

Radiotherapy Dose-Toxicity Analysis UI

By: Santiago Appiani, Alaleh Azhir, William Franceschi
Mentors: Dr. Todd McNutt, Pranav Lakshminarayanan

Goal

Develop a **web-based user interface** for refined dose-toxicity analysis:

- Compatible with existing online **SQL database** for obtaining the medically related data
- Create interactive **3D** visualizations of objects using **JavaScript** libraries such as D3.js
- Allow physician to easily use **segmenting** tools on organs, run analysis on new regions
- **Display the results** using interactive DVH histograms on the website
- Allow **new** feature analysis scripts to be **easily added** to the existing user interface
- Enable the users to **Download** the results of the analyses from the web

Significance

- Researchers and oncologists lack an easy to use way to analyze Radiation Therapy Dose-Volume data
- This web application would allow researchers not experienced with programming or a command line interface to:
 - Visualize dose-volume data
 - Segment the Regions of Interest (ROIs) using several preset masks
 - Segment the ROIs with customizable masks
 - **Run dose-volume analysis on segmented ROIs**
 - Export dose-volume data for further analysis

Resolved Dependencies

Dependency	Plan to resolve	Resolution Date
Access to database	Pranav and Dr. McNutt are emailing IT	2/13
Access to Existing DVH analysis code	Meet with Pranav	2/13
Access to Pranav's segmentation Code	Meet with Pranav	3/7

Previous Deliverables

Minimum: A UI for cutting and analyzing 3D objects in planes using manual input and existing analysis scripts.

Expected: A UI for cutting and analyzing 3D objects in planes and additional cutting features using manual input. A few additional analysis scripts are integrated into UI, future analysis scripts can be easily added.

Maximum: An interactive UI for cutting and analyzing 3D objects with draggable planes and additional cutting features. UI has additional features for regional analysis, and future analysis scripts can be easily added. Additional feature list can be used for machine learning analysis.

Updated Deliverables

Minimum: A UI for visualizing organs in 3D, calculating DVH curves, and running python analysis scripts from the javascript layer. Documenting our work.

Expected: A UI for segmenting and analyzing organs in 3D using a list of **segmentation options**. Dose-volume data analysis scripts are integrated and can be performed on segments of organs. Results of the analyses can be exported, provide code documentation.

Maximum: An interactive UI for segmenting and analyzing organs in 3D with a flexible **segmentation** as indicated by the **user** in addition to the existing ones. Dose-volume data analysis scripts are integrated and can be performed on segments of organs. An **interactive table** that can summarize the DVH analysis data.

Technical Approach

1. Set up **Python** Web Framework on the **back-end** side
2. Set up the **front end** visualizations
 - a. **Chosen** Library, for the select drop downs
 - b. D3 library for **DVH** curves and object 3D visualization
 - c. Include options, and later on text boxes enabling user to segment data
 - d. **DataTable** containing information plotted in the DVH curves
3. Enabling **communication** between back and front end using XHR requests
4. Enabling the user to **download** the results of the analyses

Back end

- Access to python segmentation and analysis tools
- Web.py for setting up python web framework
- Server runs on local machine

```
urls = (  
    "/", "index",  
    "/get_rois", "get_rois",  
    "/getp", "getp",  
    "/getd", "getd",  
    "/getdvh", "getdvh",  
    "/getvol", "getvol"  
)
```

```
class get_rois:  
    def GET(self):  
        web.header('Access-Control-Allow-Origin', '*')  
        web.header('Access-Control-Allow-Credentials', 'true')  
        return json.dumps(db.regions_of_interest.get_roi_names())
```

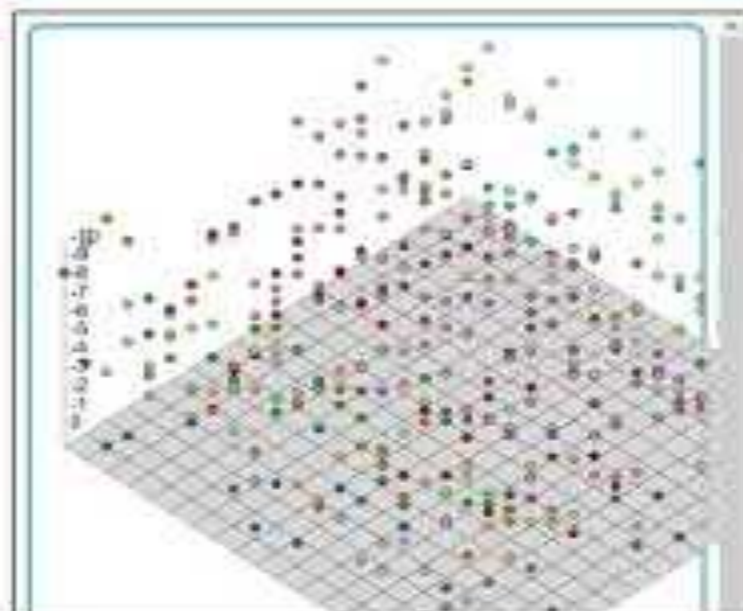

Please select an ROI and a pair for visualization and segmentation

ROI:

