**Computer Integrated Surgery Project**

**IMRT Planning**

**Data Driven Strategy For Optimized Intensity-Modulated Radiation Therapy (IMRT) Planning For Head And Neck Region**

**Yang Wuyang M.D. -- DHSI, JHSOM**

**Mentors: Dr. Todd McNutt; Dr. Russell Taylor**

19 May 2011

Final Report

# **Abstract**

This is a detailed description of the course project for “Advanced Computer Integrated Surgery Course”. The project is a clinical based piece of work, its general idea is to provide a standardized workflow and clinical knowledge maintenance tool for support of clinical decision making in Intensity-Modulated Radiation Therapy (IMRT) planning procedure for head and neck region. We focused on a new parameter (Overlapped Volume Histogram, or OVH) that represents general distance from the planning region to the Organs At Risk (OAR), and based on that distance, defined which patient are harder to plan and which are not -- compared to a new patient. At last, we built up a package in python and SQL to search through a large, completely constructed database for parameters in patients that are harder to plan than the new patient, and extract/predict a conservative best dose achievable for the new patient, this result then goes into the original planning system (Pinnacle 3 IMRT planning system) to replace prior manual planning procedure that was based on individual physician experiences. We plan to implement this planning/predicting procedure into the clinical workflow to increase efficiency and effectiveness of IMRT planning.

# **Significance And Motivation**

The motivation in doing this project is it actually greatly supports real world clinical workflow. The current planning workflow takes up time, has little information on standardized solution/practice protocol, and almost no objective (only based on physician experience) knowledge maintenance to provide clinical decision support. Eventually, planners came up with a plan that they need some interventions to improve quality of the planning process, and enhance workflow for more efficiency. Radiation Oncology Department in Hopkins proposed a way to facilitate that thought, which will be discussed in details further on. However, because each part of the facilitation was implemented by different person and was hard to interoperate between each other, the packages, instead of being used in the real world for patient planning, were used for answering research questions and produce publications . Prior efforts were made by others in the department to build up a package in matlab code. The packages can work well, but the data has to be pre-computed and pre-stored for further analyzing. This method would not work in a current clinical setting because all patient data have been moved into the MS SQL Server database in its unprocessed format (will be stored in that way in the future), and also because matlab was not optimal for a simple way to be used by physicians. The significance of this project is that it was designed to put existing unusable raw product into real-world clinical use. It does not only answer research questions, but clinical effectiveness can also be evaluated.

# **Project Accomplishment Overview**

The minimum deliverable of this project was implementing a package built in python, with the data extraction built in SQL. A working package in python that extracts data from MS SQL server, save that in a python list, and computes result within the package in a sequential method was built. Expectedly, the package takes up too much time to compute a single result, and such a delay will impact on the quality of the proposed clinical workflow. After that, our expected deliverable -- optimized package that computes 20-30 times faster was accomplished. The processing of the result of a single patient improves from 1 hour+ to about 5 minutes. We then decided to take a step further by implementing an additional feature -- the Pinnacle 3 executing script generating function. So now the package can directly generate Pinnacle 3 executable scripts based on the input of patient OVH data and output of computed results.

# **Future Work Proposal**

* Unify all packages (Head &Neck, thoracic, others) -- compile all existing packages that serves for different modules of IMRT planning into one package, the physician should be able to access all planning procedures in a single package, and built up a user-friendly for actual use in the clinical workflow.
* Optimize SQL server data structure/built pre-computed data to enhance data searching and computing time.
* Evaluate the efficiency and effectiveness of original workflow and proposed workflow. The efficiency can be evaluated by measuring how much time was saved by the change of workflow; the evaluation of effectiveness is much more complicated, it involves evaluating the easiness of method implementation process, how many physicians are willing to use the package, how does the result reflect toxicity in patients, and how can patient outcomes be improved. And most importantly how much the result is better than manual (original) planning.
* Dynamic clinical decision support and knowledge maintenance – after the package is implemented, feedbacks from patient and physicians will add knowledge to the knowledge base of the package, the method has to be adapted to achieve maximum effectiveness. This phase is very much like the phase 4 in drug development.

# **Background Overview**

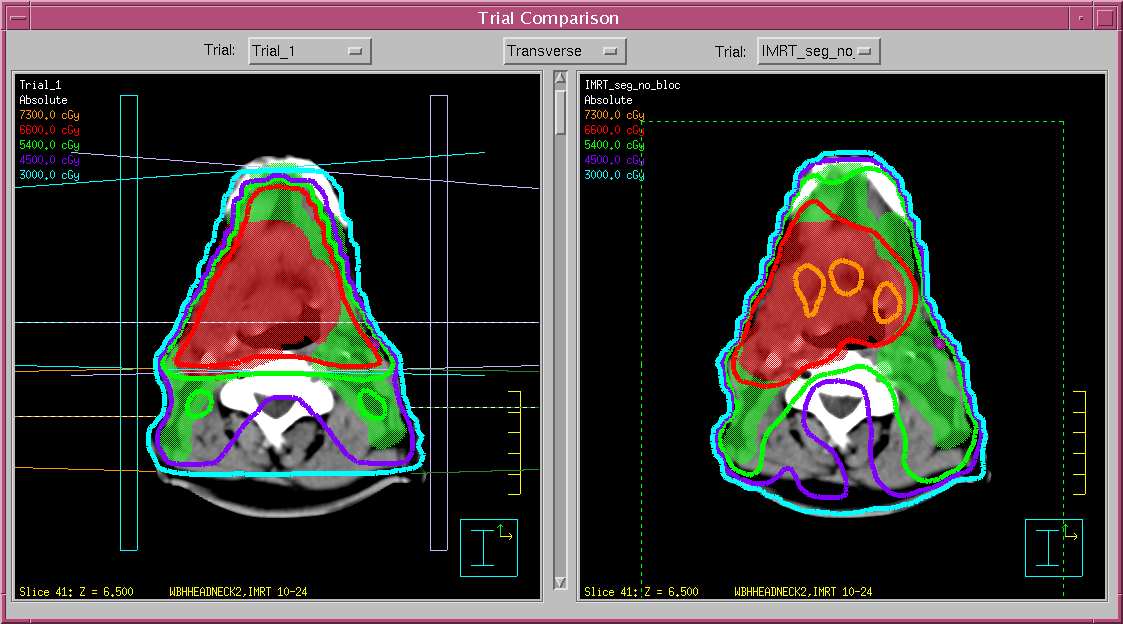
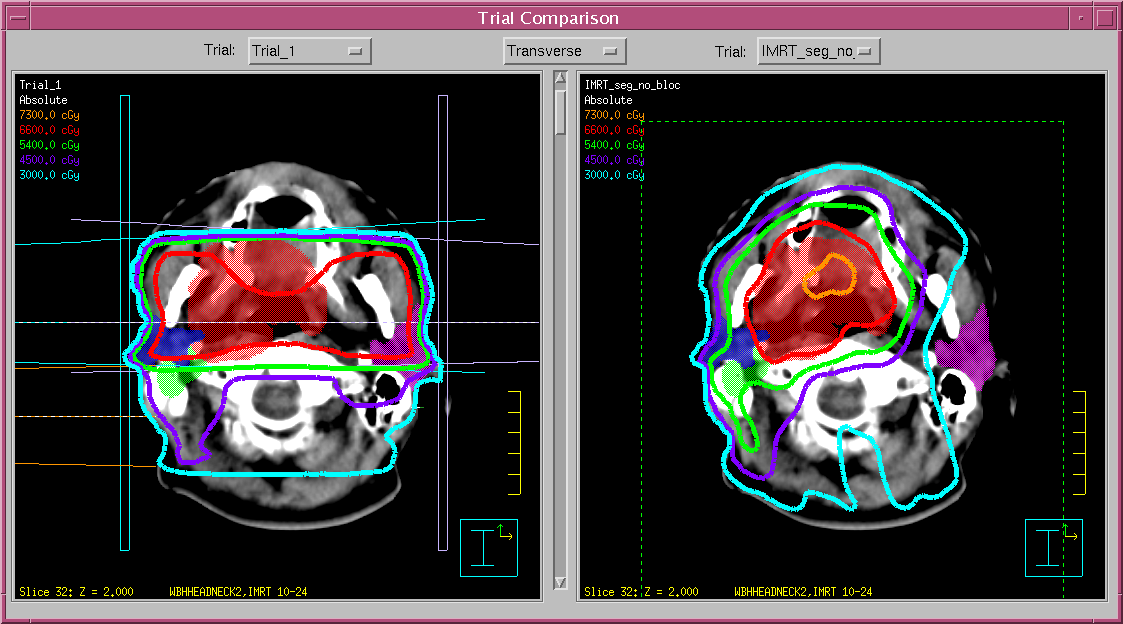
## Intensity-Modulated Radiation Therapy (IMRT)

