Decision Support System for Radiation Oncology

A Technical Report submitted to the Department of Biomedical Engineering

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Kelsie Reinaltt
Fall 2019, Spring 2020

Technical Project Team Members
Dana Schultz

On my honor as a University Student, I have neither given nor received
unauthorized aid on this assignment as defined by the Honor Guidelines
for Thesis-Related Assignments

Development of a Decision Support System for Radiation Oncology

Kelsie L. Reinaltt[a,*]. Dana K. Schultz[a,*], W. Tyler Watkins[b,1]

[a] Biomedical Engineering, University of Virginia
[b] Department of Radiation Oncology, University of Virginia School of Medicine
[1] Correspondence: ww5x@hscmail.mcc.virginia.edu, (434) 982-0161
* Authors contributed equally on this work

Abstract

The integration of artificial intelligence into radiation oncology treatment development has increased the amount of treatment plans that can be generated per patient. With more plans to evaluate, there is now a demand for tools that allow radiation oncologists and physicists to understand the differences between treatment plans. A decision support system for radiation oncology which includes visualization tools to analyze, modify, and comprehend tradeoffs between different treatment plans is being developed. The visualization tools make use of DICOM files, the industry standard for managing and communicating medical imaging information, and computational code for dose-volume histograms, which relate radiation dose to tissue volume for comparison to other plans and structures. Using programming languages such as HTML and JavaScript, and the existing industry resources, two visualization tools, a DICOM reader with dose overlay capabilities and a dynamic radar chart that displays high-correlation data from an interactive dose-volume histogram (DVH) were produced and prepared for integration into the decision support system. These tools will allow radiation oncologists and physicists to more easily determine tradeoffs between plans and to administer treatments more efficiently. As treatment development tools become more sophisticated and complex, visualization tools will ensure that the differences between each plan are well understood

Keywords: Radiation Oncology, Artificial Intelligence, Decision Support, Visualization Tools

Introduction

Radiation therapy (also called radiotherapy) is a form of cancer treatment that uses high doses of radiation to terminate cancer cells and shrink tumors. Determining the most effective radiation therapy treatment plan depends on various factors, including but not limited to: the type of cancer, the size of the tumor, the tumor's location in the body, how close the tumor is to normal tissues that are sensitive to radiation, and patient's general health and medical history. Treatment plans are created for each patient depending on a combination of such factors with the goal of maximizing radiation dosage received by the target tissue while minimizing dosage received by organs at risk (OARs). With numerous factors that must be considered, various tools have been created to aid the treatment development and decision-making process. Under current workflows, dosimetrists use treatment planning software such as Pinnacle by Philips Healthcare or Varian's Eclipse to develop and visualize radiation treatment plans[1]. Although these programs are designed to optimize the plan generation process, they still require a high level of user input. As technology has continued to improve and research has provided insight into important factors to consider in treatment planning, radiation treatment planning has become more complex and labor intensive[2].

To facilitate the process of developing treatment plans, algorithms have been developed to automate and advance multiple aspects of medical science, including the incorporation of artificial intelligence into radiation treatment planning[2]. Artificial Intelligence (AI) is a subfield of computer science which focuses on developing and improving computers to engage in human-like thought processes and capabilities such as learning, adapting, reasoning, and self-correction[3]. In the field of medical imaging, deep learning methods have demonstrated the ability to outperform humans in various pattern-recognition applications, including medical image analysis relevant to radiology[4]. Under investigation by radiation oncologists and physicists, artificial intelligence has yielded results indistinguishable from that of human experts while considerably reducing the duration of the task[5]. To that end, software packages incorporating a machine-learning approach to treatment plan generation have been released in order to further speed up the process and allow for more comprehensive treatment, such as Siris Medical's QuickMatch[6] and RaySearch's RayStation[7]. Additionally, research continues to be done in both commercial and academic settings to advance the role of artificial intelligence in radiation therapy treatment planning.

