

Dose Painting by Numbers for Head and Neck Cancer - Photons versus Protons

Eirik Ramsli Hauge



Thesis submitted for the degree of
Master of science in Biophysics and Medical
Physics
60 credits

Physics Department
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

May 2018

Dose Painting by Numbers for Head and Neck Cancer - Photons versus Protons

Eirik Ramsli Hauge

© 2018 Eirik Ramsli Hauge

Dose Painting by Numbers for Head and Neck Cancer - Photons versus
Protons

<http://www.duo.uio.no/>

Printed: Representralen, University of Oslo

*I'm going to paint you by numbers
and color you in
If things go right we can frame it
and put you on a wall*

- Lego House by Ed Sheeran

Acknowledgments

Five eventful years at the University of Oslo (UiO) is nearing an end with every sentence written in this thesis. Through these years I have met many people who have helped me in one way or another and finally, I can thank them properly.

First of all, a big thanks goes out to my wife, Grete, who has supported me and looked after me all these years. I must also thank my family and my family in-law for all the kind words, good meals and relaxing vacations. We are all stuck with each other for the rest of our lives, and I could not have ended up with better friends.

When I first started at UiO, I had no clue about which field I would choose. Luckily, I met Eli Olaus Hole already the first year and she told me about medical and biophysics. As the son who swore to never follow in his father's footsteps and become a doctor, it was a minor defeat, but my mind was set. Thank you, Eli, for telling me about this option. Through the bachelor I had several professors who have inspired me and I would like to thank Arnt Inge Vistnes and Morten Hjorth-Jensen for seeing me and other students and walking the extra mile for us. During my fourth semester, I took FYS3710 which was taught by Einar Sagstuen and whatever doubts I had about medical physics for masters were replaced with an eagerness to learn. Thank you and the whole of the BMF group for all the great lunches and showing us what this field is all about.

With regards to this thesis, I must thank Torbjørn Furre, Bernt Louni Rekstad and all the medical physicists at the radium hospital for all the help I have received. Furthermore, this thesis would not have been possible without the guidance, help and insight of my two supervisors Eirik Malinen and Marius Roethe Arnesen. Thank you both for being patient with me, teaching me and for excited talks about my thesis with me (or nerding out, as some would say). This past year has been really fun and I have discovered the potential that lies within the field of medical physics thanks to you two. And I am especially thankful for all the help in writing this thesis. We are soon free.

Looking back, it is clear to me that I would not have succeeded as a student without the help and companionship of my friends at UiO. Thank you for all the laughter, the serious and un-serious talks and all the quiz breaks, Agnes Baker, Magnus Christopher Bareid, Simen Nyhus Bastnes, Tore Markussen Høiland, Ole Gunnar Johansen, Joakim Kalsnes, Anne Marit Rykkelid and Mathias Vege.

Lastly, I want to thank my congregation, Misjonssalen, for being my spiritual home filled with talented, caring and kind people. And of course, as always, the biggest thanks goes to God.

Abstract

In Norway, treatment of head and neck tumors with radiotherapy has a success rate of about 70%. Before treatment, patients can be imaged by e.g. CT, MR or a combination of either with PET. From this information, the tumor is delineated and a radiotherapy dose plan is created. The dose planning is done by defining a set of volumes representing different regions of interest (ROI) and then defining different optimization settings to each ROI. The treatment planning software will do an inverse optimization to create the best treatment plan. One can also choose to do robust optimization e.g. patient motion is taken into account during optimization. Photons (x-rays) are normally used to deliver the dose. However, as proton therapy is known to give the same tumor dose coverage and lessen the dose to organs at risk (OAR), this might be the better option. In conventional radiotherapy, a homogeneous dose is planned for the tumor. Dose Painting by Numbers (DPBN) suggest the use of radiobiologic information to tailor the tumor dose voxel by voxel. As most treatment planning systems are optimized for planning homogeneous dose distributions to tumor, it has been a challenge to implement DPBN.

In this thesis, scripting has been used to create prescribed and inverse DPBN dose plans for 10 patients. Both plans were imported into RayStation Research and the inverse dose plan was used as a basis to plan a summed dose. This creates a secondary dose plan which is the DPBN plan. Four dose plans were created for each patient: a robust, conventional photon plan, a robust, DPBN photon plan, a robust, conventional proton plan and a robust, DPBN proton plan. The tumor control probability (TCP), the quality factor(QF) (which assesses the conformity), normal tissue complication probability (NTCP) of parotid glands and dose to OARs (parotids and spinal cord) were evaluated for each plan. Lastly, tumor motion and shrinkage during the course of DPBN was simulated.

Proton DPBN plans had a lower QF than the photon DPBN plans. Mean TCP increased with DPBN compared to conventional treatment for both beam modalities. Moreover, DPBN gave a higher increase in TCP for patients with a low TCP after conventional radiotherapy. The dose to OARs was highest for photon plans, while NTCP was seen to be lower for proton plans than for photon plans. During translation the TCP decreased and QF increased while during shrinkage both the TCP and the QF increased. If the tumor was both translated and shrunk it was seen that the translation was the dominant effect. Moreover, photons were less susceptible to change compared with protons and performed better post translation and/or shrinkage.

It was concluded that it is possible to implement DPBN in commercial treatment planning systems. Proton DPBN treatment is superior if one can ensure patient motion and other translational errors below 1.5 mm. If not, photon DPBN treatment is preferred. Due to the high dose to the cells in the wake of tumor shrinkage, one should not use DPBN through the whole treatment unless said cells are necrotic. It would be preferable to treat the e.g. first 10 fractions with DPBN and then do the rest of the treatment conventionally.

Contents

1	Introduction	1
2	Theory	3
2.1	Classification of Head and Neck cancers	3
2.2	CT	4
2.3	PET	7
2.3.1	¹⁸ FDG-PET	8
2.3.1.1	SUV	9
2.3.2	Uncertainties in PET	9
2.3.3	PET/CT	10
2.4	Radiotherapy	10
2.4.1	Volumes in Radiotherapy	10
2.4.2	Linear Acceleratos and Photons	12
2.4.3	Treatment with Protons	15
2.5	Dose Plan Optimization	18
2.5.1	Robust Dose Plan Optimization	19
2.5.2	EUD	19
2.6	The Linear Quadratic Model and TCP	20
2.7	NTCP	21
2.8	Dose Painting by Numbers	22
2.8.1	Coarseness of DPBN based upon PET	22
2.9	QF	22
2.9.1	QVH	23
3	Material and methods	24
3.1	Acquiring Data	24
3.2	Scripting Dose Painting by Numbers	24
3.3	Planning in RayStation Research	26
3.3.1	Dose Painting By Numbers in RayStation Research	29
3.3.2	Robust Treatment	30
3.4	Dose Plan Evaluation	31
3.4.1	Clinical Goals	31
3.4.2	Visual evaluation	31
3.4.3	Perturbing the Dose	32
3.4.4	Exporting Data	33
3.4.5	Evaluation of Dose Painting by Numbers	33
3.4.5.1	Finding a good TCP	33
3.4.5.2	Calculating QF, TCP and NTCP	33
4	Results	36
4.1	Creating Dose Maps	36
4.2	Quality of Treatment Plans	38
4.3	Tumor Control Probability	45
4.4	Translation and Shrinkage	46
4.5	Normal Tissue	51

5	Discussion	54
5.1	Limitations	54
5.1.1	Patients	54
5.1.2	Image acquisition	54
5.1.3	Limitations in RayStation Research	54
5.1.3.1	The Learning Process	54
5.1.3.2	Finding the right index	55
5.1.3.3	Evaluation limitations	55
5.1.4	TCP, QF and NTCP	56
5.1.5	Setup uncertainties, Translation and Shrinkage	57
5.2	Treatment Plans	60
5.2.1	Ease of Use	60
5.2.2	Robust Evaluation	61
5.3	DPBN calculations and implementation	62
5.3.1	Calculating DPBN	62
5.3.2	Making the Plans Robust	63
5.4	Comparing Treatment Plans	63
5.4.1	QF	64
5.4.2	TCP	65
5.5	Translation and Shrinkage	66
5.6	Dose to Normal Tissue	67
5.7	Clinical Use	68
5.7.1	HPV	68
5.7.2	Recurring Cancer	69
5.8	Future Works	69
6	Conclusion	69
7	Appendix A	75
8	Appendix B: Scripts	84

Abbreviations

¹⁸FDG-PET ¹⁸F-labelled fluorodeoxyglucose positron emission tomography

ALARA As Low As Reasonably Achievable

AP Anterior/Posterior

CT Computed Tomography

CTV Clinical Target Volume

DAHANCA Danish Head and Neck Cancer Group

DICOM Digital Imaging and Communications in Medicine

DNA Deoxyribonucleic acid

DOI Depth of Interaction

DPBC Dose Painting By Contours

DPBN Dose Painting By Numbers

DVH dose volume histogram

EUD equivalent uniform dose

FWHM Full Width at Half Maximum

FWTM Full Width at Tenth of Maximum

Gy Gray

GTV Gross Tumor Volume

GUI Graphical User Interface

HPV Human papillomavirus

IDL Interactive Data Language

IMPT Intensity Modulated Proton Therapy

IMRT Intensity Modulated Radiotherapy

ICRU International Commission on Radiation Units and Measurements

LET Linear energy transfer

LKB Lyman, Kutcher and Burman

linac linear accelerator

LR Left/Right

MLC Multi Leaf Collimator

MRI Magnetic Resonance Imaging
MU Monitor Units
NTCP Normal Tissue Complication Probability
OAR Organ at Risk
OS Operating System
PET Positron Emission Tomography
PTV Planning Target Volume
QF Quality Factor
QVH quality volume histogram
ROI Region of Interest
RS RayStation
RSR RayStation Research
SNR Signal to Noise Ratio
SUV Standardized Uptake Value
TCP tumor control probability
TPS Treatment planning system
VMAT Volumetric Arch Treatment

1 Introduction

In 2016 there were 32 827 new occurrences of cancer registered in Norway. Among these there were 187 persons with cancer in the throat and 107 in the mouth. A total of 637 new cases were found in the whole mouth and throat region in general [1, Table 5.1]. The cumulative risk of developing cancer in this region by the age of 75 is 1.1 % for males and 0.5 % for females [1, Table 5.4]. Over the last 50 years there has been a steady increase in incidence of cancer in this area [1, Table 5.13 and 5.14]. Among the risk factors regarding development of head and neck cancer are Human papillomavirus (HPV) infection, consuming alcohol frequently and in large quantities and the use of tobacco products [2]. The survival rate in the period 2012-2016 was 67.2 % for males and 73.7 % for females [1, Table 8.1 and 8.2].

Head and neck cancer is usually treated with surgery and/or radiotherapy while chemotherapy might be used in some cases [2]. Proton therapy is an alternative modality that can give the same or better tumor control as photon therapy while sparing normal tissue [3–9]. If the patient is treated with external radiotherapy, it is common to use X-rays (photons). As proton therapy is becoming more available worldwide [10, 11], and two new treatment centers are planned for Oslo in 2023 and Bergen in 2025 [12], it is important to consider the advantages and disadvantages of each radiotherapy modality.

To ensure sufficient radiation dose to tumor, it is common to delineate a Clinical Target Volume (CTV) and a Planning Target Volume (PTV) around the Gross Tumor Volume (GTV). While the CTV accommodates for subclinical disease, PTV accommodates for internal tumor motion, patient positioning and other planning uncertainties and errors. Up to now, it has been common to use the PTV as a main prescription volume. Modern Treatment planning systems (TPSs) allow the use of robust optimization instead of PTV. Robust optimization requires the user to define a set of possible errors and the software then takes these errors into account during optimization of the dose distribution. [13].

In modern radiotherapy the ideal is to tailor each patient's treatment to the patient specific anatomy and biology. However, there is still room for improvement. Radiotherapy through Dose Painting By Numbers (DPBN) and Dose Painting By Contours (DPBC) are treatment techniques where one utilizes radiobiological imaging to adapt the treatment to each patient [14, 15]. The former technique uses the image data to create a voxel by voxel specific dose, while the latter creates several subvolumes within a tumor and dose escalates them according to the radiobiologic information. Different image modalities can be used for this purpose, among these ^{18}F -labelled fluorodeoxyglucose positron emission tomography (^{18}F FDG-PET), which is well suited for several types of cancers [16]. Tumors with high FDG uptake before radiotherapy are found to have greater risk of lack of tumor control, regardless of treatment modality [17], and recurrence is often found within regions with high FDG uptake before treatment [18].

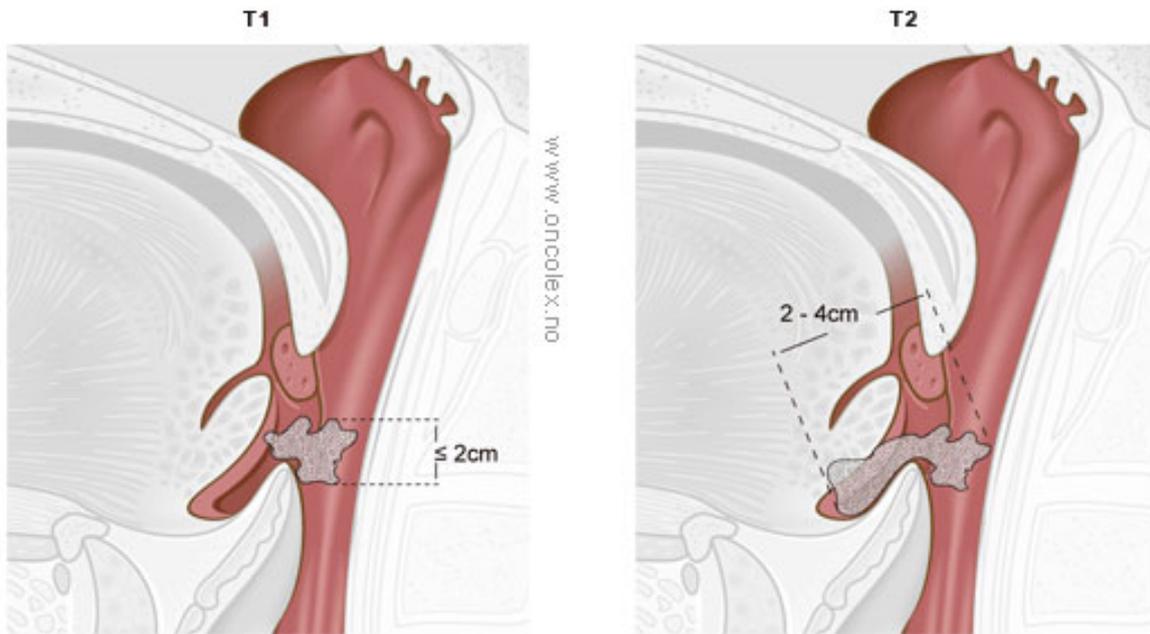
To the author's knowledge, no TPSs offer direct solutions for dose painting, neither by contours nor by numbers. Since most TPSs are based upon planning a homogeneous dose to a volume, true DPBN can be especially difficult to implement. The aim of this study is to implement robust, true DPBN in a commercial TPS for both proton and photon therapy, and compare robust DPBN treatment plans to robust photon and robust proton treatment plans. To achieve this, four treatment plans will be created for ten head and neck patients from the "Prognostic value of PET-CT before radiotherapy of head and neck cancer" study. These four plans will be: a robust Volumetric Arch Treatment (VMAT) photon plan, a robust VMAT photon DPBN

plan, a robust Intensity Modulated Proton Therapy (IMPT) plan and a robust IMPT DPBN plan. Each plan will be evaluated by looking at tumor control probability (TCP), dose volume histograms (DVHs) of the spinal cord and parotid(salivary glands), and Normal Tissue Complication Probability (NTCP) for parotids. The DPBN plans will be further evaluated through the use of quality volume histograms (QVHs) and Quality Factor (QF) values. Lastly, we will simulate translation and shrinking of the tumor during radiotherapy for the DPBN plans and evaluate QF and TCP.

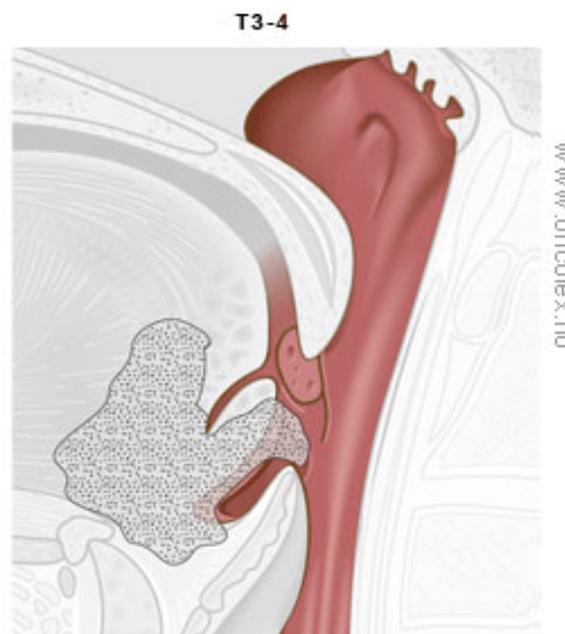
2 Theory

2.1 Classification of Head and Neck cancers

In head and neck cancer there is a standard for classifying tumors set by the International Union against Cancer (UICC). This classification will also be used in this thesis, where T1 - T4 classification for pharyngeal cancer is illustrated in figure 1. The cancer can further be classified with the letters N and M where N indicates the spread of lymph node metastasis and M gives information about whether there are distant metastasis or not [19, p.32 - 35].



(a) T1 and T2



(b) T3 and T4

Figure 1: Illustration of T1 - T4 classification of tumors in oropharynx according to the UICC [19, p.32-35] [20]. T1-T2 tumors are typically small, while T3-T4 are larger tumors that may invade neighboring organs.

2.2 CT

Computed Tomography (CT) is a commonly used diagnostic method utilizing x-rays which are taken swiftly from many angles around the patient to construct a 3D image of the patients anatomy. In principle, during a CT-scan many planar slices are constructed. These planes are traversed by X-rays which are then detected and an attenuation maps is created. In an ideal

case, i.e. there is no scattering or secondary radiation, this attenuation can be described by the following formula [21]:

$$N_L = N_0 e^{-\mu x} \quad (1)$$

where N_0 and N_L are the original amount of photons and the amount after distance x [m] traveled in a medium with attenuation μ [1/m]. The attenuation coefficient is dependent upon the energy of the x-ray beam and the density of the material it traverses [22]. Thus bone will absorb more photons than e.g. soft tissue. The detector will detect photons and provide an attenuation map to a computer which produces an image of the plane.

By producing many planes, a 3D representation of the patient can be created where each voxel is expressed in terms of Hounsfield Units [23]:

$$\text{HU} = 1000 \cdot \frac{\mu_x - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \quad (2)$$

where μ_{water} and μ_{air} are the attenuation coefficients for water and air respectively while μ_x is the attenuation registered by the detector for a given voxel. The Hounsfield Unit for water is 0 and -1000 for air. Soft tissues have about the same attenuation as water and thus the difference in HU might not be that large. When viewing a CT-image the different attenuations are presented by a gray scale. This gray scale can be adjusted digitally and one can thus view minute attenuation differences and separate different types of soft tissue. By using contrast agents these differences can be made more distinguished. An illustration of a CT-scanner can be seen in figure 2.

Modern CT-scanners are so-called third generation scanners. These scanners send out the X-ray in a fan beam wide enough to cover the whole field of view. The x-ray tube and the detector rotate around the patient while the patient's bed goes through the scanner. The rate between table increment in one revolution, T , per nominal slice thickness, b , gives us the pitch q [22]:

$$q = \frac{T}{b} \quad (3)$$

A higher pitch value means a quicker scan, but more information is lost. This information can be interpolated from the acquired data. By using a lower pitch, one gets more data, but the patient receives more radiation and the likelihood of patient movement increases with longer scans. The pitch used in a CT-scan is therefore a compromise between data quality and patient irradiation.

The data from the detector is reconstructed by a computer to create the CT-images. This reconstruction can be done by two different techniques, filtered back projection or iterative reconstruction.

Filtered back projection is the least computationally heavy of the two, but it is also the least accurate. The reconstruction is done by back projection the attenuation along the angle of which the attenuation was detected. Each back projection adds to the image and the more angles used, the better the image. However, this image appears smeared and therefore a filter is added. Different filters are used depending upon what the tissue user wishes to emphasize. E.g. an edge filter may be used if one wants clear edges. This will however increase the noise in the image and tissues which have similar densities will be hard to differentiate. The method also assumes that the focal size of the x-ray beam is infinitely small, i.e. a point source, and that each projection is assumed equal and noiseless.

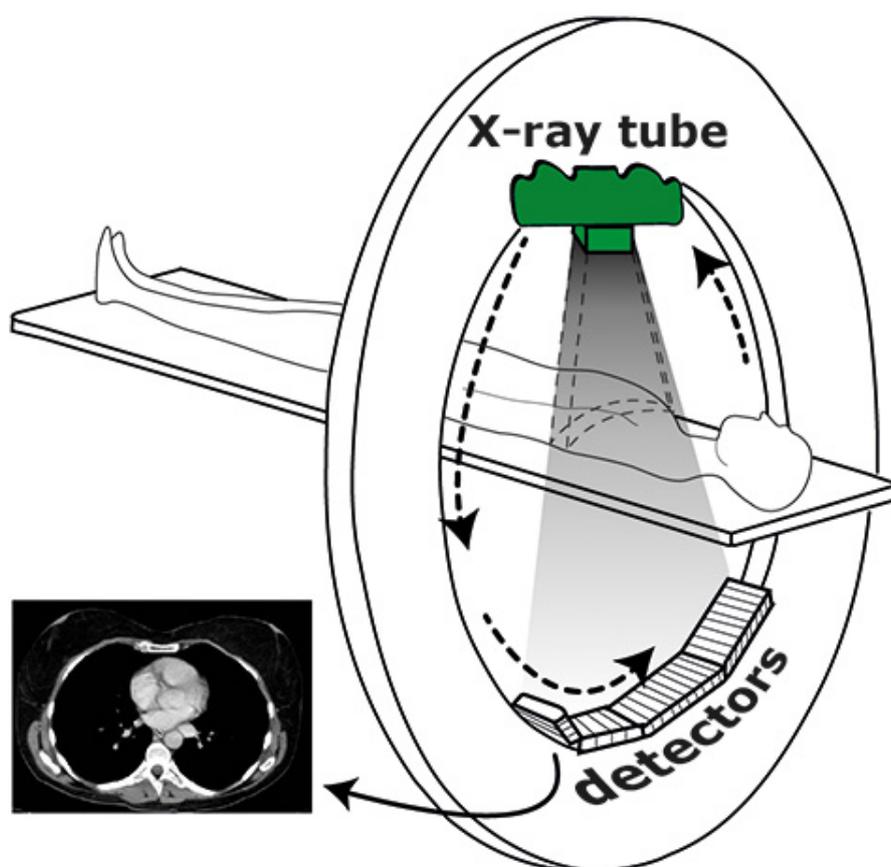


Figure 2: Illustration of a third generation CT scanner [24]. While the patient lies down in the scanner on a bed, a X-ray tube revolves around the patient with a detector array on the opposite side of the patient. The detector array detects the amount of attenuation from different angles through the patient and this information is used to reconstruct an image of the patient.

Iterative reconstruction differs from filtered back projection in that it makes a guess at the image and then compares the attenuation data with the guess before making a new guess and thus iterating itself backwards until a solution is reached, i.e. the image. However, this method does not assume the attenuation profile to be noiseless, and because of this there will not be one exact solution.

Both iterative reconstruction and filtered back projection have their pros and cons. In modern scanners one can usually choose between multiple algorithms where some are pure iterative reconstruction or filtered back projection. It is also common to offer a combination of the two techniques. By using filtered back projection to assume the original guess and then use iterative reconstruction from there, the computation time is greatly reduced and the final image draws advantage on the accuracy of iterative reconstruction.

2.3 PET

Positron Emission Tomography (PET) is a diagnostics method which differs from CT in that it does not map anatomical structures, but is a functional imaging technique. The fundamental principles behind PET is the detection of photon pairs created in a positron-electron annihilation caused by positrons emitters injected into the body. The photon pairs is emitted close to 180° angle relative to each other. Around the patient there is placed a ring of detectors which detects these photon pairs. By detecting a sufficient amount of photon pairs, one can generate an intensity map which is the PET-image.

For PET to be efficient, one must choose a good positron emitter and inject the patient with it. By attaching a positron emitter to a biological active molecule and injecting it into the patient, there will occur annihilation events along the path of the molecule. The decay of a positron emitter can be described by the following formula:



where A is the mass number and Z is the atomic number of the decaying nucleus X, and Y is the final element. e^+ is the positron and ν_e is an electron neutrino [25]. The released kinetic energy will be divided between the positron and the neutrino [21]. The amount of kinetic energy the positron receives will affect its time of flight before it is annihilated. Some positrons are annihilated in flight and the remaining kinetic energy is then transferred to the photons [21]. The annihilation of an electron-positron pair can be written as:



where e^\pm are positron and electron respectively and γ represents photons. Often the electron and positron have no kinetic energy when they interact and thus the resulting photons will both have energy equal to m_e . However, as stated above, the positron might be annihilated in flight and thus the resulting photons might have different energies and they will not go in completely different directions.

By placing a ring of detector elements around the patient we can detect the photons emitted from each annihilation. The detectors are tuned to detect 0.511 MeV photons, i.e. photons with energy equal to m_e . If two detections occur within a short time span (e.g. less than 10 ns [22]), we can assume that the annihilation happened somewhere along or near a straight line we can draw between the two detection events. The general principle is illustrated in figure 3.