The concept of radiation therapies is in general using radiation beams to deliver radiation to targeted regions that needs to be treated. Conventional radiation therapy plans include 2-D planning methods and 3-D conformable methods. The improvement in conformability of radiation beam/radiation delivery method enables more coverage of the targeted volume and in the same time spare as much critical structures as possible (or Organs At Risk, OAR). Intensity- Modulated Radiation Therapy, or IMRT, as shown in figure 1, is an advanced form of 3-D conformable radiation therapy, it uses beam grating devices to generate multiple ﬂuctuating radiation beam intensities [1], the added effect of different intensity beams then can be calculated to fit on to preoperative images in order to deliver as much Planned Target Volume (PTV) as possible, and at the same time avoid OARS.

*Figure 1 concept of IMRT*

*Courtesy from Dr. McNutt*

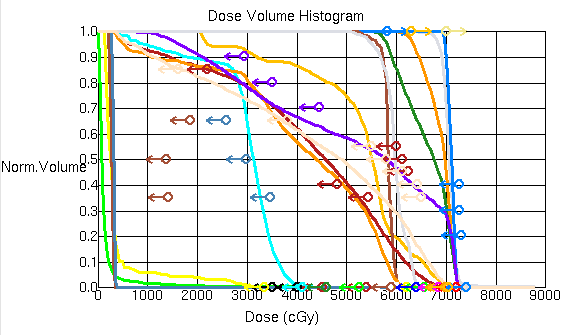
Shown in figure 2, different colors of regions represent an “Iso-dose Region”, which mean that the doses within the same color regions are the same. As the figure shows, the shape of conventional 3-D radiation planning can be conformable to fit on the CT image in order to cover the targets and spare organs, but the conformability is quite poor and as a result, the 66Gy region is covering a lot of the contra-lateral side of the head and neck region where no PTV is existing on that side, and there is no 73Gy dosage level because the dose is too high and will eventually burn the critical structures using conventional methods. In contrast, as shown on the right side of the figure, which is planned using IMRT, the “hot-zone” of the plan is very focused on the target (which in the CT image appears to be on the left side of the image), higher dosage iso-dose regions (73Gy, 66Gy, 54Gy) are converged to the target, and the highest dose level, which is 73Gy, appears on the image to be a very small area, indicating the enabling of high dose planning and focusing. Most of the critical structures are spared by the higher dose levels. For this project the target dose (PTV) would be: 58.1Gy, 63Gy, 70Gy.



*Figure 2 Comparison of Iso Dose Regions. Left: Conventional plan. Right: IMRT plan*

*Courtesy from Dr. McNutt*

## Dose-Volume Histogram (DVH)

Based on the dose of a specific point of volume an organ receives, a histogram with y-axis as volume of the dose an organ receives and x-axis the dose of the volume the organ receives can be drawn to conceptually show how much radiation a specific organ is receiving. As shown in figure 3, the shift of lines to the right generally indicates the organ is receiving a higher dose, and the shift to the left indicates a lower dose received. 

*Figure 3 Dose Volume Histogram (DVH), every color represents a different organ*

The radiation beam has penumbra regions where the radiation intensity of the beam decreases in a gradient way. Therefore, even if the dose level proposed for a certain region is fixed, in a realistic world, the dose distribution within the radiation of the beam is still slightly uneven. In addition to that, the effect of autonomous movement of the patient or other movements of the organ to the radiation distribution can add up to the whole uneven effect. Therefore, even if our target volume is assigned to splined (3 segments) dose levels, we can still plot a continuous radiation/percentage histogram based on the dose of the organs received.

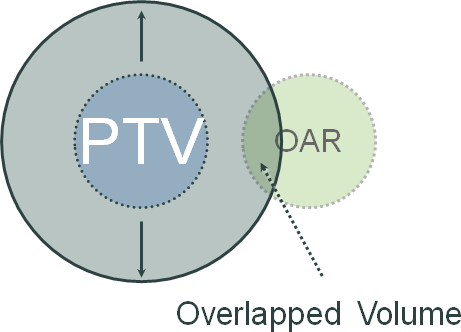
The DVH diagram is a plotting strategy to provide to way of directly viewing the following information:

1. The exact radiation dose received by each organ at each percentage point;
2. A general view of the dose level of the organ received by comparing the general tangential points, general trend of the line, and left/right shift of line;
3. General planning habits of the planner, most planners tend to focus on the 50% point, so most frequently the dose of the same organ of different patients at 50% dose point level tend to converge;
4. General overview of the relationship between tumor and certain organs, some constraining points requirement of certain organs are extremely hard to meet, this is usually because the tumor is very near the critical structure;

DVH value was stored in the “Oncospace” database as a table with a column representing the absolute and accumulative percentage points (y axis), and the respective dose received (x axis).

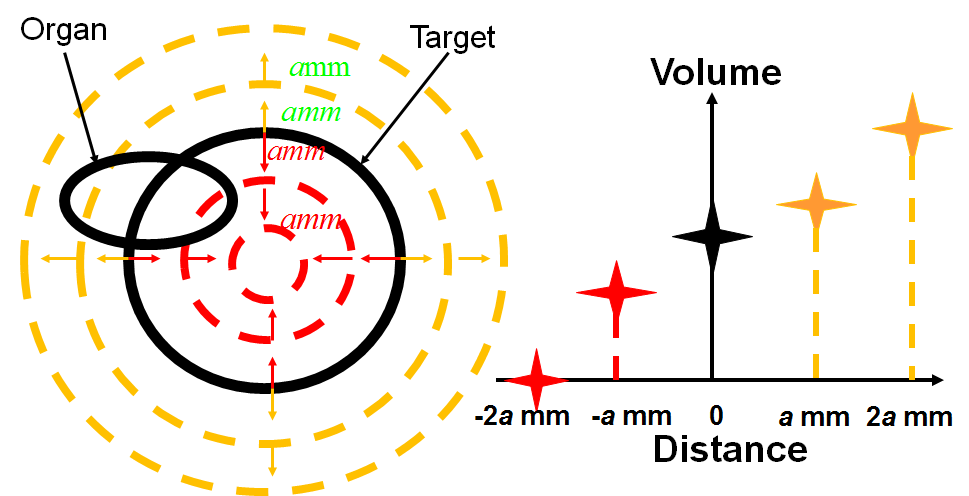
## Overlapped-Volume Histogram (OVH)

Overlapped-Volume Histogram (OVH) means plotting the percentage of the volume of the critical structure (or OAR) covered by the radiation beam by the distance between the OAR and the planned radiation concentrating focus. The general concept of the OVH is shown below:



*Figure 4 Showing the general concept of OVH, as the figure shown, the overlapped volume of the OAR increases as the radius of the PTV enlarges, therefore we can plot a continuous histogram of percentage/distance, the OVH generally represents the distance between PTV and OAR*

With the general concept illustrated, we can now plot a continuous histogram based on that concept, from the original PTV, we either extends the radius of the PTV circle or shrink it. If the OAR is far away from the PTV, the extension needs a greater radius to reach the same percentage of volume of OAR compared to OARs that are nearer to the PTV. The condition that the radius is negative and the OAR is still covered is the situation where the tumor has invaded the OAR and it’s inseparable from the PTV and the OAR, figure 5 illustrates the steps taken to complete a differential OVH (not accumulative distance). The OVH is automatically generated by pinnacle3 after the planner contours the PTV and the OARs. Below is the figure 5 showing how to plot OVH histograms:

