

User Interface for Radiation Therapy Cohort Selection

Domonique Carbajal and Keefer Chern

Mentors: Dr. Todd McNutt and Pranav Lakshminarayanan

May 9, 2018

Computer Integrated Surgery II

Dr. Russell Taylor

Spring 2019

1. Introduction
2. Technical Approach
3. Results
4. Significance
5. Management Summary
6. Acknowledgement
7. References

1. Introduction

Radiation therapy is a cancer treatment that utilizes high doses of ionizing radiation to kill cancer cells and control tumors. Each radiation therapy session for a patient requires a specifically tailored treatment, which needs to have high accuracy and guarantee to deliver the correct amount of radiation to the desired area. Typically multiple sessions are needed in order for the cancer to be treated and after each session, specific dose-volume data can be obtained. The practice of radiation therapy can significantly benefit from using previous patient data to tailor treatment for new patients; especially when considering what dose should be applied to a certain region of interest or anticipating toxicities.

There exists a SQL database which contains large amounts of data of patients with different types of cancers and a website connecting with this database (Oncospace). For example, there is data for 1,550 head and neck radiation therapy patients with up to 6 years follow up and data for 2,096 prostate cancer patients. A website written in C# is connected with this database, which has visualizations for different types of data. Some of these visualizations include distribution of clinical assessments and patient demographics. Each patient data point holds a variety of information, which can be categorized into static, longitudinal, or derived variables. Static variables are information about the patient that does not change throughout treatment. Examples of static variables are date of birth, race, gender, and diagnosis. Longitudinal variables are information that should change throughout the course of treatment and recovery. Typically these variables change as a function of time. An example of a longitudinal variable is duration and onset of symptoms (toxicities and grades). Lastly, we have the derived variables, which are information that require a mathematical computation to be applied to extract useful data from. An example of such variable is the dose volume histogram, where each patient has a set of points making up a curve that must be looked at in its entirety.

This data can be used for a significant amount of applications in regards to research and clinical care. There is a desire for doctors and researchers in applying this data for quality reporting, decision support, and studies. Specifically, quality reporting includes analyzing disparities of care, practicing quality reporting, and safety. Decision support includes toxicity prediction, data-driven quality control, and treatment adaptation. Lastly, research applications for this data include performing clinical trials and answering biological questions. Gaining all the benefits from the data available requires an easy to use system that can relay the information desired into a comprehensive format.

Currently, there is no quick and intuitive way to select patient cohorts from this database other than through a free text SQL Query. This requires knowledge of the exact variable names and some familiarity with the language. There are also no visuals of the outcomes of these patients selected by these many variables. Researchers hoping to compare groups or select a group to further study are prevented from fully utilizing the data available.

2. Technical Approach

The goal of our project was to create a user interface that would allow for easy selection of filters to apply onto the patient data and create cohorts. The user interface would also allow for visualizations of longitudinal data for exploration of significant features. Another added feature would be the saving and loading of cohort selections.

2.1. Front End (Summary)

Our implementation involves integrating a few packages to create the user interface on a web application. The code is primarily written in html and java. In addition, SQL queries were generated utilizing java. Eclipse is the development environment for the project. For the build tool, this project utilizes Maven due to its integration with Eclipse and has an easy way to handle libraries. To develop the look and feel of the user interface, this project uses Java Server Faces(JSF), which is a Java specification for building component-based user interfaces for web applications. JSF would largely be used with primefaces to generate the visuals of the application. Apache Tomcat provides a Java HTTP web server environment for the web application to run in. Spring is used for object instantiation. Hibernate ORM is an object-relational mapping tool for Java which is used to connect to databases.

2.2. Back End (Summary)

The oncospace microsoft SQL server database is the source of all of the data being provided to the front end and used in creating the visualizations. This database involves a very complex schema of connected tables and patient representations. We were not given access to all of the available information in the database and as a result limited our filtering to the available useful variables.