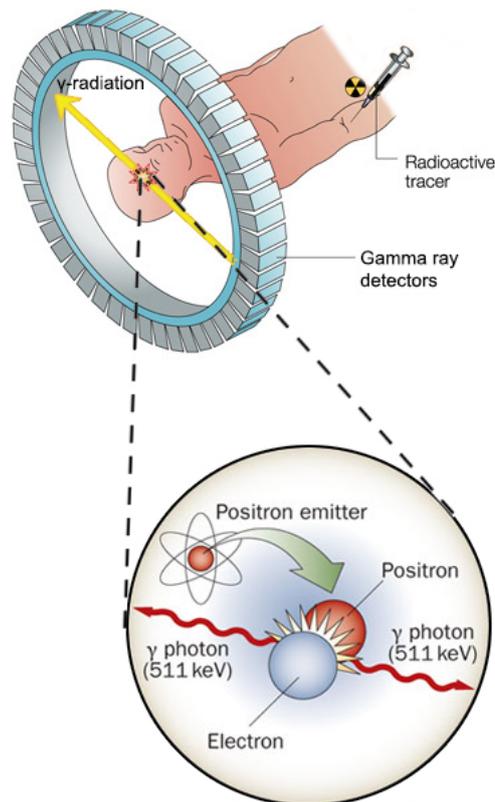


Figure 3: Illustration of the basic principle of PET [26]. A radioactive tracer (positron emitter) is injected into the patient. Once a positron is emitted, it is annihilated by an electron and results in two photons traveling in opposite directions. These photons can be detected by a detector ring and we can assume that the annihilation occurred somewhere along a line we can draw between the two detection events.

Each detector element consists of a scintillator connected to a photo multiplier tube. The scintillator absorbs the photon and is excited. Upon returning to its initial state, the scintillation crystal emits a photon with lower energy which is multiplied by the photo multiplier tube and then registered as a signal.

2.3.1 ^{18}F FDG-PET

One of the most commonly used PET tracers is ^{18}F FDG (18F-Fludeoxyglucose), i.e. the positron emitter used is ^{18}F placed in a sugar molecule. As a glucose analog, ^{18}F FDG enters the cells, but where a normal sugar would go through the entire glycolytic pathway towards glycolysis, the ^{18}F FDG can not continue past phosphorylation. This causes the ^{18}F FDG to be trapped in the cell and can therefore be used as a measurement of glucose metabolism [27]. This is an advantage in cancer treatment as many tumors have a high glucose metabolism. All PET-images used in this thesis are ^{18}F -labelled fluorodeoxyglucose positron emission tomographs (^{18}F FDG-PETs). ^{18}F FDG-PET is widely used and one of the reasons is that ^{18}F has a half-life of 110 minutes. This makes it possible to produce the radioactive sugar in the morning and still get a sufficient number of detections in the afternoon.

If the patient has received an ^{18}F FDG-PET scan prior to radiotherapy, it has also been shown that recurrence often happens in regions where the uptake of FDG was high [18].

2.3.1.1 SUV

The measured intensity from an ^{18}F FDG-PET scan depends on many factors such as the injected activity and weight of the patient. To use a semi-quantitative index of the uptake, the Standardized Uptake Value (SUV) has been introduced [22, p. 316]:

$$SUV = \frac{\text{Tumor uptake (MBq mL}^{-1}\text{)}}{\text{Injected dose (MBq)/Patient weight (g)}} \quad (6)$$

From equation (6) we can see that if the the tracer is evenly distributed throughout the patient, we would measure $SUV = 1$ everywhere. The advantage of using SUV is that it compensates for two significant variabilities between scans: The amount of injected tracer and the weight of the patient [28].

2.3.2 Uncertainties in PET

There are several contributors to uncertainty in PET scans. They vary between errors we can not do anything about to errors which are caused by hardware/software limitations. The technology surrounding the latter is always being improved upon and the error contribution from these sources are thus becoming smaller and smaller. One of the most fundamental contributions to errors is the positron range after being emitted. The range is dependent upon its initial kinetic energy. As the positron travels a certain range before annihilating, the line drawn between two detections will not necessarily go through the position of the original β -decay. Another issue is the non-collinearity of the photons, i.e. that the photons do not travel with a 180° angle between them, but with a slightly smaller angle. This effect increases as the diameter of the detector ring increases. It should also be noted that the detector size in itself will affect the spatial resolution. The contribution to the spatial resolution is usually $\simeq d/2$ where d is the scintillation crystal width.

One of the biggest contributors to the uncertainty is the Depth of Interaction (DOI). This effect is present when the emitted photons originate from a source to the side of the detectors rings field of view (see figure 4). As the emissions goes to the sides of the field of view, the photons will not hit the detector through the front, but rather come in from the side. This gives a poorer radial resolution than if the photons had entered the detector head on.

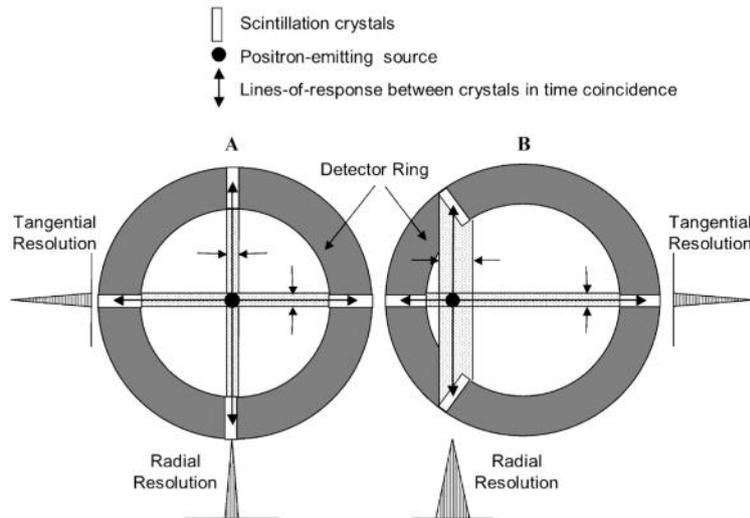


Figure 4: Illustration of Depth of Interaction (DOI). As the positron emission source is moved to the sides of the detectors field of view, the photons will hit the detector from the side rather than head on and give a poorer radial resolution.

The last contribution to mention is the one caused by the reconstruction algorithm. Because the PET-images are acquired over a finite time span there will be noise in the image. To compensate for this noise the image is smoothed to obtain a good Signal to Noise Ratio (SNR). Modern whole-body PET-scanners typically have spatial resolution between 5 and 10 mm [22, p. 295 - 296].

2.3.3 PET/CT

A pure PET-scan will give information about the different physiologic processes in the patient dependent upon the radioactive tracer used. This information alone is not optimal as the scan shows there is something happening inside the body, but gives little information about where or in which organ. To compensate for this, PET is often combined with CT; or another diagnostics method which depicts anatomy such as Magnetic Resonance Imaging (MRI); to give both anatomical and physiological information. PET/CT scanners are the most common hybrid scanners as the necessary hardware for both modalities are relatively easy to combine, they provide accurate image registration for both PET and CT, and one can use the data from the CT scan to do attenuation correction for the PET data. [22, p. 306]

2.4 Radiotherapy

2.4.1 Volumes in Radiotherapy

Today, it is common to define a set of extra volumes around the tumor to secure dose coverage of the entire tumor during the course of radiotherapy.. These volumes were defined by the International Commission on Radiation Units and Measurements (ICRU) [29] as follows:

- **Gross Tumor Volume (GTV):** is the gross demonstrable extent and location of the malignant growth.
- **Clinical Target Volume (CTV):** is a tissue volume that contains a demonstrable GTV and/or subclinical malignant disease that must be eliminated.

- **Planning Target Volume (PTV):** is a geometrical concept used for treatment planning, and it is defined to select appropriate beam size and beam arrangements to ensure that the prescribed dose is actually delivered to CTV.
- **Treated Volume:** is the tissue volume receiving at least a dose selected and specified by the radiation oncology team as being appropriate to achieve the purpose of the treatment.
- **Irradiated Volume:** is the tissue volume that receives a dose that is considered significant in relation to normal tissue tolerance.
- **Organ at Risk (OAR):** are normal tissues (e.g., spinal cord) whose radiation sensitivity may significantly influence treatment planning and/or prescribed dose.

In layman's terms one can say that GTV covers the part of the tumor one can detect by diagnostic imaging. Years of research has taught us that the tumor actually spreads a bit further, although we can not see it, except under the microscope. Therefore, one adds another volume; the CTV. Finally, while the treatment is delivered there might be errors in patient positioning, geometrical errors in treatment delivery and systematic errors within the delivery system. To account for all these possible errors one defines another volume, the PTV, to ensure that the CTV receives the intended dose. OARs on the other hand are organs that must be spared as much as possible. The treated volume is the volume that receives at least the dose intended to treat the patient while irradiated volume is the volume containing all the tissue which receives a significant dose. Each of the concepts are illustrated in figure 5

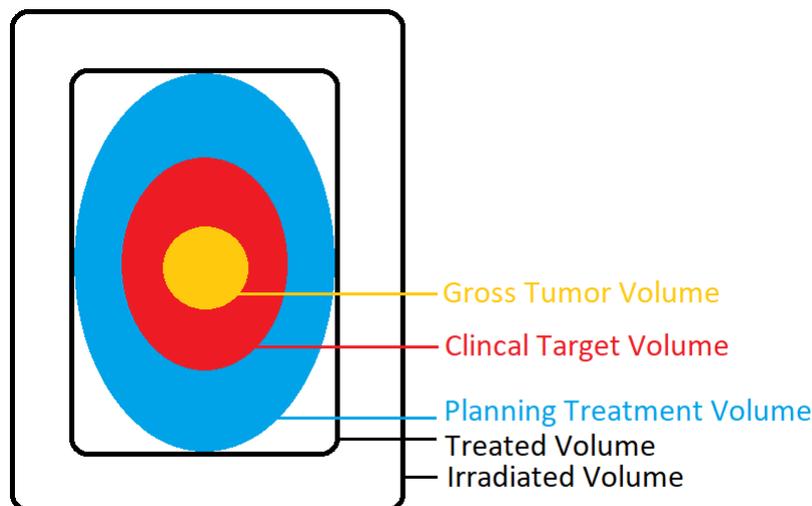


Figure 5: Illustration of volumes recommended by the ICRU for planning and reporting radiotherapy

Radiotherapy is one of several ways to treat cancer. It can be used as a stand alone treatment or together with other types of treatment such as chemotherapy or surgery. In itself, radiotherapy can be divided into two main forms of treatment; external radiotherapy and brachytherapy, where the former is the focus of this thesis.

The effectiveness of radiotherapy comes from the characteristics of ionizing radiation interactions with tissue. It is the destructive, ionizing power of the transmitted energy to the atoms and molecules along the path of the beam that we use in external radiation therapy.

Ionizing radiation can be divided into directly ionizing and indirectly ionizing. Directly ionizing radiation is particles such as electrons or protons, which almost continuously ionize atoms or molecules along their path. Indirect radiation such as x-rays transfer large amounts of their kinetic energies to mainly atomic electrons, which subsequently cause ionizations. The ionizations will in turn result in ionized atoms or molecules, which can further undergo chemical reactions leading to stabilization of the radiation damage.

The radiation sensitive target of the cell is the nucleus, where the Deoxyribonucleic acid (DNA) is stored. A cell has an impressive repair system for most types of damage, however, it has been shown that if DNA is damaged sufficiently, the cell's ability to divide is lost and the cell is considered inactivated. If the DNA is repaired, there may still be faulty repairs that can have consequences for the cell and its progeny. Radiotherapy is therefore considered being a 'double-edged sword'. On one side, the radiation will inactivate the tumor cells, however the radiation will also harm the healthy cells. In an ideal world, the treatment would only target tumor cells, but sometimes there will be damage to healthy tissue and organs. To keep the dose as low as possible while still providing tumor coverage is within the principle called ALARA (As Low As Reasonably Achievable).

Radiation dose is defined as energy-per-mass absorption of 1 Joule/kg. This means that dose is always measured as the amount of energy absorbed by the medium which the radiation travels through, and not the energy transferred from the radiation. Dose rate is defined as dose per unit of time and is also an important factor in radiotherapy. If one were to receive 2 Gray (Gy) over a period of a few minutes, the effect would be drastically different from the effect if the dose was received over the course of 60 years. Another important factor is the Linear energy transfer (LET) which is the energy transferred per unit length of the track [30], and is a measure of the ionization density.

2.4.2 Linear Accelerators and Photons

Today, the most commonly used type of radiation for cancer treatment is photon (x-ray) radiotherapy. High energy photons can be produced with a linear accelerator (linac) which needs little space (compared to e.g. accelerators used for proton therapy) and are commercially available. A linac accelerates electrons using microwaves before the electron beam is bent by a magnet and hits a target. The target then produces high energy photons used to treat the patient. Linacs are the result of decades of engineering genius where every part serves the beam in an efficient manner to produce the best beam for patient treatment.

When photons interact with soft tissue in human tissue the relative dose as a function of depth will build up and then have an exponential decline as described in equation (1) and illustrated in figure 6

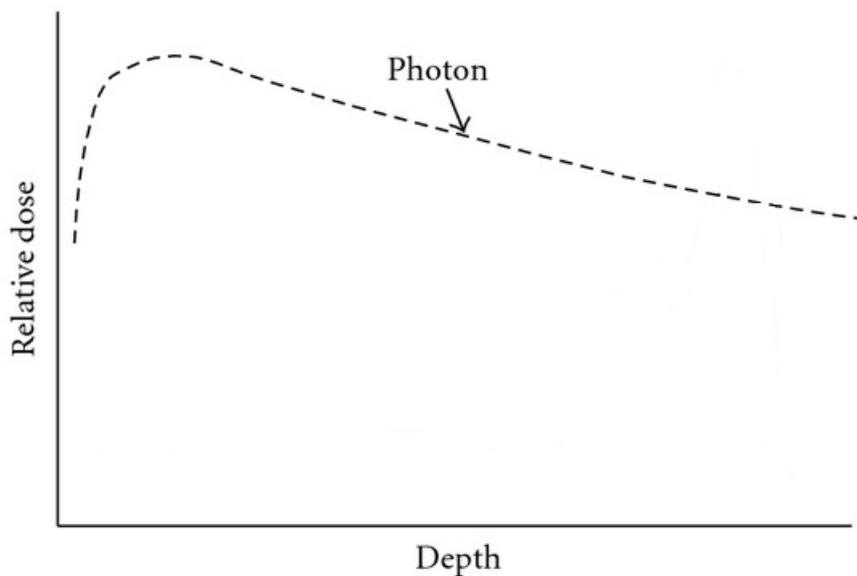


Figure 6: Dose as a function of depth for a photon beam. Adapted from [31, Fig. 1]

This characteristic dose deposition makes photon therapy problematic. A tumor is often situated deeper within a patient than the photon dose peak and there is a long tail of dose after the peak. Using a beam with a higher energy will only move the dose peak deeper and create a tail which gives a higher dose at larger depths [21].

Photon therapy is delivered using a linear accelerator (linac). The linac consists of many parts as illustrated in figure 7.

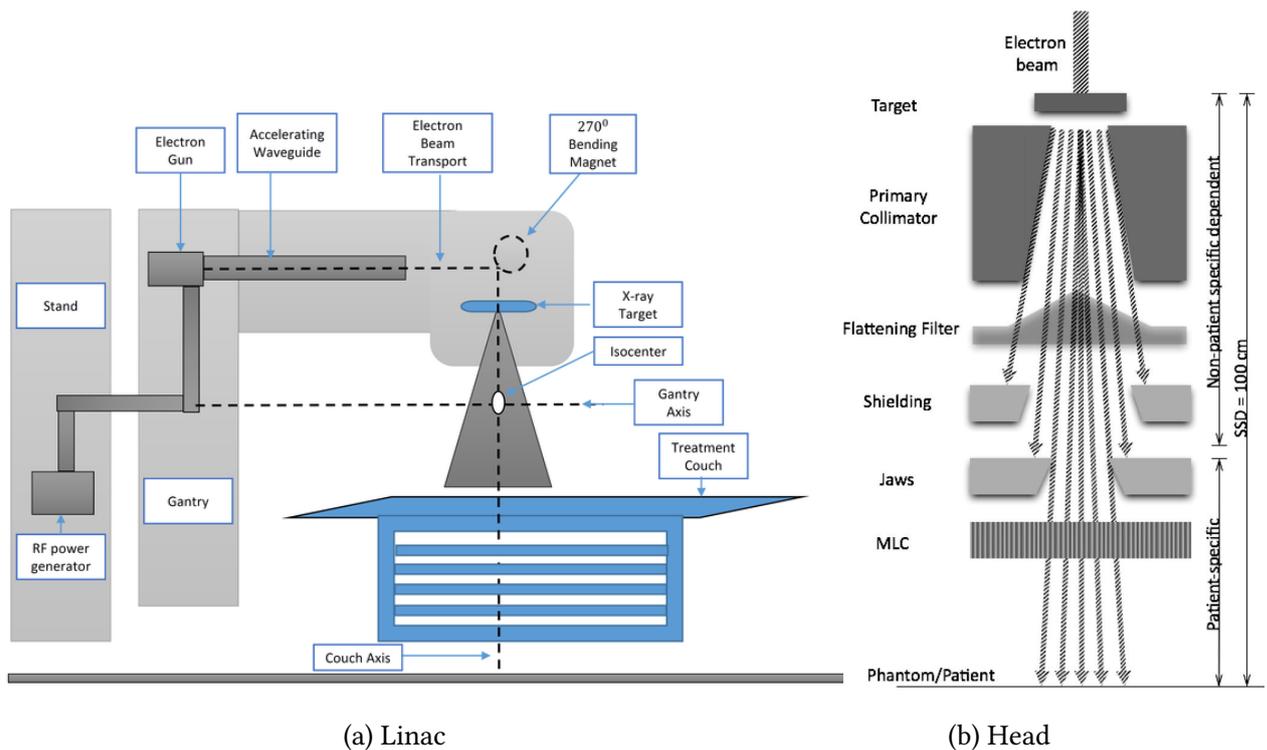


Figure 7: Illustration of a linac [32] and a closer look at its head [33]. The radio frequency (RF) generator creates RF waves which travels through a waveguide. The electron gun shoots out electrons which surf along the wave and are accelerated to near the speed of light. In the gantry head, the beam is bent 270° before hitting an high-density target to create bremsstrahlung. This produces a beam of high-energy photons which are filtered to create an even intensity across the beam. The shape of the beam is then formed by the jaws and the Multi Leaf Collimator (MLC). The shield's are used to stop scattered photons. [34, p.3].

A photon beam delivered in units of MUs (monitor units) ”which are calibrated such that 1 Monitor Units (MU) yields an absorbed dose of 1 cGy at a specific depth in water for a standardized field.” [34, p. 3].

One of the most important parts of the linac is the MLC which is used for modern treatment techniques such as Intensity Modulated Radiotherapy (IMRT) and VMAT. To give as little dose as possible to the healthy tissue and at the same time give an adequate dose to the tumor, the gantry of the linac revolves around the patient and gives the treatment from multiple angles. In the gantry head there is a Multi Leaf Collimator (MLC) (figures 7b and 8) which shapes the treatment field. By combining rotating and shaping the field using MLCs the treatment can be tailored according to each patients needs. A treatment technique which utilizes these properties is the Intensity Modulated Radiotherapy (IMRT) technique. This is a technique where the gantry rotates to a certain set of predetermined angles and radiates with a dose profile created by the MLC. Once the treatment is done from all the angles, the patient has received complementary dose profiles from each angle which merges into a single dose profile where the tumor has, ideally, received a high dose and healthy tissue has received a much lower dose.

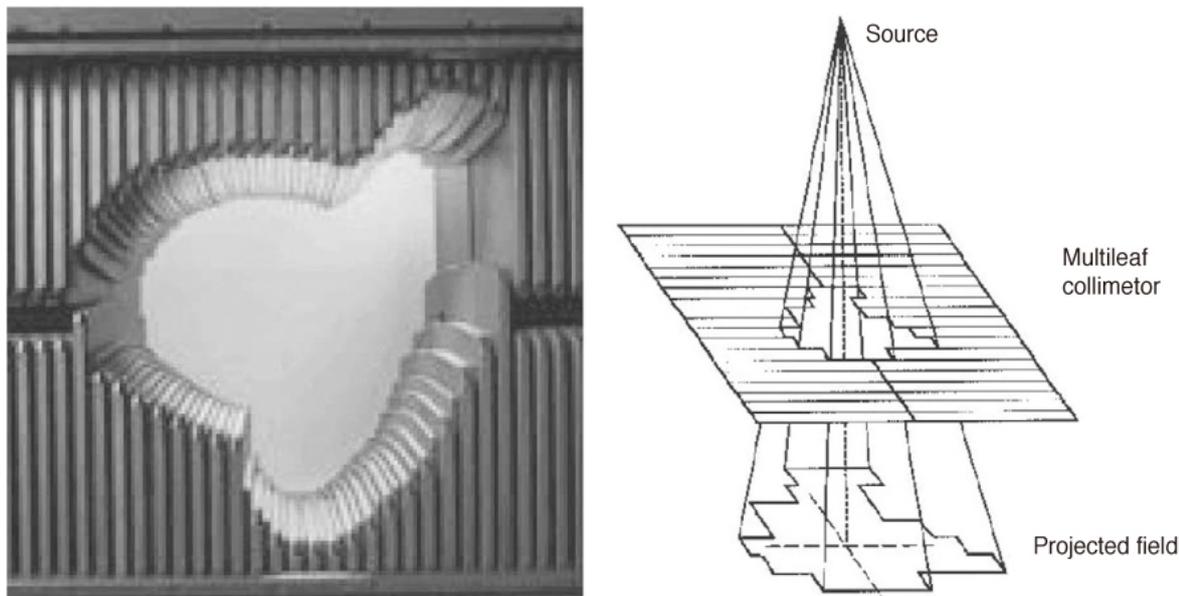


Figure 8: Picture of a MLC on the left and an illustration of how the beam is formed on the right. [35]

As technology has advanced, so has treatment techniques and IMRT is mostly replaced by the more advanced Volumetric Arch Treatment (VMAT). VMAT uses the same principles as IMRT, but instead of rotating around the patient in steps, there is a continuous rotation. Here, each leaf in the MLC can be positioned quickly with high accuracy. The advantages of using VMAT over IMRT is therefore that the treatment is given much quicker and the dose can be spread out over even more angles, but the dose profile of each angle is restricted by the velocity of each leaf.

2.4.3 Treatment with Protons

Protons behave differently than photons when interacting with tissue. As charged particles they give rise to direct ionizations. This gives a very different dose deposition in soft tissue compared to that of photons. The mean energy loss per distance traveled for a swift charged particle was described by Bethe in 1932 [36]:

$$-\left\langle \frac{dE}{dx} \right\rangle = \frac{4\pi}{m_e c^2} \cdot \frac{nz^2}{\beta^2} \cdot \left(\frac{e^2}{4\pi\epsilon_0} \right)^2 \cdot \left[\ln \left(\frac{2m_e c^2 \beta^2}{I \cdot (1 - \beta^2)} \right) - \beta^2 \right] \quad (7)$$

where m_e is the electron rest mass, e is the electron charge, c is the speed of light, ϵ_0 is the vacuum permittivity, n is the electron density of the material and $\beta = \frac{v}{c}$, where v is the particle velocity. We are especially interested in the inverse dependency upon β^2 as the charged particle slows down. With a decrease in velocity, the energy loss per distance increases and right before the particle stops there is a drastic increase as illustrated in figure 9. By aiming this large dose deposition directly at the tumor, we can, in theory, give high doses locally in the tumor, yielding a rather low dose before the target and even less behind it. The end of a protons range where this dose is deposited is called the Bragg peak.

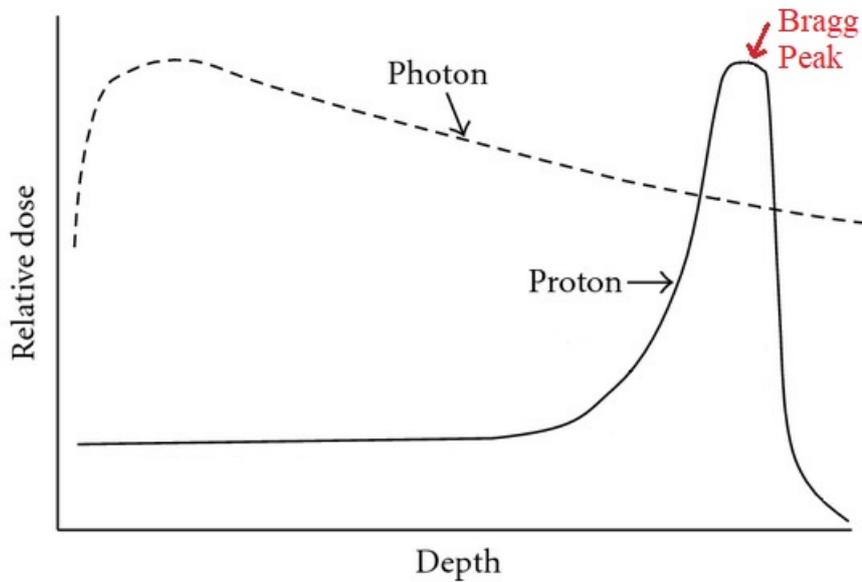


Figure 9: Dose deposited by a single energy proton beam and a photon beam as a function of depth. Adapted from [31, Fig. 1].

The depth of a Bragg Peak is dependent upon the beam energy and in a clinical setting one will use more than one energy to cover the whole depth of the tumor. This results in a more smeared out dose deposition in a so called Bragg plateau. A comparison of photon and proton depth dose curves with the Bragg plateau is presented in figure 10. Note that even though the accumulated dose from multiple proton energies increases the relative dose prior to the tumor compare to that of a single energy proton beam, it is still less than the relative dose from a photon beam.

