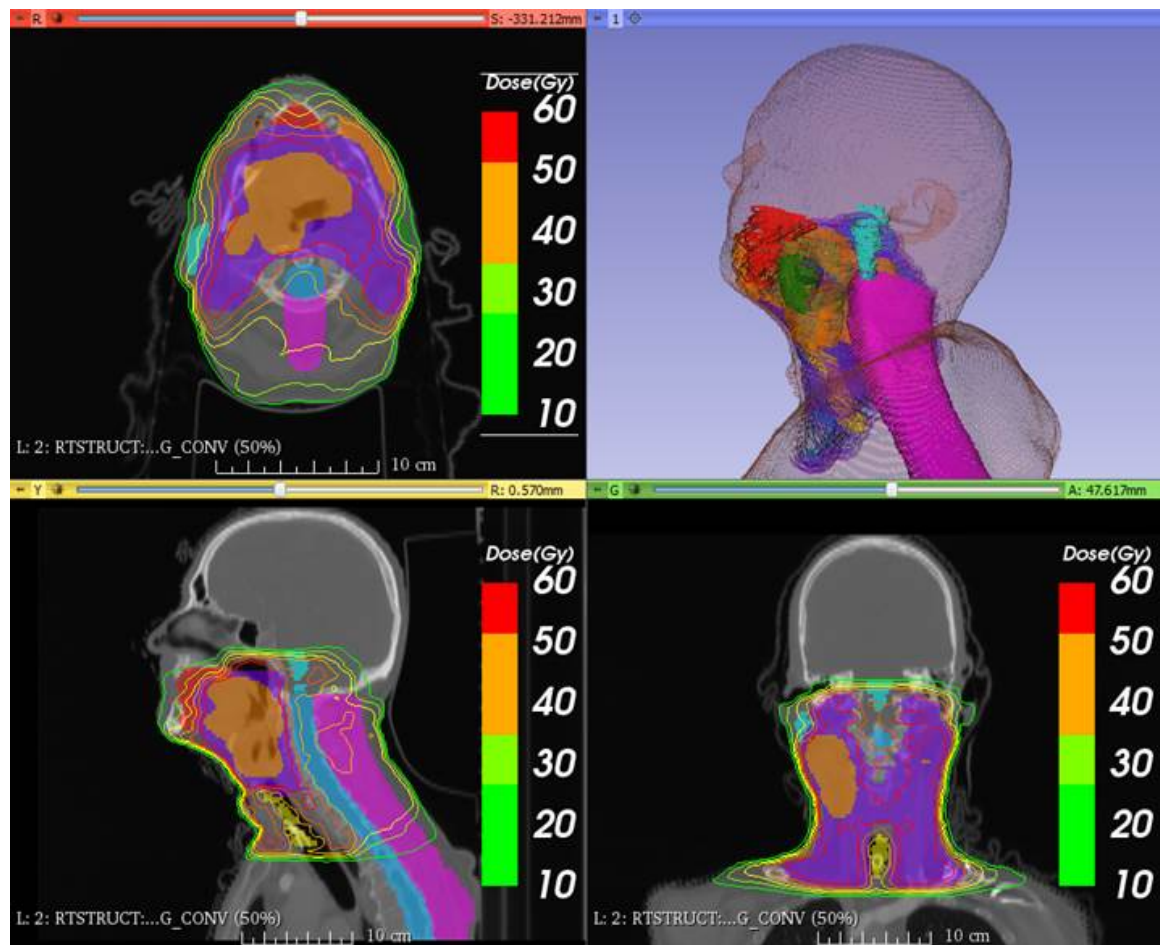


Figure 1.1: Contours and radiation dose for a head and neck case in SlicerRT

dose delivered to the target and minimize the dose delivered to healthy tissue and any nearby organs at risk (OAR). Volumetric arc therapy (VMAT) is an advanced form of IMRT that is the current state of the art therapy in clinical practice. During VMAT, the linear accelerator that is used to deliver the treatment is rotated with the radiation continuously delivered, while the shape of the MLC aperture and the dose rate are changed to deliver a dose distribution that is sculpted to the target volume. At the beginning of the treatment planning process, the patient is scanned, typically with computed tomography (CT). Contour lines are manually or semi-automatically

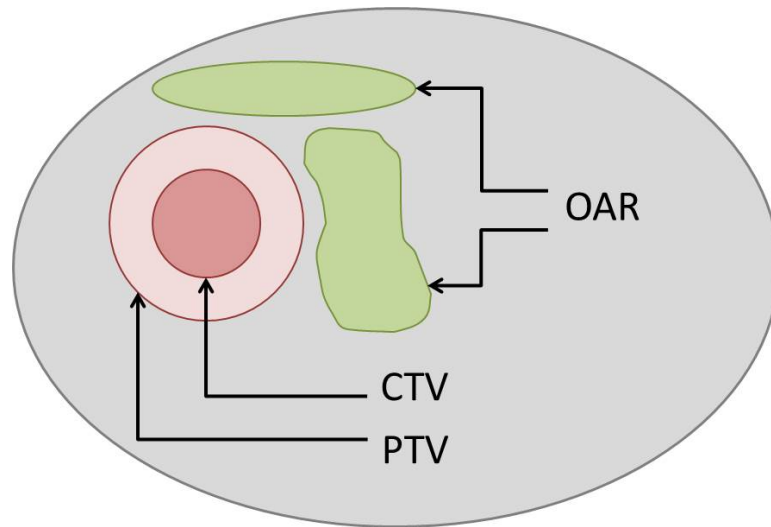


Figure 1.2: Diagram showing the clinical target volume (CTV) which contains the tumor, the planning target volume (PTV) which adds a margin of error to the CTV, and the nearby healthy organs at risk (OAR)

drawn around the structures of interest on each slice of the CT scan. The treatment plan is created using the scan, contours (as seen in Figure 1.2), and the prescribed dose, and optimized for the doses received by the target and OAR. The treatment plans are generally evaluated by examining their dose volume histograms (DVH), which as seen in Figure 1.3, relate the amount of radiation dose to the tissue volume receiving that dose. As a result, it typically takes several hours to create a plan for an individual patient [1]. The radiation dose is given in fractions over a period of several weeks [2], as giving the total dose in a single treatment results in irreparable healthy tissue damage. The patient will experience anatomical changes during the typical course of treatment, due to a number of factors including decrease in tumor size, weight loss, and change in bladder or stomach volume [2]. To attempt to correct for these changes, pretreatment imaging is performed before each dose fraction, and this image is used to position the patient to best match the treatment plan [3], which is



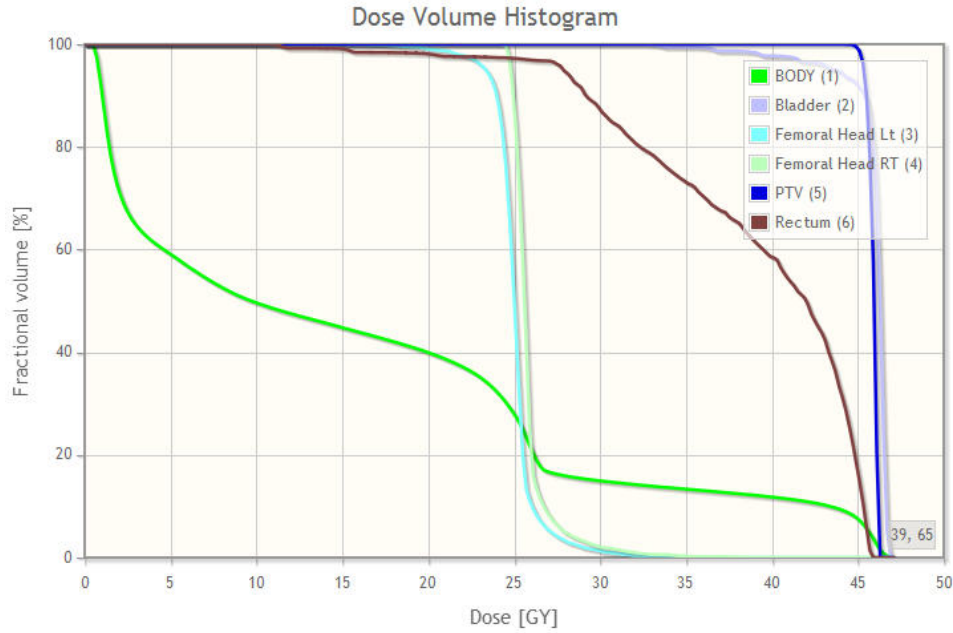


Figure 1.3: Dose volume histogram for a prostate case in SlicerRT

called image-guided radiation therapy (IGRT). However, this correction does not fully account for these anatomical changes [4], so the delivered dose might be significantly different from the planned dose distribution [2]. This can result in problems with adequately dosing the tumor, and complications with healthy tissue [2].

### 1.1.1 Adaptive radiation therapy

Adaptive radiation therapy (ART) is a strategy to produce treatment plans that best match the patient's anatomy during dose fractions, either through modifications to the original treatment plan [5] or replanning [6]. Both strategies are currently resource-intensive, as they require additional staff time [6]. Studies have shown advantages of using ART over traditional IGRT. A study conducted by Schwartz *et al.* on 24 head and neck cancer cases showed that using ART replanning resulted in lower toxicity and 95% regional disease control, as well as increased sparing to certain organs at

risk over IGRT [7]. The different ART strategies can be divided into two categories, those being offline and online methods.

Full replanning involves creating a new plan for each dose fraction delivered to the patient based on daily imaging, in order to account for their anatomical changes. Full replanning is the most versatile method of correcting for the changed anatomical geometry [8] and in theory provides the highest precision [9]. It is typically an offline method that is carried out prior to daily treatment. One benefit of full replanning can be seen in the study conducted by Wu *et al.*, who showed that full replanning results in increased sparing of organs at risk, particularly the parotid glands in head and neck cases. Additionally, replanning results in dose distributions that most closely match the original plan [10]. However, due to its high workload, full replanning has never been clinically implemented [8]. It also has disadvantages in that anatomical changes can occur between the last image being taken of the patient, and the new plan being delivered.

A variant of full replanning was investigated by Yan *et al.*, as between treatments they simulated and attempted to predict offline how the OAR and planning target volumes (PTV) would change shape and deform over time [11]. They used the daily lower quality cone beam CT (CBCT) imaging conducted using the linear accelerator's on-board imaging for patient setup prior to IGRT treatment, and created a dose distribution for the changed anatomical shapes predicted for the patient. In another study, they tested their system's ability to generate what they called a confidence-limited PTV (cl-PTV) that is patient-specific using their prediction model so that the cl-PTV would best match the patient's anatomy during treatment [12]. In addition to

using the daily CBCT setup imaging to create the prediction model, the patient also received biweekly higher quality CT simulator (sim-CT) scans that are typically used for treatment planning. The authors found that the cl-PTV had smaller margins than the conventional PTV, which would result in increased sparing of the OAR due to smaller MLC apertures being used for treatment. However, in order to achieve optimal results, their procedure required more imaging than would be clinically practical, and it did not completely handle systematic anatomical variation.

A different replanning strategy was explored by Kazhdan *et al.*, where they looked at finding the most anatomically appropriate plan match for the current patient from a database of previously treated patients [1]. This plan would then either be used to treat the current patient directly, if appropriate, or used as the starting point that would be optimized further. In theory, using this approach would eventually facilitate automated treatment planning [13]. To enable faster searching through the database, they developed a shape descriptor called the overlap volume histogram (OVH), which considers the distribution of relative distances of different nearby organs from the tumor [1]. Using the OVHs for the relevant anatomical volumes, they successfully retrieved a plan that when evaluated for the new patient, had acceptable DVHs for those volumes. Using the OVH to evaluate plans has an additional advantage in that it only needs to be calculated once for each plan in the database. In a later study, the authors showed that while creating a new plan normally requires many rounds of optimization, plans retrieved using OVHs could produce the highest quality plans in less than two optimization rounds [13]. However, the authors' intended purpose of this method was for creating plans for new patients, rather than for creating plans for later dose fractions due to changes in anatomy. Since new plans are created

from pretreatment sim-CT imaging, rather than in-room CBCT imaging prior to treatment, computation time is not as much of a concern as clinician time. Likely for this reason, the authors did not present any results on the length of time required to create an OVH, or on the time required to query the database using the OVHs.