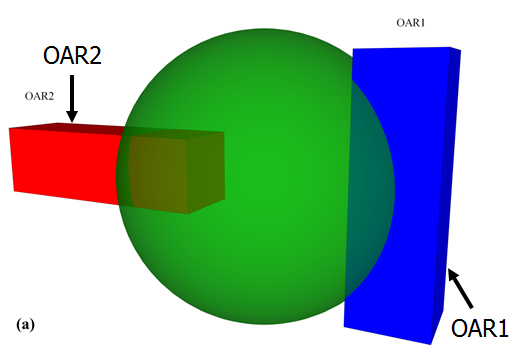
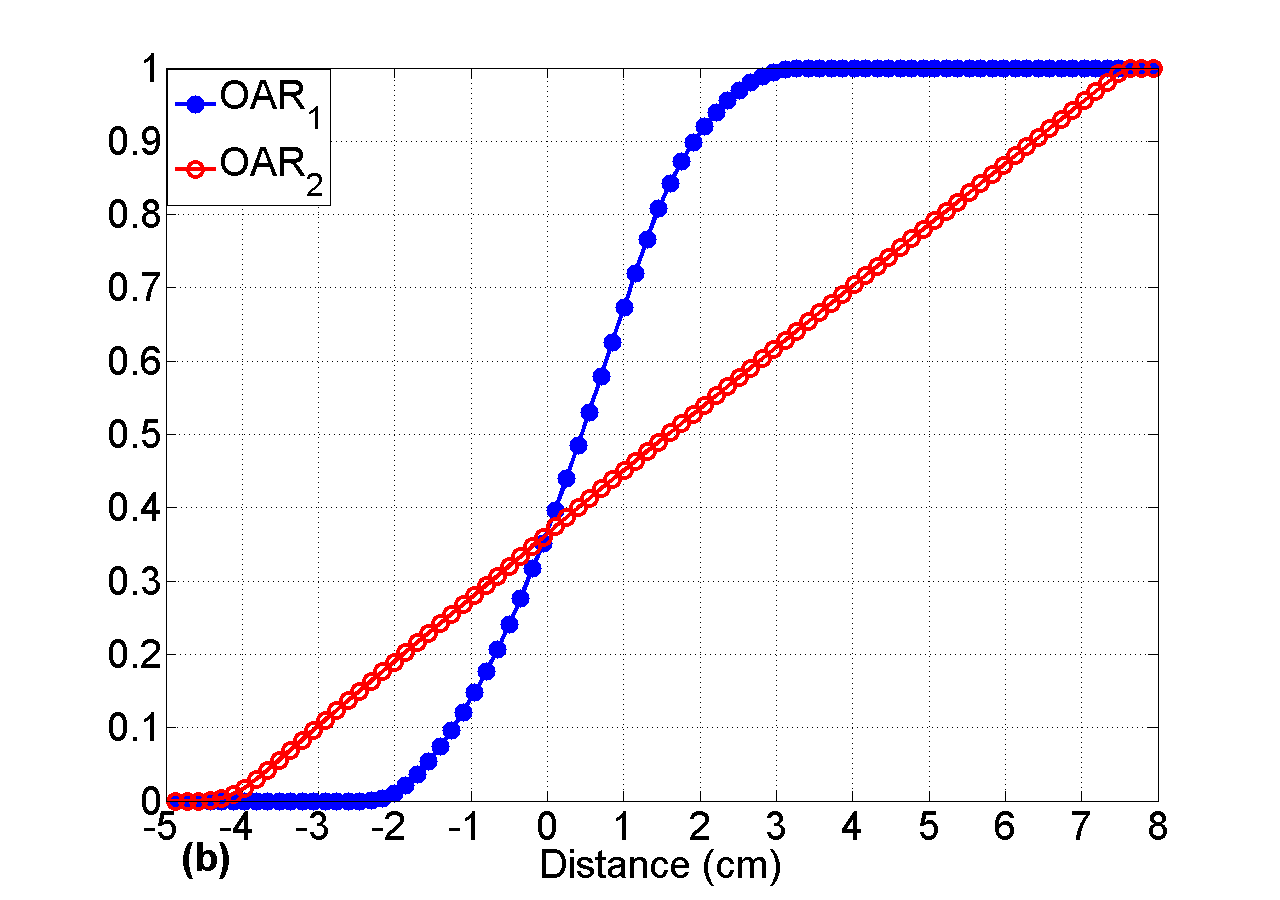


*Figure 5 the steps taken to compute histogram, OVH is calculated automatically in pinnacle3*

*Courtesy by Dr. McNutt*

From the OVH diagram, we can have quick views of a variety of useful information:

1. A general representation of distance between OAR and PTV (as explained above), the distance can roughly be interpreted as “easiness of sparing”, the further away the organ is from the PTV, the easier to plan.
2. A general representation of the shape of OAR. As the radius expands, the increasing rate of volume covered roughly represents the shape of the OAR, see figure 6 below:

*Figure 6 As shown on the left and right, different shapes of organs has a different growing curve in OVH*

*Courtesy by Dr. McNutt*

OVH value was stored in the “Oncospace” database as a table with a column representing the absolute and accumulative percentage points (y axis), and the respective distance expanded (x axis).

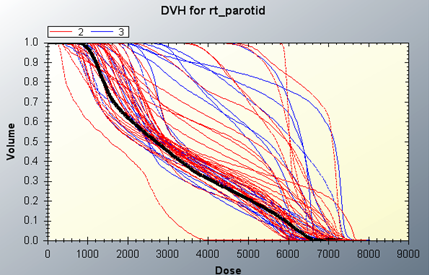
## Current IMRT Planning Clinical WorkFlow

1. Pinnacle3 Planning System

The radiation beam setup and therapy planning software currently used in Johns Hopkins Hospital Radiation Oncology Department is the Pinnacle3 planning software developed by Philips. The planner first load the patient CT, contours target and OARs, stores that into the Pinnacle3 system, the system then asks for constraints of doses at different levels of percentage of volume, the planner input the estimated constraint for each organ based on practice experiences, then returns back to the pinnacle3 planning system to stress on the algorithm to achieve the requirements.

1. Long Planning Span And Associated Consequences

Normally the planning procedure is a multitasking process, the dosimetrists start planning for multiple patients at the same time, put in constraints of doses for each organ at each dose points for each patient, and waits for the system to calculate the results. Each loop costs about 10min – 30min depending on how much pressure is stressed on the pinnacle3 system. Because of the many loops to reach a “thought to be” optimized result, and the time needed for each loop, a patient usually have to wait for two weeks before the plan is finalized. In addition to that, most hard to plan patients don’t get optimized plans because dosimetrists get impatient on the waiting, so what they do is instead of constraining on the whole DVH curve, they releases some pressure on the planning system and only focused on a few percentage points (say 50%), causing the convergence of DVH lines at that point (See figure 7), and patient therapy quality gets compromised.