The system being developed in conjunction with this project incorporates artificial intelligence with decision making software to generate radiation therapy treatment plans more efficiently. Due to the increase in efficiency, more plans are able to be generated per patient when compared to the performance of a dosimetrist using traditional treatment planning software, who generally is limited to a handful of plans per patient due to the high level of input required. However, as more treatment plan options are made available by AI algorithms, the ability to compare, adjust, and determine tradeoffs between generated treatment plans becomes increasingly valuable, especially when considering the human limitation for factor evaluation. The average human has been shown to have a capacity of handling five factors or less for any given decision. The machine learning algorithm incorporated in this system is capable of creating up to 20 treatment plans per patient, each of which has several distinct factors that must be considered. As a result, it is extremely difficult for users of the system to evaluate all of the generated plans and determine the best option for the patient in question. To address this issue, radiation dose visualization and tradeoff quantification tools have been developed for incorporation into the decision support system that will facilitate the treatment decision treatment planning process. This includes a three-dimensional DICOM (Digital Imaging and Communications in Medicine) image viewer and radar chart.

Materials and Methods

The two aims for this project were to develop an interactive tool which allows oncologists to visualize how a treatment plan manifests on a region of interest as well as the organs at risk around the region, and to develop tradeoff quantification and visualization tools. A DICOM reader was developed to address the first aim, and a dynamic radar chart associated with an interactive dose-volume histogram was developed to address the second aim.

Dose Visualization

The specifications for the DICOM reader to be developed were to allow for 3-dimensional visualization of a medical image, and to include a dose overlay feature. Multiple existing libraries were evaluated to determine if any of the desired features had already been implemented. Code which already demonstrated 3-D visualization capabilities was pulled from a GitHub repository called Papaya as a starting point. Relevant files within the Papaya library were thoroughly analyzed to provide the background necessary for deciding where and how to implement the dose overlay feature. After reviewing the code and testing different interface options, it was discovered that a dose overlay feature existed within the code dependent on certain parameters. Using JavaScript, the code was modified to include the overlay dose functionality. Once the code was working properly, a program was written to package the final result for integration into the decision support system, as the code for the completed web page was not completed at the time.

Tradeoff Quantification

The radar chart was developed in JavaScript and HTML, using the Chart.js library for animated graphs, and incorporated into existing code for the complete web-based version of the decision support system. Correlation calculations were based on an existing Python function, and were translated into JavaScript. This was done to avoid interfacing between the JavaScript and Python to pass data back and forth each time correlations needed to be calculated, thus optimizing overall run-time of the program. A program was written to take in the data from an interactive dose-volume histogram (DVH) and display the most highly correlated data on a radar chart (Figure 1).

The data pulled from the DVH chart is structured as a multidimensional array of volume values, where the dimensions corresponded to the organ, plan number and dose value, respectively. The array is first rearranged so that the second and third dimensions were switched, resulting in arrays of volumes corresponding to 20 plans for each dose level for each organ. For the correlation calculation, the array of plans corresponding to the organ and dose selected by the user from the DVH are isolated. This array is then iteratively compared to all other dose/organ combinations from the rearranged DVH array. A correlation coefficient is determined for each combination using a JavaScript function found online for calculating Pearson correlation. As the correlation coefficients are calculated, the top five correlations are stored, as well as the organ and dose level at which they occur. The absolute value of the correlation is