A similar approach to that of Kazhdan *et al.* was tested by Chanyavanich *et al.*, where they searched a database for previous cases that best match the new patient, using the radiation beam's eye view of the contoured structures [14]. The best matching case was then used to create the new patient's treatment plan, using the treatment parameters from that case, followed by minor optimization. The resulting treatment plans, which were evaluated using their DVHs against plans developed by an expert planner, were found to have comparable PTV coverage, and similar or better dose sparing of organs at risk. As the authors intended their system to be used to create initial treatment plans, while the planning times of two to three minutes were reported, the times required to query the database were not.

A similar strategy to this is case-based reasoning, which uses previous similar cases to find a solution to a new problem [15]. Berger applied case-based reasoning to radiation therapy treatment planning with the Roentgen system, which used an archive of previous patient plans to find one appropriate for a new patient, then adjusted it to match that patient [16]. The previous plans were evaluated based on the patient geometries and treatment constraints. The patient geometries were assessed by approximating each of the important tissues with a best-fit ellipse, then evaluating the ellipses on multiple features. The treatment constraints were assessed based on the results of applying each plan to the patient it was created for, which were represented

as DVHs and a metric representing how well the results matched the original plan. The metrics used for the treatment constraints can be computed once for each plan and stored with it. Once the best match was found, it was evaluated against the new patient to determine areas where the plan needed to be improved, which were then modified. However, in improving the plan, the author used an assumption that the plan would be similar enough to require little modification due to the size of the archive of plans being searched through. The author also did not report any testing results or computation times.

With the large amount of time required for full replanning for subsequent dose fractions, a number of different strategies have been tested for modifying the original treatment plan to better match the patient's daily anatomy, and are typically online methods. However, this strategy also suffers from being time consuming, labor intensive, and automated methods lack robustness in automating their subprocedures which include image segmentation and dose summation [17], so research is ongoing in an attempt to address those issues and also to find successful automated replanning methods.

Birkner *et al.* used the approach described by Yan *et al.* for calculating the patient's anatomical deformations [18]. They used it to quantify how much the patient's anatomy had changed, and track the movement of subvolumes in the image, allowing them to predict future variation. This was used to predict the biologically effective dose distribution, which was then used to reoptimize the initial treatment plan through inverse planning. The success of this strategy greatly depends on the number of daily images taken of the patient at that point, providing maximum tumor

coverage after five days of daily images were taken, but underdoses the tumor prior to the fifth day. This approach also does not correct for any patient setup errors, as it is an offline correction strategy.

Schwartz *et al.* used a different strategy for modifying the initial treatment plan, as they used deformable image registration to register the planning sim-CT to the daily CBCT taken prior to treatment, and used the resulting transform to deform the planning contours so that they also match the daily CBCT to provide some correction for smaller changes in anatomy [7]. The deformed contours are also used for plan evaluation, as in the case where original dose plan resulted in an inadequate treatment using the deformed contours, dose recalculation and replanning would be performed. Their system identified the need for ART replans at least once for all patients, and twice for 36% of patients. Their system resulted in 100% local disease control at the two-year follow up, as well as 95% regional disease control. Performing the deformable registration and contour transformation took “seconds”, however ART replanning would certainly take much longer. The dose should not currently be deformed along with the contours, as the linear accelerator cannot deform the beams to match. Additionally, there are some issues with using deformable registration to evaluate the current treatment plan on the changed anatomy, as there are no objective metrics to evaluate deformable registration results [19]. Deformable registration, which from a survey of the literature is very popular in ART planning research, is more error-prone than rigid registration, and unlike rigid registration, there is no way to correct the registration results [19]. Another study that used deformable image registration performed the registration on a graphics processing unit (GPU) for improved computation time, which was less than a minute for the registration [17]. The

dose calculation was then performed on the deformed image. However, this study fails to address the lack of metrics for objectively assessing registration success. Mohan *et al.* used a different strategy to adjust the dose distributions following deformable image registration, as they used the registration transformation to deform the intensity distributions for each individual beam, which were then transformed into leaf sequences which would be used to deliver the treatment [20]. This strategy in effect deforms the dose, but in such a way that it can be delivered by the linear accelerator. This strategy required some manual guidance and also resulted in reduced sparing of normal tissues.

The approach investigated by Zhang *et al.* used a database of expert-designed treatment plans, with the associated beam angles, tumor position, and tumor size stored with each plan [21]. These expert plans were used in automated treatment planning to make decisions about non-coplanar beam angles, by finding the most similar cases in the database, and using the five most common angles from those cases. The coplanar angles were chosen using frequency distributions of beam angles for each tumor position, with the frequency distributions having been created from the database of expert plans. The authors standardized the structures and their initial objective function parameters for the particular site, which in this study was lung. Their system then automatically adjusted the objective function parameters and beam angles so that the most optimal doses possible were delivered to the different structures in the final treatment plan. The authors found that their system consistently generated treatment plans at least as good as those created by expert dosimetrists at their institution. However, their system required the creation of a

curated database of expert-designed treatment plans, and appears to have only contained lung cancer cases. Additionally, the authors reported total computation times of twenty minutes to automatically create plans, and did not report the time required to search through their database for the most similar cases. Zarepisheh *et al.* also used a database of treatment plans, but they used it to select a reference treatment plan most similar to a new patient, which was then automatically optimized using a GPU [22]. In ART cases, the authors would use the patient's initial treatment plan as the reference plan for optimization. Their optimization algorithm produced treatment plans that were better than the plans actually created for the patients in the three cases tested. While the authors presented excellent computation times for the automated optimization algorithm, which were approximately seven to twelve seconds depending on the size of the test case, times for querying the database were not presented, and the database that was queried only contained cases of the relevant cancer type.

While many different strategies for automated ART replanning have been examined, the problem still remains unsolved. Additionally, from the literature it appears that time-efficient strategies have not yet been found for querying large databases of treatment plans to find the most similar one to either a new patient or a current patient whose anatomy has changed.

## 1.2 Cloud computing

Cloud computing is a relatively new computational paradigm, that was built on ideas originally derived for grid computing [23]. It provides services to users over the Internet, and covers a broad spectrum from virtual hardware to individual software



applications. Resources are dynamically allocated to users on-demand from a shared pool, providing scalability and flexibility. Users pay for only what resources they use for the time period they use them for, and there are no up-front infrastructure costs. Maintenance of cloud resources is handled by the cloud provider, and upgrading to newer hardware or software is cheaper and more convenient for users.

Cloud computing has three main service models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), as can be seen in Figure 1.4. With IaaS, multiple virtual machines operate on top of a single piece of infrastructure hardware, with each virtual machine (VM) isolated from the others [23]. The user can configure the resources of their virtual machines, including processing power, memory, and storage. Users can install their own operating system and applications on these machines, and then a snapshot is saved of these custom VMs. This enables faster recovery from failure, as these snapshots can quickly be copied over to other cloud nodes and started up. This process of failure recovery is typically fully handled by the cloud provider and is very fast [23]. Using IaaS avoids upfront infrastructure investment, and users do not have to worry about machines needing replacement. PaaS provides everything included in IaaS, but also provides a system platform, which allows users to develop their own system using the provided tools [23]. For example, this provided system platform could consist of a specific operating system (OS) with its associated development tools. SaaS provides software applications on the cloud, which removes the need for users to run the particular software on their local computers, and all of the infrastructure used to run the application is obscured to the user.

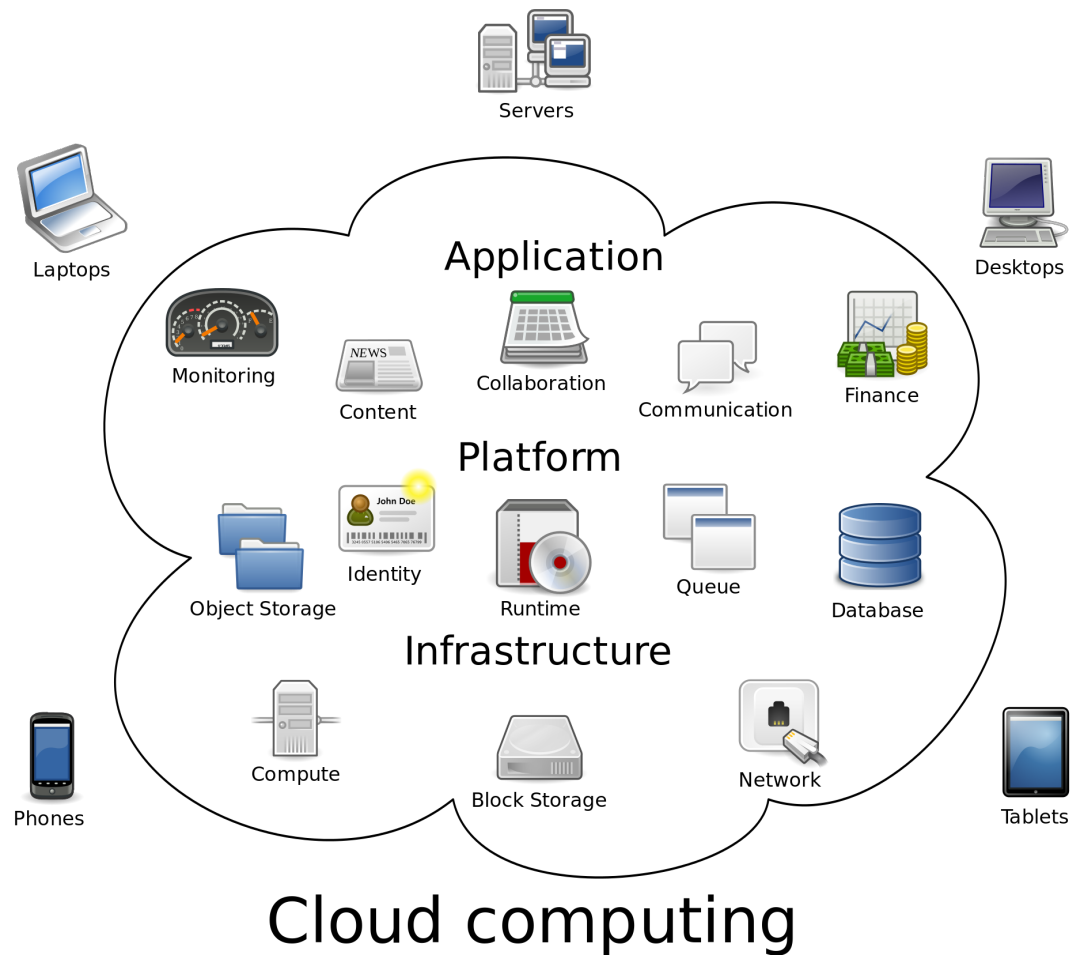


Figure 1.4: Overview of the different cloud service models and example use cases (wikipedia.org)

There are three different cloud infrastructure types, and the choice of infrastructure type depends greatly on data that will be stored and used on it. Public clouds are the most common, and provide their services including infrastructure generally via the internet, although sometimes a private connection can be purchased and used. The infrastructure is owned and maintained by the cloud provider, and is provided to the user as part of a shared pool of resources. Private clouds have infrastructure that is

completely dedicated to a single organization, and is sometimes hosted on-premise. Private clouds generally have better security than public clouds, but have higher costs, and may require users to purchase, set up, and manage the infrastructure. Hybrid clouds use a combination of public and private clouds, where for example a private cloud may be used to store sensitive data on-premise, such as health data, but uses a public cloud to analyze this data.