*Figure 7 The figure shows the DVH for right parotid organ for all patients in the current database, as seen in the figure, the more left shift the curves are, the lower generally the doses are. As observed from the figure, there seems to be a convergence phenomenon at about 50% point. This is due to the reason that most dosimetrists only focus on the 50% dose point to save time. Such a planning behaviour is a typical unintended malpractice, it compromises patient treatment quality and implicitly increases health care costs.*

*Courtesy from Dr. McNutt*

convergence

## Brief Summary On Shortfalls Of Current Planning Procedure

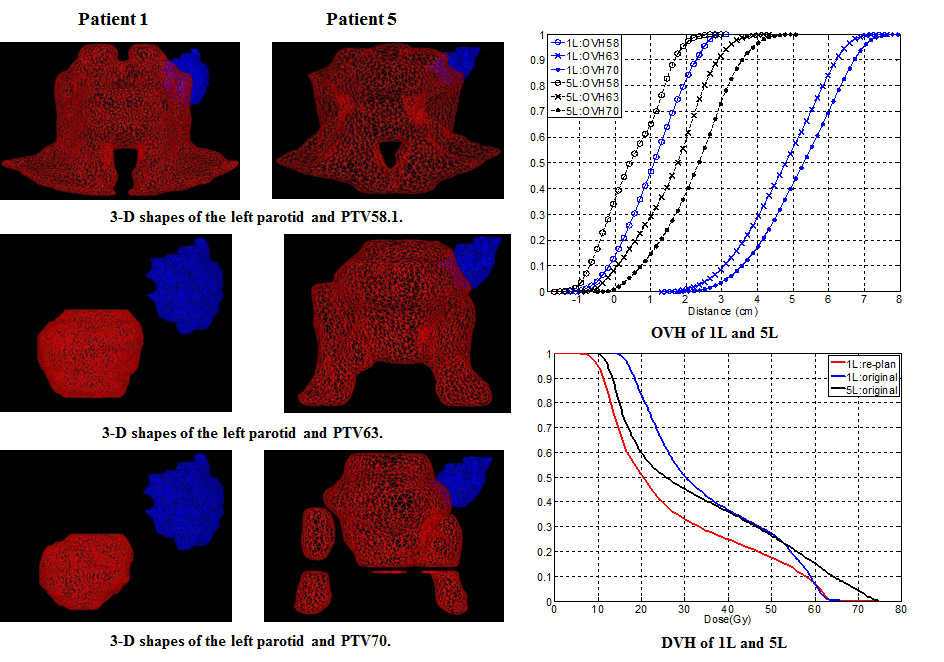
1. Long Planning Procedure – Due to number of loops and time each loop takes
2. Unoptimized Plans – Due to variation of experiences and narrow focuses on constraining points because of impatience in waiting
3. Lack Of Evidence – No reasonable level of evidence on how well a dose level a new patient should achieve
4. Waste of data on past patient plans

# Proposed Solution

## Concept: Link DVH And OVH

As explained above, the DVH and OVH data has a common y axis: the percentage of volume covered. Therefore, we can calculate the distance of the PTV extended and the Dose received at the same percentage of volume of the same organ.

Because the OVH can be roughly interpreted as the “easiness of sparing OAR”, and DVH interpreted as “dose received by OAR”, therefore we use this convenience to utilize a hypothesized causal relationship established as “OAR that are easier to plan receive a lower dose in general” or “OAR that are harder to plan receive a higher dose in general” (see figure 8 for 3D comparison of “harder to plan”).



*Figure 8 Figure showing the relationship of the same organ and PTV between two patients, the parotid on the right is harder to plan (5L, black line), as appears to be a general left shift compared to 1L (blue) in the OVH as shown in the upper right.*

*Courtesy by Dr. Todd McNutt*

The meaning of such a utilization is that we can now take the OVH of a new patient, compare that to the OVHs of patients in the database, find all that are “harder to plan”, take the doses of those patients, find the minimum values in the series of lists, and take those as the predicted value that the new patient can receive. That is the other way of generating a conservative prediction of the “worst case the new patient can achieve”, if the “worst case” is lower than that the pinnacle3 predicted, then it is by logic reasonable that pinnacle3 algorithm can be stressed more by our intended constraints to at least reach the dose defined by the “worst case”. The logic algorithm is shown below (figure 9):

C:\Users\Yang\Documents\Johns Hopkins\Radiation Oncology\Presentation\algorithm.png

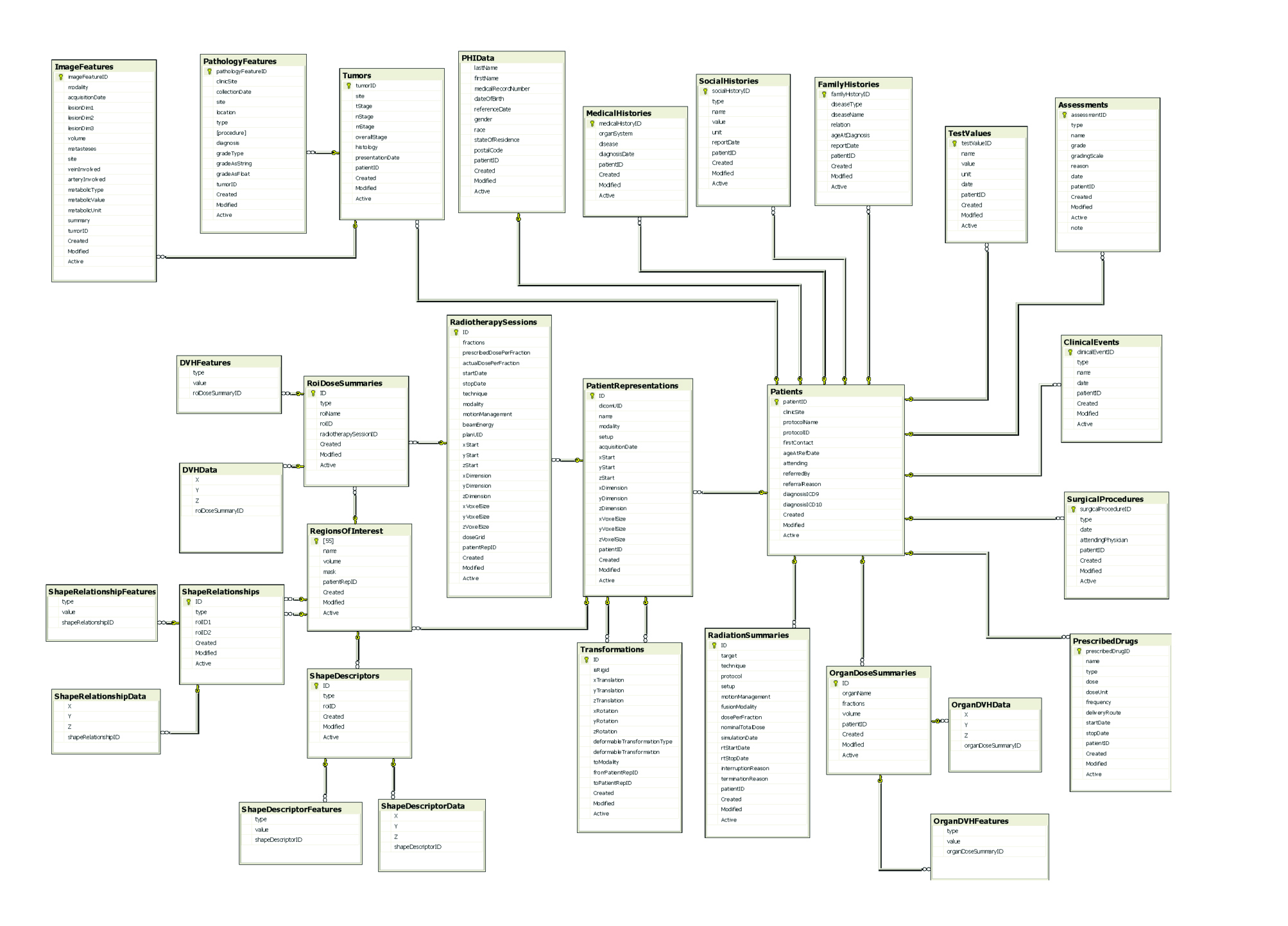
*Figure 9 This algorithm takes the minimum dose from patients that have a smaller distance between OAR and PTV, and and the same time the coverage of PTV is not compromised.*

*Courtesy by Dr. Binbin Wu and Dr. Todd McNutt*

This algorithm is the basic logic we use for searching through the database for a conservative prediction of the dose the new patient can possibly receive.