Parents:

Doors:

ROI:



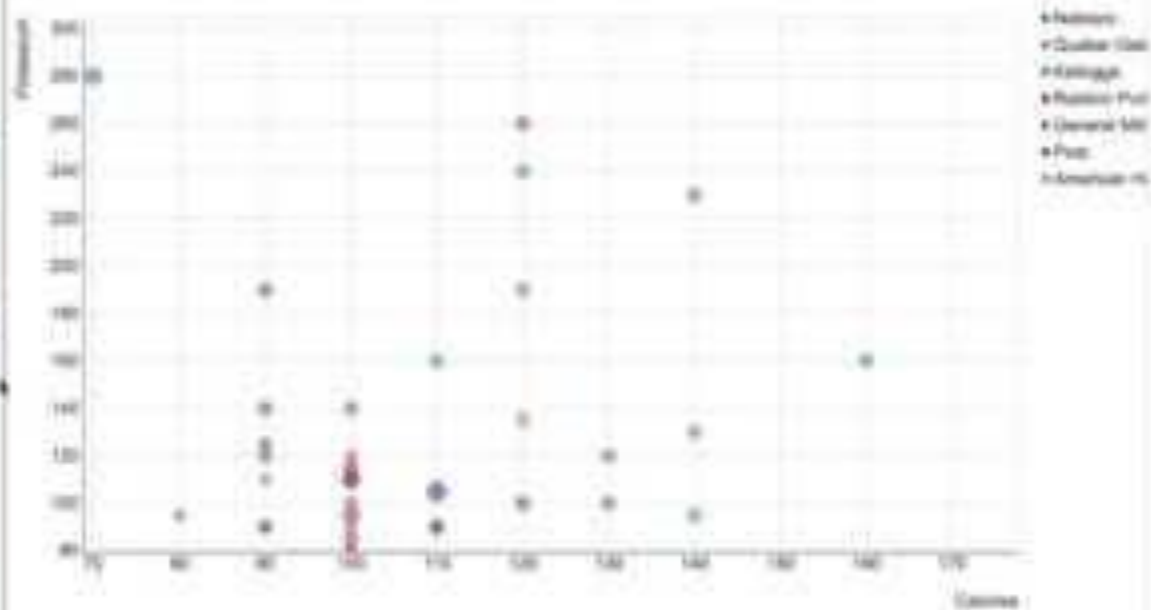
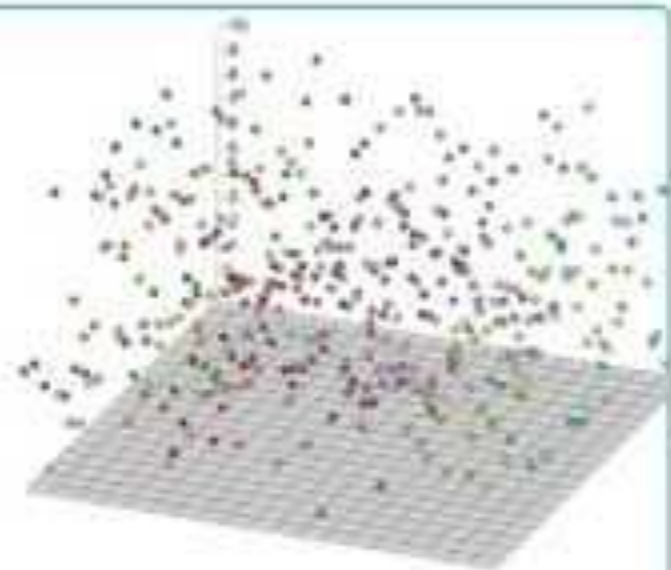
Front end

- We used **Chosen** library for the select boxes, to allow for multiple choices.