The cloud presents a number of advantages over traditional computing in addition to those mentioned above. It is well suited for information-rich computations, in part due to the high parallelization of processing available through using the cloud. Additionally, data and services can be stored on the cloud, reducing storage and processing requirements of the local computer. The cloud can provide variable amounts of computational power as chosen by the user, making it a good choice for applications with variable workloads or those that require a large amount of computational power for a short period of time, among others.

However, cloud computing presently has concerns around privacy and security of data, since this is under the control of the cloud provider [23]. The main concerns related to this include unauthorized data access, and data transfer, both in terms of the security of the data during the transfer, and that the data will not be transferred at any point outside of a particular jurisdiction [23]. Additionally, the country the data is stored in can have an impact on the privacy and security of data stored on the cloud, as laws in different countries differ in terms of data protection and local government access to data stored on the cloud in that country. From a survey of the literature, a number of solutions are being explored to address these privacy and

security concerns. Additionally, the use of an on-premise private cloud would take care of most of these issues. Another disadvantage of using the cloud is the lack of standardization between cloud providers that makes it currently difficult to migrate to a different cloud platform, which would present a significant challenge if the cloud provider used were to go out of business [23].

Due to the continuously decreasing cost of cloud computing, it is slowly being integrated into medical processes [23]. One major area of medical research that benefits from cloud computing is bioinformatics, as analyzing large amounts of biomolecular data can be performed faster on the cloud, with its ability to greatly scale up the amount of computation power in use [24]. This has successfully been applied to a number of different bioinformatics applications, including sequence processing, genome annotation, and single nucleotide polymorphism identification [24]. The cloud also is well-suited to storage and sharing of large biomolecular datasets [24]. Electronic health records are another application that can be improved through cloud computing, as it enables the records to be accessed from any location, and it is easy to increase the size of the database [25]. Cloud computing has also been used in telemedicine, such as the study conducted by Hsieh *et al.* for electrocardiography telemedicine [26]. In this study, the cloud was used to enable access of the collected data via the internet from a number of devices, including mobile devices, as well as for data processing and visualization. Medical image storage requirements are constantly growing, which cloud storage addresses well and provides the additional advantage of not needing to transfer large datasets to workstations in hospitals for viewing, as the visualization can be performed on the cloud [23]. Kagadis *et al.* proposed a

cloud-based picture archiving and communication system (PACS), which would provide location independence, enabling physicians to access medical images at other institutions, as well as device independence [23].

Using the cloud is slowly being introduced to advance research in radiation therapy. One application in radiotherapy that has shown promise in using cloud computing is in Monte Carlo dose calculations. These calculations are long and resource intensive, which prevents them from being integrated into routine clinical use, despite their increased dose calculation accuracy, and using the cloud to decrease the computation time of these calculations has been explored by a number of different authors [27] [28] [29] [30]. In each case, the authors found that using the cloud sped up the computation, although most expressed concerns about data privacy and security when using the cloud. In one of those studies, the authors not only performed Monte Carlo dose calculations, but additionally performed inverse planning on the cloud to generate a treatment plan [30]. In the three cases tested, the plans took less than three minutes to create, including data transfer time. However, the authors primarily focused on computation time results and did not compare the plans to those created manually by expert physicians, so the plan quality was not measured. Meng *et al.* tested a different application in radiation therapy for potential improvements using the cloud, as they examined cone-beam CT reconstruction [31]. They were able to parallelize the computation on the cloud, and found it reduced computation time from approximately an hour down to about five minutes.

### 1.3 Thesis contributions

The contributions in this thesis include the design and implementation of a system to analyze a database of radiation therapy studies to find the one with the most anatomical similarity to a new patient. The system uses the cloud to decrease the computation time required to find it, as the similarity analyses are performed in parallel on the cloud instances. The database of studies is also stored on the cloud. The system is controlled by a user-friendly module which obscures the cloud operations.

The system was tested against previous methods of performing this computation to see how it compared in terms of computation time. Its computation time was also measured for larger numbers of studies in the database to analyze, to test how the system's capabilities could scale up. As the system has a parameter to give the user some control over the number of cloud instances that will be used, each level available in that parameter was tested for computation time and contrasted against the others. The system was validated for the accuracy of the similarity values it returns by manually comparing the results from the cloud-based system to those from a previous method of performing the computation.

## Chapter 2

### System design and implementation

#### 2.1 System overview

As can be seen in the work by Berger, Kazhdan *et al.*, and Chanyavanich *et al.*, one method for generating a new radiation treatment plan is to optimize a plan selected from a database of previously computed plans, that best matches the new patient, rather than starting with a generic treatment plan [16] [1] [14]. The plan with the most similar anatomy to the new patient would be a suitable initial guess, however the similarity analysis computation to find this plan is long and resource intensive. To reduce the computation time required, hopefully to the point where it could be done clinically, we propose using the cloud to speed up the process of finding the plan with the most similar anatomy. This is made possible by the unique advantage of the cloud that enables the similarity analyses to be performed in parallel.

The similarity analysis computation is performed on virtual server instances, called cloud instances. The anatomical information about the new patient, which includes a CBCT and structure set, is stored in a study. The instances are directed by a master local computer, which uploads the new patient study to storage in the cloud, launches

the instances, and directs the instances on which previously computed plan located in the cloud storage to compare to. The instances may perform one or multiple similarity analysis computations, depending on the user's preference, due to the tradeoffs between cost and computation time. Once the cloud instances are finished, they self-terminate, and the local computer downloads the results from the cloud storage and selects the plan with the best similarity value.

## 2.2 Software infrastructure

### 2.2.1 3D Slicer

3D Slicer<sup>1</sup> is popular and well-maintained medical imaging platform that is professionally engineered and supports a wide variety of features [32]. It provides advanced tools for image analysis and visualization. 3D Slicer is versatile and extensible, and has over 200,000 downloads worldwide for the most recent version. Its BRAINS general image registration was used in the similarity analysis computation, which will be discussed in section 2.5.

### 2.2.2 SlicerRT

SlicerRT<sup>2</sup> is a toolkit for 3D Slicer and contains extensive tools for RT research which leverage the features provided by 3D Slicer [33]. It is designed to facilitate rapid implementation of advanced experimental radiotherapy workflows. It supports RT research in a number of ways, including providing an environment in which new techniques can be developed and comparative validation can be performed. The core features provided by SlicerRT were chosen through discussion with RT researchers,

---

<sup>1</sup>[www.slicer.org](http://www.slicer.org)

<sup>2</sup>[www.slicerrt.org](http://www.slicerrt.org)



and includes support for the clinical standard DICOM-RT file format, as well as contour morphology and DVH computation among others. As will be described in detail in section 2.5, in this work we used SlicerRT to perform the similarity analysis computation using its tools for contour comparison, which use SlicerRT’s infrastructure for automatic conversions between the data representations of the contours. Its functionality for reading DICOM-RT files was used as the radiation therapy studies were stored in this format, which also functions as a non-relational database for the information in those studies.

### 2.2.3 Amazon Web Services and Boto

The cloud platform used was Amazon Web Services (AWS), which is comprised of a set of different services [34]. The Elastic Compute Cloud (EC2) service provided the cloud instances, on which the similarity analysis computation was performed. In particular, the instance type used was “c3.xlarge”, which was a compute-optimized configuration, which have higher performing processors and lower price per compute performance. The Simple Storage Service (S3) was used to store the previously computed treatment plans and associated anatomical data, as well as for intermediate storage of the new patient’s anatomical data and the comparison results prior to being transferred to the instances and the local computer, respectively. The Simple Queue Service (SQS) was used for sending messages between the local master computer and the instances. The Boto library is a python-based interface to AWS, which was used to interact with the AWS components [35].

## 2.3 System architecture

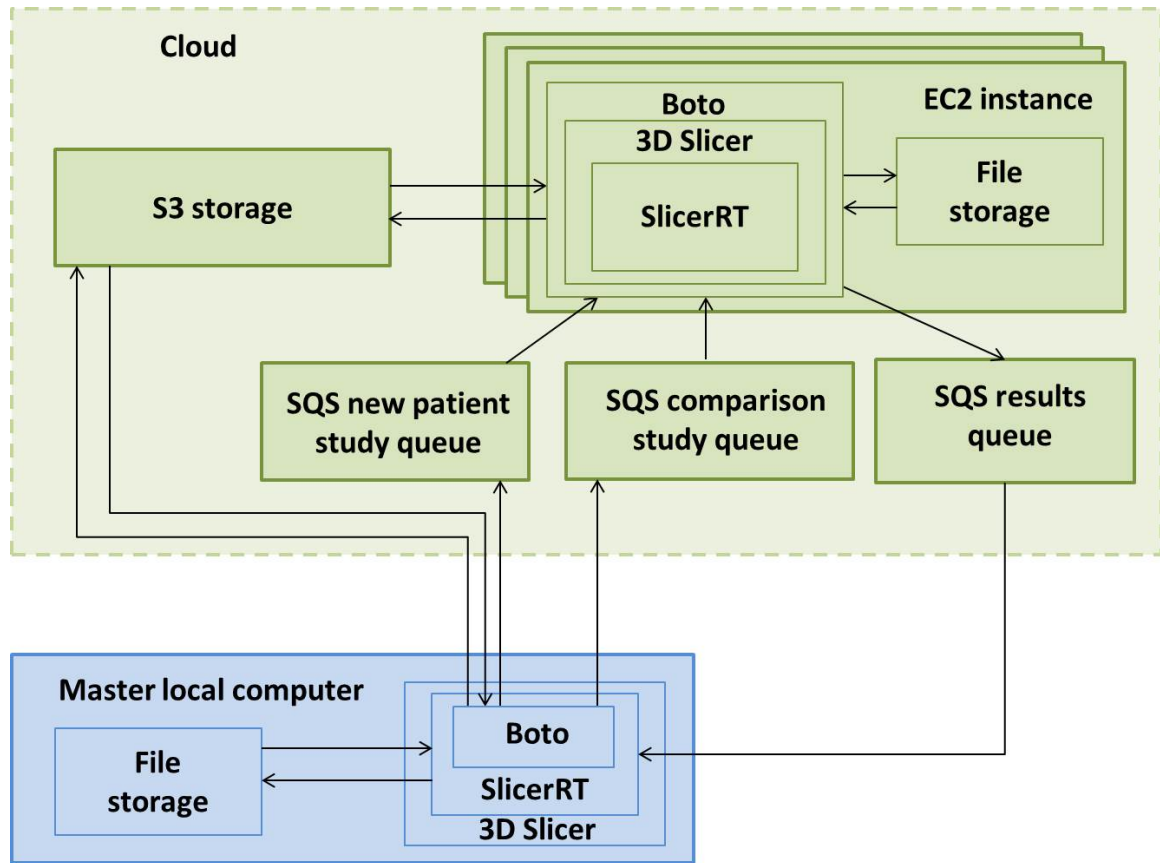


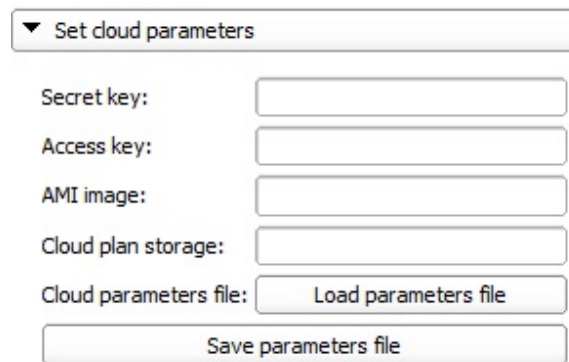
Figure 2.1: Overview of the system architecture

The overall architecture of the system can be seen in Figure 2.1, which shows how the master local computer interacts with the major components in the cloud, those being the S3 storage, EC2 instances, and the three SQS queues. This will be explained in more detail in section 2.4.