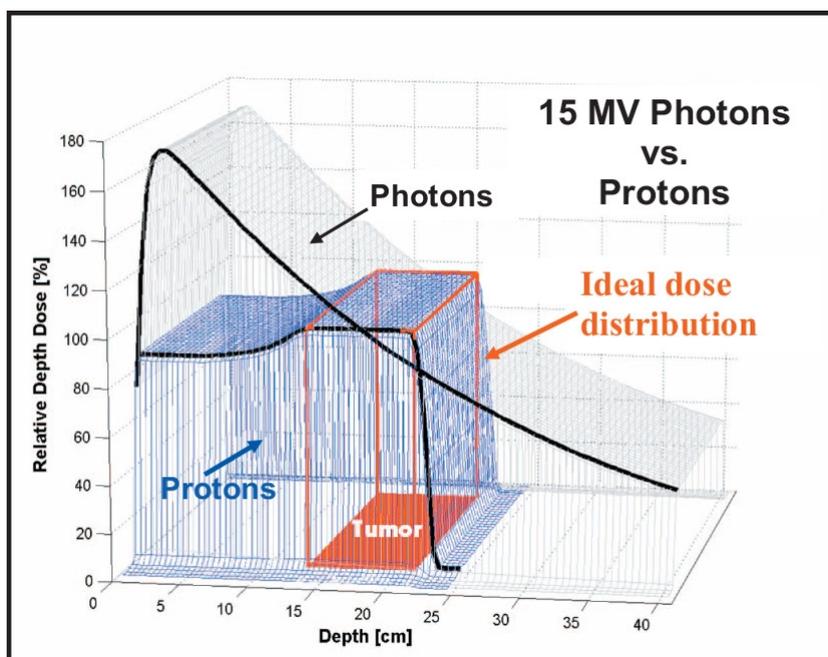


Figure 10: Depth dose curve comparison of photons and spread-out protons [37]

Photon radiotherapy delivered with a linac has become the modern standard, for protons,

however, there is no well established standard yet and there are many options. One of the biggest challenges regarding proton therapy is to build accelerators that accelerates the protons to sufficient energies, but at the same time are small enough to fit inside a hospital. As proton therapy is still in rapid development with many companies trying different approaches, there are many solutions on how to guide the beam to the patient. Figure 11 is an illustration of a treatment center. The treatment center has three treatment rooms where the protons are delivered to the patients. Two of them contains gantries which revolve around the patient, while in the last room the beam is fixed.

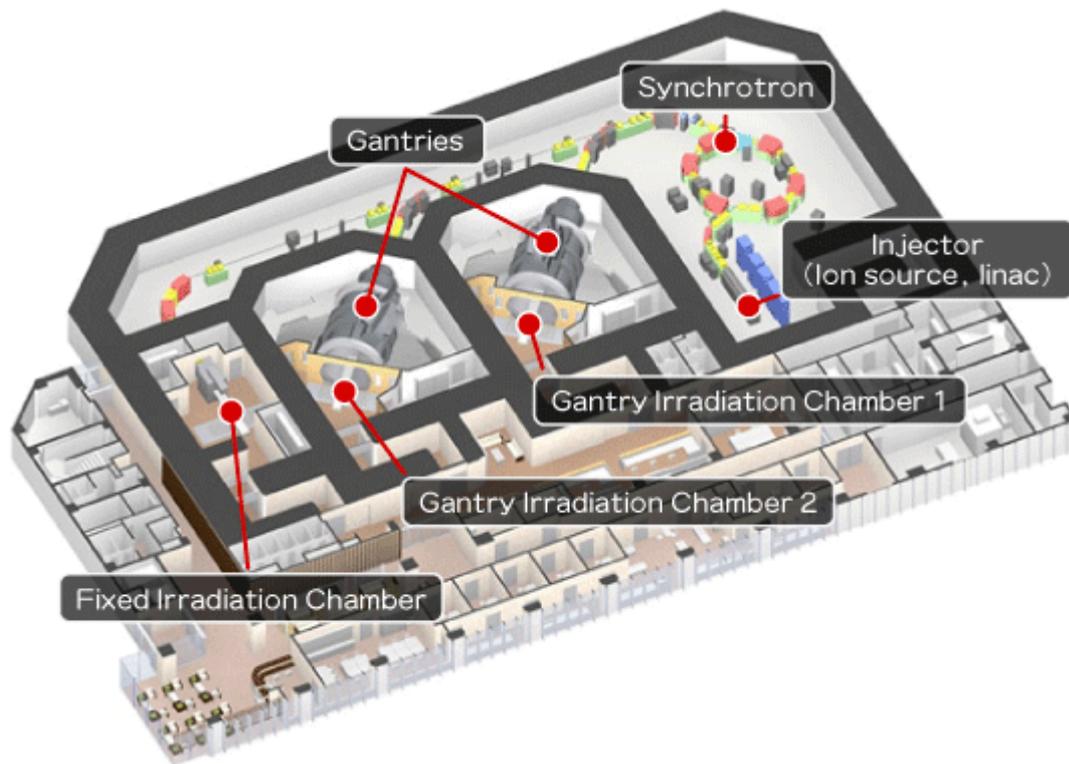


Figure 11: Illustration of a Japanese proton treatment center [38]. The protons are injected from an ion source into a linac to increase their velocity. They are then injected into a synchrotron which increases the energy even further. The beam is then lead to one of three treatment chambers where two contains gantries which can revolve around the patient and in the third the beam is fixed.

In pencil beam scanning the desired dose distribution is delivered to the tumor by scanning the volume with a modulated, focused proton beam so that the Bragg peaks (or spots) are placed evenly throughout the volume. The modulation is done by scanning magnets and by changing the energy, and therefore the range, of the beam. It is typical to scan the spots of the deepest range first and then the second deepest range etc [37]. Figure 12 illustrates the concept of pencil beam scanning.

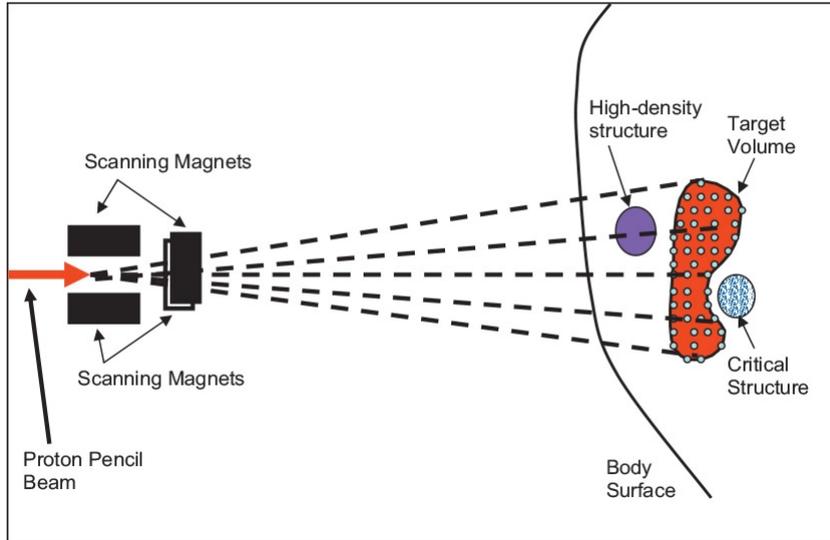


Figure 12: Illustration of the pencil beam scanning technique [37]. The proton beam scans the target volume by depositing Bragg peaks (spots) at depths dependent upon the energy of the proton beam. Scanning magnets and beam energy is modulated to scan the whole target. Usually one scans the deepest spots first and then the second deepest etc.

Intensity Modulated Proton Therapy (IMPT) is analogous to IMRT and utilizes pencil beam scanning. Treatment with IMPT is done by optimizing the Bragg Peaks from all fields to deliver the desired dose distribution to the patient while sparing OARs and healthy tissue as much as possible. As with IMRT one can set optimization functions for all Region of Interests (ROIs) and the optimization is done as described in section 2.5. Both IMRT and IMPT changes the intensity of the beam during dose delivery, but IMPT can also change the energy of the beam.

2.5 Dose Plan Optimization

Previously, when computation power was not as easily accessible as it is today, treatment planning was formed by forward optimizing. The planner would manually set the position of the gantry, how collimators should be positioned and other settings to create a dose profile. Once all settings were defined, the planner would evaluate the dose to the patient and then tweak the settings until the optimal profile was found. However, in a modern scanner you are able to fine tune such settings as gantry speed of rotation, couch position, MLCs position and speed and many more. Because of the share amount of time needed to optimize all these settings, radiotherapy has moved away from forward planning and over to inverse planning. In the latter process, the planner defines structures inside the patient according to the ICRU's definitions described above. Each volume is given at least one optimization setting with a certain weight and the computer optimizes the treatment thereafter. The optimization in itself is a variant of least square methods where the optimization function for volume r , f_r , compares the given dose to the preferred dose for each voxel:

$$f_r(d) = \sum_{i \in V_r} \Delta_{i,r} (d_i - d_i^{presc})^2 \quad (8)$$

where the relative volume of voxel i in volume r is $\Delta_{i,r}$, so that $\sum_{i \in V_r} \Delta_{i,r} = 1$. The entire volume of the structure is defined as V_r . d_i is the dose to voxel i while d_i^{presc} is the prescribed dose to the same volume [39].

Weighting each optimization function (f_1, f_2, \dots, f_n) with a weight factor (w_1, w_2, \dots, w_n) one finds the optimal solution by:

$$x \in X \underset{\text{minimize}}{\sum_{r=1}^R w_r f_r(d(x))} \quad (9)$$

where $d(x)$ is the dose distribution as a function of optimization variables x in the set of feasible optimization variables X (e.g. MLC leaf positions and segment weights for IMRT or spot weights for IMPT). [13, 34, 39, 40].

For photons, $d(x)$ is found by using the fluence $\varphi(x)$ [34]:

$$d(x) = P\varphi(x) \quad (10)$$

The photon beam is divided into rectangular bixels. The incidence matrix P has one column for each bixel and one row for each voxel. For pencil beam scanning with protons we find the dose distribution by using the spot weights as the optimization variables x , the columns in P will therefore represent each spot [34]:

$$d(x) = Px \quad (11)$$

2.5.1 Robust Dose Plan Optimization

In recent years, instead of using PTV as a safety margin, there is a possibility of using robust optimization. The robust optimization used in this thesis "takes into account the effects of possible errors and then strives for the plans which are robust against these effects" [40]. This optimization method is based upon minimax optimization. Similar to the optimization described above, minimax optimization optimizes r functions, which are all required to be robust over all possible scenarios $s \in S$. The minimax optimization method optimizes for the worst case scenario:

$$\underset{x \in X}{\text{minimize}} \max_{s \in S} \sum_{r=1}^R w_r f_r(d(x; s)) \quad (12)$$

where $x \geq 0$ and $d(x; s) = P(s)x$ as each scenario has a different dose distribution. It should also be noted that S is set to an interval as letting it contain improbable scenarios can compromise the plan quality [13]. The set of scenarios S represents different errors possible in patient treatment. In this thesis these errors are setup errors, i.e. uncertainties in patient position, and range errors, i.e. the density of the patient. The plan should be robust against all of these uncertainties.

2.5.2 EUD

According to Henríquez and Castrillón [41], the equivalent uniform dose (EUD) is the absorbed dose which, when homogeneously given to a tumor, gives the same mean surviving clonogen number as a heterogeneous, irradiated dose [41] and is defined as:

$$EUD = \left(\sum_j v_j \cdot D_j^a \right)^{\frac{1}{a}} \quad (13)$$

where D_j is the absorbed dose of partial volume v_j while a is a tissue specific constant. The EUD is useful when comparing two dose plans.

2.6 The Linear Quadratic Model and TCP

The Linear Quadratic (LQ) model is a commonly used model to describe cell survival in radiobiology. This model assumes that there are two ways to inactivate a cell when it is exposed to radiation, one proportional to dose and one proportional to the square of the dose. This model was first suggested by Chadwick and Leenhouts in 1973. In the 1960s it was shown that cell survival was proportional to chromosome breakage. Chromosomes are made of DNA (figure 13) which is a double stranded helix where each strand is built up by nucleotides which complements the nucleotide on the other strand. If an event occurs which breaks both strands, there will be a double strand break. And if only one of the strand breaks there is a single strand break. The breaking of chromosomes which were observed in the 1960s were the results of double strand breaks in the DNA. However, it was also shown that these chromosome damages were dependent upon dose in both a linear and quadratic way. Chadwick and Leenhouts suggested that double strand breaks could occur in two ways: either a particle broke both strands (linear) or two particles broke two strands (quadratic). This last event had to occur close enough in time and space so that the DNA did not have enough time to repair potential lethal damage. By their model, the expression for cell survival is:

$$S = \exp(-\alpha D - \beta D^2) \quad (14)$$

where S is survival, D is dose, α is the linear coefficient and β is the quadratic coefficient. Figure 14 plots the survival model in equation (14). It is not evident from the figure, but many tumor cells gain less from the shoulder than healthy tissue. Therefore a fractionated treatment gives the healthy tissue a better healing opportunity than tumor tissue.

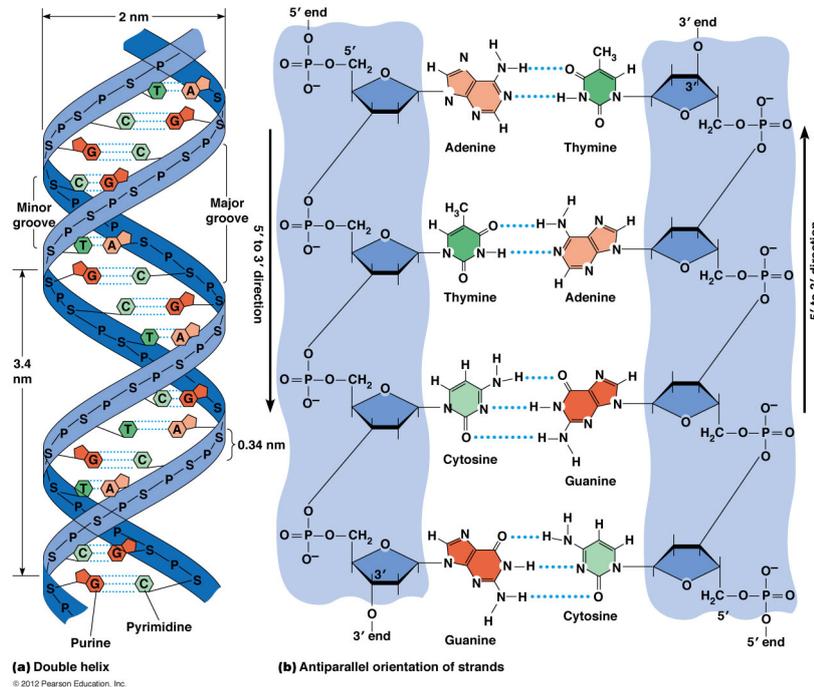


Figure 13: Illustration of DNA [42]. The nucleotides build upon each other to create pairs of Guanine, Cytosine, Thymine and Adenine in center of the DNA double helix. The light blue areas illustrates the strands in the double helix.

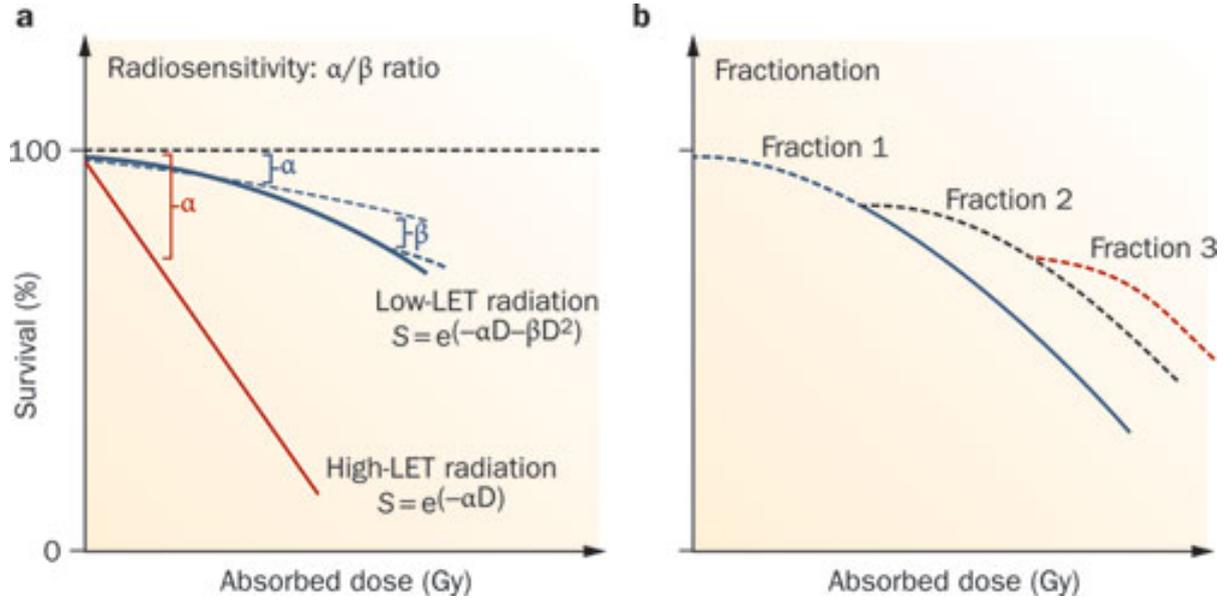


Figure 14: Illustration of the LQ model [43] where survival is a function of dose. The left plot shows high LET radiation and low LET radiation while the figure to the right shows the advantage of fractionated treatment as you regain the initial shoulder in cell survival.

α and β from equation (14) are constants and it should be noted that if the contribution from the linear and quadratic terms are equal, i.e. $\alpha D = \beta D^2$ then $D = \frac{\alpha}{\beta}$.

A good indication of how successful a treatment might be is the tumor control probability (TCP). For this thesis the following model, based on the linear quadratic formalism [44], was used:

$$TCP = \prod_i TCP_i, \quad TCP_i = \exp[-\rho_i V_i \exp(-\alpha'_i D_i + \gamma_i \Delta T)] \quad (15)$$

where $\alpha'_i = \alpha_i [1 + d_i / (\alpha_i / \beta_i)]$, ρ_i is the clonogenic density, V_i is the volume, α_i and β_i are the linear quadratic coefficients, and d_i is the fraction dose for voxel i . The TCP signifies the probability of inactivating all cancer cell, so $TCP=1$ implies that the tumor is eradicated.

2.7 NTCP

Normal Tissue Complication Probability (NTCP) of the parotid has been chosen to be a measure of how well the treatment spares OARs in this thesis. Complication is defined by Roesink et al. as stimulated parotid flow rate 25% of the preradiotherapy rate [45]. The formula used for NTCP calculation is the Lyman, Kutcher and Burman (LKB)-model [46–48]:

$$NTCP = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^u e^{-t^2/2} dt \quad (16)$$

where

$$u = \frac{d_{ref} - TD_{50}(v)}{m \cdot TD_{50}(v)}$$

$$TD_{50}(v) = TD_{50}(1) \cdot v^{1/n}$$

$$v = \sum_i \left(\frac{d_i}{d_{ref}} \right)^{1/n} \cdot \Delta V_i$$

where d_{ref} is the maximum dose to the parotid, $TD_{50}(1)$ is the tolerance dose for a homogeneous dose to an organ where 50% of patients are likely to experience a defined toxicity within 5 years, $TD_{50}(v)$ is the tolerance dose for a partial volume v . m is the steepness of the dose-complication curve for a fixed partial volume, n is the parameter which represents the "volume effect", i.e. how parallel or serial the dose response of the organ is. [48]. For parotid this parameter will be set to $n = 1$, i.e. the volume is assumed to be organized entirely in parallel.

2.8 Dose Painting by Numbers

In traditional radiotherapy the dose is planned using inverse planning where each ROI is defined and given a desired dose (or dose limit for OARs) and a weight factor according to its importance. An optimization function then tries to iterate itself to the best dose plan based upon these parameters. Dose Painting can also be done by the same optimization, but instead of giving a homogeneous dose to the tumor ROI one can use PET-images and either delineate the most aggressive sub-volume and elevate the dose given there or give a specific dose to each voxel based on the voxel's PET-intensity. The former technique is called Dose Painting By Contours (DPBC) and the latter is called Dose Painting By Numbers (DPBN) [49] [50].

The main focus of this thesis is DPBN. After the patient has taken a PET/CT-scan, an oncologist or nuclear medicine specialist delineates the tumor and malignant lymph nodes along with other structures of interest. It has been proposed [51] that the PET-intensities can be transformed to a dose prescription using the following formula:

$$D_i = \begin{cases} D_{low} & \text{for } I_i < I_{low} \\ D_{low} + \frac{I_i - I_{low}}{I_{high} - I_{low}} (D_{high} - D_{low}) & \text{for } I_{low} < I_i < I_{high} \\ D_{high} & \text{for } I_i > I_{high} \end{cases} \quad (17)$$

where D_i is dose voxel i , I_i is the PET-value for voxel i , I_{low} and I_{high} are the 5% and 95% percentage of the FDG uptake in tumor and lymphnodes, and D_{low} and D_{high} are the minimum and maximum doses respectively [16].

2.8.1 Coarseness of DPBN based upon PET

Radiotherapy planning software optimize the different optimization variables to create the optimal treatment for a patient placed in a certain machine (see section 2.5). This optimization is based upon the attenuation data found from the patient CT-scan.

2.9 QF

To compare the DPBN treatment plan with the prescribed DPBN-plan, we calculate the Quality Factor (QF) [16]:

$$QF = \frac{1}{N_T} \sum_i |Q_i - 1|, \quad Q_i = \frac{D_i^{plan}}{D_i^{pres}} \quad (18)$$

where N_T is the total number of tumor voxels, Q_i is the quality index for tumor voxel i which is the ratio of the planned dose D_i^{plan} to the prescribed dose D_i^{pres} . A prescribed dose plan which is identical to the planned dose plan will give a QF-value of 0.

2.9.1 QVH

A QVH is defined as the decumulative histogram of Q_i [52]. In contrast to QF a QVH centers around 1 and an ideal QVH is a descending step function where all values before 1 are given at 100% volume and all values after 1 is 0%, i.e. all voxel doses are the same in both plans. The QF thus becomes a measurement of the spread of the QVH curve around 1. [52]. An important value in this setting is the $V_{Q=X.X\%}$ which is volume percentage receiving at least XX% of the prescribed dose.

3 Material and methods

3.1 Acquiring Data

All patient data is part of the Prognostic value of PET-CT before radiotherapy of head and neck cancer study and have been anonymised. The study contains PET/CT-images for around 230 patients and 10 of these were chosen for this thesis. The criteria each patient had to fulfill to be chosen were that the tumor was larger than 12 cm^3 and that the max SUV value was higher 8 and the tumor could be separated from the background. Additional positive lymph nodes were also chosen to be treated if they had a mean dose larger than 70 Gy in the DPBN plan. Tumor size and SUV values were known from previous work with this data set. Table 1 contains an overview of all patients:

Patient	Gender	Age	Diagnosis	TNM	Lymph Nodes	HPV	MTV (cm^3)	SUV _{max}	Loco-regional Control
Patient 1	M	39	Tonsil	T4aN1M0	1	1	15.089	20.2608	0
Patient 2	M	64	Hypo-pharynx	T4aN3M0	0	0	46.528	13.5845	1
Patient 3	M	60	Tonsil	T3N0M0	2	1	16.321	15.7162	0
Patient 4	M	51	Tonsil	T4aN2bM0	2	1	24.397	15.3702	0
Patient 5	M	59	Hypo-pharynx	T4bN2bM0	2	2	30.75	19.301	1
Patient 6	M	66	Tonsil	T4aN0M0	0	2	29.657	13.5634	1
Patient 7	F	46	Base of tongue	T2N3M0	3	2	15.023	18.4358	0
Patient 8	M	64	Base of tongue	T1N2bM0	1	2	21.221	16.0566	0
Patient 9	M	68	Tonsil	T3N0M0	2	1	16.555	15.1179	0
Patient 10 (11)	M	59	Lower gum	T4bN2cM0	8	2	73.426	8.95632	0

Table 1: Patient information, most columns are self explanatory. MTV = Metabolic Tumor Volume and HPV = Human papillomavirus. The numbers in the HPV column stands for negative (0), positive (1) and unknown (2). Likewise locoregional control either is (0) or is not (1). Loco-regional control means that there has been no recurrence in either the tumor (local control) or the lymph nodes (regional control).

The PET scans were acquired on a Siemens Biograph 16 (Siemens Healthcare, Munich,

Germany). After more than 6 hours of fasting, about 300 MBq FDG was injected, and the patient rested in bed in a quiet, dimly lit room for about 1 hour before scanning. The imaging was performed on a flat bench top with head support, and a radiation therapy fixation mask. A

Gaussian post-reconstruction filter with full width at half maximum of 3.5 mm was used to reconstruct the image. This is similar to the scans performed by Dale et al. [53].

3.2 Scripting Dose Painting by Numbers

Dose Painting By Numbers (DPBN) can be implemented by converting PET data to a dose map. In this thesis, this is done by an IDL script, originally created by Marius Røthe Arnesen, and then developed further by adding the option to treat more than one volume at the same time with

DPBN. This features lets the user plan DPBN for both lymph nodes and tumors in the same process. The complete script can be found in appendix 8 in script `dpbn_max_norm_erh181017.pro`. Before running the script the user define which patient for whom he wishes to create dose plans. The program then reads the Digital Imaging and Communications in Medicine (DICOM) PET- and structure files. The RT structure files contains all the delineated regions of interest (ROIs) such as the GTV. A list of the structure names is displayed and the user is asked which ROIs the script should plan DPBN for. Once the structures are specified, the script creates a dose map by applying equation (17) to the PET information. This dose map is called the prescribed dose. The script also creates an inverse dose prescription file to create a heterogeneous basis to plan treatment upon as described in section 3.3.1. Equation (19) is used to calculate the inverse dose:

$$d_{inv} = d_{low} + d_{base} - d_{pres} \quad (19)$$

where d_{low} is 68 Gy, d_{base} is 50 Gy and d_{pres} is the prescribed dose. These numbers were chosen because 68 Gy is the standard guideline for head and neck cancers at Oslo University Hospital, and 50 Gy because the number had to be large enough to not get negative values when calculating d_{inv} . The PET-image along with both the prescribed and inverse dose plan is presented in figure 16.

The concept behind the implementation of an inverse dose map is illustrated in figure 15. By starting out with a prescribed dose (figure 15a) one can calculate the inverse dose (figure 15b) using equation (19). The summed dose of the inverse dose and a new dose distribution is optimized for 118 Gy (figure 15c). By only delivering the planned dose (figure 15d) to the patient, we deliver a voxel by voxel dose.