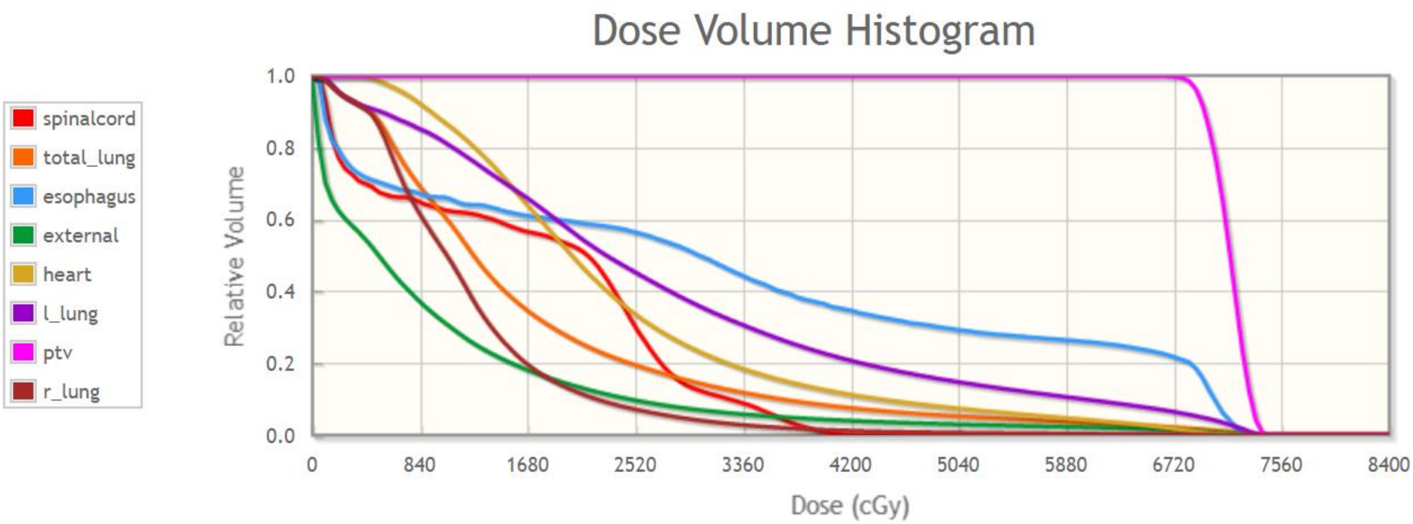

Fig. 1. Interactive Dose-Volume Histogram. Dose-Volume Histogram graphically summarizes the radiation distribution per organ for a proposed treatment plan[8]. Each curve on the graph represents an organ's exposure to radiation at different doses. The user can move a curve for an organ to a different volume and dose, and the other organ-dose combinations update accordingly.

used to include both positive and negatively correlated changes. For each of the highest correlated organ/dose pairs, absolute change in volume is determined by computing the range of the array of plans for that specific organ and dose levels.

Once the top correlations are calculated, the radar chart, which was programmed to be initialized with placeholder data upon loading of the page, is updated. The axis labels are updated to the stored organ and dose levels of the top correlations, as well as the organ and dose level originally selected by the user. On each axis, the value plotted corresponds to the absolute volume change value calculated. The radar chart was programmed dynamically so as to execute the correlation calculation, compute absolute change, and reflect updates in the radar chart axes and plotted values each time a curve on the DVH is moved.

Results

Dose Visualization

The first tool that was developed was an interactive medical image viewer. The tool displays DICOM formatted images obtained through computed tomography (CT) imaging in grayscale (Figure 2). These images can then be overlaid with dose structure images in color aligning with the tissue structures. In the dose structure image, the color corresponds to dosage. Several included color scheme options are available to visualize dose structure. For instance, with the color scheme chosen in Figure 2, the white in the center of the image indicates a higher level of dosage relative to the surrounding orange/red portions. Additionally, the viewer enables the user to see the images from three cross-sectional viewpoints to achieve a three-dimensional understanding of the structures and associated dose.

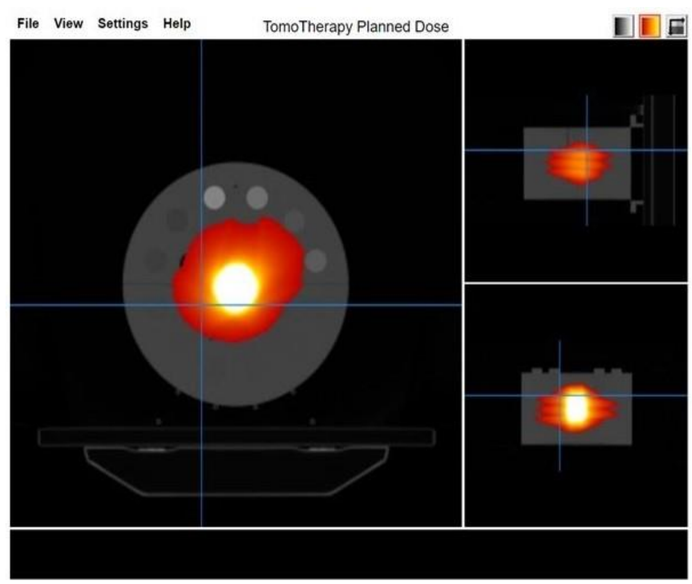

Fig. 2. DICOM Viewer for Dose Visualization. The DICOM viewer was downloaded from the Papaya repository on GitHub and modified. A CT image of a phantom was loaded in grayscale along with a sample radiation dose file in color. The viewer includes multiple frames as well as the ability to scroll through individual cross sections to allow for visibility of the full 3-dimensional structure.