## 2.4 Workflow

### 2.4.1 Set cloud parameters

The operations performed on the local master computer are done through a 3D Slicer module in SlicerRT which was created for this project. The module has three



▼ Set cloud parameters

Secret key:

Access key:

AMI image:

Cloud plan storage:

Cloud parameters file:  Load parameters file

Save parameters file

Figure 2.2: “Set cloud parameters” section of the module prior to the parameters being set

main sections, which comprise the critical operations. The first section which can be seen in Figure 2.2 allows the user to input the values for the parameters required to connect to the cloud, which are the access key, secret key, the name of the Amazon Machine Image (AMI) to be run on the instances, and the name of the cloud storage bucket where the radiation studies are stored. The names of the cloud queues and cloud storage subfolders, the geographic region to launch the instances in, the instance configuration type, the security group, and the name of the key pair are hard-coded in the system. The user can input the parameters either by hand or from a file, and the data can be saved to a file.

### 2.4.2 Upload study to cloud

The second section allows the user to upload a study containing a previously computed treatment plan to the cloud storage. The user first loads the study into 3D Slicer, as seen in Figure 2.3, and selects the names of the CT and structure set. The module then locates the DICOM-RT files in the file storage on the local computer, zips them up to compress them, uploads this zip file to the cloud storage, and deletes the zip file from the local computer. A descriptor file is generated containing relevant

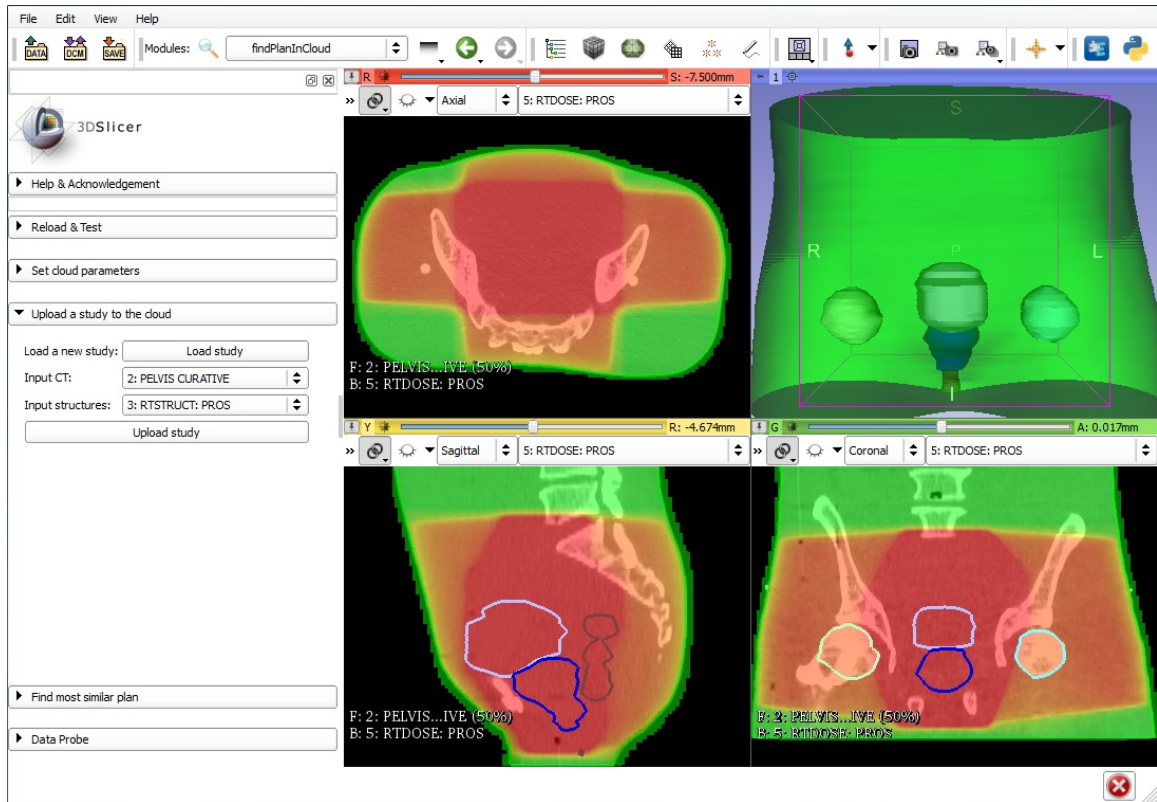


Figure 2.3: “Upload study to cloud” section of the module with prostate cancer CT and contours loaded and parameters set

information about the study, including the names of the CT and structure set, is uploaded to the cloud storage with the zip file, and is also deleted from the computer. The studies in the cloud storage are stored in separate folders with a specific naming convention, and the module determines the name of the new folder to contain the new study, which is automatically created when the study is uploaded to the cloud.

### 2.4.3 Find most similar plan

The third section of the module is the section for actually performing the similarity analysis computation on the cloud to find the plan most anatomically similar to the new patient by analyzing the new patient’s anatomy against all of the studies in the

cloud storage. Similar to the previous section, in this one the user is first required to load a radiation treatment study, in this case the study to be compared to, and select the names of the CT and structure set. The user is also asked to select a folder for the results to be downloaded to, the type of similarity analysis to be used (in this phase of development, the only option is the Dice similarity coefficient [36], or Dice for short), and the level of number of instances to compare to, which has three options: “high, time-efficient”, “medium, time and cost balanced”, and “low, cost-efficient”. Selecting the level “high, time-efficient” results in as many instances being launched as there are studies stored in the cloud to be compared to, so each instance performs exactly one comparison. The formula for the number of instances to use for this level was chosen to provide maximum parallelization and the fastest possible computation time. The level “low, cost-efficient” calculates the number of instances as the square root of the number of comparison studies, rounding to the nearest positive integer. This formula was chosen so that the number of comparisons per instance scales up and down with the number of studies to analyze, while maintaining a low number of instances used. The level “medium, time and cost balanced” calculates the number of instances as twice the square root of the number of studies to be compared to, and this formula was chosen to provide a midpoint between the high and low levels, while still scaling the number of comparisons per instance by the number of studies to analyze. In calculating this, the system checks that the number of instances is not higher than the number of comparison studies, and that the number of instances is greater than zero if there is at least one comparison study in the database. As in the previous section, the study is automatically zipped up and a descriptor file containing relevant information is automatically generated, both are uploaded to the

cloud and are then deleted from the local computer. The local computer launches the number of instances determined by the level selected along with the specified AMI, instance type, and geographic region, along with the secret key and access key in order to obtain permission to launch the instances. After starting the instances up, the local computer obtains a reference to these instances, which can be used to shut the instances down in case of failure. When testing the system, it was observed that if an EC2 instance is not fully started up by the time the computer attempts to obtain a reference to it, it causes an error, which occurred in 1.21% of tests. Adding in a small wait in between starting up the instances and attempting to get a reference to them may solve this problem. The local computer adds messages into the comparison study queue on the cloud to specify which studies stored in the cloud to compare to, with one message per study placed in the queue. Once the new patient study is uploaded to the cloud, messages are added into the new patient study queue to indicate this, with one message per instance launched placed into this queue.

The instances first poll the new patient study queue to determine if the study has been uploaded, and once they receive a message indicating this, they download the study from the cloud and extract the data from the zip file. The instances then poll the comparison study queue to determine which study they should compare to, with each instance pulling a single message from that queue. If the comparison study queue is empty of messages, the instance will check for messages five times over the next few seconds, then stop the process, delete the new patient study, and automatically self-terminate. If the instance gets a message from this queue, it checks which study to compare to, downloads it from the cloud, and extracts the data. Then 3D Slicer is launched with a script containing the similarity analysis process, which is described

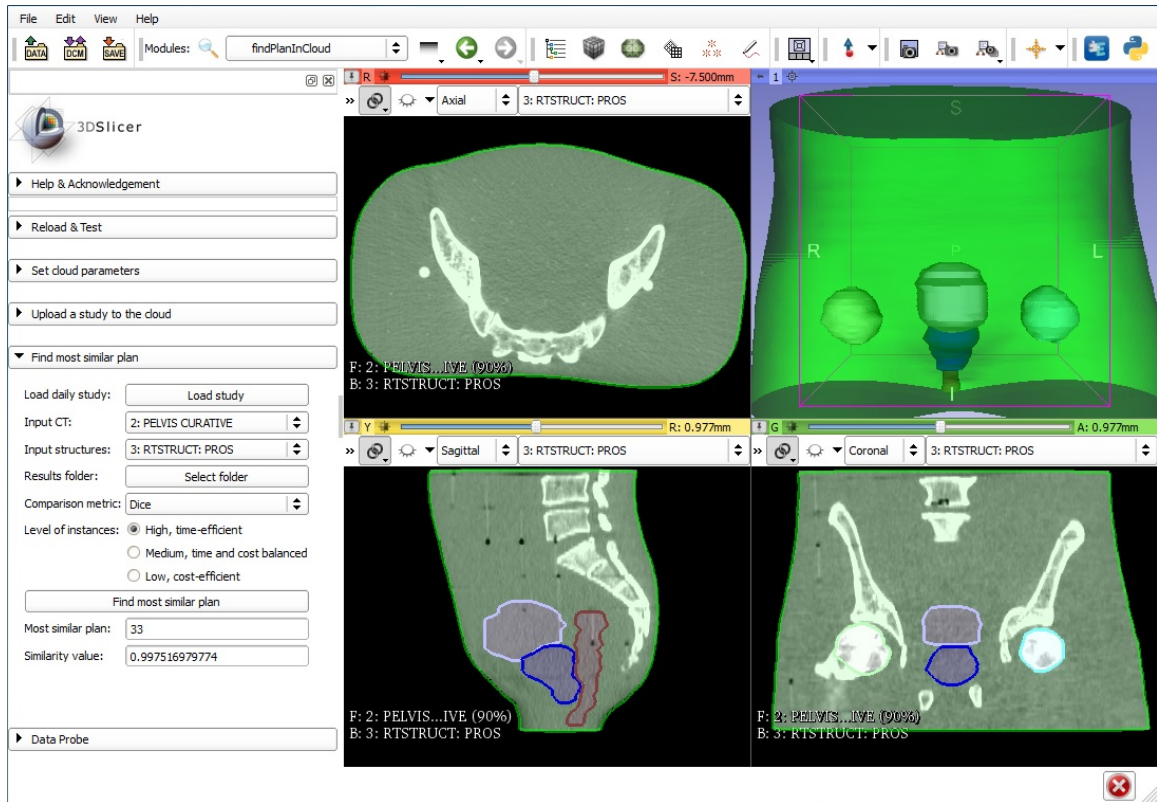


Figure 2.4: “Find most similar plan” section of the module at the end of the computation

in detail in section 2.5. Once the similarity analysis is complete, this is signaled to the instance through the creation of a “semaphore” file, which is deleted by the instance. The similarity analysis script will have saved the results to a text file in a known location, which the instance will upload to the cloud, in a folder named for the comparison study used. It will also put a message into the results queue on the cloud to indicate that the selected comparison study has been analyzed and its result uploaded. The data associated with the comparison study, as well as the results and semaphore files, are deleted. The instance will then poll the comparison study queue again, following this process until the queue is empty, at which time, as mentioned above, it will stop the process and self-terminate.

After populating the new patient study and comparison study message queues, the local computer starts polling the results queue on the cloud for messages. Each time the local computer receives a message through this queue, it checks which study has been compared to, and notes this. It then quickly checks whether all of the studies have been analyzed, meaning that the similarity analysis process is complete. Once all studies have been analyzed, the local computer downloads all of the results from the cloud, iterates through the results, and notes the study with the best similarity value in a new results file. This information (both the name of the study and its similarity value) are displayed in the graphical user interface (GUI), which can be seen in Figure 2.4. The best study is then downloaded from the cloud into a subfolder within the user-specified results folder.