## Database Structure

In order to implement the concept, a database has to be pre-constructed to store all patient relevant data and OVH/DVH data. The radiation oncology department has already built such a database, see figure 10 for database schema:

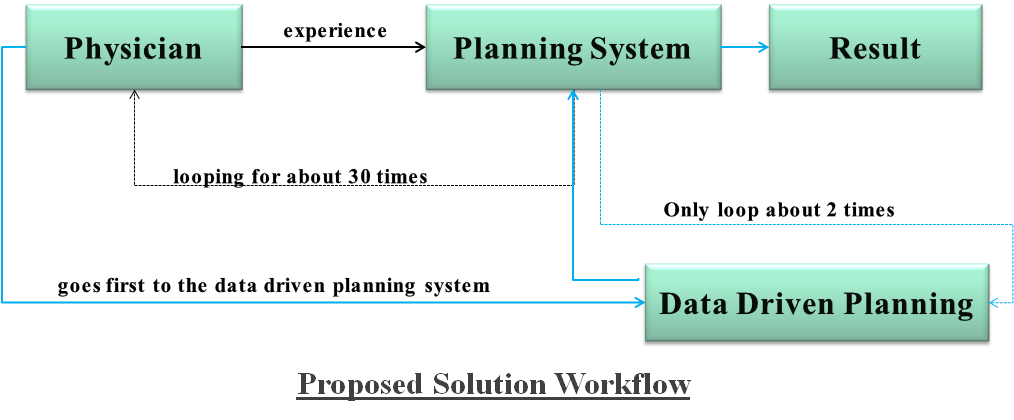


*Figure 10 Showing the whole schema of the database, the selected part is the needed part for this project, containing patient representation data, DVH and OVH data.*

*Courtesy by Dr. Todd McNutt*

## Proposed Solution Workflow

Changing of workflow is critical for efficiency of planning, the proposed workflow is shown below (figure 11):



*Figure 11 Showing the modified workflow, supposed elevate efficiency*

The general idea of this workflow is we inserted a data-driven planning procedure between the actual pinnacle3 planning and physician, so the initialization of every planning is based on our data-driven planning procedure instead of physician experience. We think that this strategy shall improve the quality of planning because there is transparent evidence in how we generate the initialization data, and the result shall be reasonable in that it explicitly expresses the past treating experience of the planners. In addition to that, because the result is reasonable, and it represents the experience of past planners, it shall also be time-saving because the planners do not have to tweak too much on the constraints to reach a dose point.

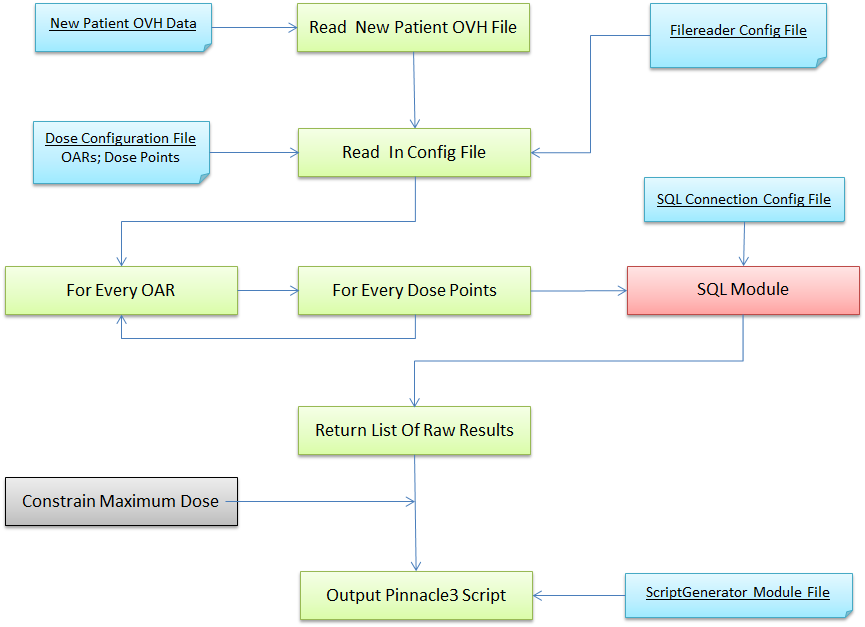
## Implementation – Package Built In Python And SQL

The implementation process for Data-Driven Planning is developing a package for the department, so before going to the step of pinnacle3 planning, the physician goes to the package first to generate results for initial prediction. The package then searches the database to calculate reasonable results. Former efforts have been made to implement this idea. Dr. Binbin Wu from the department developed a Head and Neck planning package in matlab. He used precomputed data of OVH and DVH, store them into matrices to serve as a database, and compute the results. The result comes out good, but the workflow using matlab and precomputed OVH and DVH to generate result is not effective because it is too “complicated” for the planners to use. In addition, the database is built in a way that the matlab code is incompatible with it. Packages for the other regions, such as pancreas and thoracic regions, are all built in python. To make the whole project usable in clinic, it is necessary to implement the workflow in python instead of matlab.

# Approach

## Program Structure

The program UML is shown below:



## SQL Module:

The SQL Code For Query is shown below:

**params = targetName1, dose1, percent, targetName1, oarName, point, r\_point1, targetName2, dose2, percent,targetName2, oarName, point, r\_point2, point**