Tradeoff Quantification

The second tool that was created was a radar chart to assess the tradeoffs between different treatment plans (Figure 3). The radar chart corresponds to the interactive DVH that is also included in the decision support system. The DVH includes a curve for each region of interest (ROI) for the selected patient. The user is able to adjust the chart by choosing an individual curve at a specific dose level and clicking and dragging to a higher or lower volume level. As the DVH updates based on user input, the radar chart updates to represent the most highly correlated changes to organ/dose pairs resulting from the adjusted organ/dose curve on the DVH. The top axis on the final radar chart corresponds to the organ and dose level adjusted on the DVH. The remaining axes consist of the name of the organ and dose of the five highest correlated organ-dose pairs. The value plotted on each axis corresponds to the absolute change in volume received by the organ at the specified dose level as a result of the change made to the specified organ and dose level. Each time a new curve on the DVH is selected and the volume is adjusted the radar chart updates automatically to display the new highest correlated organ and dose levels and their associated change in volume.

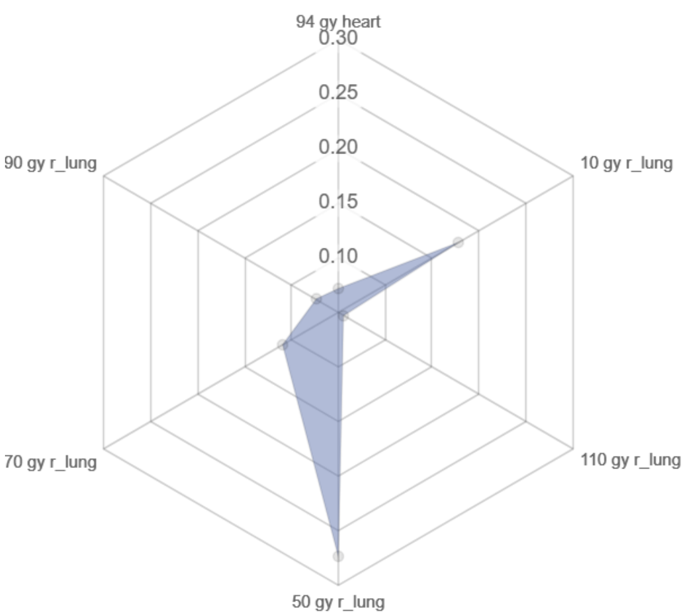

Fig. 3. Radar Chart for Tradeoff Quantification. The radar chart was created using the Chart.js library and was coded to update dynamically as DVH volume updates. The top axis represents the absolute change for the selected organ and dose, while the remaining 5 axes illustrate the absolute change for the organ of the organ-dose pairs most highly correlated to the selected organ and dose.

Discussion

The tools developed for this project greatly enhance the ability of radiation oncologists to quickly and intuitively compare treatment plans. The DICOM viewer allows physicians to visualize the dose structures in relation to the target area ahead of time, enabling them to plan out their strategy ahead of time and improve the chances of success. In future iterations of the decision support system, the viewer will be further modified to easily scroll through the various treatment plans generated by the machine learning algorithm. This will facilitate easy comparison of the structural differences between plans, and give physicians an idea of which plans are more feasible.

The second tool that was developed presents the data from the DVH in a much more interpretable manner. The radar chart is designed to minimize the number of factors a user needs to consider while making maximum use of the several factors that go into selecting a treatment plan. By displaying only the volume change in the organ/dose pairs most highly correlated to the selected organ and dose level, the radar chart filters for the most important data from the DVH and presents it clearly. This drastically reduces the amount of time required to analyze the different plans, as the data to be interpreted is reduced from eight to ten organs with 200 dose values and 20 plans per dose value to six organ/dose pairs with a single volume change value associated with each. Additionally, by displaying the absolute change in volume of an ROI receiving radiation, the user is able to see the tradeoffs of any given change made to the DVH, and adjust the plan based on which organs are of higher priority. The combination of increased efficiency and tradeoff comparison of the system has the potential to speed up the process of choosing a treatment plan for a given patient, which could allow doctors to help more patients in the same amount of time.