While testing the system, some failures were observed where the computation process would fail on an instance. This occurred in 3.25% of tests, and 0.14% of instances. In some cases, the process would crash due to the instance being unable to successfully download one of the two studies from the cloud storage. To handle this error, a check will be put in after downloading the studies to check that they have been successfully downloaded, and if not, to attempt it again. The other cause of the process failing on an instance occurs when 3D Slicer crashes on the instance during the contour comparison step described in section 2.5. At present, the crash is nondeterministic and is likely connected to the way the automation of the similarity analysis algorithm is implemented.

## 2.5 Similarity analysis computation



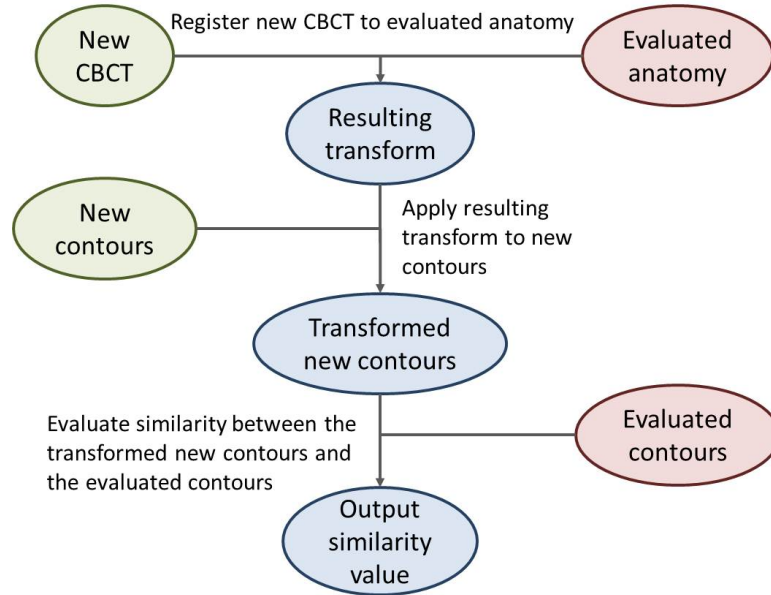


Figure 2.5: Workflow of the similarity analysis computation

Prior to this project, the main way to perform the comparison process in SlicerRT was manually through the GUI, so this process was first automated for this work. To evaluate the similarity of two radiation treatment studies, where one contains anatomical information about a new patient, represented as a cone-beam CT (CBCT) and structure set, and the other contains a previously computed treatment plan and its associated anatomical information, the CBCT of the new patient is first registered to the planning sim-CT in the selected study under evaluation. The resulting transformation is applied to the new patient's contours. Then, pairs of matching contours, one from the new patient study, and one from the study under evaluation, are compared and their similarity is measured using the Dice similarity metric. The Dice similarity coefficient measures the similarity of two contours by measuring the amount of overlap between them, with values ranging from 0 (no overlap) to 1 (complete overlap) [36]. The Dice similarity coefficient of two segmented structures, A and B, can

be calculated as seen in Equation 2.1 [36].

$$DiceSimilarityCoefficient(A, B) = \frac{2(A \cap B)}{A + B} \quad (2.1)$$

The overlap between A and B can be computed in a number of different ways, including converting the stack of contours representing them to volumes, and checking for overlap in each voxel of the structures. As contouring is a very time-consuming step, the comparison of contoured structures was only used as a proof-of-concept for using the cloud with SlicerRT to evaluate the similarity of radiation treatment studies. The overall similarity of the two studies is calculated as the average of the Dice values of the contour pairs. The workflow of this process can be seen in Figure 2.5. This approach was used over intensity-based registration as it is much faster, and the computation time of this process needed to be low in order to be clinically feasible.

## 2.6 User interface

As can be seen in Figure 2.6, the user interface has three main components, those being “Set cloud parameters”, “Upload study to cloud”, and “Find most similar plan”. The functionality of these components is described in section 2.4. The parts of the user interface are ordered in the sequence that would most likely be followed by the user. The buttons “Find most similar plan” and “Upload study” are greyed out and cannot be clicked until all of the required information is input by the user. The user interface was designed to be fairly straightforward and easy to use.

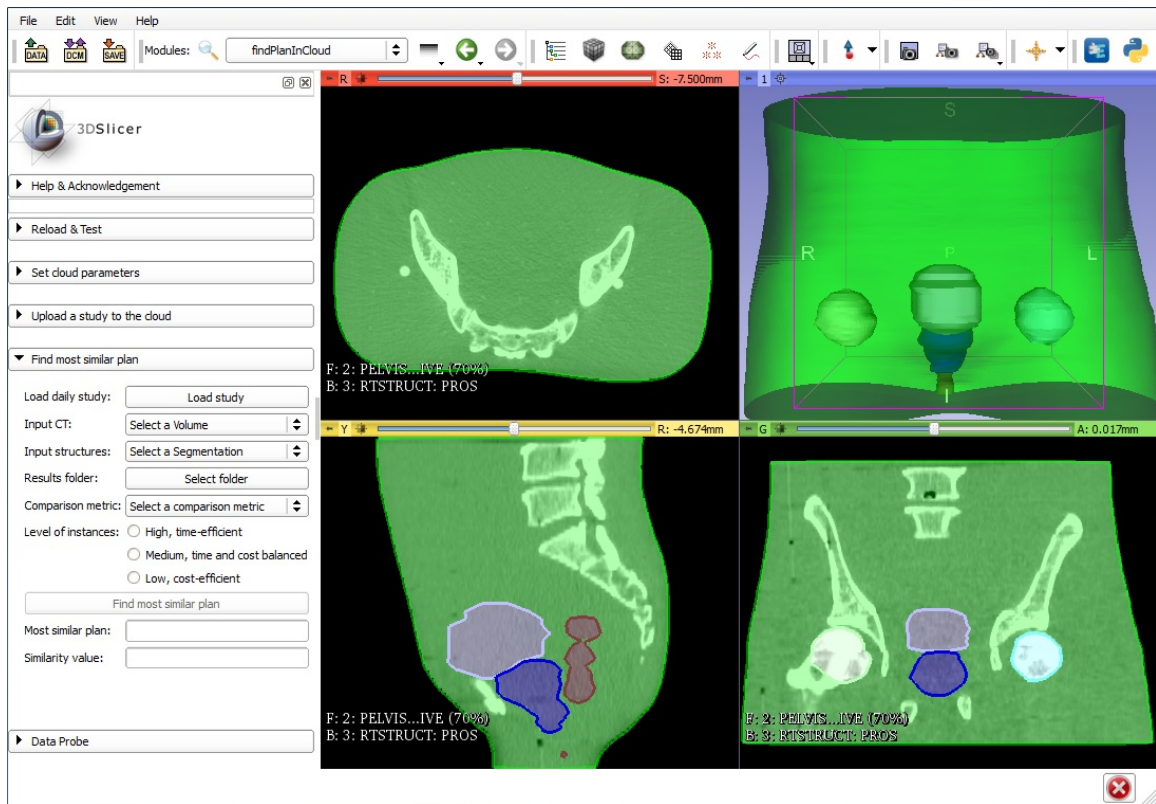


Figure 2.6: “Find most similar plan” section prior to parameters being set

## Chapter 3

### Results, validation and discussion

#### 3.1 Testing methodology

##### 3.1.1 Test data

The data used in this work for both the new patient study and the comparison studies was simulation data that was created for this purpose, as it was not possible to get enough anonymized patient data to adequately test the system. The data was generated by warping a radiation treatment plan for a RANDO phantom [37] prostate dataset with deformation fields. These deformation fields were created with displacement vectors in all directions that were randomly assigned values in the range -10 to 10 mm. The displacement vectors were assigned to the dataset every 125 mm in the left-right and anterior-posterior directions, and every 57.5 mm in the inferior-superior direction. The original radiation treatment plan was provided by the Cancer Centre of Southeastern Ontario at the Kingston General Hospital.

### 3.1.2 Testing hardware

The local computer used for this project was a desktop computer running 64-bit Windows 7, with Intel Core 2 Quad central processing units (CPU) and 8GB of random access memory (RAM). It was running on a local area network (LAN) with a measured download speed of 94.89 Megabits per second (Mbps) and upload speed of 20.86 Mbps. The EC2 instances used the ‘c3.xlarge’ configuration, which used 4 virtual CPUs, 8GB of RAM, and had Intel Xeon E5-2680 v2 processors. The instances were running Windows Server R2 Datacenter, and had clock speeds of 2.80GHz. In this work, different instance types were not experimented with due to the workload involved in migrating from one instance type to another. The instance type used was chosen to best match the local computer for the purposes of demonstrating the advantage in using the cloud over just the local computer.

### 3.1.3 Computation time

A number of different tests were performed to evaluate how the system performed in terms of time. First, a test was conducted to compare performing the similarity analyses using the cloud system to performing them consecutively on the local computer, both manually and using an automated script. The two different methods of performing the computation on the local computer were both tested, as prior to this project the main way of performing this computation was manually through the GUI. To test how long it took to perform the comparisons on the local computer in the two different methods, each method was used to perform three comparisons. Then, the times were averaged to determine the average time required to perform one comparison with each method.

To measure the time performance of the cloud system, the number of studies to compare to in the database was gradually increased from 5 up to 100, increasing the number by 5 each time. Each number of studies was tested three times, as the length of time required to perform each test meant that performing more than three tests was not feasible, and then averaged to get the final time result. Each instance performed exactly one similarity analysis, meaning that the instance level “high, time efficient” was used. The time was measured from when the “Find most similar plan” button was clicked to the point where the results had been received, and the GUI updated with them. Then, the same procedure was repeated for the levels medium and low, with the number of instances used in each test also being recorded. The time required to start up 3D Slicer, load the data, and set the relevant parameters was also measured three times.

Next, the overhead of using the cloud system was measured for three different instance loads, those being 5 instances, 50 instances, and 100 instances. It was measured from the point at which the “Find most similar plan” button was clicked to the point where the last message from the database study queue is retrieved, so the time period measured includes starting up the instances, uploading the new patient study to the cloud, and populating the two message queues with messages for the instances. Each test was run three times.

#### 3.1.4 Accuracy

To test whether the system accurately identified the study with the most anatomical similarity, 10 studies from the cloud database were randomly chosen and used in separate tests as the new patient study. The study returned by the cloud system as

the best match was then manually analyzed with the new patient study on the local computer, to check whether the returned Dice value was correct. While in theory the returned study should be the exact same as the one used as the new patient study, the BRAINS general registration algorithm from the base 3D Slicer platform has some difficulty in registering the same CT image to itself (as occurs when the two studies are identical). This results in a lower Dice value, and in some cases means that the study chosen by the system as the best match is not the identical study. In cases where this happened, the identical study was also manually analyzed against the new patient study, to demonstrate that this lower Dice value resulted from the registration and not the cloud system. Additionally, the case of analyzing a radiation therapy study against itself would not actually happen in clinical practice.