2.3. Step by Step Technical Procedure

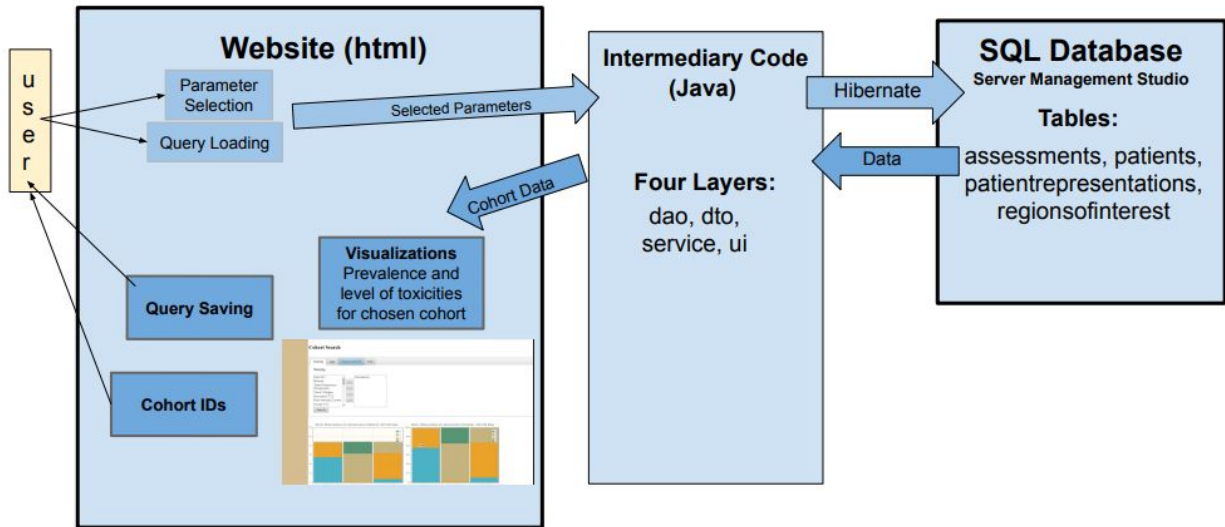


Figure 1. Shows how the code communicates between each component of the application. The user interacts with an html page, which then transfers user input to the intermediary code. The intermediary code process the input, queries for information with hibernate, and then sends user's desired information back to the html page. The user then has the option to save the query and extract cohort IDs.

2.3.1. Setting up the Framework

To start off we set up the developing environment using the installation process as discussed in the documentation. Afterwards, we worked towards connecting our application to the database using a hibernate file, which contains all the login information required to connect to the SQL server. After that we set up four separate intermediate layers with each providing a different function to our application. The first layer is the Data Transfer Object layer (dto), which creates an object that would represent each table that our application interacts with in the database. The second layer is the Data Access Object layer (dao), which would generate SQL Query scripts that would then be sent to hibernate to extract data; the returned data would be stored into dto objects. Next is the service layer, which would contain the inputs from the user and call on the dao layer in order extract data. The final layer is the User Interface layer (ui) which would connect .html files to the other intermediate layers and perform the majority of the business logic. With all the intermediate layers setup, we then create the .html files that would be the front end of our implementation and display all our visuals.

For all the visuals we will be using primefaces, which is an open source framework for JavaServer Faces. This completes the set up for the framework of our application.

2.3.2 Developing Algorithm for Static and Longitudinal Variables

The algorithms are developed in SQL to produce the patient data in the case of static variable filtering and to produce patient counts in the case of longitudinal variable. The complexities of static variable algorithms involve the connection of tables in the database schema. An example of this can be seen in **Figure 3** with the connection required between the tables Patient and Patient Representations to access both ROI and Assessments. The longitudinal algorithms require subqueries of patient IDs to assure that the patients have instances of data with the dates in each time period(<7, between 180 and 365, and >540). It also uses the patient IDs to make a count of the number IDs for each specific grade. **Figure 2** just gives the grade counts for the earliest date for each patient and is an example of the complexity of the algorithms required to gather the data to make visualizations.

```
WITH CTE
AS
(
SELECT
    ROW_NUMBER() OVER(PARTITION BY patientID ORDER BY patientID, [date])
    AS rn , patientID, name, grade, [date]
FROM OncospaceHeadNeck.dbo.Assessments
WHERE [type]= 'toxicity' AND name = 'Xerostomia'
--patient has baseline score for Xero
AND patientID in (SELECT patientID FROM OncospaceHeadNeck.dbo.Assessments
    WHERE [type]= 'toxicity' AND name = 'Xerostomia' AND [date]<7 AND
    grade >= 0)
--patients that have record between 6-12months
AND patientID in (SELECT patientID FROM OncospaceHeadNeck.dbo.Assessments
    WHERE [type]= 'toxicity' AND name = 'Xerostomia' AND grade >= 0 AND
    [date] BETWEEN 180 AND 365)
--patients that have record between 18-36months
AND patientID in (SELECT patientID FROM OncospaceHeadNeck.dbo.Assessments
    WHERE [type]= 'toxicity' AND name = 'Xerostomia' AND grade >= 0 AND
    [date] > 540)
--Will give the First for Each Patient that has all time periods
)
--just give counts in ea grade category
SELECT COUNT(DISTINCT patientID), grade
FROM CTE
WHERE rn = 1
GROUP BY grade;
```

Figure 2. Example of the longitudinal algorithms that are required to provide count data of each grade.