**cursor.execute("""**

**SELECT DISTINCT MIN(dvh.X) OVER(PARTITION BY dvh.roiDoseSummaryID) as dose**

**FROM DVHData AS dvh**

**INNER JOIN (**

**SELECT ds.type, ds.ID**

**FROM RoiDoseSummaries AS ds**

**INNER JOIN (**

**SELECT ID1.ID**

**FROM (SELECT DISTINCT roi2.ID, MIN(srf.X) OVER(PARTITION BY roi2.ID) as X\_high1**

**FROM ShapeRelationshipData AS srf**

**INNER JOIN ShapeRelationships AS sr**

**ON srf.shapeRelationshipID=sr.ID**

**INNER JOIN (**

**SELECT rds1.roiID AS 'ID', rds1.roiName AS 'name'**

**FROM roiDoseSummaries rds1**

**INNER JOIN DVHData dvh1**

**ON dvh1.roiDoseSummaryID = rds1.ID**

**Constraining Coverage Of PTV1**

**WHERE rds1.roiName = ?**

**AND dvh1.x >= ?**

**AND dvh1.y >= ?**

**AND rds1.type = 'Cumulative DVH, Norm Volume'**

**) AS roi1 ON sr.roiID1=roi1.ID**

**INNER JOIN RegionsOfInterest AS roi2**

**ON sr.roiID2=roi2.ID**

**WHERE roi1.name= ?**

**AND roi2.name= ?**

**AND type = 'Cumulative OVH, Norm Volume'**

**AND srf.Y > ?**

**Harder To Plan For PTV1**

**AND srf.X < ?**

**GROUP BY roi2.ID, srf.X, srf.Y) AS ID1**

**INNER JOIN(**

**SELECT DISTINCT roi2.ID, MIN(srf.X) OVER(PARTITION BY roi2.ID) as X\_high1**

**FROM ShapeRelationshipData AS srf**

**INNER JOIN ShapeRelationships AS sr**

**ON srf.shapeRelationshipID=sr.ID**

**INNER JOIN (**

**SELECT rds2.roiID AS 'ID', rds2.roiName AS 'name'**

**FROM roiDoseSummaries rds2**

**INNER JOIN DVHData dvh1**

**ON dvh1.roiDoseSummaryID = rds2.ID**

**Constraining Coverage Of PTV2**

**WHERE rds2.roiName = ?**

**AND dvh1.x >= ?**

**AND dvh1.y >= ?**

**AND rds2.type = 'Cumulative DVH, Norm Volume'**

**) AS roi1 ON sr.roiID1=roi1.ID**

**INNER JOIN RegionsOfInterest AS roi2**

**ON sr.roiID2=roi2.ID**

**WHERE roi1.name= ?**

**AND roi2.name= ?**

**AND type = 'Cumulative OVH, Norm Volume'**

**Harder To Plan For PTV2**

**AND srf.Y > ?**

**AND srf.X < ?**

**GROUP BY roi2.ID, srf.X, srf.Y)**

**AS ID2 ON ID1.ID = ID2.ID**

**) AS x ON ds.roiID=x.ID**

**) AS d ON dvh.roiDoseSummaryID=d.ID**

**WHERE d.type = 'Cumulative DVH, Norm Volume'**

**Constraining Dose Percentage**

**and dvh.Y <= ?**

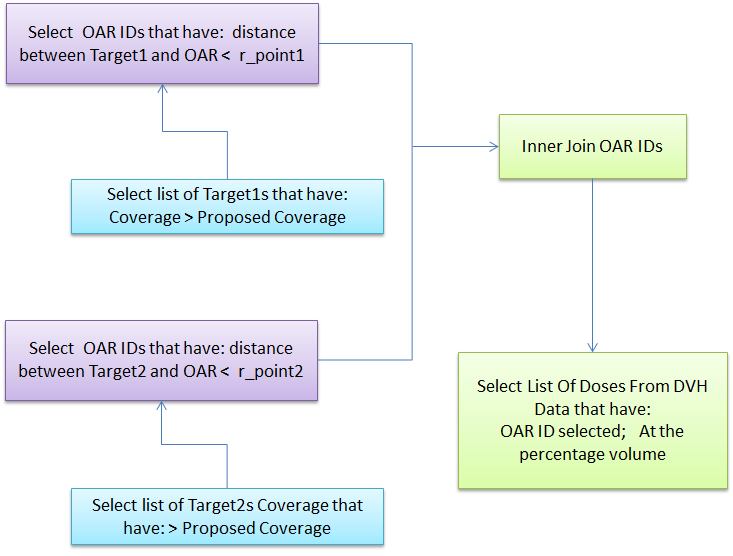
**GROUP BY dvh.X, dvh.roiDoseSummaryID**

**ORDER BY dose**

**""", params)**

For each question mark in SQL, a representing variable in “params” will be fit into. The ‘targetName 1’ means the name of ptv1; ‘dose 1’ is the dose of ptv1, ‘percent’ is the coverage of the volume, ‘oarName’ is the name of Organ At Risk, ‘r\_point1’ is the distance between OAR and ptv1, ‘targetName 2’ means the name of ptv2; ‘dose 2’ is the dose of ptv2; ‘r\_point2’ is the distance between OAR and ptv2.

Summary of SQL query process:



## Pinnacle3 Script Module:

For each OAR and each percentage point, we need to generate a Pinnacle3 executable block, the block has to have the descriptions of the OAR, the dose it should receive, and some basic beam setups. Here’s an example of some of the blocks:

Store.FloatAt.ZeroTest = 0.0;

PluginManager.InversePlanningManager.AddObjective="Add Objective";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.ROIName="lt parotid";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#25"="Max DVH";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="4139.33007812";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.UserPercent="35.0";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0.01";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager .InversePlanningManager .CombinedObjectiveList .Last .ROIName = "PTV70\_eval";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#25"="Max Dose";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="7350";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0";

Store.FreeAt.ZeroTest="";

Store.FloatAt.ZeroTest = 0.0;

PluginManager.InversePlanningManager.AddObjective="Add Objective";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.ROIName="lt parotid";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#26"="Max DVH";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="3675.11010742";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.UserPercent="40.0";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0.01";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager .InversePlanningManager .CombinedObjectiveList .Last .ROIName = "PTV70\_eval";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#26"="Max Dose";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="7350";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0";

Store.FreeAt.ZeroTest="";

Store.FloatAt.ZeroTest = 0.0;

PluginManager.InversePlanningManager.AddObjective="Add Objective";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.ROIName="lt parotid";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#27"="Max DVH";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="4173.68017578";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.UserPercent="45.0";

IF.RoiList.#"lt parotid".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0.01";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager .InversePlanningManager .CombinedObjectiveList .Last .ROIName = "PTV70\_eval";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#27"="Max Dose";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="7350";

IF.RoiList.#"lt parotid".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0";

Store.FreeAt.ZeroTest="";

The red circle is the name of the OAR, the blue circle is the initial prediction of the dose which we calculated from, and the brown circle means the weight of such a constraint (the weight is predefined in the python package).

# Results

## Target Contours Successfully Loaded:

After we input the pinnacle scripts generated from the package, we loaded all the constraints to the system. The system first defines the contour of targets for the new patient, the contours were manually drawn before. Figure 12 shows the contours of PTVs for the patients:

**C:\Users\Yang\Documents\Johns Hopkins\Radiation Oncology\Presentation\2x3_2d_targets_colorwash.tif**

*Figure 12 The Green Contour represents PTV58.1; Brown Contour represents PTV63; Purple Contour represents PTV 70*

As we can see from the pictures, each section shows the coverage of different PTVs. The purple part, which is the PTV 70, generally represents the highest dose level received (70Gy), so it is also another way of saying where the tumor was in this patient.

## Beams Successfully Setup:

After target contours were defined, we now setup the beams and the intensity modules of each direction of beams to meet the requirements of each constraint. The Pinnacle3 system automatically computes a suggested continuous dose with the consideration of each dose level the planner wants to constrain. The intensity of the beams has to be modulated to meet the requirements. Figure 13 and Figure 14 shows the simulation of the beams in which direction should the beam radiate and in how much amount of radiation should the beam release. We can view iso-dose lines in figure 15, as we can see that the high doses are trying to avoid the critical structures and in the same time ensures PTV coverage. For this patient particularly, because the tumor has invaded the right parotid, so the right parotid is very hard to spare, but the dose requirements of other critical structures are basically met.

C:\Users\Yang\Documents\Johns Hopkins\Radiation Oncology\Presentation\3d_fields.tifC:\Users\Yang\Documents\Johns Hopkins\Radiation Oncology\Presentation\3d_fields_2.tif

*Figure 13 and 14, 3D simulation image on how the beams should radiate*

C:\Users\Yang\Documents\Johns Hopkins\Radiation Oncology\Presentation\Isodose_Results.tif

*Figure 15, Iso-dose lines of the first round computation. The result is reasonable as most of the PTVs have a satisfactory coverage, and OARs are very much spared. The red line is the highest dose line, as one can see, it is trying very hard to cover the tumor which is at the right parotid gland, but also stressing to spare as much as possible on esophagus and other OARs. The red arrow indicates where the tumor was, and the yellow arrow points at a dent in 70Gy isodose line to spare some of the esophagus.*

## DVH ResultC:\Users\Yang\Documents\Johns Hopkins\Radiation Oncology\Presentation\DVH_Results.tif

*Figure 16, DVH Result*

As shown in figure 16, the DVH result shows a continuous dose curve for the organs (in this project, there are 13 organs, brain, brainstem, cord, mandible, right parotid, left parotid, right inner ear, left inner ear, larynx, oral mucosa and esophagus, please see appendices for dose point configurations) involved in the project. The arrows indicate a constraining objective defined by the planner which was read into the python package. Almost all objectives are met, except for the right parotid, which was invaded by the tumor. Compromise of the PTVs are very low, coverage objectives are basically met.

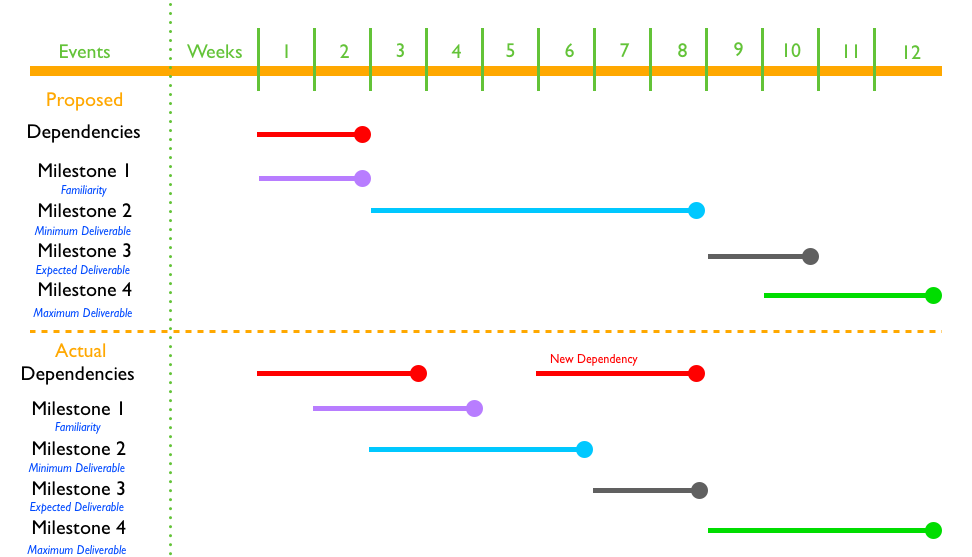
## Planning Time:

The package running time is 5 minutes, and a single round of Pinnacle3 running time is about 20 minutes. As the results is reasonably good enough, we can anticipate the general planning time to be greatly lesser than 2 weeks, although the actual planning time/clinical efficiency needs the package to be widely used in clinics and evaluate the general time used for planning.