## 3.2 Results and discussion

### 3.2.1 Results for computation time

Using the cloud produced better computation times than using either the automated script on the local computer or the manual comparison through the GUI. This was true of each of the three instance levels used on the cloud. The three comparisons using the automated script and manual comparison were each averaged, then used to project the computation time required for larger numbers of studies to analyze, as the comparisons on the local computer can only be performed consecutively. These projected computation times, along with the average computation times for the three different cloud instances can be seen in Figure 3.1 for 5 to 25 studies to analyze. The computation times for the cloud instance levels have the average time required to set parameters and the start the computation on the local computer added to them. This

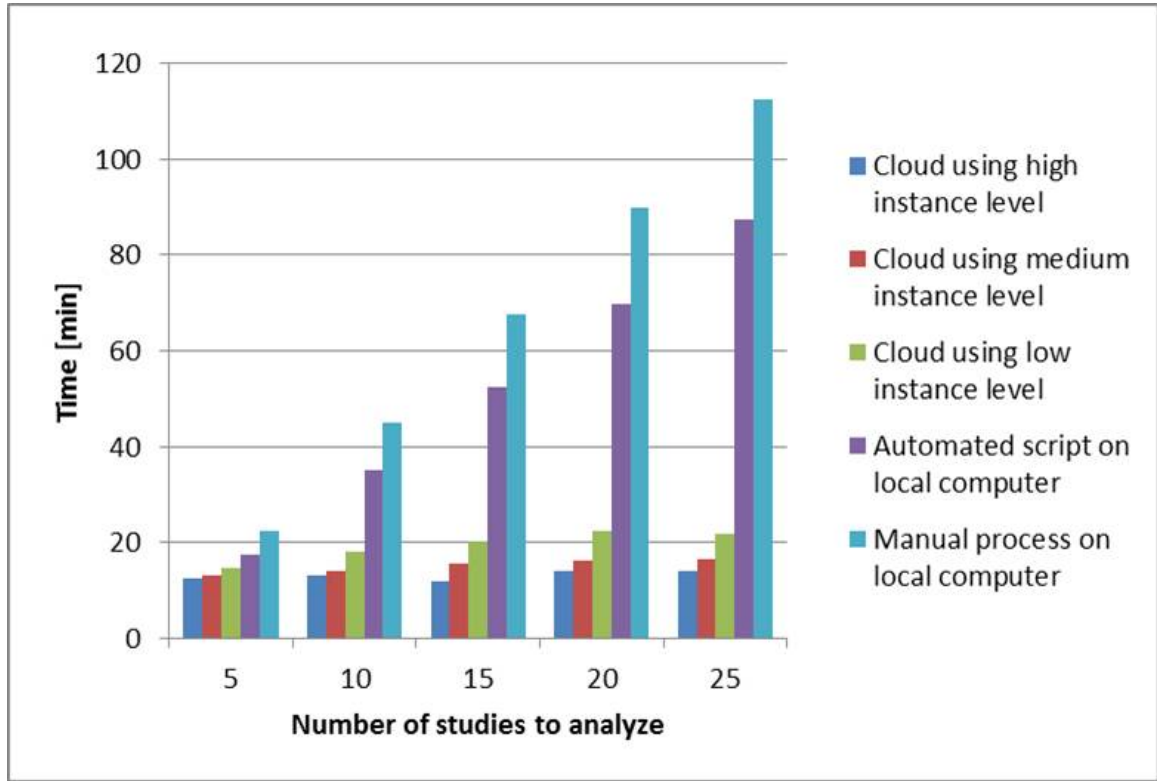


Figure 3.1: Average computation times for different analysis methods

demonstrates that as the number of studies grows, the cloud performs increasingly better than using either method on the local computer.

Additionally, this test provided the ability to observe how the system performed as the number of studies to analyze was scaled up from 5 studies to 100. For the high instance level, this also meant that the number of instances used was also scaled up from 5 to 100. The results for this can be seen in Figure 3.2, which shows a slight increase in computation time for the largest numbers of studies, but is generally fairly flat. This demonstrates that the system does not experience too much delay when having to start up larger numbers of instances or put larger numbers of messages in the queues. Some variability in the computation times can be observed, and is primarily



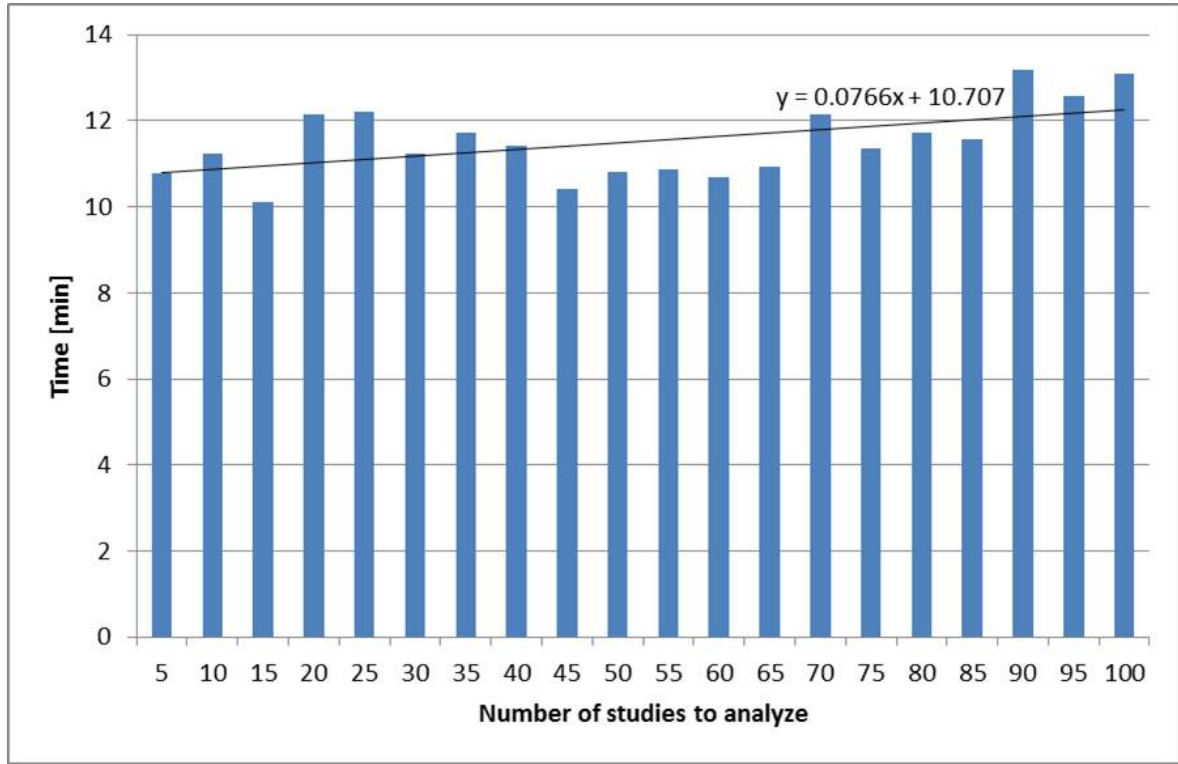


Figure 3.2: Average computation time using high cloud instance level

due to the variable loads on the AWS services caused by their other customers, which fluctuates over the course of the day. This variability in computation times can be seen in Figure 3.3, which shows the times for performing the same test three times for each number of studies, using the instance level high.

Figure 3.4 shows the detailed comparison between each of the different instance levels and their computation times for the different numbers of studies to be analyzed, from 5 to 100. This figure shows that the instance level high in each case has lower computation times than the instance level medium, which in turn has lower times than the instance level low. This shows that full parallelization of the system produces faster computation times, and that the formulas used to calculate the number of

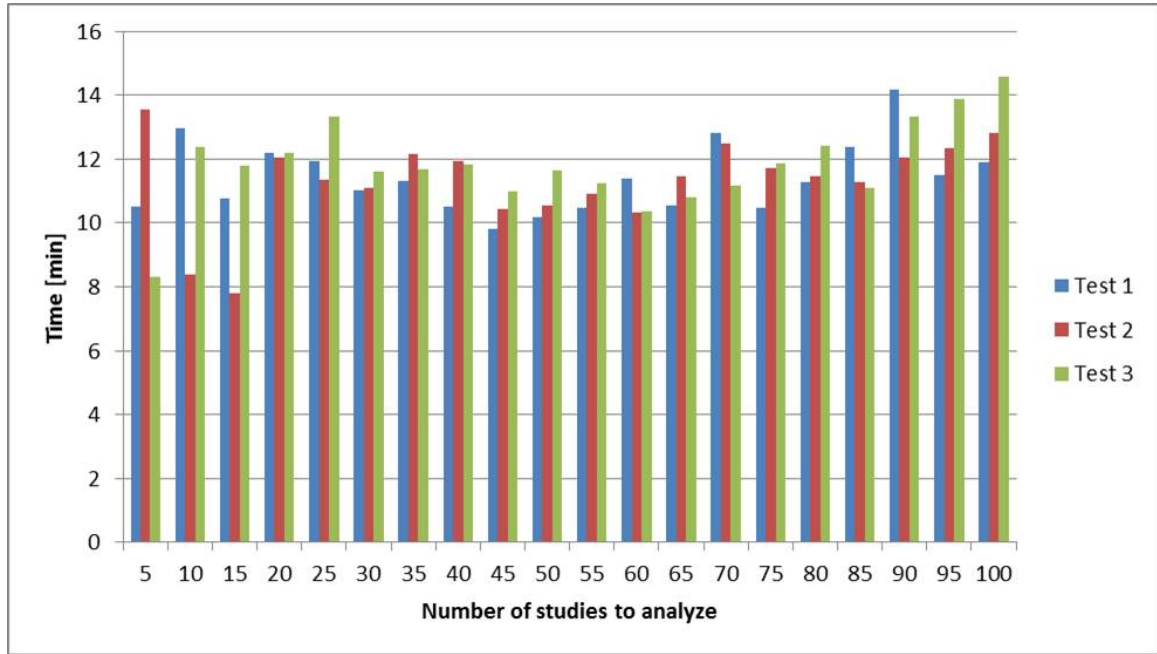


Figure 3.3: Variability of similarity analysis computation times when using high cloud instance level