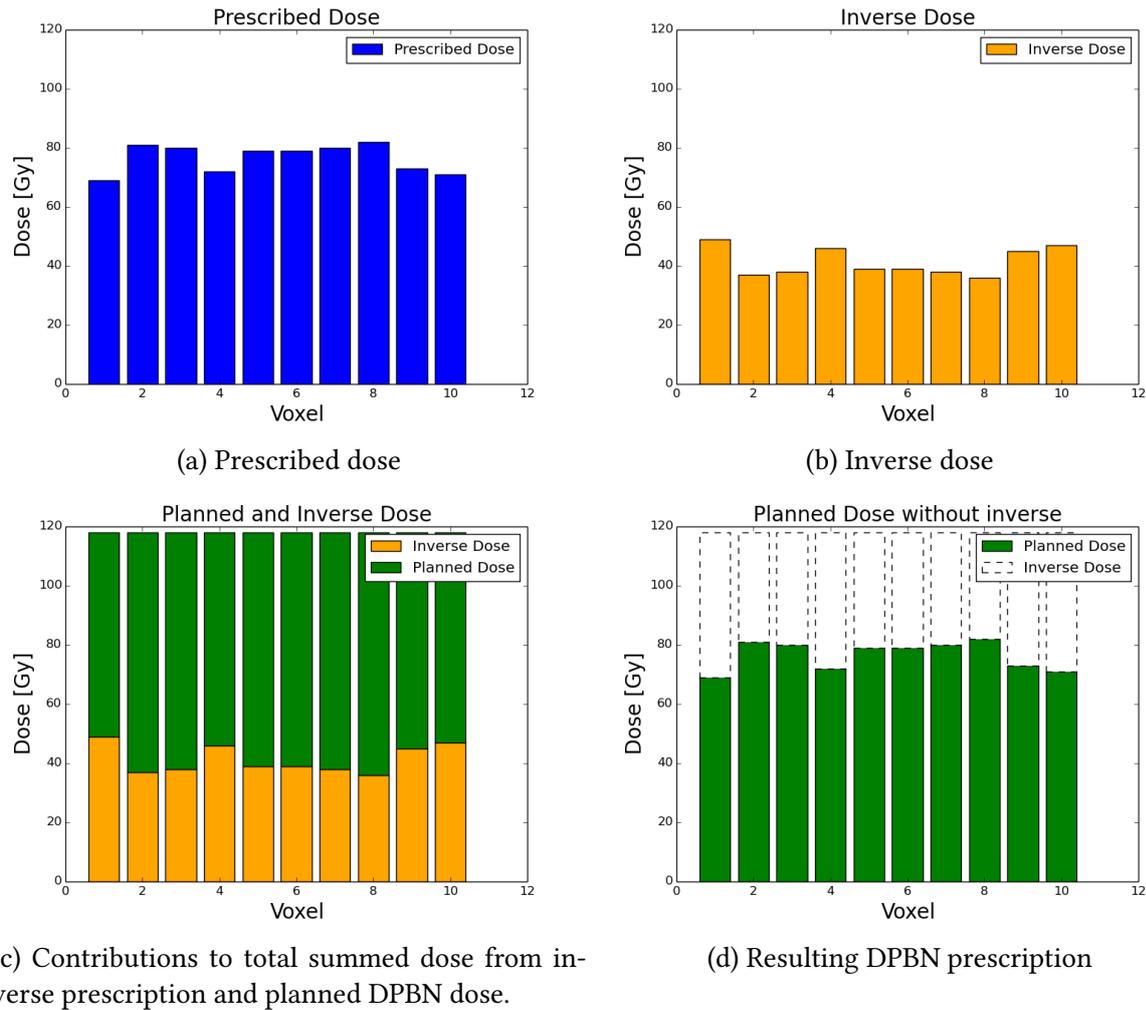


Figure 15: Illustration of the concept of using inverse prescription to create a DPBN plan.

3.3 Planning in RayStation Research

The prescribed and inverted dose plans were imported into RayStation Research (RSR) version 4.99. For all patients, there were multiple ROIs defined and many had unconventional naming. In this thesis, I made a script where the ROIs to be used as the GTV_{68} , CTV_{64} and CTV_{54} were chosen among the preexisting ROIs. Once chosen, the ROIs were correctly named and all ROIs of the same type were created according to table 2. The number behind ROIs indicates the dose they would be optimized for in a homogeneous dose plan according to standard guidelines at Oslo University Hospital. E.g. GTV_{68} is optimized for 68 Gy in a conventional plan. ROIs with "_5mm" behind the name and dose were corrected so that there was a minimum of 5 mm between the ROI and the outer contour. Likewise, ROIs with "_eks" behind the name and dose were made into a torus shape by extracting all GTVs, CTVs and PTVs contained inside it. Even though PTVs were not used, they were removed from CTVs to create room for dose gradients which were not to be constricted by optimization settings for CTVs.

The script also asked if the following OAR were present and named them thereafter: Submandibularis_L, Submandibularis_R, Parotis_L, Parotis_R, External, SpinalCord and SpinalCord_PRV. In the few cases where even more OARs were defined, they were manually renamed. The script was named `ROIcreator.py` and can be found in appendix 8.

For each patient, four plans were created in RSR; one robust (see section 3.3.2) photon plan, one robust photon DPBN plan, one robust proton plan and one robust proton DPBN plan. For each beam type the program requests the user to specify which machine the plans were intended for. The Varian TrueBeam STx (SB5) linear accelerator (linac) was chosen for photon treatment and the IBA_dedicated for proton treatment. Photons were given at 6 MV. The SB5 has 120 MLC leaves where the central 8 cm leafs have a thickness of 2.5 mm and the outer leaves are 5 mm thick. The maximum IMRT field size is 34 cm x 22 cm. During optimization, a constrain was put on the MLC leaves, only letting them to move 0.5 cm per degree rotation. The IBA_dedicated is a fictional machine within RSR with a nominal energy range from 69.80 MeV to 230.80 MeV with a spotsize of 0.37 cm x 0.37 cm at 160.20 MeV. For each plan the patient was irradiated from three angels: 70°, 180° and 290°. All beams were given the same isocenter in the middle of the tumor.

Each plan was created with a set of clinical goals (see table 4) which were the same for all plans. To reach these goals RSR uses inverse optimization as described in section 2.5 and the user defines a set of optimization settings with different weighting for each volume. Each beam module had an initial optimization setting that was the same for each plan. This means that the plans were created with a consistent optimization setting within each beam modality, thus largely avoiding user bias in the different plans. The initial settings were chosen in close collaboration with personnel at Oslo University Hospital and are presented in table 2. Initially, RSR would run 40 iterations upon these settings before the user was allowed to tweak them. Changes included changing the weight of optimization settings, removing or adding optimization settings and defining new ROIs where needed. Once the changes were made, the optimization would continue for as many iterations as the user wanted or until the optimal solution was found. For photons, the true beam dose was always calculated after the first 7 iterations.

If the clinical goals for GTV and CTVs were reached or about to be reached, the weighting of optimization settings for OARs were usually increased until the dose to OAR was well below the limit set as the clinical goal. In cases where accommodating for OARs dose restrictions were in dispute with dose coverage to tumor, the OAR dose restriction was ignored. The one exception was the Spinal Cord which where the clinical goal was always reached.

It should be noted that clinical goals differs from optimization settings in that the clinical goals are what we wish to achieve, while the optimization settings are what we work with to achieve these goals. Thus, the clinical goals presented in table 4 are achieved through adapting the optimization settings throughout the optimization.

ROI	Function	Optimization setting photon	Weight	Optimization setting proton	Weight
Parotid_R	Contains the right parotid	Max EUD 26.00 Gy, Parameter A1	1	Max EUD 26.00 Gy, Parameter A1	1
Parotid_L	Contains the left parotid	Max EUD 26.00 Gy, Parameter A1	1	Max EUD 26.00 Gy, Parameter A1	1
Submandibular_R	Contains the right sub-mandibular	Max EUD 39.00 Gy, Parameter A1	1	Max EUD 39.00 Gy, Parameter A1	1
Submandibular_L	Contains the left sub-mandibular	Max EUD 39.00 Gy, Parameter A1	1	Max EUD 39.00 Gy, Parameter A1	1
SpinalCord	Contains the Spinal Cord	Max Dose 50.00 Gy	1	Max Dose 50.00 Gy	1
SpinalCord_PRV	A security margin around the SpinalCord	Max Dose 52.00 Gy	1	Max Dose 52.00 Gy	1
GTV68	Contains the tumor and all lymph nodes to be treated with DPBN	Unfirm Dose 68.00 (118.00) Gy	50	Unfirm Dose 68.00 (118.00) Gy	50
GTV68		Min DVH 66.00 Gy to 99% volume*	100	Min DVH 65.50 Gy to 99% volume*	100
GTV68		Min DVH 64.60 Gy to 99% volume	0	-	-

Table 2: ROIs used during treatment planning. Each ROIs function is described to explain why it was added in the second column. Third and fifth column contains the optimization functions applied to each ROI with the weighting in the column to the right. Note that each ROI can have more than one optimization function applied to it. If an optimization setting has * behind it, it was optimized robust. If an optimization has a dose inside brackets, that is the dose planned for in a DPBN plan. Parameter A1 means that $a = 1$ for the EUD. Continued in table 3.

ROI	Function	Optimization setting photon	Weight	Optimization setting proton	Weight
CTV64_eks_5mm	Margin around the tumor to ensure adequate dose to the subclinical disease around the tumor	Min DVH 62.30 Gy to 99% volume*	100	Min DVH 62.00 Gy to 99% volume*	100
CTV64_eks		Uniform Dose 64.00 Gy	1	Uniform Dose 64.00 Gy	10
CTV64_eks		Max Dose 68.00 Gy	1	Max Dose 68.00 Gy	10
CTV54_eks_5mm	Margin around the tumor to ensure adequate dose to the subclinical disease around the tumor	Min DVH 52.30 Gy to 99% volume*	100	Min DVH 52.00 Gy to 99% volume*	100
CTV54_eks		Uniform Dose 54.00 Gy	1	Uniform Dose 54.00 Gy	10
CTV54_eks		Max Dose 64.00 Gy	1	Max Dose 64.00 Gy	5
External	Margin around the external contour of the patient	Max Dose 68 (varied)	1	Max Dose 71 (varied)	10
External		Dose Fall-Off: Dose should go from 68 Gy to 34 Gy over 3.00 cm	1	Dose Fall-Off: Dose should go from 68 Gy to 34 Gy over 2.00 cm	10
External		-	-	Dose Fall-Off: Dose should go from 68 Gy to 0 Gy over 6.00 cm	10

Table 3: Continuation of table 2.

3.3.1 Dose Painting By Numbers in RayStation Research

RayStation Researchs (RSRs) graphical user interface allows optimizing for each ROI in the patient. The software in itself can access each voxel and assign individual doses, but the interface restricts the user to optimize a homogeneous dose to each ROI. Dose Painting By Num-

bers (DPBN) is a voxel by voxel method and RSR needs to be "tricked" into treating each voxel independently, while still using the interface normally. The solution to this problem was published by Arnesen et al. [16]. RSR gives the user the option to implement a dose map of previously given treatment to create a new, summed plan. In clinical practice, this is used if the patient has already been treated and the planner wants the total dose given to include the previously given dose. Thus, the total patient dose can be optimized by creating a second dose plan which accommodates for the previously deposited dose. By importing the inverse dose map described above (section 3.2) and selecting this dose map as a previously given dose, this provides the basis to plan a new, heterogeneous dose prescription map. If one plans a homogeneous summed dose with RSR, the secondary plan will become heterogeneous if the first plan is heterogeneous. The summed dose was set to be $d_{base} + d_{low} = 118\text{Gy}$ as this gives the prescribed dose. By shaping the inverse dose map correctly, the secondary map becomes the intended heterogeneous DPBN dose distribution. One can choose to treat the patient with only the secondary dose and thus give a voxel by voxel treatment as illustrated in figure 16.

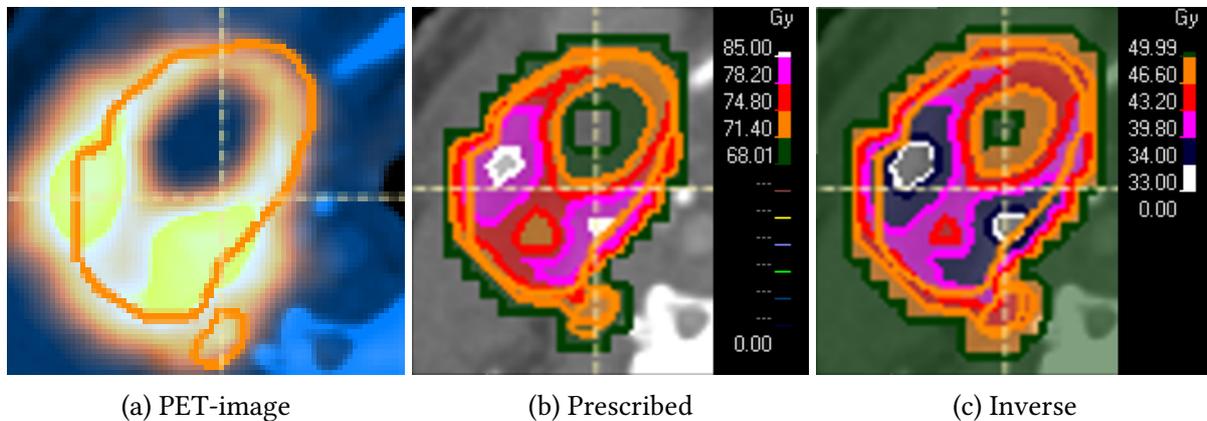


Figure 16: The PET-image, prescribed dose map and inverse dose map for patient 7. The orange line is the GTV border.

3.3.2 Robust Treatment

One of the main issues in a standard treatment plan is that the patient will never be positioned exactly the same between each treatment fraction. However, the treatment is usually planned upon CT data where one assumes that the patient lies approximately in the same place each time. There are a number of actions taken to prevent this, but there will most likely be a difference in the patient position from fraction to fraction. To accommodate for this movement and other non-biological uncertainties, a treatment plan would normally be planned with a PTV volume around the CTV. However, this is not the only option and in this thesis robust treatment as described in section 2.5 is used. RSR allows each ROI to be optimized robustly, and the hypothetical motion was set to 6 mm to ensure adequate dose coverage to all CTVs up to a displacement of 3 mm. The displacement was set to the double of the needed distance as the optimization tends to fail near the edge. Another advantage of planning robust in RSR is that you can also add an uncertainty in density (e.g. from uncertainties in CT HU to density conversion). This uncertainty was set to 3% for all patients except patient 1. All robust optimization settings were defined according to recommendations from experienced staff at Oslo University Hospital. GTV_{68} , CTV_{64} and CTV_{54} were set to be robustly optimized with a minimum dose as optimization goal as described in table 2.

3.4 Dose Plan Evaluation

After a set number of optimization iterations, the optimization stops and the user can evaluate the dose distribution to decide if the plan is adequate, if the optimization process should continue or if it should start over. This evaluation is done in several ways. The quickest way to evaluate is to see how many clinical goals are fulfilled. Another method is to scroll through each slice of the patient and visually evaluate the conformity and dose coverage. Lastly, one can use RSR's perturbed function to move and/or change the density of the patient.

3.4.1 Clinical Goals

The clinical goals are a measurement of how well the optimization functions performs, i.e. how close we are to the desired result. Clinical goals used in this work are presented in table 4 and a good dose plan will fulfill every goal. Note that PTVs are defined here, but not optimized for in tables 2 and 3. This is because these clinical goals were used as an evaluation of the robustness of the plan. It is important to note that some goals are more important than others. E.g. coverage of GTV and CTV and low dose to the spinal cord is very important. One of the Submandibular and parotid glands on the other hand, may be sacrificed if this helps the previous mentioned clinical goals to be achieved. As many of the head and neck tumors in this thesis were placed on one side of the head, one would try to spare the glands on the opposite side (contralateral) as much as possible. The spinal cord on the other hand will cause paralyzation if the dose is too high, which therefor must be restricted..

ROI	Clinical Goal
SpinalCord_PRV	At most 52.00 Gy dose at 0.0 cm ³ volume
SpinalCord	At most 50.00 Gy dose at 0.0 cm ³ volume
Parotid_L	At most 26.00 Gy average dose
Parotid_R	At most 26.00 Gy average dose
Submandibular_L	At most 39.00 Gy average dose
Submandibular_R	At most 39.00 Gy average dose
GTV68	At least 64.60 Gy dose at 98.00% volume
CTV64_eks	At least 60.80 Gy dose at 98.00% volume
CTV64_eks_5mm	At least 60.80 Gy dose at 98.00% volume
CTV54_eks	At least 51.30 Gy dose at 98.00% volume
CTV54_eks_5mm	At least 51.30 Gy dose at 98.00% volume
PTV68	At least 64.60 Gy dose at 98.00% volume
PTV64_eks	At least 60.80 Gy dose at 98.00% volume
PTV64_eks_5mm	At least 60.80 Gy dose at 98.00% volume
PTV54_eks	At least 51.30 Gy dose at 98.00% volume
PTV54_eks_5mm	At least 51.30 Gy dose at 98.00% volume
External	At most 70 (varied) Gy at 2.0 cm ³ volume

Table 4: The clinical goals for the treatment plans

3.4.2 Visual evaluation

By viewing each image slice in RSR one can easily discover any irregularities in the dose map. Common problems to discover this way are nonconformity, hot/cold spots and dose spillage.

Nonconformity happens when the decrease in dose beyond the border of the ROI is not steep enough such that the dose does not conform to the borders of the intended ROI. To make the dose more conformal we increase the weighting of the "dose fall-of" optimization setting (table 2). Hot spots occur when the optimization algorithm creates excess dose in small areas where the optimization functions are less strict. The algorithm might do this to achieve satisfactory dose coverage of a nearby ROI, but the result is a spot in the dose distribution that receives too much dose. The opposite might also happen inside ROIs and then it is called a cold spot. Dose spillage is similar to hot spots in that excess dose is stored elsewhere, but instead of a small area receiving the dose, the dose is spilled over a larger area. Both hot spots and dose spillage can be restricted by creating helper ROIs over the affected area and setting stricter optimization criteria on the newly created ROIs. Dose spillage can also sometimes be fixed by increasing the weighting of the "Dose Fall Off" optimization function. Cold spots can be dealt with by introducing or increasing the weight of an optimization function demanding uniform dose to the ROI. Examples of hot spots and dose spillage are presented in figure 17

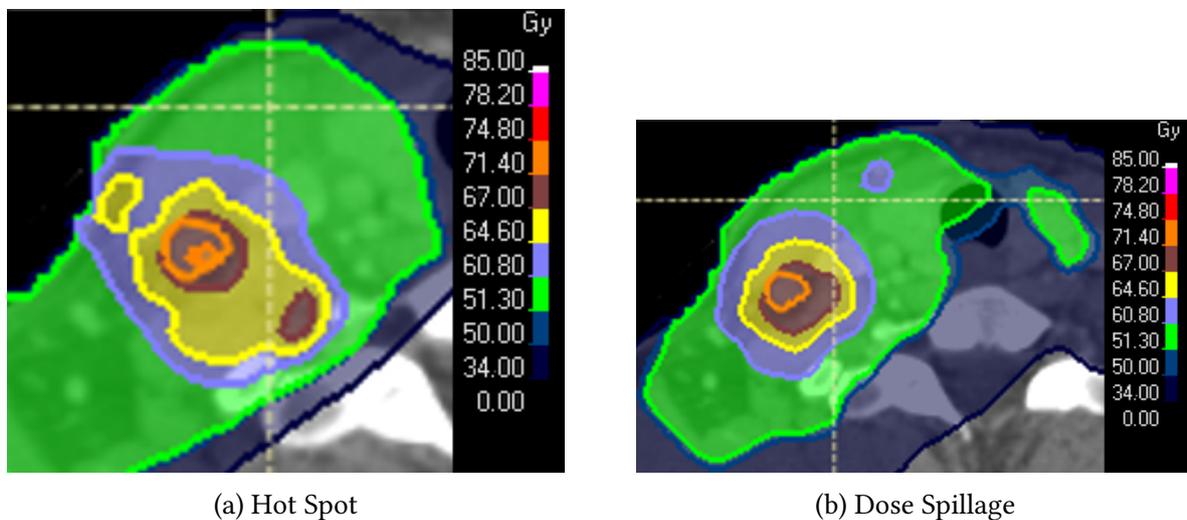


Figure 17: Hot spot and dose spillage from two of the plans for Patient 7. The brown spot in (a) is a hot spot while the green dose to the right in (b) is dose spillage. In the latter case there is no apparent reason to why the dose should be so high in the spillage region.

3.4.3 Perturbing the Dose

RSR is equipped with a number of handy evaluation scripts. One of these is the perturbed dose option. By using this function one can perturb the planning basis (CT and ROIs) in x, y or z direction as well as change the density of the patient. This can be used to evaluate how robust the dose distribution is. During the optimization the clinical goals for PTV_{68} , PTV_{64} and PTV_{54} were used as an indication of how robust the treatment was. However, this was not an adequate evaluation in the long run and to truly see if the dose was robust one should perturb the dose and then look at deviations from clinical goals in the perturbed dose maps clinical goals. This way one can see which dose the patient would receive in each ROI if the patient moved e.g. 2 mm to the left. To call a dose plan robust, the plan had to be perturbed in different directions and still fulfill the clinical goals.

3.4.4 Exporting Data

To evaluate our dose plans even further, we exported the dose from RSR and made the data available for analysis in Interactive Data Language (IDL). Exporting dose from RSR was done by the python script `export_dose.py` (section 8) who wrote the dose matrix to a `.dat` file along with other relevant information such as dimensions of dose matrix, voxel size, position of dose matrix's top left corner, number of fractions and total number of voxels. The dose was stored in cGy. The index of each ROI was also stored in `.txt` files for each patient.

3.4.5 Evaluation of Dose Painting by Numbers

3.4.5.1 Finding a good TCP

TCP gives the probability of inactivating enough clonogenic cells in the tumor such that the tumor will not come back post treatment. This values gives a good indication about the quality of the plan when one evaluates the dose to tumor. TCP can be found from equation (15). However, there are some issues with using this function directly. The model assumes that a full voxel-by-voxel map of both clonogen density and radiation sensitivity is available. In this thesis one assumes that the FDG-uptake is a representation of the clonogen density, and therefore one can express ρ as $SUV \cdot k$ where k is a constant. To find this constant, a known survival rate of the patients used in this thesis ($\simeq 70\%$) were used as a measure of TCP. By setting TCP equal to 0.70 and the dose to tumor to 68 Gy in each tumor voxel for each patient, one can find k through the least squares method. 68 Gy (in 34 fractions) was used since this was the total dose delivered to the patients.

3.4.5.2 Calculating QF, TCP and NTCP

How well the Dose Painting by Numbers doses were delivered were evaluated using the QF. This value was calculated using equation (18) as described in section 2.9. All plans were also evaluated by calculating the TCP (equation (15)) for the tumor and the NTCP (equation (16)) for the parotids. To calculate the equations mentioned above, one needs the dose distribution. As mentioned above, the dose and other information was stored in `.dat` files. These files were read by IDL using the script `read_dat.pro` whenever used. For most of my calculations I used the script called `dat_treater2.pro`. This script reads inn the dose for all patients and then calculates QF and TCP among others and writes the results to file. The results were then read by a Python script and plotted. NTCP was calculated in a separate script, `NTCP.pro`, which works in a similar fashion. The script takes a list of patients and a list of ROIs, reads the `.dat` file for each patient and calculates the NTCP for each ROI. It should be noted that the NTCP was not found for Patient 5 as that patient had no clear ipsilateral or contralateral parotid gland.

When finding TCP, SUV was read through `read_dat.pro`. Using this script works even though we do not read a dose matrix, as the PET-matrix and the dose matrix are the same size. V_i was read in with the dose, D_i , using `read_dat.pro`. $\gamma_i \Delta T$ was set to 0, α/β was set to 10 and α was calculated using equation (15) with $\gamma_i \Delta T = 0$, where TCP_i was set as a survival fraction of 0.48 after irradiating one fraction with 2 Gy and $\phi_i V_i = 1$ according to recommendation from professor Eirik Malinen at Oslo University. This gave a α of 0.31 which was assumed to be the same for all voxels. For NTCP, d_{ref} was set to the max dose in the tumor, $TD_{50}(1)$ was set to 40.07, m to 0.41 and n equal 1.

During radiotherapy, tumor shrinkage and patient movement may occur. To evaluate QF and TCP for a moving patient, the tumor was moved up to 3 mm in each direction. This was done post

planning by moving the dose matrix using the script `translate_volume2.pro` which utilizes the function `transform_volume.pro` to translate the dose matrix. As the voxels are 3x3x2 mm in size, the function interpolated all data where the movement was not a multiple of 3 (x- and y-direction) or 2 (z-direction). The translation was done by moving the whole dose matrix for either the photon DPBN plan or the proton DPBN plan. QF and TCP were then evaluated for the same indexes which would contain the tumor in a non-translated dose map to look at the change compared to a non-moving tumor. Moving the tumor and calculating the QF also gives an indication of if one can keep the DPBN GTV non-robust and still call the treatment robust.

Similarly, the script `shrink_volume2.pro` accepts the prescription maps and dose maps from RSR and shrinks them using a numerical procedure. The DPBN dose maps can be the original dose maps or translated - as described above. An illustration of the shrinking process can be found in figure 18. To be able to find the tumor center of mass throughout the whole shrinking process, there was created a zeros-and-ones matrix where each tumor voxel was given a value of 1 and the rest were given 0 (figure 18a). This zeros-and-ones matrix was shrunk along with the original dose plan. Once both matrices were shrunk, the change in center-of-mass was found in the zeros-and-ones matrix. As the shrunk dose map was smaller than the original, but one still wants the same dose map dimensions, we moved the shrunk dose map into an empty matrix with the same dimensions as the original dose map (figure 18b). The shrunk dose map was then positioned such that the center of mass in the shrunk tumor was placed in the same spot as the original one (figure 18c).