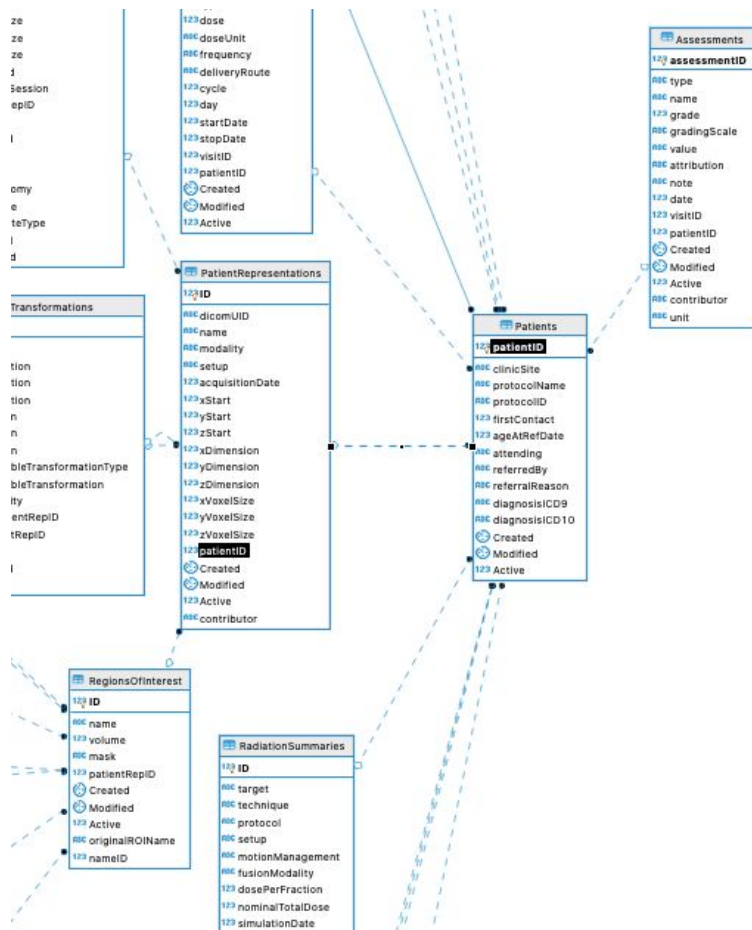


Figure 3. Example of portion of database schema exhibiting complex interconnectedness of the tables. Ex. ROI is associated with a Patient Representation ID and Patient Representation table, Patient Representation ID each has an associated Patient ID, and Patient ID is needed to access Assessments which includes toxicities.

2.3.3. Implementing each Algorithm

For each of the developed algorithms, we would apply them onto our framework. To do this we would need to modify our intermediate layers. Our dao layer would be modified to be able to generate the types of SQL queries that would extract the data the user wants. In the dto layer, new objects will be created to represent new tables that the algorithm would interact with. The service layer would be modified to be able to call on the newly modified dao layer. Lastly new business logic will be added in the ui layer to process the queried data. On the front end a .html file is created, following the algorithm, to show the data and UI to the user.

3. Results

We were able to develop almost all of the features that we had initially planned on implementing. This includes filtering by: age, diagnostic ROI, Toxicity, and Toxicity grade during a given time period. There are also features to download the patient IDs obtained through filtering and save and load filters applied. Visualizations of the toxicity data are also produced (with a continuous and discrete option depending on the measure of grade). For comparing two selections of cohorts there is a double search option which allows for side by side visualizations of the cohorts of interest.