When these tools are combined with machine learning techniques, the overall decision support system also has the potential to improve the accuracy of treatment planning. The machine learning algorithm included in the system will be capable of generating more treatment plans per patient than a dosimetrist can with traditional planning software. By increasing the number of plans available for a given patient, doctors have much more opportunity to personalize the treatment plan to their patient to meet their specific needs. Thus, the incorporation of automatically generated plans could potentially improve the chances of success of radiotherapy, as there is a greater chance for finding a treatment plan that meets all of the specific needs of the patient.

Limitations

Throughout the duration of the project, the aims shifted due unanticipated changes in the availability of resources. Initially we expected to receive code for the first version of the application's front-end by November 2019. Due to setbacks in the development of the front-end by third-party developers, the tools developed were not integrated into the official version of the decision support system. Instead, the DICOM reader was packaged and prepared for integration into the final application, and the radar chart was integrated into a partially complete version of the application's front-end.

Another limitation of the functionality of the decision support system stems from the programming language used. The tools developed were primarily coded in JavaScript, a client-side scripting language[9]. As an interpreted language, JavaScript programs and functions do not need to be compiled prior to running, and are ideal for task automation in a program or procuring information from data sets[10]. Interpreted languages, however, do not execute as quickly as compiled programming languages: instead of being converted into machine code before execution, programs are executed line-by-line. In preparing the data to be processed by the correlation function, the average of each bin of organ-dose arrays was taken and passed to the correlation calculation. Running the correlation calculation on the processed data caused the website to freeze, as the calculations iterated for multiple arrays became time consuming. To address the slow run-time correlations were estimated on the median of a bin instead of the average, but at the cost of precision of the correlation. With no compiler, errors in calculating the absolute change could only be discovered after the code was executed[11]. Moreover, JavaScript is a loosely typed language, meaning it implicitly converts to different types two types do not match in an operation or function. If developers are not careful to check data types and structures when using JavaScript for computation, the program might not produce the desired results. Despite the potential drawbacks of programming with JavaScript, the language provides benefits over other programming languages crucial for an application. JavaScript scripts are able to run on the computer's browser, as opposed to on the server wherever it may be[9], thus allowing for immediate responses to user action[9]. Most if not every web browser is compatible with JavaScript, and JavaScript applications can run on any device, making the development of a cross-platform application much simpler[12]. Finally, scripts are not able to read or write files on a user's hard disk to protect users against damaged hard disks and stolen data[9]. Based on the need for the support system application to be interactive, dynamic, functional in all web browsers, and safe for users, JavaScript served as an adequate language in the development of this application. The JavaScript language meets the needs of the application at its current stage, but as the system incorporates more sophisticated tools and capabilities, introducing a compiling language might be beneficial for ensuring the quality of the system.

Future work

In addition to the reported results, we intended to perform in-person usability testing on the tools created. Usability testing is the process of evaluating target user experience while using an application to gauge how accessible and intuitive the application is for the user[13]. Such analysis is conducted to ensure the expectations of the user are being met, reduce or eliminate interface flaws, discern user reactions to the application, and observe how successful users can complete tasks in the current state of the application. The testing would have focused on problem discovery, a study focused on pinpointing complications, and evaluating learnability by keeping record of user actions, reactions and errors to quantify the ease at which participants learn how to use the application and accomplish specific tasks. As a result of the COVID-19 crisis, it was no longer possible to connect with medical professionals to evaluate the tools and application. As the application continues to advance, gathering quantifiable feedback from target users will be beneficial to evaluating progress and ensuring user satisfaction. Going forward, the next step in this project would be to collect this feedback, to ensure that the website has the potential to be useful for oncologists before continuing development.