# Project Management Summary

## Project Management And Timeline:

The project was solely finished by myself, although Dr. Patricio and Dr. Binbin Wu helped out in the initialization and the matlab code translation part. The proposed timeline and actual timeline can be seen below:



Original And Adjusted Timelines And Deliverables

## Prospective Views:

* Currently, the ”harder to plan” is defined as “harder to plan in PTV58.1” And “harder to plan in PTV63”, which is a very conservative solution, further work can be done to specify this uncertain area, a much more detailed rule can be set up to define which is harder to plan and which is not.
* As defined in clinical epidemiology, clinical effectiveness means the generalization of result or method. The efficiency of this method is already discussed in Dr. Binbin Wu’s paper. Effectiveness, however, is not yet addressed. A future step to take this study further is to introduce patient outcome measures into the database and do a outcome assessment of the result. In addition to the outcome assessment, clinical implementation can also be evaluated.
* The database can serve as a dynamic decision support solution for IMRT planning, further research can be done to implement decision support ideas into the region.

## Lessons Learnt:

* Information and hardware/software integration is the future of automated clinical/surgical medicine
* Being conservative is better than being aggressive in the beginning of designing a product, especially a product related to medicine/surgery
* Project management is a very important ability that we didn’t put a lot of focus before, the way to present to people and the way we criticize about other peoples’ presentation was indeed helpful
* A project is easy to start, hard to maintain, and even harder to finish
* Documentation is very important

# Appendices

## Code

1. Main Driver Program:

'''

Yang Wuyang

Todd McNutt

Division of Health Science Informatics

Department of Radiation Oncology and Molecular Radiation Sciences

The Johns Hopkins University School of Medicine

March 2011

'''

import sys

import os

import subprocess

import pyodbc

import datetime

import fileargsreader

import scriptgenerator

import re

# Load params for SQL DB connection

dbConnectArgs = fileargsreader.load('sqldbconnection.config')

# define connection string

connString = "DRIVER={{SQL Server}};SERVER={0};DATABASE={1};UID={2};PWD={3}".format(dbConnectArgs['host'], dbConnectArgs['database'], dbConnectArgs['user'], dbConnectArgs['password'])

# ------------------------------------------------------------------------------

def getDistance(filename, point):

"""Read OVH files and return the distance"""

#list y is the list of overlapped volume; list x is the list of distance

y = []

x = []

#incremental variables

m = 0

p = 0

j = 0

distance = 0

increment = 30.0/199.0

#read lines in file(overlapped volume), store them in list y

f = open(filename,'r')

for line in f:

s = float(line)

m = m + s

y.append(m)

total = y[199]

#calculate cumulative OVH

for i in y:

y[p] = i/total

p += 1

#calculate distance

while j <= 199:

distance = -10 + j\*increment

x.append(distance)

j += 1

#find in list y the max value that is less than point

k = 0

while k <=199 and y[k] <= point:

k += 1

#return corresponding distance

return x[k]

# ------------------------------------------------------------------------------

def achievableDose(targetName1, targetName2, oarName, point, r\_point1, r\_point2):

"""The actual sql querying code for finding list of minimum doses, first constrain

V95 to percent1 for targetName1, select from OVH that are closer than r\_point1 for

targetName1 and oarName; then constrain V95 to percent2 for targetName2, select from

OVH that are closer than r\_point2 for targetName2 and oarName; inner join these

two tables on target IDs, and get list of minimum doses based on the point in DVHData

table"""

# Establish DB connection

conn = pyodbc.connect(connString)

cursor = conn.cursor()

# Find doses of V95

doses = [5500,5985,6650]

targets = ['ptv58.1\_eval','ptv63\_eval','ptv70\_eval']

# Constrain V95 to percentage points

if oarName == 'inner ear' or 'cord4mm' or 'brainstem' or 'brain' or 'oral mucosa':

percent = 0.99

elif oarName == 'rt\_parotid' or 'lt\_parotid' or 'larynx for edema' or 'esophagus':

percent = 0.98

elif oarName == 'rt\_brachial plexus' or 'lt\_brachial plexus':

percent = 0.975

elif oarName == 'mandible':

percent = 0.995

# Get V95 doses according to target names

dose1 = doses[targets.index(targetName1)]

dose2 = doses[targets.index(targetName2)]

params = targetName1, dose1, percent, targetName1, oarName, point, r\_point1, targetName2, dose2, percent,targetName2, oarName, point, r\_point2, point

cursor.execute("""

SELECT DISTINCT MIN(dvh.X) OVER(PARTITION BY dvh.roiDoseSummaryID) as dose

FROM DVHData AS dvh

INNER JOIN (

SELECT ds.type, ds.ID

FROM RoiDoseSummaries AS ds

INNER JOIN (

SELECT ID1.ID

FROM (SELECT DISTINCT roi2.ID, MIN(srf.X) OVER(PARTITION BY roi2.ID) as X\_high1

FROM ShapeRelationshipData AS srf

INNER JOIN ShapeRelationships AS sr

ON srf.shapeRelationshipID=sr.ID

INNER JOIN (

SELECT rds1.roiID AS 'ID', rds1.roiName AS 'name'

FROM roiDoseSummaries rds1

INNER JOIN DVHData dvh1

ON dvh1.roiDoseSummaryID = rds1.ID

WHERE rds1.roiName = ?

AND dvh1.x >= ?

AND dvh1.y >= ?

AND rds1.type = 'Cumulative DVH, Norm Volume'

) AS roi1 ON sr.roiID1=roi1.ID

INNER JOIN RegionsOfInterest AS roi2

ON sr.roiID2=roi2.ID

WHERE roi1.name= ?

AND roi2.name= ?

AND type = 'Cumulative OVH, Norm Volume'

AND srf.Y > ?

AND srf.X < ?

GROUP BY roi2.ID, srf.X, srf.Y) AS ID1

INNER JOIN(

SELECT DISTINCT roi2.ID, MIN(srf.X) OVER(PARTITION BY roi2.ID) as X\_high1

FROM ShapeRelationshipData AS srf

INNER JOIN ShapeRelationships AS sr

ON srf.shapeRelationshipID=sr.ID

INNER JOIN (

SELECT rds2.roiID AS 'ID', rds2.roiName AS 'name'

FROM roiDoseSummaries rds2

INNER JOIN DVHData dvh1

ON dvh1.roiDoseSummaryID = rds2.ID

WHERE rds2.roiName = ?

AND dvh1.x >= ?

AND dvh1.y >= ?

AND rds2.type = 'Cumulative DVH, Norm Volume'

) AS roi1 ON sr.roiID1=roi1.ID

INNER JOIN RegionsOfInterest AS roi2

ON sr.roiID2=roi2.ID

WHERE roi1.name= ?

AND roi2.name= ?

AND type = 'Cumulative OVH, Norm Volume'

AND srf.Y > ?

AND srf.X < ?

GROUP BY roi2.ID, srf.X, srf.Y)

AS ID2 ON ID1.ID = ID2.ID

) AS x ON ds.roiID=x.ID

) AS d ON dvh.roiDoseSummaryID=d.ID

WHERE d.type = 'Cumulative DVH, Norm Volume'

and dvh.Y <= ?