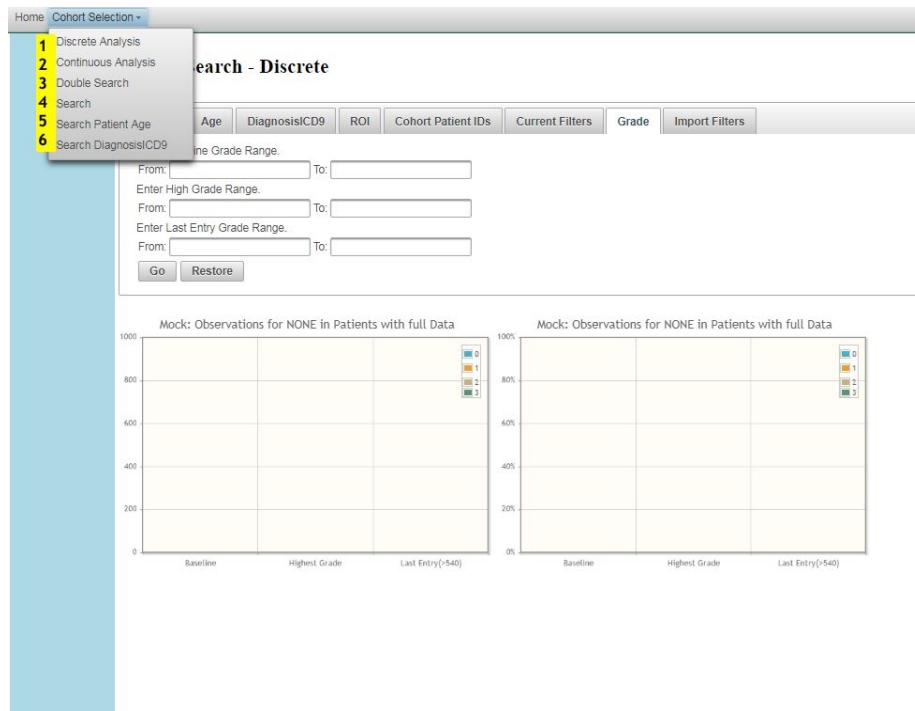


Figure 4. View of application with highlighted dropdown menu of options

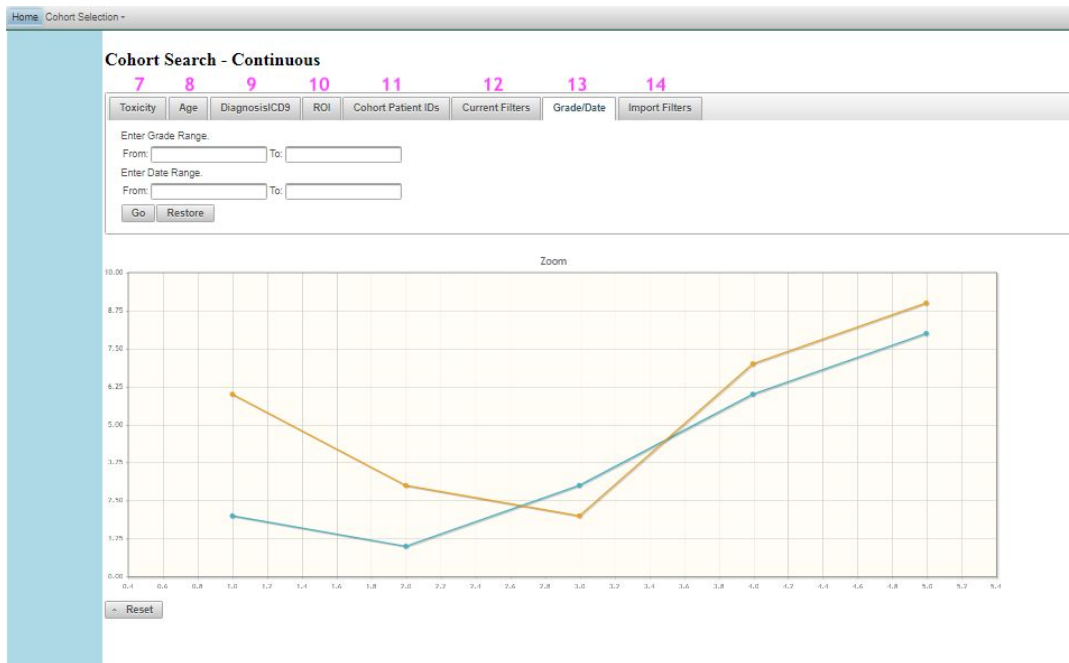


Figure 5. Continuous search with filter tabs. Line plot of toxicity data

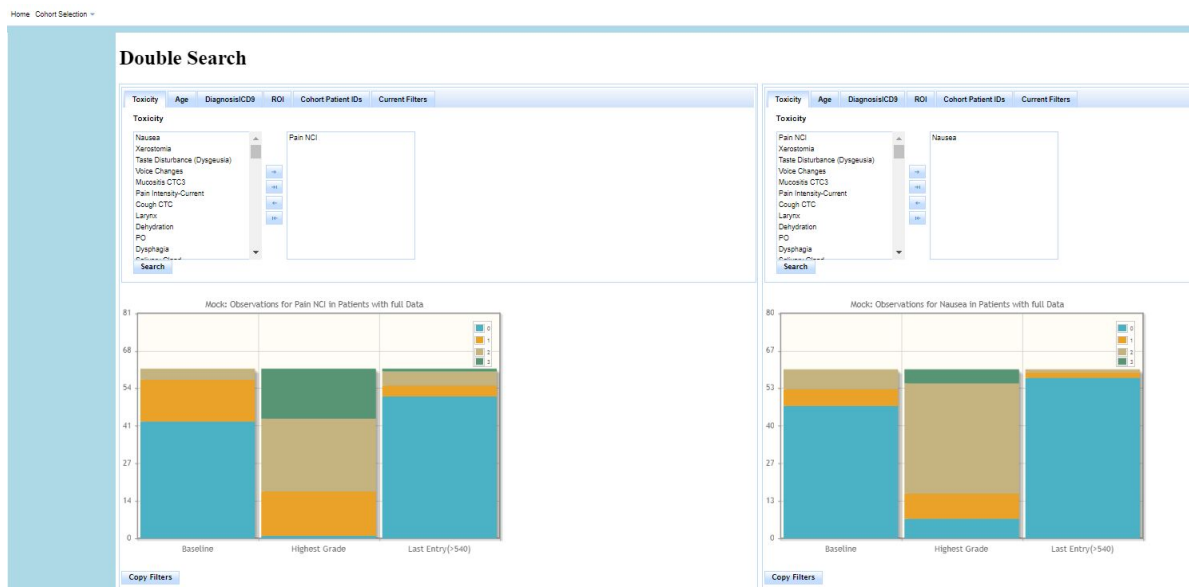


Figure 6. View of the Double Search with two separate toxicities. Can compare cohorts of patients differing by specific features.

Menu Selections

Number	Dropdown Name	Description	Filters Included:
1	Discrete Analysis	Provides visualizations for discrete grade toxicities in the form of a stacked bar chart where the patient count for each grade is a stacked component.	Toxicity, Age, Diagnostic ICD9, ROI, Cohort Patient IDs, Current Filters, Grade, Import Filters
2	Continuous Analysis	Provides visualizations for continuous grade toxicities in the form of a line plot where each patient is a line and the axes are Date vs. Grade.	Toxicity, Age, Diagnostic ICD9, ROI, Cohort Patient IDs, Current Filters, Grade, Import Filters
3	Double Search	Provides comparison of discrete visualizations of two separate searches with "Copy Search" button that can transfer the first search to the second so that a comparison can be made on an additional filter with the initial filters the same.	Toxicity, Age, Diagnostic ICD9, ROI, Cohort Patient IDs, Current Filters
4	Search	Static variable filtering that does not provide visualizations but produces table of cohort data.	Age, Diagnostic ICD9, ROI, Toxicity.
5	Search Patient Age	Static variable search of just patient's age.	Age
6	Search Diagnostic ICD9	Static variable search of just Diagnostic ID9's.	Diagnostic ICD9

Filters/Options

Number	Filter Name	Description	User Specifications
7	Toxicity	Ability to choose the toxicities present in patient assessments. First toxicity chosen has grade filters applied to it.	Can select from table of top 30 toxicities and choose to insert as a required filter.
8	Age	Ability to filter patient cohort by selecting the upper and lower bounds of age.	To input values, click the integer accepting boxes in From # To #. Where # is the integer accepting boxes. If no values are put in, then the filter assumes that there are no bounds.
9	Diagnostic ICD9	Ability to filter patient cohort by Diagnostic ICD9 code.	Drag all the ICD9s from the left box to the right box. Once search is clicked all the patients that are left contain data for that ICD9 code. (AND)