The shrinking can be done either relatively or absolutely. If one shrinks relatively, the script is given the fraction of the original volume one wants to remove. E.g. by asking for a 0.4 shrinking, there will remain $0.6 \cdot V$ where V is the original volume. In contrast, shrinking the absolute volume is done by first encompassing the whole tumor with a hypothetical sphere. The user can then specify how much the radius of this sphere should be reduced, and the tumor is shrunk according to this. When shrinking, the prescribed dose map was shrunk. However, for translation, the dose maps were translated.

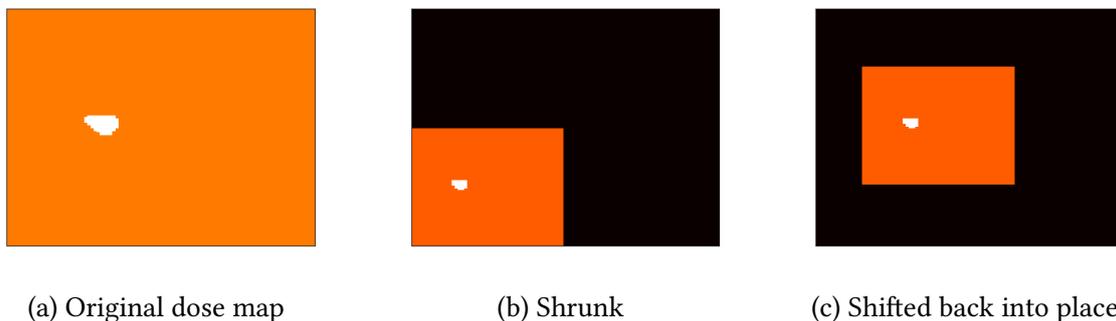


Figure 18: Numerical shrinking of a zeros-and-ones matrix. The white represent the tumor and the orange represents the original dose map.

By shrinking and moving both the prescribed dose map and the ones-and-zeroes dose map as described above, one can find QF and TCP by comparing the planned DPBN dose maps with the shrunk prescribed dose maps in the voxels belonging to the shrunk tumor, i.e. where the ones-and-zeroes dose map is equal to 1. Keeping the DPBN dose maps the same and shrinking the

prescribed dose map is the equivalent of using the same treatment plan throughout the whole treatment while the tumor shrinks.

4 Results

4.1 Creating Dose Maps

In this work, four treatment plans were created for each patient using RSR. Figure 19 and 20 show the PET image and consequent dose maps for patient 7. Both the CTV_{64} and CTV_{54} are corrected for distance to the outer contour (not shown) and GTVs, CTVs and PTVs within them are extracted.

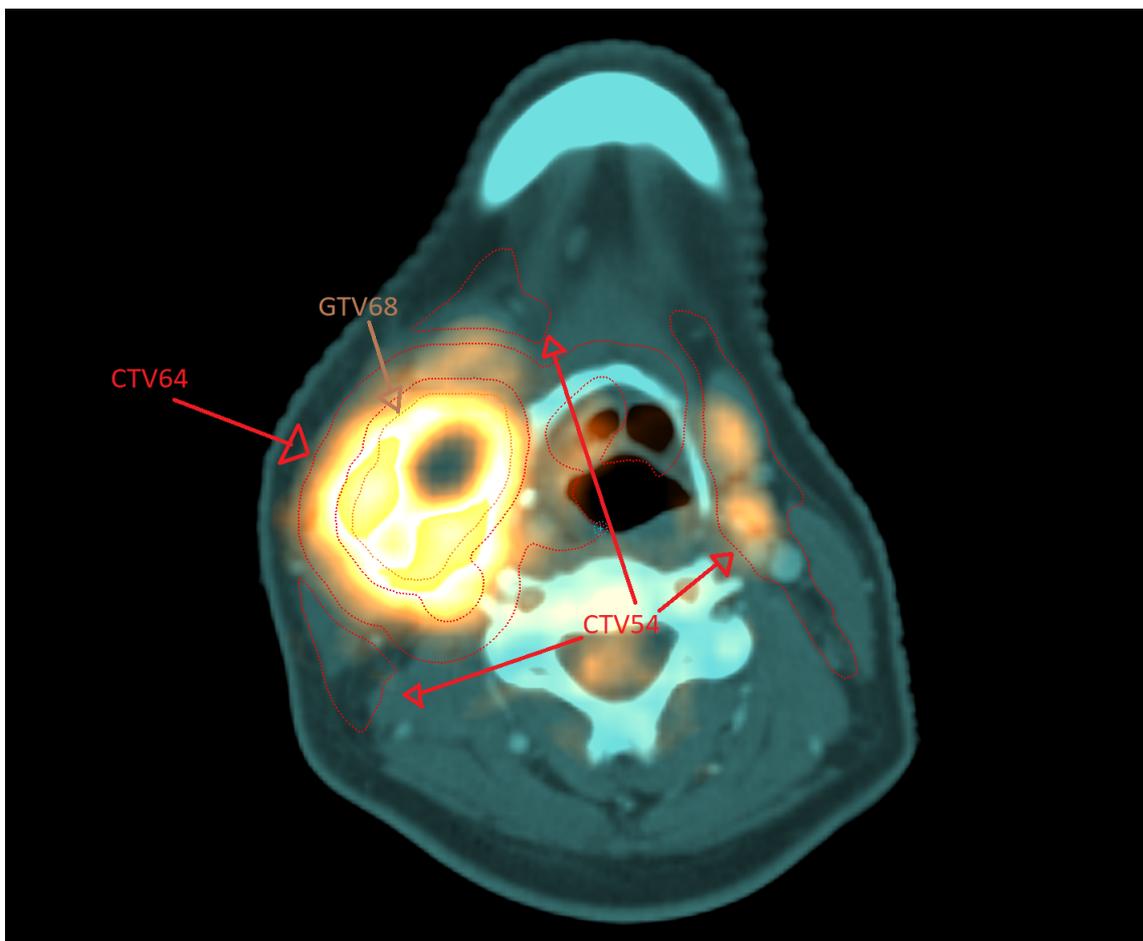


Figure 19: PET information for Patient 7 superimposed upon the CT data. The orange line outlines the GTV_{68} while the red lines outlines CTV_{64} and CTV_{54}

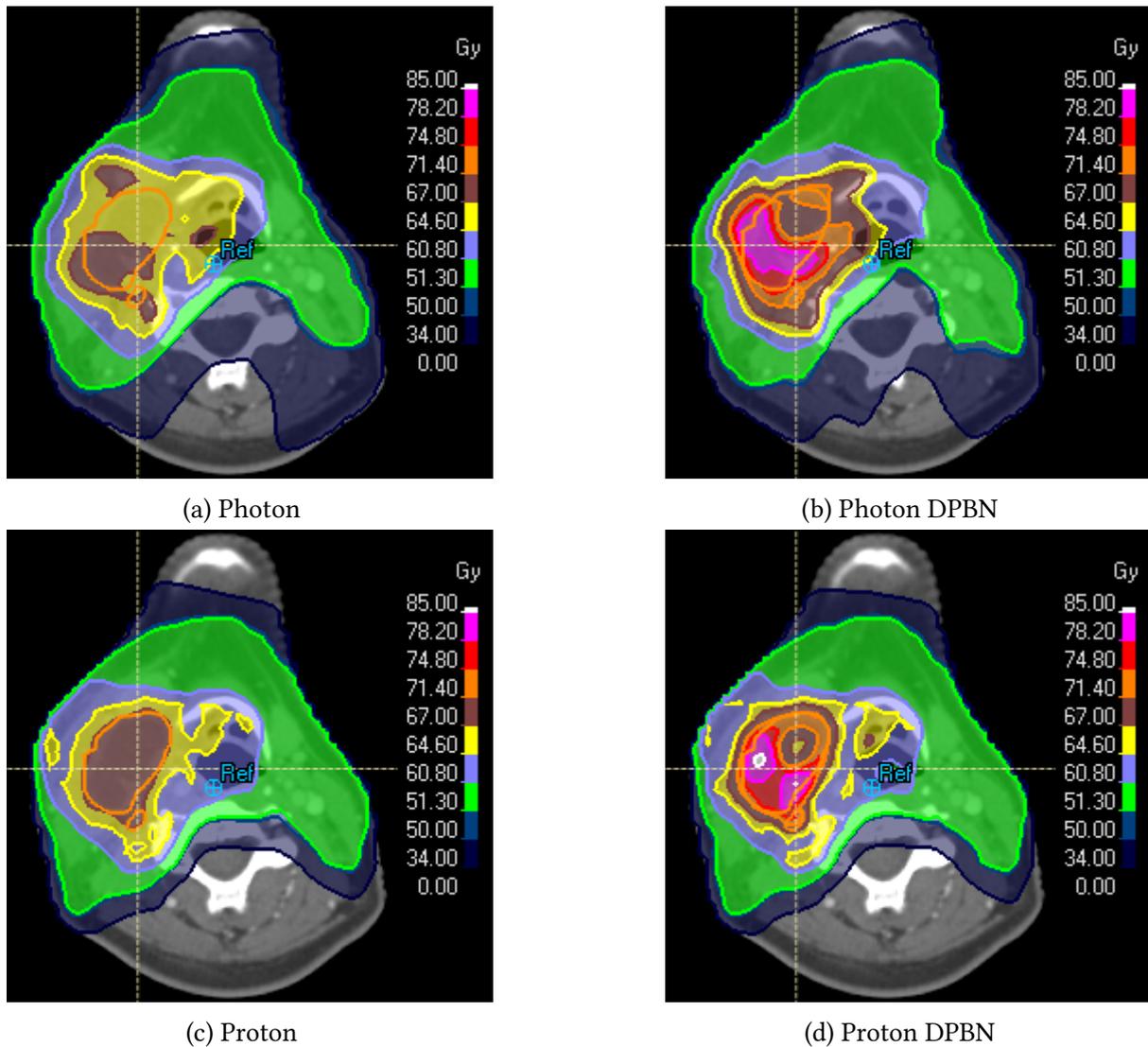


Figure 20: Four complete dose plans for Patient 7 at the same slice as for figure 19. The orange line outlines the GTV and the shaded contours show isodose levels indicated in the color bars.

As seen in figure 20, the full dose distribution in the patient reflects the spatial location of the tumor and affected lymph node areas. Moreover, for DPBN we see that the dose distribution within the tumor is more or less a mirror image of the ^{18}F FDG-PET uptake shown in figure 19. Also, it may seem like protons conform better to the PET distribution. It should also be mentioned that there seems to be a hot spot towards the middle of the patient, this is not so as there is a ROI directly above this slice which starts gaining dose already in this slice. Images of the dose plan can only provide info for one slice.

To achieve a better understanding of the dose distribution in each plan one can create DVHs. The DVHs for GTV_{68} , CTV_{64} and CTV_{54} are plotted in figure 21 where the following can be observed: for conventional radiotherapy protons and photons give virtually the same tumor dose distributions. Furthermore, the DPBN distributions are very similar in GTV_{68} . However, for CTV_{64} and CTV_{54} , photons tends to give a tail of high doses to these volumes.

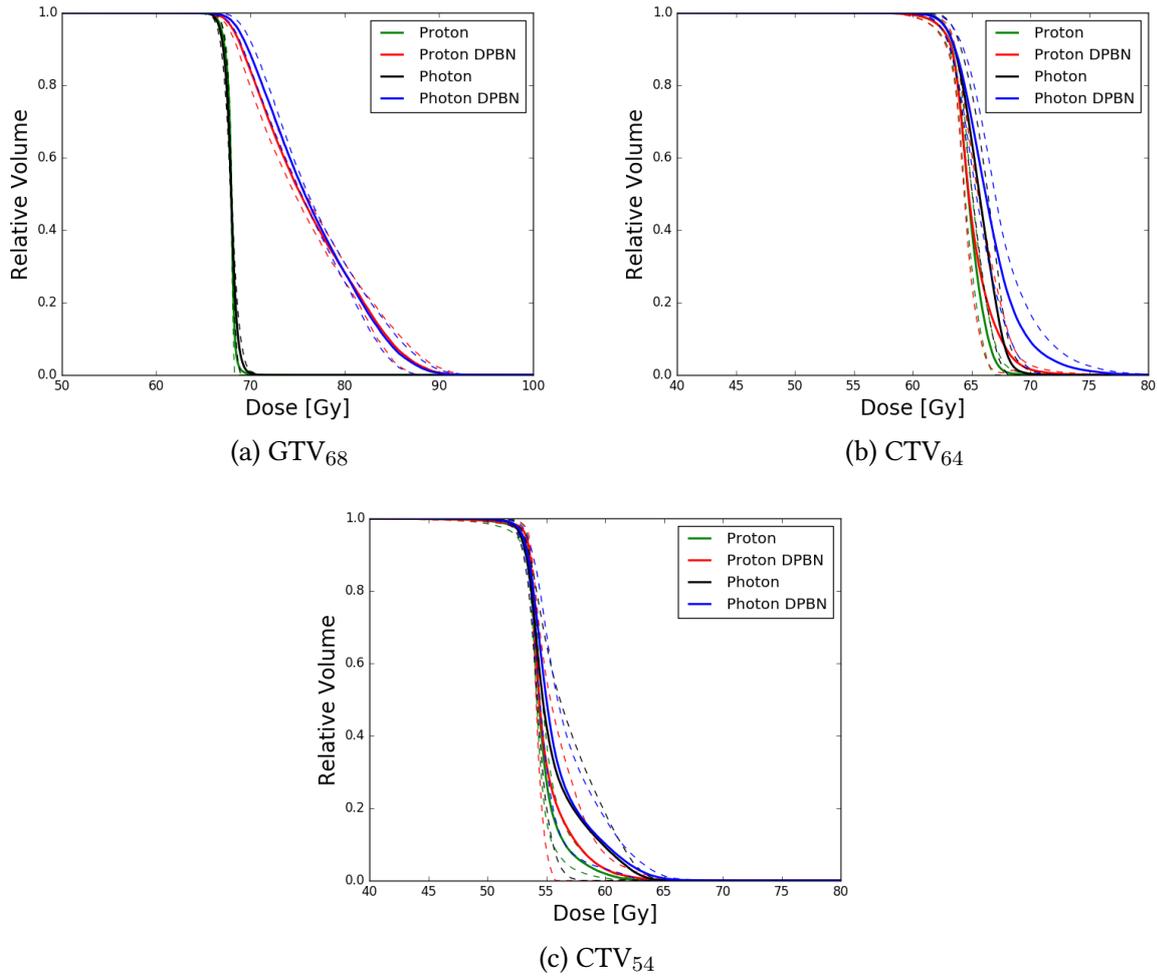
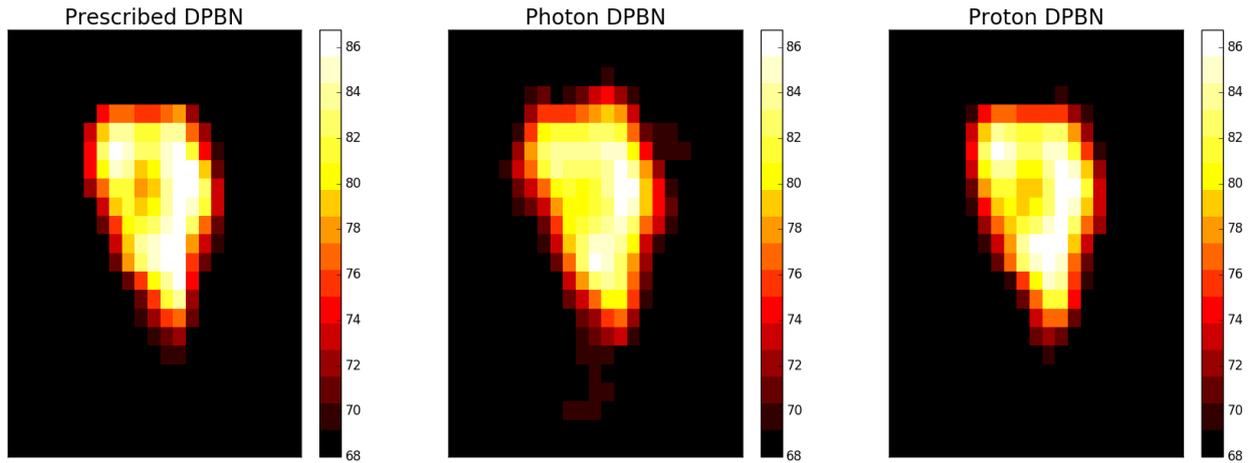


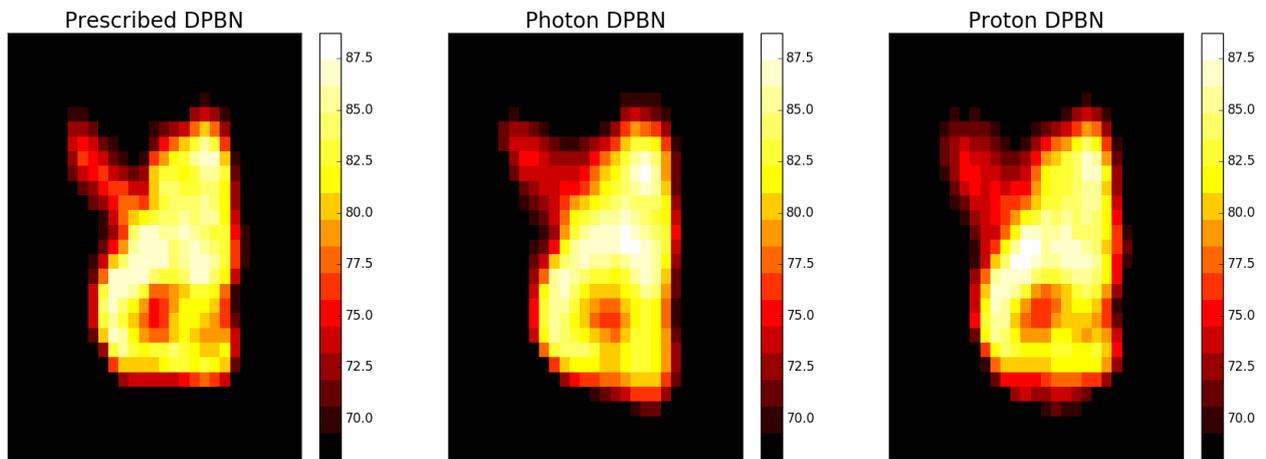
Figure 21: Mean dose volume histograms over all patients for different ROIs. The whole lines are the mean value and the dotted ones display one standard deviation.

4.2 Quality of Treatment Plans

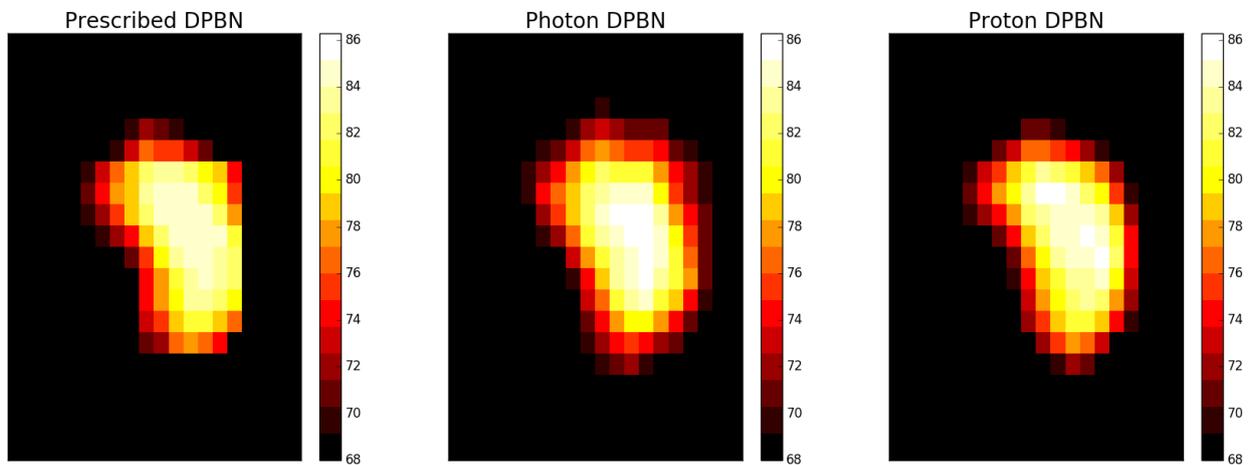
To see the differences between each DPBN plan and the prescribed dose to each patient, each tumor was plotted in the slice containing the prescribed dose plan's center of mass, i.e. the center of mass of the tumor. A comparison for all patients can be seen in figure 22. Here it is apparent that both photons and protons can provide DPBN with quite high spatial accuracy. For some of the patients, the differences in dose between photons and protons is minimal (patients 1, 2, 3, 7, 9 and 10). For the other patients, some differences are visible. Where protons tend to conform better to the PET.