Please select an ROI and a patient for visualization and segmentation

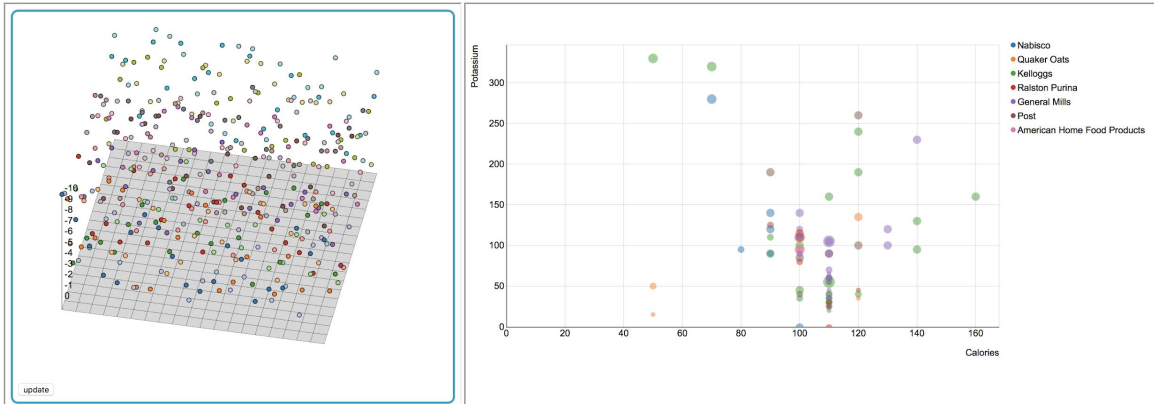
ROI: Patients: Doses:

- To display the organ, we used an interactive 3d object, created by **d3.js** library. It requires d3 4th version.
- To display the DVH curve, we use an interactive scatter plot, also created by d3.js library. It requires d3 3rd version.



A conflict

- **Problem:** Each of our plot used a different version of d3 library. Both are on the same page, and HTML does not allow both version (2nd one overwrites first one).
- **Initial Try?** Make one of the plots adaptable to the other version of d3 code. This did **not work**, as they used different zooming and dragging mechanisms, and for it to be compatible, we needed to recalculate the coordinates.
- **Solution?** Used **iframe**: a nested browsing: embedding another HTML into the current page, created **2 separate** HTML files, 1 for each plot



Merging Front and Back End

- XHR requests from JS front end to specific urls
- For given ROI, return DVH and ROI point cloud data in JSON format for analysis and 3D visualizations

Front End JS

```
var dose = $(selectid3).chosen().val();
var roi = $(selectid4).chosen().val();
var xhr = new XMLHttpRequest();
xhr.open('GET', 'http://127.0.0.1:8080/getdvh?do');
xhr.onreadystatechange = function() {
    if(xhr.status == 200 && xhr.readyState == 4)
    {
        var dvh = JSON.parse(xhr.responseText);
        dvh_data = [dvh.prob, dvh.dose];
    }
}
xhr.send();
```

Back End Python

```
urls = (
    "/", "index",
    "/get_rois", "get_rois",
    "/getp", "getp",
    "/getd", "getd",
    "/getdvh", "getdvh",
    "/getvol", "getvol"
)

class getdvh:
    def GET(self):
        return json.dumps(dvh)
```


Milestones

Milestone	Date - End	Status	Measureable
Familiarity with code and database	2/20	Completed	Run the basic analysis notebooks
Basic website framework	2/27	Completed	See patient ROIs and ID numbers on website
Choose Js Library	2/27	Completed	Successfully visualize sample data
Basic UI with DVH	3/15	Completed	Display DVH for any selected patient anatomy
Exporting Data	3/27	Completed	Download DVH data as .csv file
Interact with 3D visual representation of ROIs	4/1	Ongoing	Segment objects that are compatible with analysis scripts
Integrate user interface with existing tools	4/10	Ongoing	Get matching results to command line analysis
Add table, customizable segmentation tools	4/25	Not Started	Segment objects by moving planes in 3D representation of ROIs, search data with table

Extra Slides

Management Plan

- Weekly **meetings** with Pranav and/or Dr. McNutt
- **Alex** -
 - 3D rendering,
 - JavaScript and Front-end Rendering
 - Data Visualization, Interactivity Management
- **Willie and Santi** -
 - Focus on passing data between JavaScript, Python and SQL;
 - Adding new modules to front end and integrating with back end
 - Back-end Management

Reading List

- Lakshminarayanan, P. (2017). Radio-morphology: Parametric Shape-Based Features in Radiotherapy (Unpublished master's thesis). Johns Hopkins University.
- McNutt, T., PhD., & Lakshminarayanan, P. (2018, February 6). User Interface to Extract radio-morphologic features for refined dose-toxicity analysis in radiotherapy. Lecture presented at CIS II Lecture in Hackerman B17, Baltimore, MD.
- Chen R, Gabriel P, Kavanagh B, McNutt T, “How will big data impact clinical decision making and precision medicine in radiation therapy?” Int’l J. of Radiation Oncology, Biology, Physics. Published online: November 27 2015