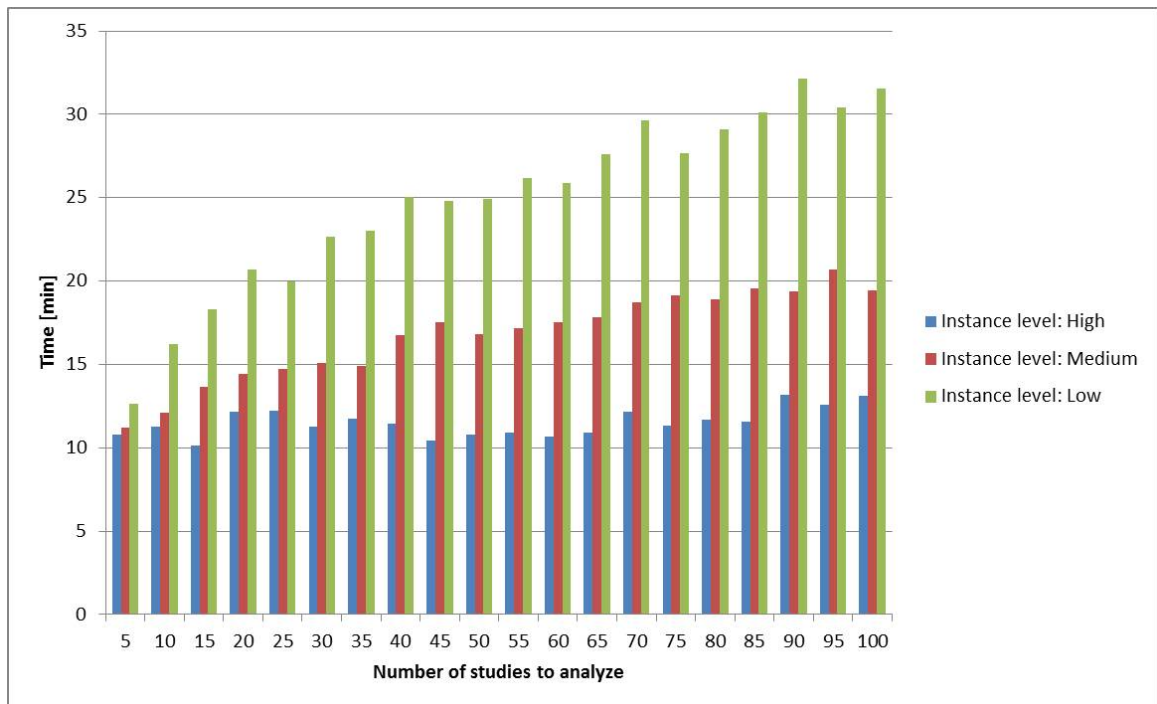


Figure 3.4: Average computation times for different analysis methods

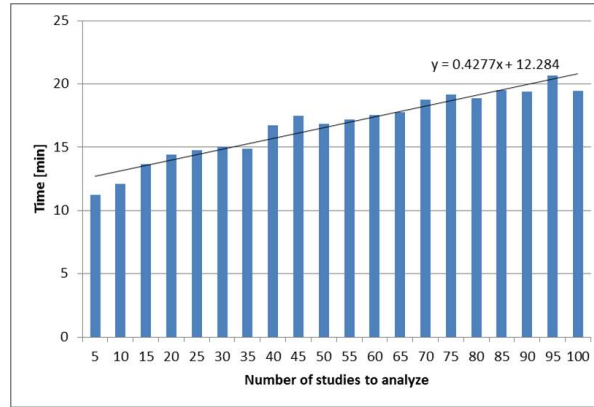


Figure 3.5: Average computation time using medium instance level

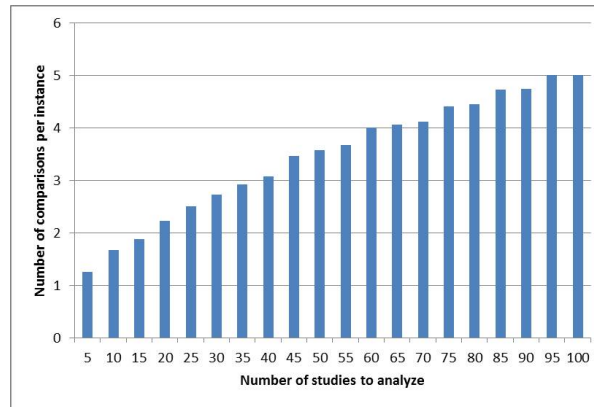


Figure 3.6: Average number of comparisons per instance for medium instance level

instances for each of the three instance levels produce computation times that match their level.

As an additional note, analyzing the average computation times for the medium and low instance levels indicate that the computation time typically correlates with the average number of study comparisons being performed on each instance. This can be seen in Figure 3.5 and Figure 3.6 for the medium instance level, and Figure 3.7 and Figure 3.8 for the low instance level.

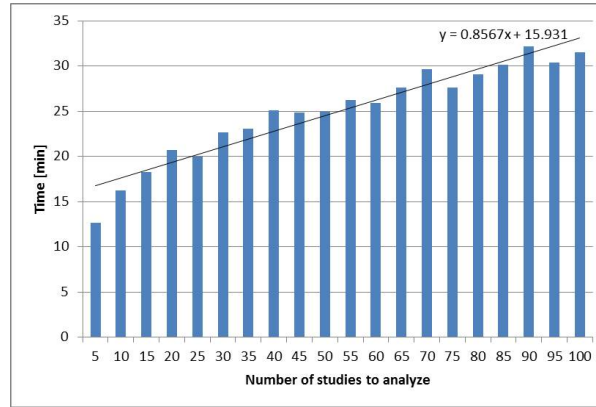


Figure 3.7: Average computation time using low instance level

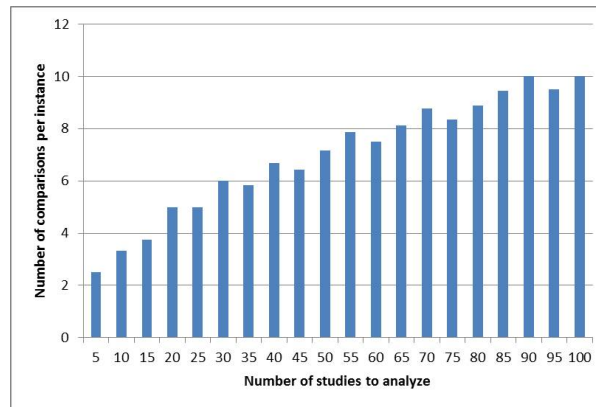


Figure 3.8: Average number of comparisons per instance for low instance level

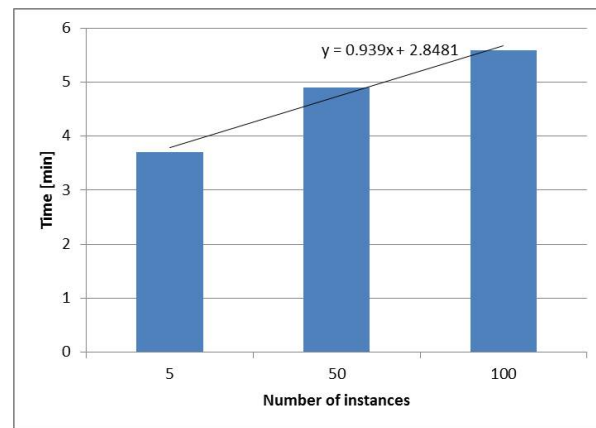


Figure 3.9: Cloud overhead time for different numbers of instances

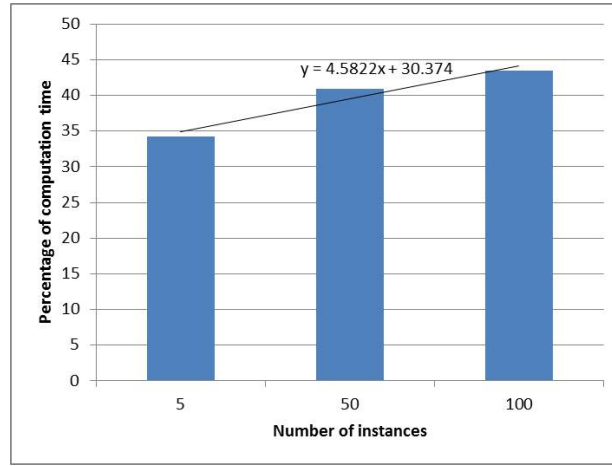


Figure 3.10: Overhead measured in percentage of computation time for different number of instances

Finally, the results for the test measuring the overhead in starting up the cloud-based system is shown in Figure 3.9, with Figure 3.10 showing the overhead time as a percentage of the total computation time. These figures indicate a small increase in overhead as the number of instances increases. This is due to the longer time required to start up more instances, which took from on average 23 seconds to start up 5 instances, to approximately 2.5 minutes to start up 100. This can be seen in Figure 3.11, which shows that the instance startup is a large contributor to this increased overhead time, as it shows the largest time difference between the different numbers of instances. Additionally, the greater number of messages to be added to the queues also increased the overhead for the larger numbers of instances, as the time required increased from 7 seconds for 5 instances to approximately 2 minutes for 100 instances. This time increase has less impact on the total overhead than the time to start up the instances, as for the smaller numbers of instances, the queues are filled long before the instances are ready to start picking up the messages, whereas for the larger numbers of instances, they are ready to pick up the messages as they are

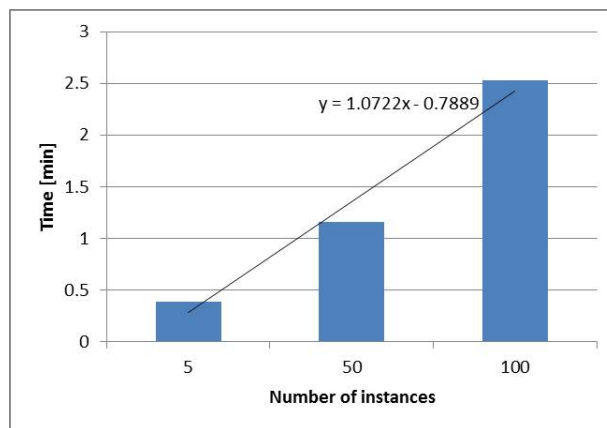


Figure 3.11: Average time required to start up cloud instances

added to the queues. The remaining major component of the system overhead is the time required to zip up and upload the new patient study to the cloud, which takes on average 41 seconds, and consistently takes that time regardless of the number of instances used.

### 3.2.2 Results for accuracy

As can be seen in Table 3.1, the Dice value returned by the cloud-based system for the chosen study matched the Dice value that was produced by manually comparing the new patient and chosen studies on the local computer. In half of the cases, the registration algorithm resulted in the identical study in the cloud database not being chosen, due to the registration being unable to handle the case of registering an image to itself. To verify this result, in each case the new patient study was compared to itself manually using the same analysis procedure as on the cloud, and the same Dice value was calculated for each case. Additionally, the result of the identical study not always being returned is not a concern, as the situation of comparing a study to itself would not occur in actual practice, and is an artifact of the data available at this time.

Manual Dice value matches cloud Dice value for study chosen by cloud [/10]	10
Study chosen by cloud matches new patient study [/10]	5
Manual Dice value matches cloud Dice value for identical study [/10]	10

Table 3.1: Results of accuracy testing

These results indicate that the cloud-based system returns the same Dice values for the similarity analyses as could be achieved using the original method available prior to this work.

### 3.2.3 Earlier iterations

An earlier version of the system [38] used a single SQS queue for all of the messages, with the messages being tagged with the instance ID of their intended target, in order for all instances to receive the appropriate combination of messages (with one message to say that the new patient study had been uploaded, and another to direct the instance on which study in the database to compare to). Results messages were also placed in the same queue, using an instance ID of 0 (which would never be assigned to an EC2 instance) to denote the local computer. However, this strategy was found to have bottlenecks whenever the instances needed to pick up their messages from the queue, as they could not only pull out their messages from the queue, so they had to grab a message, check its instance ID, and then either read it or release it back to the queue, as was appropriate. This resulted in a large time delay for the instances to sort out which messages were for them, and would only result in larger delays as the number of instances was scaled up.

Another feature that was added later was the ability to perform multiple comparisons on a single instance. Before, the only option was to perform a single similarity

analysis on an instance, resulting in maximum parallelization and the lowest time computation time. However, this strategy is not the most cost-efficient, so this feature was added. The user was given three options to direct the number of instances to be used, in effect controlling the number of comparisons performed on each instance. The options of high, medium, and low choose the number of instances based on formulas listed in section 2.4.3.



## Chapter 4

### Conclusion

#### 4.1 Summary

In this thesis, an examination of the literature surrounding adaptive radiation therapy treatment planning was conducted, and it was found that this is an outstanding problem that could benefit from the application of cloud computing. The approach explored in this thesis was to analyze a database containing treatment plans created for other patients, with each plan and its associated CBCT and structure set stored in a study, to find the one that was the most similar anatomically to a new patient. This selected plan would be used as the starting point for the new patient's plan, and would be optimized further.

To perform the similarity analyses of the studies in the database, the cloud was used to enable the analyses to be performed in parallel, which was found to greatly reduce the computation time. The database of studies was also stored on the cloud, and the procedure was directed by a master local computer through the use of message queues and the Boto interface. An interface was built in 3D Slicer and SlicerRT, which streamlined the process of starting the similarity analysis for the user, and also

provided them with an easy way to upload new studies to the cloud database.

Several tests were conducted to evaluate the system performance in terms of time and accuracy. The system was found to produce Dice values which matched those calculated by performing the analysis manually through the GUI, and only had difficulty handling the case of evaluating a study against itself, which would not occur in actual use. The system showed large reduction in computation time relative to performing the same analyses consecutively on the local computer. The tests also indicated that the system scales up well in terms of the number of instances used, with low increases in computation time and overhead for the larger numbers of instances.