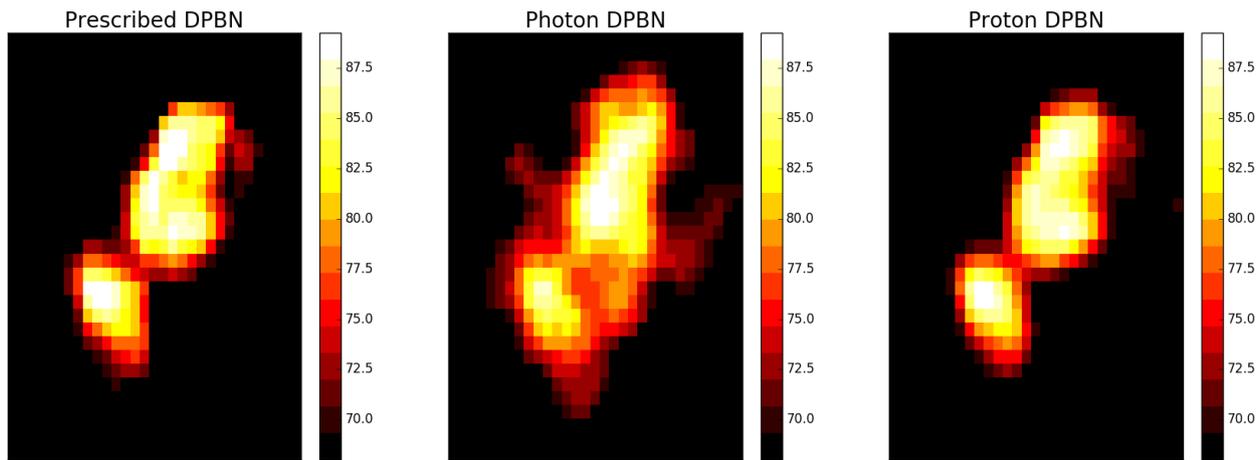
(a) Patient 1



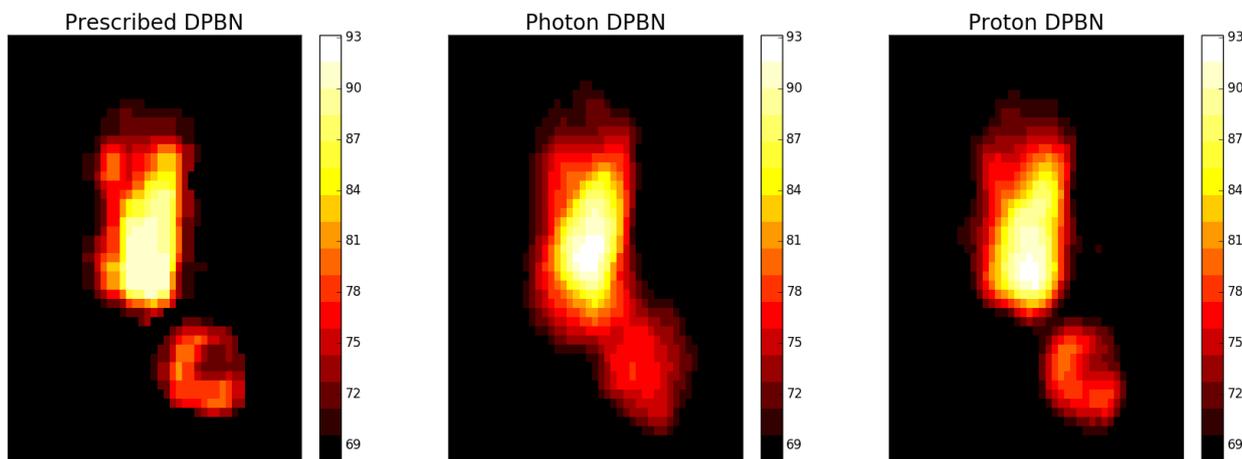
(b) Patient 2



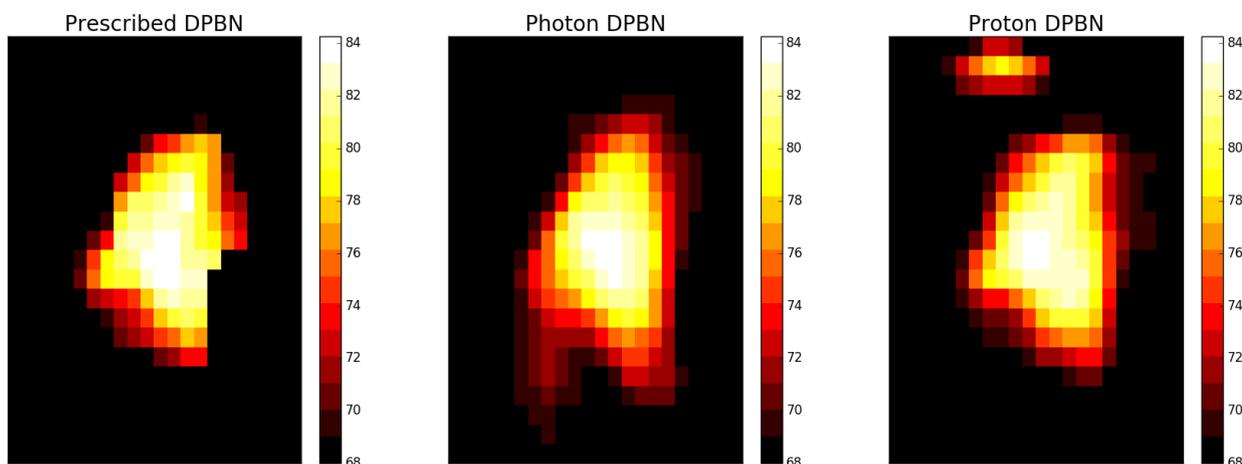
(c) Patient 3



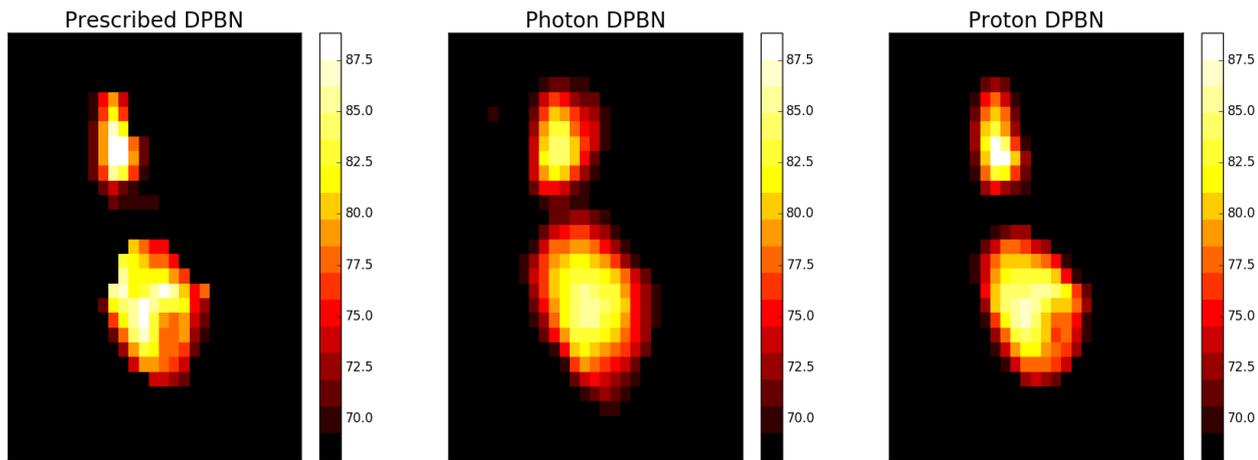
(d) Patient 4



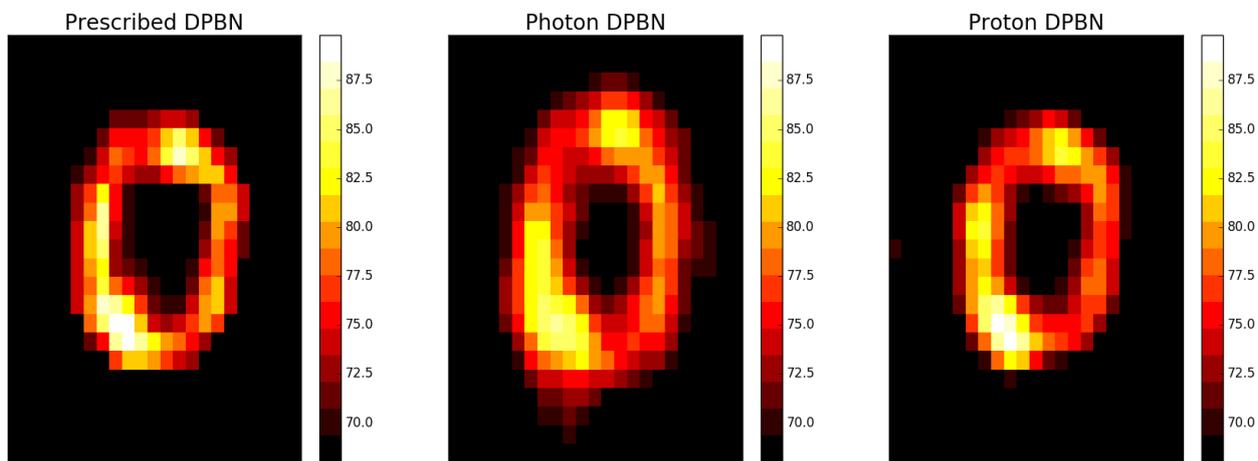
(e) Patient 5



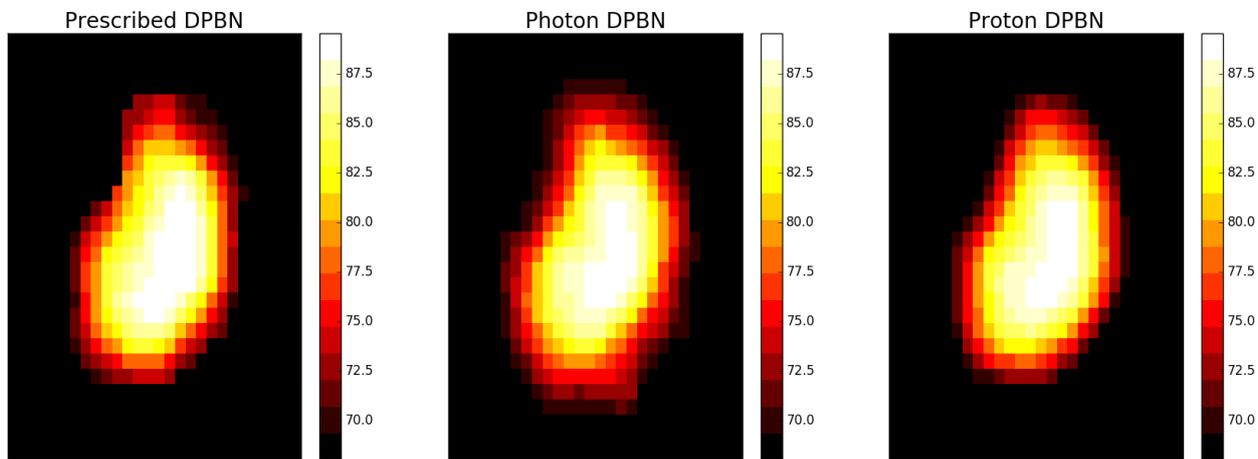
(f) Patient 6



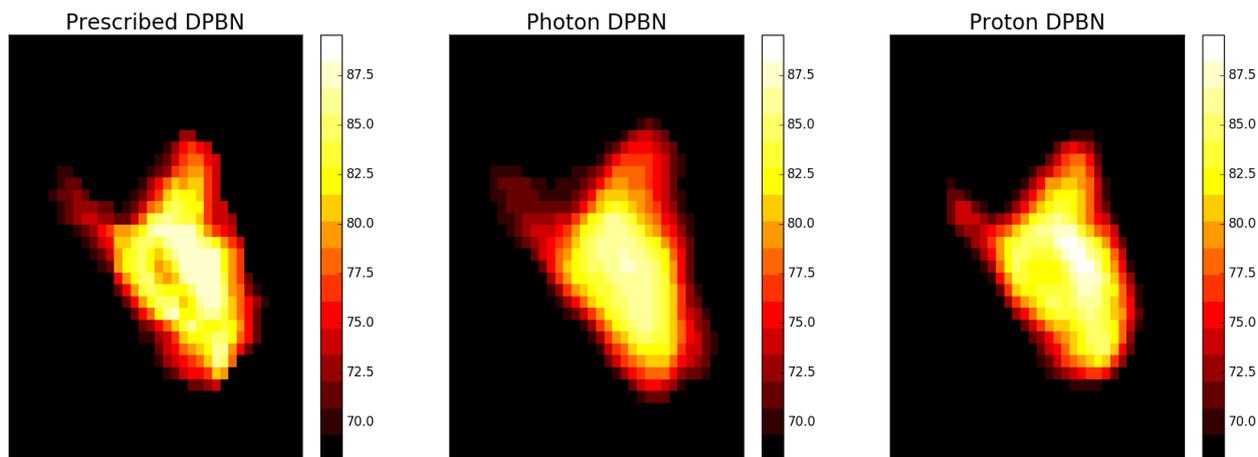
(g) Patient 7



(h) Patient 8



(i) Patient 9



(j) Patient 10

Figure 22: Comparison between prescribed plan, photon plan and proton plan for each patient's tumor. The image slice is the slice where one finds the prescribed dose plan center of mass. For the photon and proton plans the plans were cut of at 68.01 Gy.

It is apparent that the dose is not perfectly given according to the prescribed plan. Still, protons are seen to conform more closely to the prescription map compared to photons. However, the figures in figure 22 only illustrates one tumor slice. As the dose map is a 3D structure, one slice may give indications, but one needs more information to make any conclusions. For this purpose, the voxel dose of the proton DPBN and photon DPBN plan were plotted against the the same voxel's dose in the prescribed dose plan. These results are presented in figure 23.

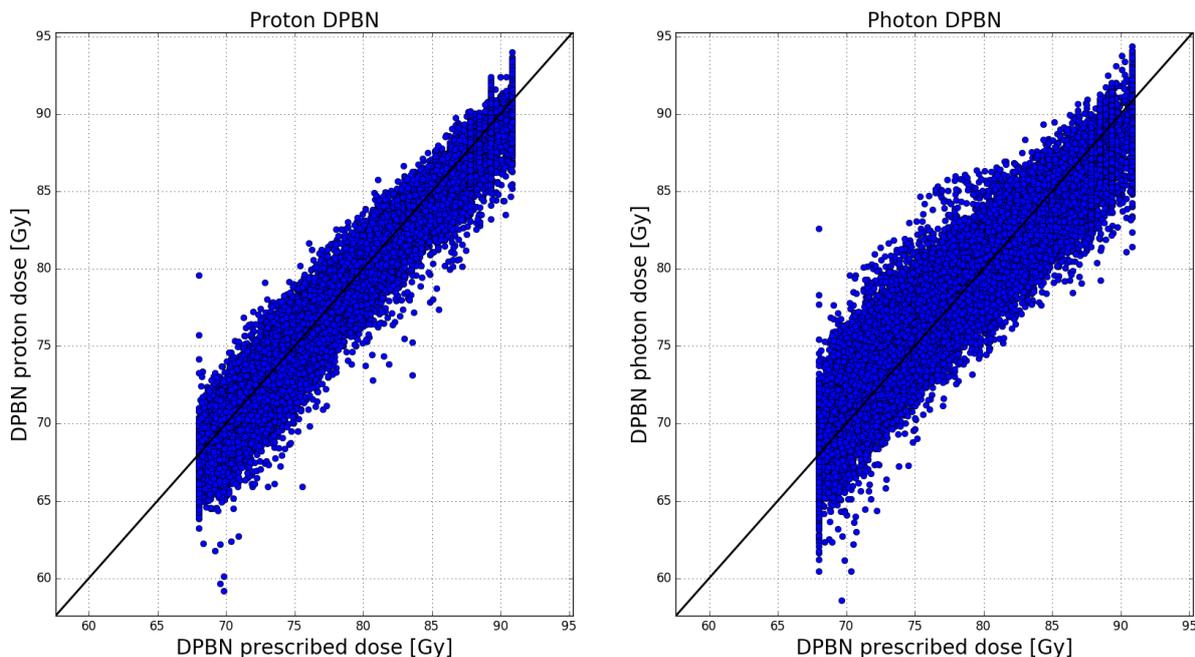


Figure 23: Comparison between prescribed plan, photon plan and proton plan for all voxels in all patients. Each dot represents a voxel. The black line represents the identity line, i.e. where the plan is equal to the prescription.

Ideally, all dots in figure 23 should lie atop of the identity line. This is not the case, and substantial scatter is observed showing that the prescription can not be fully reconstructed by protons or photons. For a given proton prescription dose, we typically see a variation in the planned dose of around 4-5 Gy. While for photons, the variation is larger, about 5-6 Gy. Also, for photons there is a slight tendency for the data not being symmetricly distributed around the identity line.

A measurement of the quality, i.e. how close the DPBN plans are to the prescribed dose, is given by the Q_i as described in section 5.4.1. Figure 24 illustrates the QVH for all patients while table 5 contains the $V_{Q=95\%}$ and $V_{Q=105\%}$. Again, it appears that photons have an inferior dose distribution compare to protons, and again we see a tail of voxels with high doses. This is also illustrated by the much higher $V_{Q=105\%}$ for photons (table 5).

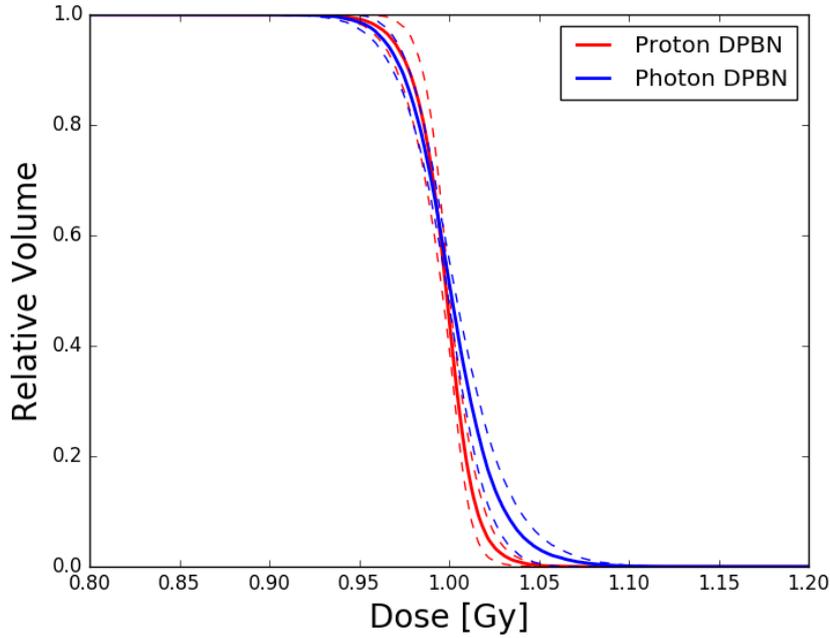


Figure 24: Quality Volume Histogram over the average Q_i values for all patients. The dotted lines represent one standard deviation.

Plan	$V_{Q=95\%}$	$V_{Q=105\%}$
Proton DPBN	0.992 ± 0.009	0.002 ± 0.002
Photon DPBN	0.99 ± 0.01	0.03 ± 0.03

Table 5: $V_{Q=95\%}$ and $V_{Q=105\%}$ for the QVH over all patients in figure 24.

Since the QVH in figure 24 represents all patients, the QF for all patients were found using equation (18). The results are presented in table 6. From the QVH the QF was calculated for each patient (table 6). As seen, there can be substantial variation between the patients and their QFs. Furthermore, all patients have lower proton QFs compared to photon planning.

Patient	Proton DPBN	Photon DPBN
Patient 1	0.008	0.016
Patient 2	0.012	0.017
Patient 3	0.009	0.015
Patient 4	0.013	0.025
Patient 5	0.016	0.022
Patient 6	0.009	0.014
Patient 7	0.015	0.022
Patient 8	0.017	0.021
Patient 9	0.008	0.013
Patient 10	0.013	0.017
Total Mean	0.012 ± 0.003	0.018 ± 0.004

Table 6: QFs for all patients.

4.3 Tumor Control Probability

While QF provides a measure of the dose conformity for the DPBN plans, the TCP may provide us with an estimate that is closer to the treatment outcome. Following the procedure described in Material and Methods, a constant relating the clonogen density to the SUV was found by requiring a patient averaged TCP of 0.7 for a total dose of 68 Gy in 2 Gy fractions. Following this procedure a constant of 22975 was found. The resulting TCPs for individual patients are given in table 7.

As seen in the table 7, conventional photon and proton radiotherapy gives a population averaged TCP of around 0.7. Furthermore, if the DPBN prescription can be reproduced in the patient, a TCP of 0.97 is expected. This TCP was more or less retained after photon and proton DPBN planning. It should also be noted, that there are substantial variations in TCP between the patient following conventional planning, but that these variations are minimized when moving to DPBN.

Patient	Photon	Photon DPBN	Proton	Proton DPBN	Prescribed	2Gy Plan
Patient 1	0.76	0.98	0.76	0.98	0.98	0.77
Patient 2	0.50	0.94	0.40	0.92	0.93	0.51
Patient 3	0.83	0.98	0.83	0.98	0.98	0.84
Patient 4	0.70	0.97	0.71	0.96	0.97	0.71
Patient 5	0.55	0.96	0.55	0.94	0.95	0.56
Patient 6	0.73	0.98	0.73	0.98	0.97	0.73
Patient 7	0.75	0.98	0.75	0.97	0.98	0.77
Patient 8	0.69	0.97	0.70	0.96	0.97	0.72
Patient 9	0.82	0.98	0.81	0.98	0.98	0.82
Patient 10	0.58	0.95	0.58	0.94	0.95	0.59
Total Mean Value	0.69 ± 0.11	0.97 ± 0.02	0.68 ± 0.13	0.96 ± 0.02	0.97 ± 0.02	0.70 ± 0.11

Table 7: TCP for different plans for all patients. Most plans are self explanatory, the last column represents the plans where each voxels dose per fraction was set to 2 Gy, i.e. a total of 68 Gy.

By dividing the patients into two groups, the ones where treatment was successful and the ones who had recurrence, the TCP values in figure 25 were found for the 2 Gy per fraction plan. As seen, patients who were cured had on average higher TCPs although, the data are too scarce to conclude statistically.

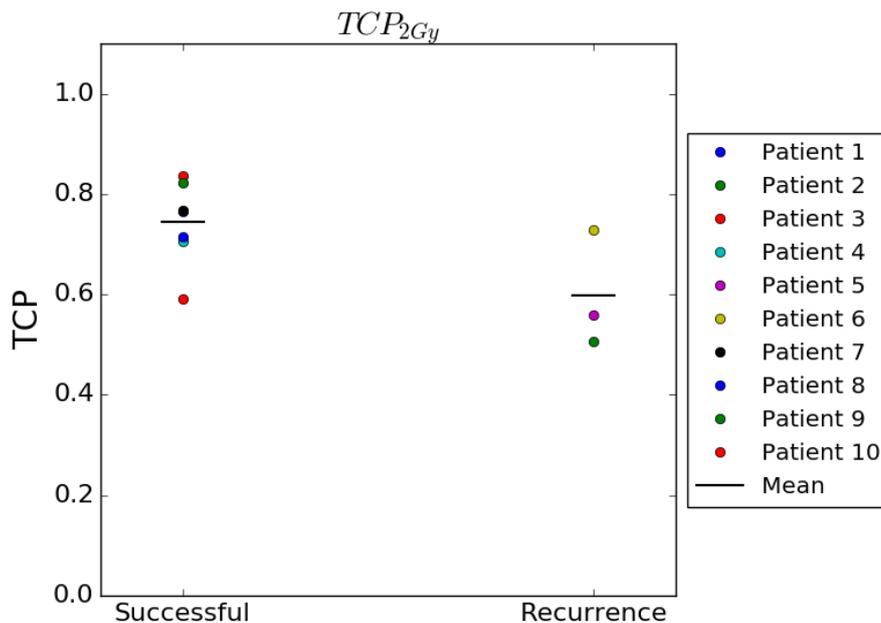


Figure 25: TCP values for the 2 Gy per fraction plan for the patients with successful treatment and the ones who had recurrence.

4.4 Translation and Shrinkage

To simulate patient movement or setup errors the DPBN dose maps were translated. A similar figure to the one in figure 23 can thus be created where one compares the dose prior to movement to the dose in each voxel after movement. Figure 26 contains such a plot where data from patients 1 through 8 is displayed.

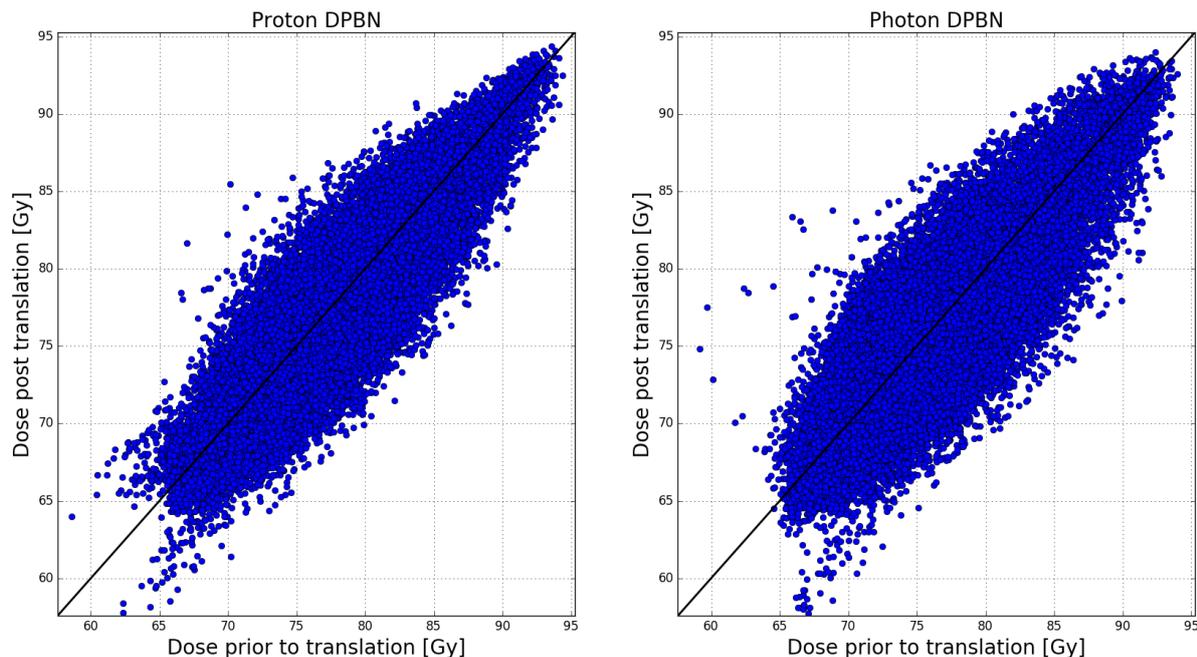


Figure 26: Comparison of the DPBN dose to each tumor voxel for translated photon and proton plans for patients 1 through 8. Each dot represents a voxel. The black identity line represents the situation where the dose is the same before and after translation.

As seen compared to figure 23, we again see substantial variations around the identity line, which again are larger for photons compared to protons. Still, it should be noted that the dose variations for the high dose voxels is not that great compared to voxels prescribed to intermediate doses. Moreover, there is also a small cluster of voxels prescribed to low doses which are at risk of being further underdosed for both protons and photons.

By translating the DPBN dose maps one will achieve different TCP and QF values. The plot in figure 27 plots the average TCP and QF value for each translation with one standard deviation. It is seen that QF increases as the tumor is translated. Also, the interpatient variations increase with increasing translation. Moreover, the increase in QF is greater for protons than for photons. These changes are also reflected in the TCP plot (right in figure 27). Here it is seen that TCP is relatively unchanged for translation less than 2 mm. For larger translations, a marked drop in TCP may be observed. The drop is greater for protons compared to photons. Again, as apparent from the standard deviations, some patients retain their TCP regardless of translations, while others show a substantial drop in TCP. Similar QF features are also seen with respect to tumor shrinkage (figure 28). The impact of shrinkage on TCP however, increases the TCP although the impact of shrinkage is substantially smaller compared to translations. Shrinking each tumor with a set decrease in radius (figure 29), one observes the same effect on both QF and TCP.

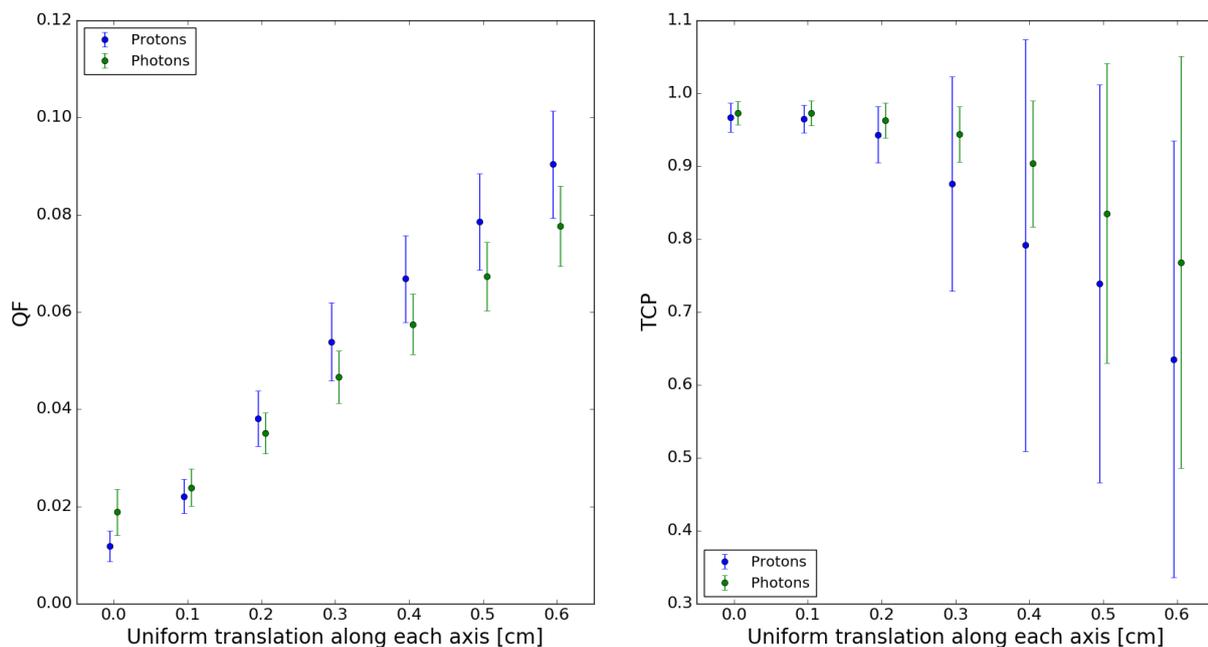


Figure 27: Mean QF and TCP with one standard deviation after translating the DPBN dose maps. Found for patients 1 through 8. It should be noted that the dots are separated so that they do not lie on top of each other, but the translation is the same for both. E.g. it may look like the first two points are at $x = -0.005$ and 0.005 , but they are represent $x = 0$.