GROUP BY dvh.X, dvh.roiDoseSummaryID

ORDER BY dose

""", params)

rows = cursor.fetchall()

cursor.close()

conn.close()

return rows

# ------------------------------------------------------------------------------

def minDose(targetName1, targetName2, oarName, point, r\_point1, r\_point2):

"""=Find minimum dose in the list returned from the achievableDose() function"""

t = achievableDose(targetName1, targetName2, oarName, point, r\_point1, r\_point2)

m = []

i = 0

while i < len(t):

m.insert(i,t[i][0])

i = i+1

if min(m)\*5 < max(m):

m.remove(min(m))

if m:

return min(m)

else:

return 0

# ------------------------------------------------------------------------------

def setTarget(oarName):

""" This function constraints the target

Rules:

1. databasePatients--OVH(target#1 to OAR) < newPatient--OVH(target#1 to OAR)

2. databasePatients--OVH(target#2 to OAR) < newPatient--OVH(target#2 to OAR)

"""

target = [58.1,63]

if oarName == 'mandible':

target = [63,70]

elif oarName == 'rt\_brachial\_plexus' or oarName == 'lt\_brachial\_plexus':

target = [58.1,70]

elif oarName == 'esophagus':

target = [58.1,70]

return target

# ------------------------------------------------------------------------------

def checkDose(oarName, mDose):

""" This function contraints the maximum dose recieved by each organ """

if oarName == 'brain':

if mDose > 5700:

mDose = 5700

elif oarName == 'cord4mm':

if mDose > 4100:

mDose = 4100

elif oarName == 'brainstem':

if mDose > 5100:

mDose = 5100

elif oarName == 'mandible':

if mDose >= 7050:

mDose = 7050

elif oarName == 'rt\_brachial\_plexus' or oarName == 'lt\_brachial\_plexus':

if mDose >= 5900:

mDose = 5900

elif oarName == 'esophagus':

if mDose >= 7050:

mDose = 7050

return mDose

# ------------------------------------------------------------------------------

def changeName(oarName):

""" This function changes the oarName to a readable format in script """

name = oarName

if oarName == 'rt\_parotid':

name = 'rt parotid'

elif oarName == 'lt\_parotid':

name = 'lt parotid'

elif oarName == 'rt\_brachial\_plexus':

name = 'rt brachial plexus'

elif oarName == 'lt\_brachial\_plexus':

name = 'lt brachial plexus'

elif oarName == 'rt\_inner\_ear':

name = 'rt inner ear'

elif oarName == 'lt\_inner\_ear':

name = 'lt inner ear'

elif oarName == 'larynx\_for\_edema':

name = 'larynx for edema'

elif oarName == 'oral\_mucosa':

name = 'oral mucosa'

return name

# ------------------------------------------------------------------------------

# MAIN

# ------------------------------------------------------------------------------

"""Main driver component"""

#print to file

f = open('SetupPatientSpecificObjectives.txt','w')

#load dosepoints from configuration file

dosepoints = fileargsreader.load('head\_and\_neck.config')

number = 25

for oarName in dosepoints.keys():

targets = setTarget(oarName)

#Apply target constraint rules

targetName1 = 'ptv' + str(targets[0]) + '\_eval'

targetName2 = 'ptv' + str(targets[1]) + '\_eval'

for percent in eval(dosepoints[oarName]):

#For every percent in the config file

point = percent/100.0

filenameDist1 = targetName1 + '-' + oarName

filenameDist2 = targetName2 + '-' + oarName

path1 = 'OVH/' + filenameDist1 + '.OVH'

path2 = 'OVH/' + filenameDist2 + '.OVH'

r1 = getDistance(path1, point)

r2 = getDistance(path2, point)

#Get minimum calculated dose

mDose = minDose(targetName1, targetName2, oarName, point, r1, r2)

#Check if minimum calculated dose exceeds max acceptable dose

m = checkDose(oarName, mDose)

if mDose == m:

#Normal constrain of weight of result if mDose has not been reset

weight = scriptgenerator.setWeight(oarName, percent)

elif mDose != m:

#Forcing constrain of weight if mDose was reset

weight = scriptgenerator.forceSetWeight(oarName, percent)

print 'Current OAR: ', oarName

changedOarName = changeName(oarName)

if m != 0:

#Modify print to file strings

t = scriptgenerator.modifyString(changedOarName, 100\*point, weight, m, number)

print 'Point: D(%d) Minimum Dose: %.2f' %(100\*point, m)

elif m == 0 :

t = scriptgenerator.modifyString(changedOarName, 100\*point, weight, checkDose(oarName, 10000), number)

print 'Point: D(%d) Minimum Dose: %.2f' %(100\*point, checkDose(oarName, 10000))

for s in t:

print >>f, s

print >>f

number = number + 1

print

1. Scriptgenerator:

# Module to printout the script

#

# Written by:

# Yang Wuyang

# Division Of Health Science Informatics

# Johns Hopkins University School Of Medicine

# 29 April 2011

# ------------------------------------------------------------------------------

def setWeight(oarName, point):

# Constrains weight of beam setup for each organ if doses are normal

weight = 'No Weight'

if oarName == 'brain':

weight = '0.5'

elif oarName == 'cord4mm':

weight = '0.5'

elif oarName == 'brainstem':

weight = '0.5'

elif oarName == 'mandible':

weight = '0.05'

elif oarName == 'rt\_brachial\_plexus' or oarName == 'lt\_brachial\_plexus':

weight = '0.05'

elif oarName == 'esophagus':

weight = '0.05'

elif oarName == 'larynx\_for\_edema':

if point == 0 or point == 20 or point == 30 or point == 40:

weight = '0.05'

elif point == 85:

weight = '0.1'

if oarName == 'lt\_inner\_ear' or oarName == 'rt\_inner\_ear' :

weight = '0.01'

elif oarName == 'lt\_parotid' or oarName == 'rt\_parotid':

if point == 35 or point == 40 or point == 45 or point == 55:

weight = '0.01'

elif point == 50:

weight = '0.05'

elif point == 65 or point == 85:

weight = '0.1'

elif oarName == 'oral\_mucosa':

if point == 70 or point == 80:

weight = '0.01'

elif point == 90:

weight = '0.1'

return weight

# ------------------------------------------------------------------------------

def forceSetWeight(oarName, point):

# Constrains weight of beam setup for each organ if doses exceeds normal

weight = 'No Weight'

if oarName == 'brain':

weight = '1'

elif oarName == 'cord4mm':

weight = '1'

elif oarName == 'brainstem':

weight = '1'

elif oarName == 'mandible':

weight = '0.1'

elif oarName == 'rt\_brachial\_plexus' or oarName == 'lt\_brachial\_plexus':

weight = '0.05'

elif oarName == 'esophagus':

weight = '0.05'

return weight

# ------------------------------------------------------------------------------

def modifyString(oarName, point, weight, minimumDose,number):

# Modify and store the strings to a list, return the list

t = []

t.append('Store.FloatAt.ZeroTest = 0.0;')

t.append('PluginManager.InversePlanningManager.AddObjective="Add Objective";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.ROIName="' + oarName + '";')

t.append('IF.RoiList.#"' + oarName +'".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#'+ str(number)+'"="Max DVH";')

t.append('IF.RoiList.#"' + oarName +'".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="' + str(minimumDose) + '";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.UserPercent="' + str(point) + '";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.NOTEQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="' + str(weight) + '";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager .InversePlanningManager .CombinedObjectiveList .Last .ROIName = "PTV70\_eval";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.SetObjectiveType.#"#' + str(number) + '"="Max Dose";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Dose="7350";')

t.append('IF.RoiList.#"' + oarName + '".CurveCount.EQUALTO.Store.FloatAt.ZeroTest.THEN.PluginManager.InversePlanningManager.CombinedObjectiveList.Last.Weight="0";')

t.append('Store.FreeAt.ZeroTest="";')

return t

## Configuration Files

Percentage Points Configuration (oar names on the left, and percentage points on the right)

cord4mm [0]

brain [0]

brainstem [0]

mandible [0]

rt\_brachial\_plexus [0]

lt\_brachial\_plexus [0]

esophagus [0]

larynx\_for\_edema [0,20,30,40,85]

rt\_inner\_ear [35,50,65]

lt\_inner\_ear [35,50,65]

lt\_parotid [35,40,45,50,55,65,85]

rt\_parotid [35,40,45,50,55,65,85]

oral\_mucosa [70,80,90]