10	ROI	Ability to filter patient cohort by Regions of Interest.	Drag all the ROIs from the left box to the right box. Once search is clicked all the patients that are left contain data for that ROI. (AND)
11	Cohort Patient IDs	Provides a table of patient ID's present in current cohort.	Gives options to download patient ID's into .xls, pdf, .xml or .csv files.
12	Current Filters	Provides a view of all the currently applied filters in the cohort and to export the filters.	User can view the currently applied filters and click a button exporting the filters into .xls, pdf, .xml or .csv files.
13	Grade	Ability to filter the toxicity by grade for a specific time period.	<i>Discrete</i> : 3 sections for each of the different periods(First Entry, Highest Entry, Last Entry) with 2 integer receiving boxes of grade upper and lower bound <i>Continuous</i> : 2 sections, the first for specifying upper and lower bound for grade, the second for specifying upper and lower bound for date
14	Import Filters	Provides opportunity to import previously exported filters.	Select upload button and select from local files the downloaded .xml file of exported filters.



Figure 7. View of Import Filters selection tab with information from imported file visible.

4. Significance

The resulting interface is significant as the database has largely untapped potential in improving patient care and research goals. Given the UI developed in the project, clinicians can use the large amounts of data they have to obtain a patient cohort similar to their current patient. This data on patient cohort can be used to anticipate expected outcomes. Researchers can use the UI developed in this project to easily

explore the data for interesting patterns or relationships between variables. Age, toxicities, and regions of interest can be examined in new ways with the severity of outcomes quickly visualized. Knowledge of SQL and query calling is no longer necessary with our implementation, expanding the population of those able to analyze database of information and the population of those who can benefit from its results.

5. Management Summary

5.1. *Who did what?*

Keefer Chern: Built UI Framework, developed frontend selections and visualizations, connected developed frontend to backend.

Domonique Carbajal: Managed documentation of features, developed SQL algorithms for data extraction from backend

5.2. *What was accomplished vs planned?(Our Deliverables and their status)*

Minimum: Develop SQL algorithm allowing for static variable(age, race, gender, diagnosis) selection of cohort with code and documentation. Framework setup for the implementation of the UI.

Status: Completed

Expected: Algorithm for longitudinal variable (duration/onset of symptoms) selection of cohort with code and documentation. Creating a UI for static variable cohort selection along with code and documentation.

Status: Completed

Maximum: Algorithm for derived variable selection of cohorts with code and documentation. Query saving and loading and UI for longitudinal variable Cohort Selection implementation along with code and documentation.

Status: Completed (with exception to Derived algorithm as the pertinent information was not able to met and discussed over given limited final timeline and scheduling conflicts)

These are the updated deliverables presented in our Midterm Presentation.

5.3. Possible Future

Future improvements include development of grade filters for multiple toxicities, filtering from interaction with visual, filtering of other variables we were not given access to, and exploration of derived variables. After usage by oncologists and radiologists, their feedback could help improve the tools by adding other visualization and analysis features. Lastly, an extension of the application to mobile devices. Keefer hopes to continue working on the project in the future.

5.4. Lessons Learned

We obtained a significant amount of knowledge from working on this project and presenting our progress to an audience. This project allowed us to gain experience working with SQL, JSF, HTML, developing web frameworks, and connecting to databases. The process and importance of documentation, specifically as a planning tool, was something that we utilized as a team and led to a more thoroughly developed interface. We have honed our presentation skills from receiving feedback from Professor Taylor and have a better understanding of the best forms to present technical information to an audience.

6. Acknowledgement

We would like to thank Dr. McNutt and Pranav for their amazing guidance and support for this project. Thanks to their vision and direction we were able to create a interface that would provide the most use to clinicians. Thank them for continued patience in helping to establish connection to the database and forming visualization recommendations to guide us. We would also like to thank Professor Taylor whose critics helped develop our presentation skills and performance.

7. References

- Mayo, Charles S., et al. "The Big Data Effort in Radiation Oncology: Data Mining or Data Farming?" *Advances in Radiation Oncology*, vol. 1, no. 4, 2016, pp. 260–271., doi:10.1016/j.adro.2016.10.001.
- Mcnutt, Todd R., et al. "Needs and Challenges for Big Data in Radiation Oncology." *International Journal of Radiation Oncology*Biology*Physics*, vol. 95, no. 3, 2016, pp. 909–915., doi:10.1016/j.ijrobp.2015.11.032.
- Mcnutt, Todd R., et al. "Using Big Data Analytics to Advance Precision Radiation Oncology." *International Journal of Radiation Oncology*Biology*Physics*, vol. 101, no. 2, 2018, pp. 285–291., doi:10.1016/j.ijrobp.2018.02.028.