The same analysis can also be done after a tumor has been shrunk. By shrinking the tumor according to the method described in section 3.4.5.2, one can find the QF and TCP presented in figure 28.

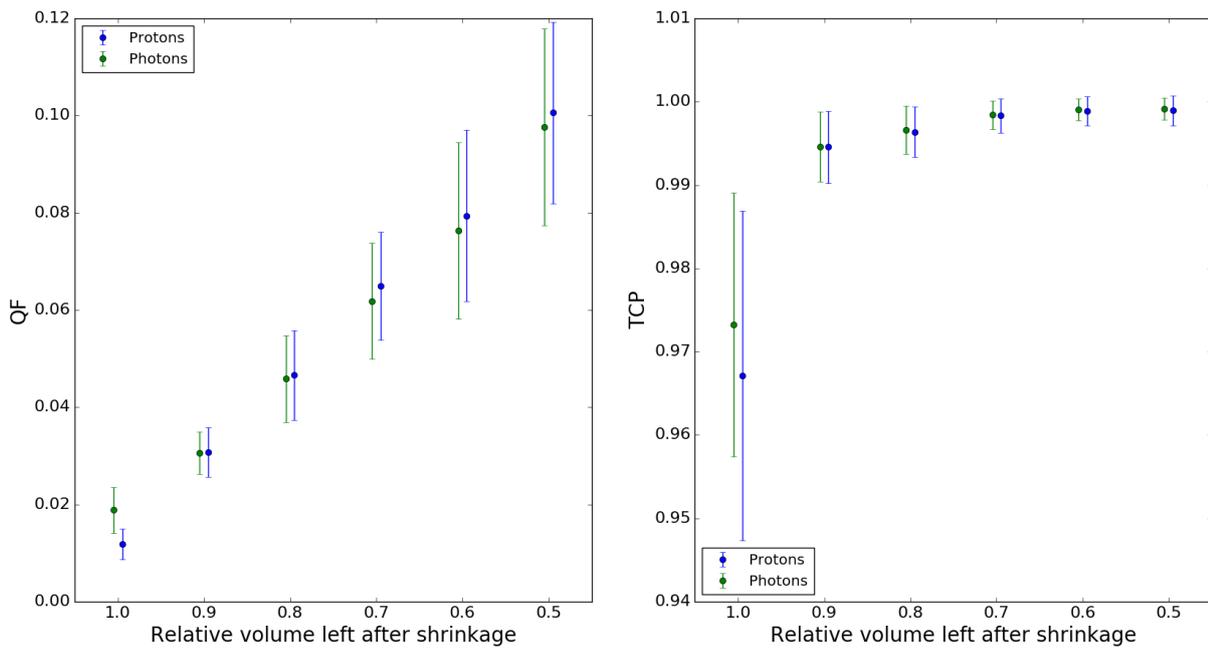


Figure 28: Mean QF and TCP with one standard deviation after shrinking the tumor volume relative to a static DPBN plan. Found for patients 1 through 8. Also here the dots are shifted so that they do not overlap.

However, as the tumors are different size, one can also shrink all tumors the same amount of mm. Figure 29 illustrates the changes in QF and TCP where this is done.

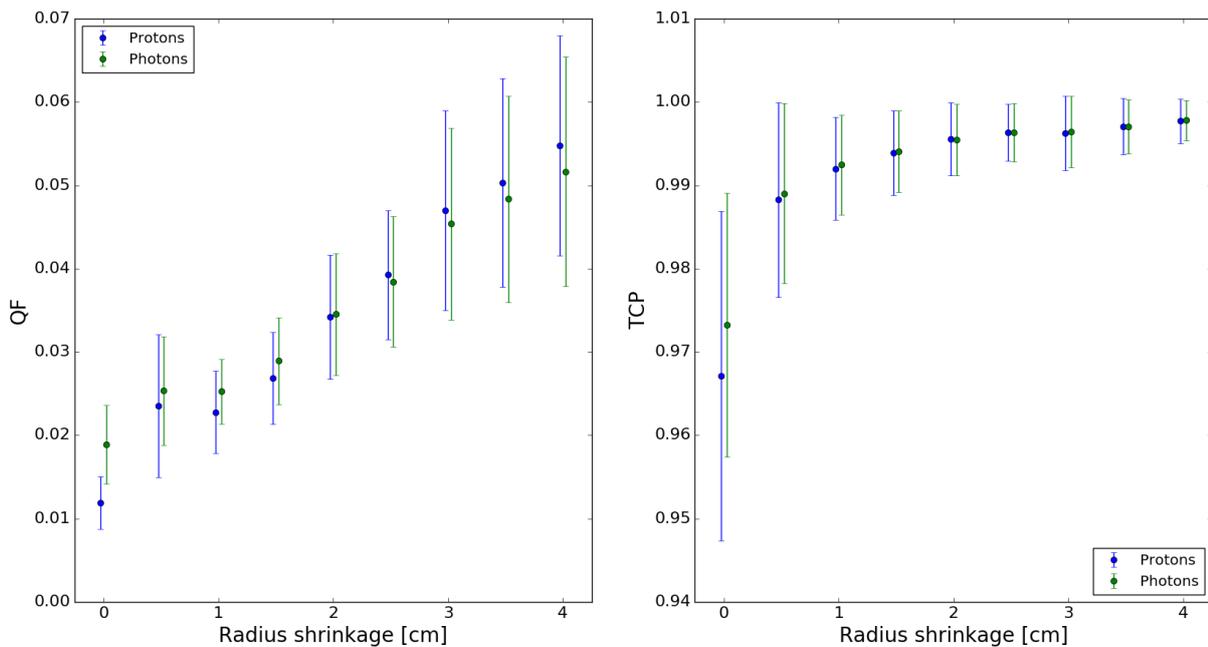
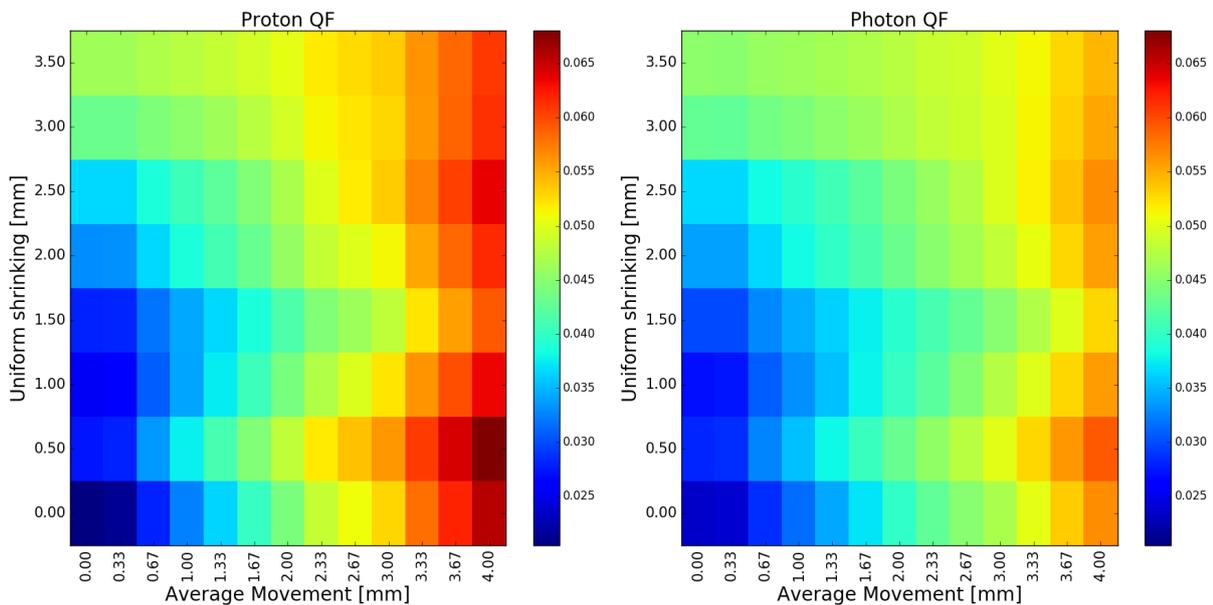
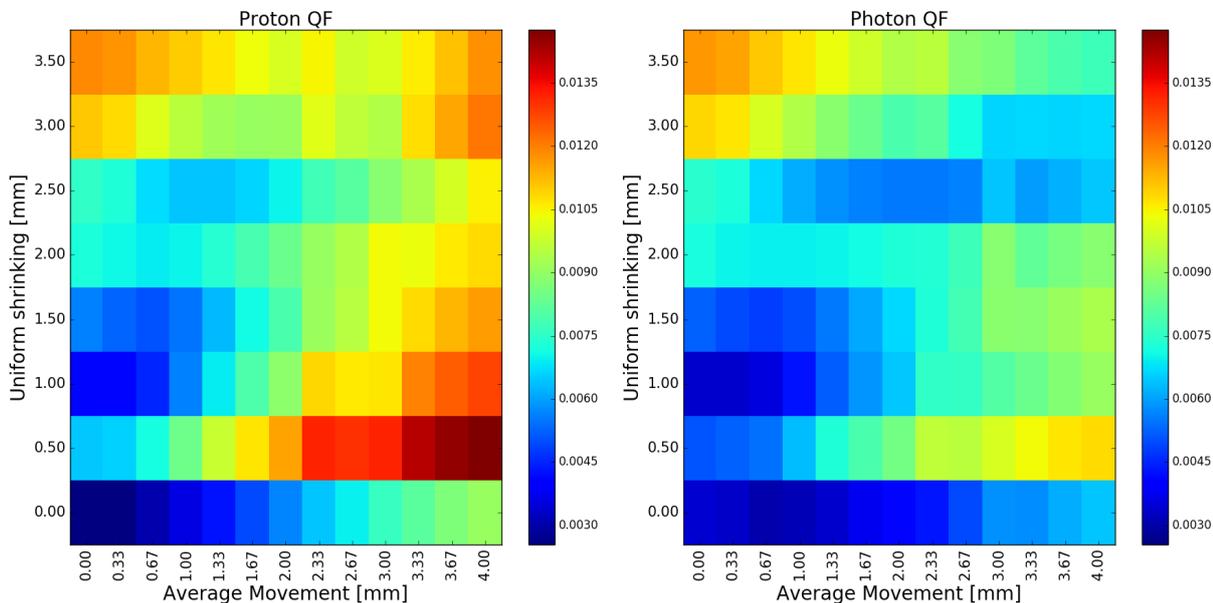


Figure 29: Mean QF and TCP with one standard deviation for the DPBN plans after shrinking the radius of a sphere encompassing the tumor in a static dose map. Found for patients 1 through 8. All dots are shifted a small amount to avoid overlap.

The translation done above was uniform along each axis. By translating along each axis individually and shrinking the tumor for each translation, one finds the QF and TCP color plots presented in figure 30 and 31 respectively. What was seen in the univariate figures above, are also reflected in these color plots. However, it is clearly seen that the greatest impact, especially on TCP, is caused by tumor translation. For QF, the impact of shrinkage and translation is more balanced, although translation is still the dominant perturbation.



(a) Mean



(b) STD

Figure 30: Color plot of mean QF-values and standard deviation of said QF-values as the dose maps is moved up to 4 mm and the sphere around the tumor is shrunk up to 3.5 mm

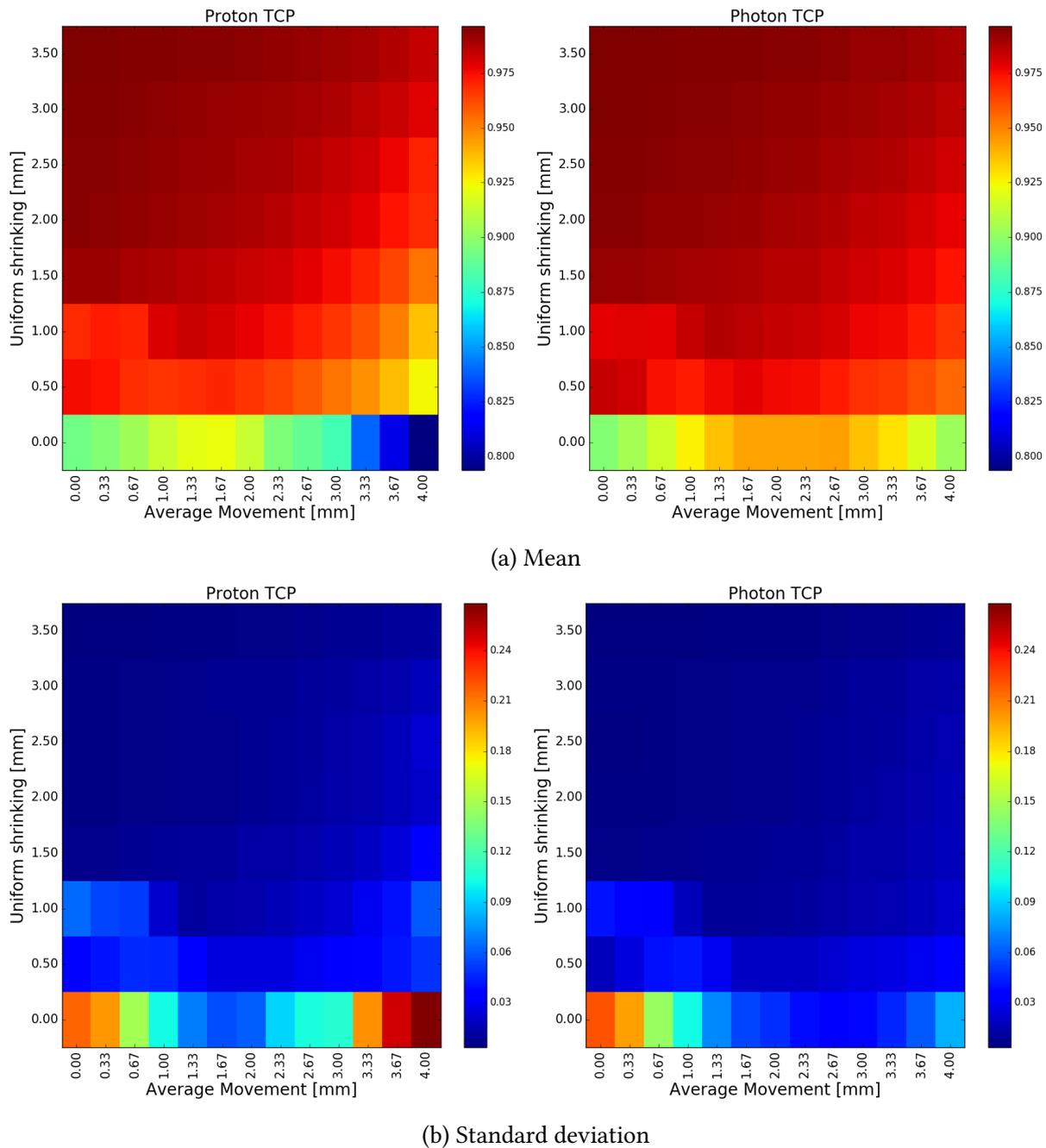


Figure 31: Mean TCP-values and standard deviation for all patients as the dose maps is moved up to 4 mm and the tumor is shrunk up to 3.5 mm.

4.5 Normal Tissue

Dose to normal tissue was assessed by producing DVHs for the spinal cord, the ipsilateral parotid and the contralateral parotid. By summing all DVHs for all patients the figures presented in figure 32 were achieved. It is seen that protons tends to give lower doses to the OARs compared to photons, with the impact being greatest for the spinal cord. 2. DPBN does not seem to increase the dose to any of OARs compared to conventional treatments. 3. There are substantial variations between patients, especially in the dose to the ipsilateral parotid and the spinal cord.

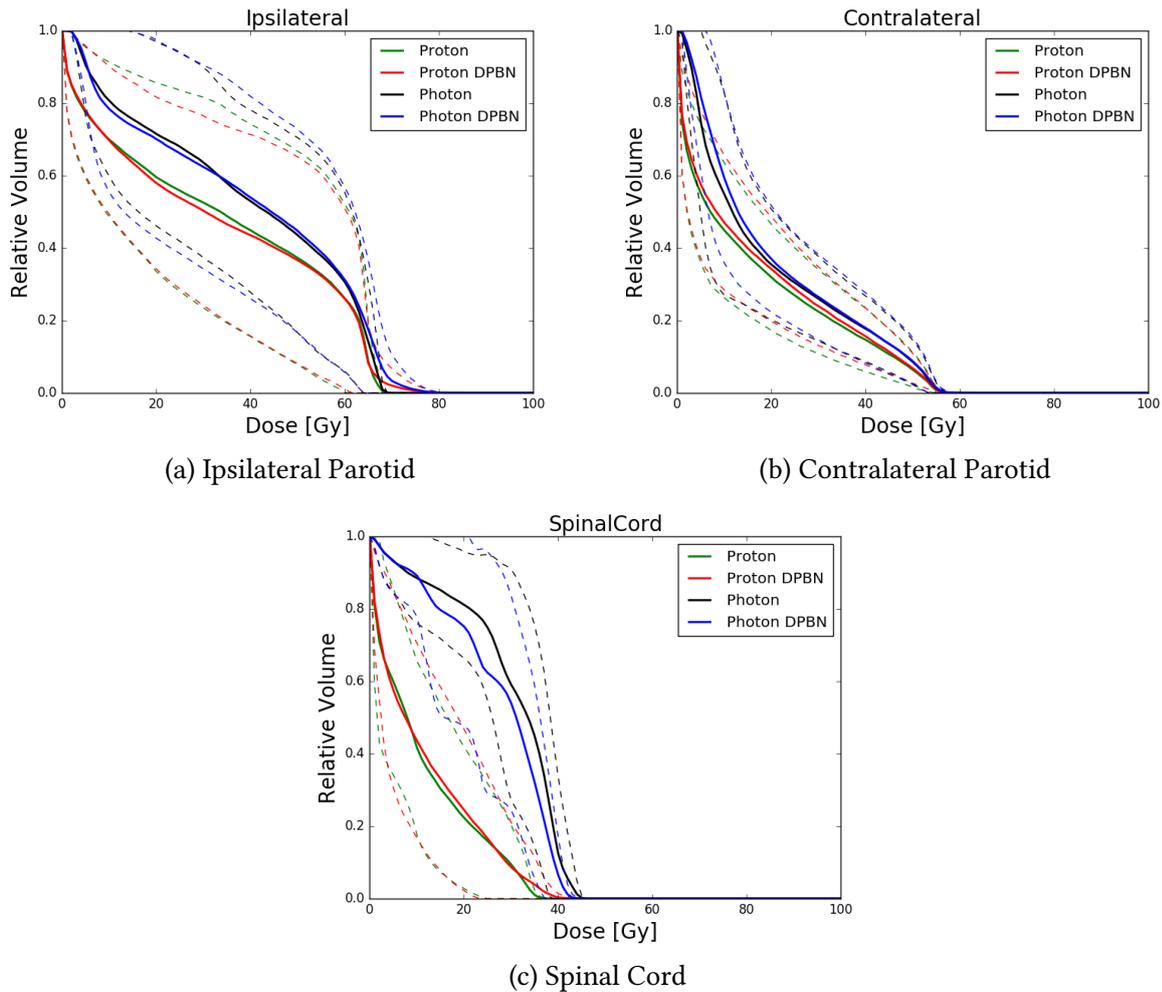


Figure 32: Mean dose volume histograms over all patients, except patient 5, for different OARs. The whole lines are the mean value and the dotted ones display one standard deviation.

Each DVH's mean and max dose was also found and can be found in table 8

Mean/Max Dose	ROI	Photon	Photon DPBN	Proton	Proton DPBN
Mean	Ipsilateral	41.53 ± 13.19	41.95 ± 14.69	34.77 ± 16.61	41.95 ± 14.69
	Contralateral	19.01 ± 7.13	20.21 ± 6.30	14.99 ± 6.69	20.21 ± 6.30
	Spinal Cord	28.93 ± 6.45	26.86 ± 7.30	9.47 ± 4.84	26.86 ± 7.30
Max	Ipsilateral	67.78 ± 2.04	74.13 ± 7.34	67.20 ± 1.90	74.13 ± 7.34
	Contralateral	53.14 ± 12.90	54.66 ± 10.33	51.96 ± 16.59	54.66 ± 10.33
	Spinal Cord	40.43 ± 5.82	37.57 ± 9.98	31.80 ± 12.75	37.57 ± 9.98

Table 8: Mean of the max and mean dose to the parotids and spinal cord for all patients.

By using the formula for Parotid NTCP, equation (16), the results in table 9 and 10 were found for the ipsilateral parotid and the contralateral parotid respectively. From the tables, we again see similar features as for the DVHs. Mean NTCP for ipsilateral parotid was 0.5 while it was 0.12 for contralateral parotid, following both conventional and DPBN photon treatment. Also, small differences are found in NTCP conventional and DPBN for individual patients. This was also the case for protons. Moreover, protons seems to reduce the NTCP to both the ipsilateral and contralateral parotid glands.

Patient	Photon	Photon DPBN	Proton	Proton DPBN
Patient 1	0.70	0.72	0.19	0.18
Patient 2	0.14	0.16	0.05	0.04
Patient 3	0.17	0.14	0.14	0.14
Patient 4	0.53	0.61	0.43	0.58
Patient 6	0.81	0.86	0.80	0.58
Patient 7	0.58	0.54	0.65	0.63
Patient 8	0.40	0.33	0.21	0.21
Patient 9	0.50	0.47	0.18	0.18
Patient 11	0.92	0.94	0.92	0.93
Total Mean Value	0.5 ± 0.3	0.5 ± 0.3	0.4 ± 0.3	0.4 ± 0.3

Table 9: NTCP of ipsilateral parotid

Patient	Photon	Photon DPBN	Proton	Proton DPBN
Patient 1	0.15	0.15	0.04	0.07
Patient 2	0.13	0.18	0.06	0.05
Patient 3	0.14	0.12	0.07	0.11
Patient 4	0.01	0.02	0.01	0.01
Patient 6	0.16	0.17	0.09	0.14
Patient 7	0.13	0.11	0.09	0.09
Patient 8	0.08	0.08	0.04	0.05
Patient 9	0.19	0.18	0.16	0.16
Patient 11	0.05	0.13	0.12	0.07
Total Mean Value	0.12 ± 0.05	0.13 ± 0.05	0.08 ± 0.04	0.08 ± 0.05

Table 10: NTCP of contralateral parotid

5 Discussion

5.1 Limitations

Any research is only as good as the data the research is based upon. The results presented thus far are the numbers found through calculations, but the truth is not as black and white. There are many limitations with the different methods used for DPBN treatment. These limitations and shortcomings will be discussed in greater detail here as an introduction to the general discussion of the thesis. By having a foundation of knowledge about the limitations of the results, it is easier to both understand and discuss the results as what they are: data found through a specific method which can provide insight into cancer treatment in some aspects, but certainly not all.

5.1.1 Patients

As seen in table 1 the patients were few and in some aspects quite similar. There is an overweight of men as there is only one woman, many are in their 60s, most have the tumor in the same area (tonsil) and the tumor grade of most patients is T3 or higher. The number of patients considered could also have been higher. Ten patients is sufficient in a study like this as the work involved in creating four robust plans for each patient was demanding. However, as with all studies like these, more would be better.

5.1.2 Image acquisition

A limiting factor is that all the patient PET-scans are done with the same procedure and the same scanner, although it is well known that the measured uptake distribution depends on scanner technology and software. It should be noted that the error in SUV is quite large. de Langen et al. [54] found that the error was about 20 % for SUV_{max} . The uncertainty was lower for tumors with higher ^{18}F uptake. Such a large uncertainty in SUV is problematic as the DPBN calculations are based upon these data. Calculations of TCP is also dependent upon the SUV and thus this error will also have an effect on TCP.

A general limitation in all image formation processes whether it is MRI, CT or PET is the reconstruction algorithm used. Each reconstruction algorithm has its own shortcomings and advantages. By using only one reconstruction algorithm as has been done for the PET-data used in this thesis, one limits the results as all findings are only valid for scans done with the same type of reconstruction algorithms. Thus it could have been relevant to include data acquired at different PET-scanners to look at the difference in sensitivity of DPBN depending on scanner type and reconstruction technique.

5.1.3 Limitations in RayStation Research

5.1.3.1 The Learning Process

Due to inexperience with TPS prior to this work, one should ideally have repeated the planning process at this final stage of the thesis work to evaluate the impact of training. However, as time did not permit more planning, the data already created had to be used. This creates a small bias on the quality of the plans. The plans were planned in the following order: Patient 1, Patient 6, Patient 2 - 5, Patient 8 - 10.

5.1.3.2 Finding the right index

When the dose prescription was imported to RSR the number of voxels in the dose matrix was not changed. This in turn caused the dose matrix imported back into IDL to be the same size as the one originally exported. During the calculation of the initial dose prescription in IDL, indexing was used to identify which voxels that were contained in the DPBN ROIs. This index was also used early on to indicate the same ROIs in the dose matrix exported from RSR.

Ideally, this indexing would be retained when exporting the corresponding matrices from RSR back into IDL. However, we saw that some small shifts/accuracies/translations were present in the maps exported from RSR. Thus, it was necessary to develop a method to identify the correct ROI indices in the exported maps. In the present work three methods were employed. 1) The indices from the original IDL script was used. 2) Use voxels with dose less than the baseline in the inverse RSR dose map to determine index transformation. 3) The exported RSR indices from RSR using a custom made script. Ideally one would use only one method, but because of computational restraints, both method two and three were used. Method two was used for all calculations regarding the GTV while method three was used when data from other ROIs were used in the calculation. The third method was also used when one needed the tumor indices and the GTV contained both tumor and lymph nodes.