## 4.2 Future Work

There are a number of ways in which this system could be improved and extended further. One present system shortcoming is that it does not handle cases where an instance crashes. This could be fixed by, in cases where all other instances have finished and one is still remaining on the cloud much later, terminating the remaining instance, starting up a new one, and adding new messages to the two queues so that it could perform the remaining similarity analysis computation. Additionally, 3D Slicer has a capability to run without displaying the user interface, and it would be worth testing whether running the similarity analysis computation on the cloud with this setting decreases computation time.

This system could be extended further by increasing the portfolio of similarity analysis methods and metrics the user can choose from. For example this could include the Hausdorff metric, which is available for evaluating contour pairs in SlicerRT. The

metrics could also include some of those discussed in the literature, such as the overlap volume histogram discussed by Kazhdan *et al.* [1]. The current similarity analysis method which uses the Dice metric could also be improved by experimenting with weighting the Dice values of the individual contour pairs in the final similarity value, to give higher weight to more important anatomical structures and targets.

Another useful addition to this system would be to extend it to other cancer types, as presently the only one used is prostate cancer. To add other kinds of cancer, pre-filtering would need to be added into the system to select out cases that do not match the cancer site before the studies are analyzed, as for example a head and neck case will not be a good match for a patient with prostate cancer. To enable pre-filtering, the studies could be stored in an Structured Query Language (SQL) database, which would facilitate fast retrieval of the relevant cases. Other methods of pre-filtering could also be considered, such as removing cases with the wrong prescribed dose. The similarity analysis computation could be modified so as to eliminate dissimilar cases earlier, for example if the contours delineating the bodies did not match enough to meet a predetermined threshold, the study could be eliminated rather than finishing the computation. The lack of similarity between the body contours indicates that the remaining structures will not likely match enough for the plan being analyzed to be a good fit for the new patient.

As the cloud was used to optimize for computation time, which is most critical for online ART replanning, this scenario could be also addressed by not relying on pre-contoured datasets, but rather by doing contour propagation as seen from Schwartz *et al.* [7].

To use the cloud-based system clinically, or even to test it on patient data, the privacy and security concerns around storing patient data on the cloud will need to be addressed. This could be done using an on-premise private cloud, which is a growing focus in the cloud computing industry. It is also anticipated that with privacy and security in the cloud being an area in which much research is targeted at improving, advancements in cloud technology will emerge to solve this problem.

Finally, the treatment plan selected as the most similar anatomically to the new patient would need to be dosimetrically optimized further to create the new patient's plan. This optimization process would be a logical next step to extend this work with the hopes of being able to use it clinically.

## Bibliography

- [1] Michael Kazhdan, Patricio Simari, Todd McNutt, Binbin Wu, Robert Jacques, Ming Chuang, and Russell Taylor. A shape relationship descriptor for radiation therapy planning. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 100–108. Springer, 2009.
- [2] Pierre Castadot, John A Lee, Xavier Geets, and Vincent Grégoire. Adaptive radiotherapy of head and neck cancer. In *Seminars in radiation oncology*, volume 20, pages 84–93. Elsevier, 2010.
- [3] Bruno Sorcini and Aris Tilikidis. Clinical application of image-guided radiotherapy, IGRT (on the Varian OBI platform). *Cancer/Radiothérapie*, 10(5):252–257, 2006.
- [4] Tiezhi Zhang, Yuwei Chi, Elisa Meldolesi, and Di Yan. Automatic delineation of on-line head-and-neck computed tomography images: toward on-line adaptive radiotherapy. *International Journal of Radiation Oncology\* Biology\* Physics*, 68(2):522–530, 2007.
- [5] Di Yan, Frank Vicini, John Wong, and Alvaro Martinez. Adaptive radiation therapy. *Physics in medicine and biology*, 42(1):123, 1997.

- 
- [6] Elizabeth Brown, Rebecca Owen, Fiona Harden, Kerrie Mengersen, Kimberley Oestreich, Whitney Houghton, Michael Poulsen, Selina Harris, Charles Lin, and Sandro Porceddu. Predicting the need for adaptive radiotherapy in head and neck cancer. *Radiotherapy and Oncology*, 116(1):57–63, 2015.
- [7] David L Schwartz, Adam S Garden, Jimmy Thomas, Yipei Chen, Yongbin Zhang, Jan Lewin, Mark S Chambers, and Lei Dong. Adaptive radiotherapy for head-and-neck cancer: initial clinical outcomes from a prospective trial. *International Journal of Radiation Oncology\* Biology\* Physics*, 83(3):986–993, 2012.
- [8] Marcel van Herk. Different styles of image-guided radiotherapy. In *Seminars in radiation oncology*, volume 17, pages 258–267. Elsevier, 2007.
- [9] Derek Schulze, Jian Liang, Di Yan, and Tiezhi Zhang. Comparison of various online IGRT strategies: The benefits of online treatment plan re-optimization. *Radiotherapy and Oncology*, 90(3):367–376, 2009.
- [10] Qiuwen Wu, Yuwei Chi, Peter Y Chen, Daniel J Krauss, Di Yan, and Alvaro Martinez. Adaptive replanning strategies accounting for shrinkage in head and neck IMRT. *International Journal of Radiation Oncology\* Biology\* Physics*, 75(3):924–932, 2009.
- [11] Di Yan, DA Jaffray, and JW Wong. A model to accumulate fractionated dose in a deforming organ. *International Journal of Radiation Oncology\* Biology\* Physics*, 44(3):665–675, 1999.
- [12] Di Yan, David Lockman, Donald Brabbins, Laura Tyburski, and Alvaro Martinez. An off-line strategy for constructing a patient-specific planning target

- volume in adaptive treatment process for prostate cancer. *International Journal of Radiation Oncology\* Biology\* Physics*, 48(1):289–302, 2000.
- [13] Binbin Wu, Francesco Ricchetti, Giuseppe Sanguineti, Michael Kazhdan, Patrio Simari, Robert Jacques, Russell Taylor, and Todd McNutt. Data-driven approach to generating achievable dose–volume histogram objectives in intensity-modulated radiotherapy planning. *International Journal of Radiation Oncology\* Biology\* Physics*, 79(4):1241–1247, 2011.
- [14] Vorakarn Chanyavanich, Shiva K Das, William R Lee, and Joseph Y Lo. Knowledge-based IMRT treatment planning for prostate cancer. *Medical physics*, 38(5):2515–2522, 2011.
- [15] Li D Xu. Case based reasoning. *IEEE potentials*, 13(5):10–13, 1994.
- [16] Jeffrey Berger. Roentgen: radiation therapy and case-based reasoning. In *Artificial Intelligence for Applications, 1994., Proceedings of the Tenth Conference on*, pages 171–177. IEEE, 1994.
- [17] X Sharon Qi, Anand Santhanam, John Neylon, Yugang Min, Tess Armstrong, Ke Sheng, Robert J Staton, Jason Pukala, Andrew Pham, Daniel A Low, et al. Near real-time assessment of anatomic and dosimetric variations for head and neck radiation therapy via graphics processing unit–based dose deformation framework. *International Journal of Radiation Oncology\* Biology\* Physics*, 92(2):415–422, 2015.

- 
- [18] M Birkner, D Yan, M Alber, J Liang, and F Nüsslin. Adapting inverse planning to patient and organ geometrical variation: algorithm and implementation. *Medical physics*, 30(10):2822–2831, 2003.
- [19] Timothy E Schultheiss, Wolfgang A Tomé, Colin G Orton, et al. It is not appropriate to “deform” dose along with deformable image registration in adaptive radiotherapy. *Medical Physics*, 39(11):6531–6533, 2012.
- [20] Radhe Mohan, Xiaodong Zhang, He Wang, Yixiu Kang, Xiaochun Wang, Helen Liu, K Kian Ang, Deborah Kuban, and Lei Dong. Use of deformed intensity distributions for on-line modification of image-guided IMRT to account for interfractional anatomic changes. *International Journal of Radiation Oncology\* Biology\* Physics*, 61(4):1258–1266, 2005.
- [21] Xiaodong Zhang, Xiaoqiang Li, Enzhuo M Quan, Xiaoning Pan, and Yupeng Li. A methodology for automatic intensity-modulated radiation treatment planning for lung cancer. *Physics in medicine and biology*, 56(13):3873, 2011.
- [22] Masoud Zarepisheh, Troy Long, Nan Li, Zhen Tian, H Edwin Romeijn, Xun Jia, and Steve B Jiang. A DVH-guided IMRT optimization algorithm for automatic treatment planning and adaptive radiotherapy replanning. *Medical physics*, 41(6):061711, 2014.
- [23] George C Kagadis, Christos Kloukinas, Kevin Moore, Jim Philbin, Panagiotis Papadimitroulas, Christos Alexakos, Paul G Nagy, Dimitris Visvikis, and William R Hendee. Cloud computing in medical imaging. *Medical physics*, 40(7):070901, 2013.



- 
- [24] Jiajia Chen, Fuliang Qian, Wenying Yan, and Bairong Shen. Translational biomedical informatics in the cloud: present and future. *BioMed research international*, 2013, 2013.
- [25] Gonzalo Fernández-Cardenosa, Isabel de la Torre-Díez, Miguel López-Coronado, and Joel JPC Rodrigues. Analysis of cloud-based solutions on EHRs systems in different scenarios. *Journal of medical systems*, 36(6):3777–3782, 2012.
- [26] Jui-chien Hsieh and Meng-Wei Hsu. A cloud computing based 12-lead ECG telemedicine service. *BMC medical informatics and decision making*, 12(1):1, 2012.
- [27] Roy W Keyes, Christian Romano, Dorian Arnold, and Shuang Luan. Radiation therapy calculations using an on-demand virtual cluster via cloud computing. 2010.
- [28] Christopher M Poole, Iwan Cornelius, Jamie V Trapp, and Christian M Langton. Technical Note: Radiotherapy dose calculations using GEANT4 and the Amazon Elastic Compute Cloud. 2011.
- [29] James CL Chow. Cloud computing in preclinical radiation treatment planning. *Parallel & Cloud Computing*, 1(1):10–15, 2012.
- [30] Yong Hum Na, Tae-Suk Suh, Daniel S Kapp, and Lei Xing. Toward a web-based real-time radiation treatment planning system in a cloud computing environment. *Physics in medicine and biology*, 58(18):6525, 2013.

- 
- [31] Bowen Meng, Guillem Pratz, and Lei Xing. Ultrafast and scalable cone-beam CT reconstruction using MapReduce in a cloud computing environment. *Medical physics*, 38(12):6603–6609, 2011.
- [32] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic resonance imaging*, 30(9):1323–1341, 2012.
- [33] Csaba Pinter, Andras Lasso, An Wang, David Jaffray, and Gabor Fichtinger. SlicerRT: Radiation therapy research toolkit for 3D Slicer. *Medical physics*, 39(10):6332–6338, 2012.
- [34] Amazon Web Services. [aws.amazon.com](http://aws.amazon.com). Accessed: 2016-08-01.
- [35] Boto, a Python interface to Amazon Web Services. [github.com/boto/boto](https://github.com/boto/boto). Accessed: 2016-08-01.
- [36] Kelly H Zou, Simon K Warfield, Aditya Bharatha, Clare MC Tempany, Michael R Kaus, Steven J Haker, William M Wells, Ferenc A Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index 1: Scientific reports. *Academic radiology*, 11(2):178–189, 2004.
- [37] Alderson Radiation Therapy phantom and Alderson RANDO phantom. [www.rsdphantoms.com/rt\\_art.htm](http://www.rsdphantoms.com/rt_art.htm). Accessed: 2016-08-15.
- [38] Jennifer Andrea, Csaba Pinter, and Gabor Fichtinger. Measuring radiation treatment plan similarity in the cloud. In *World Congress on Medical Physics*

---

*and Biomedical Engineering, June 7-12, 2015, Toronto, Canada*, pages 432–435.  
Springer, 2015.