In addition to collecting feedback, the decision support system has a lot of potential for improvement. Additional visualization or analytical tools can be developed to further enhance the comparison capabilities of the software and support the decision-making process of its users. As these tools become more complex, a compiled programming language can be used in addition to JavaScript to optimize the speed of the program and allow for more robust calculations to be performed. Additionally, work still remains in connecting the interface with the algorithm for plan generation. The interface used for development in this project included a set of previously generated treatment plans for various patients. These datasets were used for the testing of each of the tools throughout the development process to ensure that they functioned correctly. Once the machine learning algorithm is incorporated into the decision support system, the result will be a working prototype that is able to take in patient data, automatically generate treatment plans, and allow the physician to efficiently evaluate those plans with various analytical tools and proceed with treating their patient.

End Matter

Author Contributions and Notes

K.L.R, D.K.S, and W.T.W designed research, K.L.R, and D.K.S performed research, K.L.R, D.K.S, and W.T.W wrote software, K.L.R. and D.K.S. analyzed data; and K.L.R and D.K.S wrote the paper.

The authors declare no conflict of interest.

Acknowledgments

We would like to thank Dr. W. Tyler Watkins for his advising and contribution to our work. We would also like to thank Dr. Timothy Allen and Dr. Shannon Barker, University of Virginia Department of Biomedical Engineering, for their guidance throughout our project.

References

1. Taschetta-Millane M. Treatment Planning Systems Overview. Imaging Technology News. https://www.itnonline.com/article/treatment-planning-systems-overview. Published July 6, 2016. Accessed November 18, 2019.
2. Wang C, Zhu X, Hong JC, Zheng D. Artificial Intelligence in Radiotherapy Treatment Planning: Present and Future. Technol Cancer Res Treat. 2019;18. doi:10.1177/1533033819873922
3. Kok JN, Boers EJW, Kosters WA, Peter vander P, Poel M. ARTIFICIAL INTELLIGENCE: DEFINITION, TRENDS, TECHNIQUES, AND CASES. In: Encyclopedia of Life Support Systems (EOLSS). https://www.eolss.net/Sample-Chapters/C15/E6-44.pdf. Accessed April 20, 2020.
4. Thompson RF, Valdes G, Fuller CD, et al. Artificial Intelligence in Radiation Oncology Imaging. International Journal of Radiation Oncology*Biology*Physics. 2018;102(4):1159-1161. doi:10.1016/j.ijrobp.2018.05.070
5. Kiser KJ, Fuller CD, Reed VK. Artificial intelligence in radiation oncology treatment planning: a brief overview. Journal of Medical Artificial Intelligence. 2019;2(0). http://jmai.amegroups.com/article/view/4996. Accessed April 20, 2020.
6. Valdes G, Simone CB, Chen J, et al. Clinical decision support of radiotherapy treatment planning: A data-driven machine learning strategy for patient-specific dosimetric decision making. Radiotherapy and Oncology. 2017;125(3):392-397. doi:10.1016/j.radonc.2017.10.014
7. Automated Treatment Planning | RaySearch Laboratories. https://www.raysearchlabs.com/automated-treatment-planning/. Accessed November 18, 2019.
8. Drzymala RE, Mohan R, Brewster L, et al. Dose-volume histograms. Int J Radiat Oncol Biol Phys. 1991;21(1):71-78. doi:10.1016/0360-3016(91)90168-4
9. JavaScript. https://people.clarkson.edu/~vdubrovs/xxx/javascript/js_tutorial/index.htm. Accessed April 26, 2020.
10. What's the difference between Scripting and Programming Languages? GeeksforGeeks. March 2016. https://www.geeksforgeeks.org/whats-the-difference-between-scripting-and-programming-languages/. Accessed April 26, 2020.
11. Compiled Versus Interpreted Languages. https://guide.freecodecamp.org/computer-science/compiled-versus-interpreted-languages/. Accessed April 26, 2020.
12. Introduction to JavaScript. https://web.stanford.edu/class/cs98si/slides/overview.html. Accessed April 26, 2020.
13. Udoh (Inioge) I. ALL YOU NEED TO KNOW ABOUT USABILITY TESTING. Medium. https://uxplanet.org/all-you-need-to-know-about-usability-testing-b754158e195f. Published June 11, 2019. Accessed April 25, 2020.