It was thought that each method was an improvement over the previous one, however, there were some indications that this was not true. Table 11 compares the mean dose to the DPBN volumes which should be 76 Gy in the prescribed dose plan:

Patient	PET method	Inverse method	Export method
Patient 1	76 ± 6	76 ± 6	74 ± 6
Patient 2	76 ± 6	76 ± 6	75 ± 6
Patient 3	76 ± 5	76 ± 5	76 ± 5
Patient 4	76 ± 6	76 ± 6	76 ± 6
Patient 5	76 ± 7	76 ± 7	76 ± 7
Patient 6	72 ± 5	76 ± 5	72 ± 5
Patient 7	76 ± 6	76 ± 6	76 ± 6
Patient 8	76 ± 6	76 ± 6	76 ± 6
Patient 9	76 ± 6	76 ± 6	76 ± 6
Patient 10	76 ± 6	76 ± 6	76 ± 6

Table 11: Comparing the mean dose value to the volumes treated with DPBN in the prescribed dose plan where the mean value is supposed to be 76 Gy. The first method found the indices by using the original voxel indices from IDL, the second finds the indices with dose less than d_{base} in the inverse plan and the last method exports the indices from RSR.

As seen in table 11, all methods are similar and there is no clear indication to which method is the best. Thus one can assume that method two and method three are similar enough that both can be used, although it is not ideal.

5.1.3.3 Evaluation limitations

The different evaluation techniques in RSR all have their advantages and disadvantages. Evaluation using clinical goals is the easiest way to quickly assess the dose plan according to the goals. However, the clinical goals should not be blindly followed and one should evaluate visually as well during optimization. The plans were evaluated visually to find non-conform dose,

hot spots and cold spots. Especially in proton plans there tended to appear hot spots around DPBN volumes.

As every dose plan was planned robust, it was important to evaluate the plan if the patient moved or changed density. RSR lets the user move the patient by using the perturbed dose option as described in section 3.4.3. Ideally one should check many movement scenarios and density changes before accepting the dose plan. As this thesis looks into proof of concept for DPBN in planning software, these perturbed evaluations were not performed to their full extent. However, eight plans for two patients were tested to see how well the plans performed under perturbed conditions. Most of the plan failures seen were due to excess dose to OARs. These issues would most likely be easily solved by spending more time on planning and evaluating and should be done if plans are to be used clinically.

Each dose plan must therefore be thought of as a proof of concept and is an indication on whether or not it is possible to give DPBN plans robustly. The plans in this thesis should not be given as a clinical treatment, but the knowledge behind them should be read and considered as they describe the planning of DPBN.

5.1.4 TCP, QF and NTCP

All dose plans were evaluated with TCP of tumor, QF of the dose plan and NTCP of parotids. These values are all useful in describing the success of the dose plans created, but they too have their limitations.

TCP is based upon the premise that SUV for each voxel represents the clonogenic density, and that the correct α and α/β values are given. These variables are all questionable and the uncertainties can be quite large. For alpha the uncertainty is around 30% [55]. These uncertainties will effect the results and the TCP presented in the results should therefore be thought of as indications rather than absolute truths. The α can be thought of as a measure of radiation sensitivity and it will differ for each patient. Therefore some patients are likely to achieve the TCP's presented in this thesis while other, more radiation resistant tumors will not achieve the same results.

Another limitation of the TCP is the use of the constant k multiplied with SUV. This constant was found by setting a dose of to 68 Gy in each tumor voxel and then finding this constant which gave 70% mean TCP for all patients. From table 7 we can see that the mean TCP of the 2 Gy per fraction plans is 0.70. However, there is also a rather large standard deviation. Ideally, one would use a vast array of patients to find this constant, instead I have used the same ten patients presented in this thesis. The more patients used to find this constant, the closer it would be to an ideal true constant. While it is a good estimate of the dose actually given to the patients when they actually were treated, no treatment will manage to give exactly 68 Gy to every voxel within the tumor. A better way to estimate this constant would have been using Monte Carlo simulation where each dose plan is evaluated with an unique combination of dose to each voxel and where said dose is within a predefined span. This approach would demand a lot of computational power and should be done on a super computer where one utilizes parallel processing. If the computational cost is worth the gain in accuracy however, is debatable. It would probably be wiser to expand the number of patients used to find the constant before one starts a heavy Monte Carlo analysis.

QF is a measure of how well the plan created by the TPS is compared to the desired plan. In itself, the QF is quite stable as the prescribed dose is considered without error. However, it would be interesting to have a factor comparing the planned dose to the original SUV uptake, a SUV-to-Plan factor. With this factor one would also have to consider the uncertainty in the PET-data, which would give an indication of the uncertainty in dose prescription resulting from the PET.

The NTCP model used in this work, is dependent on three parameters: m , n and $TD_{50}(1)$. In the literature there are many different approaches to finding these factors and as it is beyond the scope of this thesis to explain them all, a brief presentation is presented in table 12:

Study	m	$TD_{50}(1)$ [Gy]	n	Time after radiotherapy
Gupta et al. [56]	0.48	35.1	1	3 months
	0.38	41.3	1	12 months
	0.44	55.9	1	24 months
	0.38	64.3	1	36 months
Dijkema et al. [57]	0.40	39.9	1	12 months
Miah et al. [58]	0.25	26.3	1	12 months
Eisbruch et al. [59]	0.18	28.4	1	12 months
Roesink et al. [45]	0.54	31	1	1.5 months
	0.46	35	1	6 months
	0.45	39	1	12 months

Table 12: Different m , n and $TD_{50}(1)$ values found from different authors.

It is evident that there is a difference from study to study on which values are to be used. In this thesis the time after radiation calculations were set to 12 months as all articles have parameters for this time period. As Gupta et al., Dijkema et al. and Roesink et al. are all relatively close to each other for both $TD_{50}(1)$ and m , an average of these three were used in the calculations of NTCP. Eisbruch et al. and Miah et al.'s data could have been used as well, but there seems to be such a big difference that it was decided to only use the parameters within one of these groups. A deeper study into what might cause these differences and whether or not one should use all values to find a mean would be advantageous, but this is beyond the scope of this thesis.

5.1.5 Setup uncertainties, Translation and Shrinkage

It should be mentioned that the data for figures 26, 27, 28 and 29 are for patients 1 through 8 as patient 9 and 10 were not possible to add in the current script version.

During fractionated radiation therapy, the patient will be treated five or six times a week until the total dose is delivered. To deliver the treatment correctly it is important to position the patient the same way as when the patient was scanned. The positioning errors are examples of random errors. These errors might occur every time a patient is given a fraction, but are different in extent. On the other hand, one has systematic errors which might also occur. These errors are the same throughout the treatment and may e.g. be caused by a faulty calibration of the treatment machine. Both of these errors are possible to minimize. One can use restraining masks or positioning systems to ensure correct positioning of the patient and one should regularly perform quality controls of the treatment machine to minimize systematic errors.

A study by Park and Park [60] found that the mean random error of Left/Right (LR) and Anterior/Posterior (AP) translational shift of the cervical spine were -0.39 ± 3.24 mm and -0.57 ± 0.99

mm respectively. The same study also found that the mean systematic error of the translational shift in the same directions were -0.64 ± 2.57 mm and -0.33 ± 1.22 mm, respectively. Another study by Kaur et al. [61] used Cone Beam CT to track setup uncertainties and found that the systematic error were 1.5 mm, 2.1 mm, and 1.9 mm in the x, y, and z-directions respectively. The random errors were 0.9 mm, 1.2 mm, and 0.9 mm, respectively in x, y, and z-directions. To meet the uncertainties produces by these errors, one usually use PTVs, but in this thesis, robust treatment was used instead. To discuss the success of the robust treatment plans one must look at the translation of the dose plan as this can simulate both tumor movement or patient setup errors.

For every dose plan the translation and shrinkage was simulated post optimization in IDL. The shrinkage and translation was set to an interval which varied from 0 to 4.0 mm translation in each direction and an uniform shrinkage of the tumor. As this shrinkage and translation was purely theoretical and not observed within the patients planned upon, it is important to compare our theoretical assumptions with that found in patients during treatment. One should also consider the setup uncertainties and compare them to both translation and shrinkage of tumor.

Barker et al. [62] found that the median total relative loss of tumor volume was 69.5 % (range 9.9% - 91.9%) of the initial GTV over the range of 6-7 weeks with treatment of head and neck cancer. The shift of median mass center of the tumor was found to be 3.3 mm (range 0 - 17.3 mm).

Figure 33 contains some of the plots presented in Barker et al.'s paper. The results presented in figure 29 gives the shrinkage in mm shrinkage. To see this reduction in relative terms as well, the relative shrinkage was plotted for each tumor in figure 34.

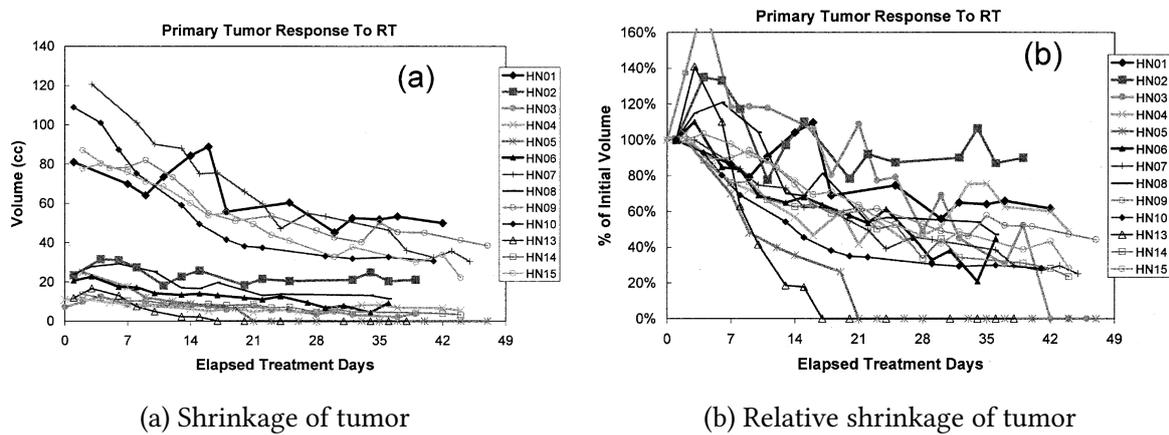


Figure 33: Barker et al.'s [62, Figure 4] figures displaying both relative and absolute shrinkage of tumor and lymph nodes during radiotherapy.

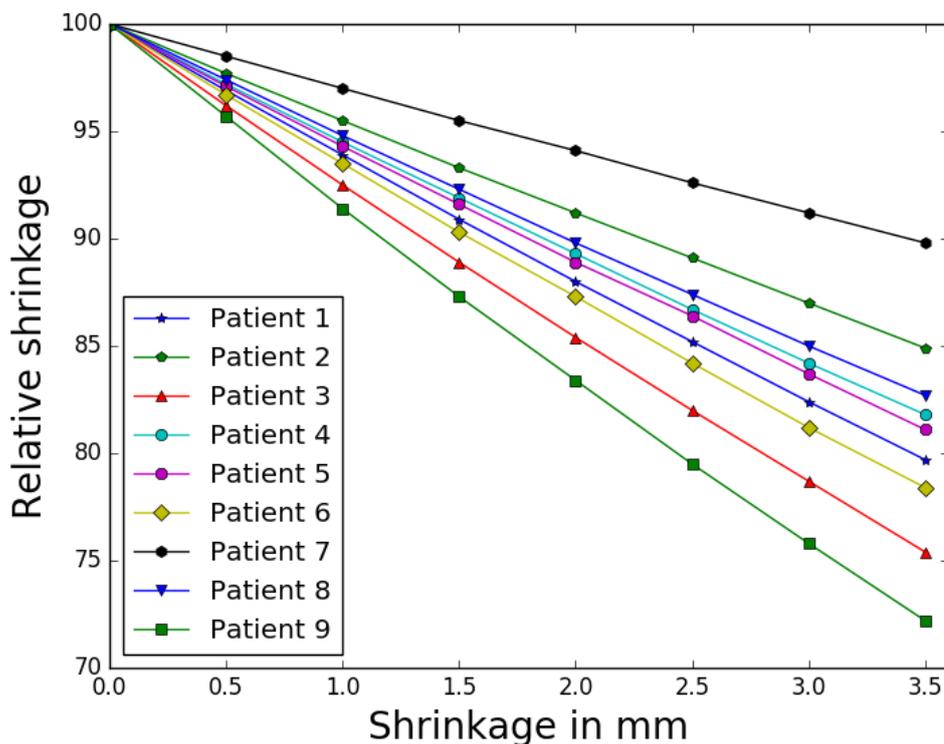


Figure 34: New volume size after shrinkage of each tumors' radius expressed as percentage of original volume.

Barker et al. observed a median shrinkage of 69.5%, i.e. 30.5% of the original volume remained. It is evident from figures 34 and 28 that the shrinkage simulated in this work is not as profound as it should be. This means that the shrinkage proposed within this thesis is not highly representative of the real shrinkage which may occur during treatment. However, shrinkage does not happen all at once and one should also consider the shrinkage as part of the tumor reaction process in fractionated treatment.

Figure 33a and 33b indicates that there is a big variation in whether the tumor shrinks or grows initially, but at the end of treatment most of the tumors have shrunk. It would therefore be wise to consider if one should use DPBN throughout the whole treatment. From the results in this thesis it seems that a little shrinkage would not affect the QF of the DPBN plan, but the more a tumor shrinks, the higher the QF. This also leads to an important point: It seems as set up uncertainties and uncertainties in PET data are minimal compared to the shrinkage occurring during radiotherapy. If one were to take the results at hand and use the technique suggested by this thesis clinically, it would be wise to only do so for patients who are likely to have a predictable shrinkage and only use DPBN for as long as the shrinkage is minimal. By combining DPBN with traditional VMAT treatment where one use DPBN for the first e.g. 10 days, seems to be the most logical route to choose. However, what to do clinically is also dependent upon what is left in the wake of the tumor. If there is no healthy tissue and only necrotic tissue, one could continue with the intense radiotherapy and hopefully achieve better TCP of the remaining tumor.

The translation used in the current study seems to be in accordance to other findings. By translating the tumor up to 4 mm in each direction one covers the median movement found by Barker et al. [62] and the setup uncertainties found by Kauer et al. [61] and Park and Park [60].

However, it should be noted that the script used to find the average movement does so by translating the tumor along the x, y and z-axis in a triple for-loop. This means that there are few data points where the tumor is translated to its maximum distance (4.0 mm) and there are more data points as the average movements goes towards zeros. The maximum distance of 4.0 mm was chosen to ensure a sufficient amount of data points at 3.0 mm. Ideally one should translate the tumor even further or collect more data points in every point, but the computational cost outweighs the benefits after a certain number of translations and 4.0 mm was therefore chosen.

5.2 Treatment Plans

Creating robust treatment plans for 10 patients is time consuming. In theory it is an easy process where the quality of the clinical goals and optimization functions determine how efficiently one can create a plan. However, this method is not fool proof and as described previously there are many ways to evaluate a dose plan and its properties.

5.2.1 Ease of Use

In RSR the treatment planning user interface is intuitive and easy to learn. The software allows the user to construct templates for setup of beams, clinical goals and optimization functions. Early planning were done by utilizing these options and using standard treatment techniques, goals and optimization in head and neck cancer. These standards were set up in close collaboration with qualified staff at Oslo University Hospital. However, the true potential of RSR treatment planning lies in the option to add self made scripts. By implementing a script which named all ROIs correctly and two scripts which set up beams, clinical goals and optimization settings for photon and proton treatment respectively, the planning went even faster. These scripts also ensured consistent naming, beam set up, clinical goals and optimization functions for all plans within each beam type. The scripts therefore complemented the otherwise efficient treatment system in two notable manners. Firstly, the automation of otherwise time consuming, repeating processes ensures that the set up is correct and that user errors due to inattentiveness are of no concern. Secondly, the automation behind consistent naming ensures that each type of ROI is named the same regardless of planner and treatment plan.

This second point proved especially important as the patient data contained many naming conventions which all seemed to differ from oncologist to oncologist. In RSR each ROI can be given both a name and a color which is displayed beside the ROI, as well as being used in the visual presentation of the ROI in the patient. E.g. the GTV was given an orange color as illustrated in figure 19. By implementing a consistent naming and coloration for each ROI across different plans, it was easier for the planner, and every person consulted about the different cases, to know which volumes were discussed. The naming script was so efficient that it was implemented clinically at the Radium Hospital in Oslo.

However, there are also some restrictions in the treatment planning. A common error when planning proton therapy was that the optimization function would find the optimal solution too early. That is, the optimization would have found a local minimum. If an optimization setting was not sufficiently weighted from the beginning, enforcing the optimization function to accommodate a higher weighting late in the optimization process proved difficult. This problem also occurred when planning with photons, but where the proton plans would sometimes be

virtually unaltered after 80 or 100 iterations, photon plans were still open for change.

Another recurring problem regarding the proton plan was that the Dose Fall-off optimization function seemed to require enormous amounts of computational power and at the same time did little to change itself. Often the Dose Fall-Off was what contributed most to the error which the optimization tried to minimize. Usually, one would find that what gave the highest error was the DPBN distribution. The ineffectiveness of the Dose Fall-Off caused hot spots and non-conformity. It was a struggle to remove hot spots and ensure conformity for nearly all proton dose plans.

All these effects are important to mention as they reflect the quality of the plans created. The creation of a good plan requires an attentive planner who is patient and has an eye for details. As the treatment planning, especially robust planning, is a time consuming process, it might be easy to lose focus and quickly go on in the optimization. Every aspect of a treatment planning process which helps the planner focus on the task at hand is positive and helps in the creation of a good dose plan. The most limiting factor to the quality of the treatment plans used in this thesis is the time. A good plan can always become better, but it is important to find the balance between the quality of a plan and the time used to create it.

5.2.2 Robust Evaluation

While using PTVs to indicate plan robustness mostly worked for photon plans it was a bad indication when it came to proton plans. The reason for this might be that since the protons are much more precise in their energy deposition, the PTV was not sufficiently covered even though the plan was considered robust. Photons on the other hand have such a broad dose deposition that the robust optimization also gives enough dose to the PTV.

It was evident that a better evaluation of the robustness of the dose plan was needed. As mentioned in section 3.4.3 this was provided by the perturbed dose function. Ideally one should move the dose the equivalent of the setup uncertainties in each direction and for a number of translations in between. As this is rather unpractical and time consuming, it was suggested to only use the extremes as proposed by the staff at Oslo University Hospital. Initially, it was suggested to translate to all eight corners of a hypothetical cube around the point of origin. I.e., move the patient $[0.3, 0.3, 0.3]$ mm in x, y and z-direction respectively, then $[0.3, 0.3, -0.3]$ mm and continue until you reach $[-0.3, -0.3, -0.3]$ mm translation. However, this translation was deemed improbable by the staff and it was suggested to translate the dose to the center of each side of the cube instead. I.e. $[0.3, 0, 0]$ mm, $[0, 0.3, 0]$ mm ... $[0, -0.3, 0]$ mm, $[0, 0, -0.3]$ mm. This movement will by far cover the setup uncertainties found by Park and Park [60] and by Kaur et al. [61]. However, it does not cover the typical translations found by Barker et al. [62]. If the latter translation is to be considered, it would indeed be wiser to evaluate in the corners of the cube and extend the evaluation up to 3.5 mm.

Because of the impracticality and time restraints, only a few plans were tested by perturbing the dose distribution. Most clinical goals were fulfilled once the dose was perturbed, but the dose to the PRV of the spinal cord sometimes exceeded the clinical goal. This is a very serious issue as the spinal cord is considered a serial organ, and damaging it in one place will destroy its entire function. The robust optimization is therefore a useful tool and should be used, but it is important with an extensive quality control to ensure that the dose to vital organs are not exceeded.

5.3 DPBN calculations and implementation

One of the main problems when planning DPBN using modern treatment systems is that even though the system in itself can access each voxel, there are no modules for planning each voxel independently. The implementations done in this thesis is based on that of Arnesen et al. [16], however there have been other approaches such as adapting the in house treatment planning system at Ghent University Hospital to treat each voxel independently based upon their PET/CT [63]. Meijer et al. used a research version of a commercial treatment planning treatment software to modify the optimization function and add in a plug-in which combined DVH objectives with biological voxel based objectives [64]. While Thorwarth et al. at Tübingen University Hospital used their HYPERION system which uses pencil beams for hypoxia DPBN [65]. A last DPBN approach to create a dose plan with equispaced dose levels where the dose to the whole structure was the mean dose of all voxels in that volume [52,66–68]. The latter approach becomes a sort of combination of DPBN and DPBC, a dose map based on the voxel intensity is created, however, the dose deliverance is done by contours.

The strength of the method used in this thesis is that it is possible to implement it in any treatment planning system as the inverse plan can be created elsewhere. In its current state, the user must have access to IDL and some programming knowledge to be able to use the full potential of this method. One could rewrite the script in a more accessible language such as MatLab or Python. The latter would be the preferred choice as it is versatile and free. By writing the same script in Python and creating a simple Graphical User Interface (GUI) the script should be easily accessible to everyone. Especially as Windows 10 now offers the possibility to create an Ubuntu like terminal inside their Operating System (OS). Another option is utilizing the fact that many TPSs allows the user to add in self written scripts or macros. In RSR these scripts are written in IronPython for every version up to 7 and thereafter one have to write in Python. RSR has a wide array of built in functions which are easily utilized through scripting. By writing the IDL script used in this thesis to an RSR module would be a small task with a great benefit. A script which would utilize a GUI to ask the user which ROIs DPBN should be planned for and then creates the inverse dose plan, would only have to be written once, but could be used by all RSR users world wide. If one would aim to make DPBN more accessible than it is to day, this would probably be an effective way to do it.

5.3.1 Calculating DPBN

Usage of PET/CT data may seem like a good approach in theory, but this assumes that the information we gain from the PET is accurate. The spatial resolution described in the theory section aims at giving the reader an understanding about how spatial accurate the PET is, but this is not the only error one should consider. According to a metaanalysis performed by de Langen et al. [54] it was found that there was an uncertainty to the mean SUV of 20%. As DPBN bases itself directly on the SUV values for each voxel, this uncertainty will profoundly affect the quality of the method. This uncertainty will also affect the error estimates of TCP and NTCP which were discussed in section 5.1.4.

The 5th and 95th percentile for I_{low} and I_{high} were chosen after the recommendation of my two supervisors. The function of these percentile were to avoid unrealistic high dose peaks or

voxels receiving less dose than they should to acceptable levels. While Arnesen et al. [16] used the 2nd and 98th percentile, the more conservative percentiles used in this thesis might create a more feasible treatment plan, but on the other hand it might also cause the most PET intense clonogenic cells to receive less dose than they should have.

Calculating D_{high} from a set D_{low} and D_{mean} presented an interesting approach to DPBN. Initially defining D_{low} and D_{high} was suggested and the resulting D_{mean} would thus vary from patient to patient. By opting for a fixed D_{mean} instead all patients receive the same mean tumor dose, but a different maximum dose. This approach will be in contrast to a plan with a homogeneous dose increase to the entire GTV. A last option would be to have a fixed d_{high} and d_{mean} and let d_{low} vary, but this could cause the dose becoming too low for some voxels. The tumor dose should at least be at least 68 Gy according to guidelines at Oslo University Hospital.

5.3.2 Making the Plans Robust

As RSR does not allow robust optimization of doses where another dose has already been given, the GTV were not planned robust in the DPBN plans. This, however, is not considered a problem. Given the relatively large spatial resolution of a PET-scan compared to CT, a DPBN dose plan based upon PET will inherently be a more coarse dose distribution than the one based upon CT. This will cause any volume planned with DPBN to be naturally more robust than a volume planned using normal treatment planning. Thus, one can argue that our DPBN plans are robust in the sense that the dose is spread out.

5.4 Comparing Treatment Plans

Although four dose plans have been created for each patient, the DPBN plans are the main focus of this thesis. In the following discussion, all dose plans will be discussed, but the DPBN plans will be the ones in focus.

Figure 22 depicts the dose to each voxel above 68 Gy for each patient in the slice of the center of mass. Viewing these images tells us nothing exact, however, they give an indication. By using the images for patient 5 as an example (figure 22e), we can observe some of the characteristics found in this thesis. To the left the prescribed dose plan illustrates what we want to give based on the PET-data. Each plan replicates the main part of the tumor in the middle of the image quite well, although it looks as if the proton DPBN plan is more conform. The smaller part of the tumor below changes dose rapidly between each voxel, creating a sort of half moon. This shape is partly recreated by the proton DPBN plan while the photon plan creates a circle. Finally, it should be noted that the space between the main part and the smaller part is supposed to receive less than 68 Gy. This is a too steep dose gradient for the photon DPBN plan to handle, while the proton beam seems to handle it better.

Further, one can view the scatter plots in figure 23. The protons are closer to the ideal identity line than the photons. It does also seem as though the photons are more warped around the identity line than the protons. The few voxels which are placed far from the identity line should be noticed as well. Especially the ones given a low dose to voxels which should have had 70 Gy, indicating that one or more plans gives a too small dose to many of the voxels. This is quite serious as in a clinical situation, giving too little dose might give a lower TCP and thus lower the chance of inactivating all clonogenic cells.

In general, it seems as though the proton plans are closer to the prescribed DPBN plan than the photon plans. This is further confirmed by the QVH in figure 24 where the QVH for all proton plans is closer to the ideal step function than the one for all photons. Since the QVHs represents of the sum of all plans, it is better to discuss individual QFs for an in depth study.

5.4.1 QF

An overview of QF values from different paperson dose painting is presented in table 13. The table also contains a QF which compares the planned dose with given dose. If one were to continue work on this thesis, it would be natural to measure dose given to a phantom and compare the delivered dose to the planned dose.

Article	Modality	Site	QF (plan/pres) [%]	QF (del/pres) [%]
Skorksa et al. [68] DPBN	Photon	Head and Neck	2.10	-
			3.04	-
			2.72	-
Arnesen et al. [16] DPBN	Photon	Cervix	2.60	-
		Head and Neck	1.80	-
		Lung	2.0	-
Papoutsis's communi- cation with Malinen E., Svestad G. DPBC [69]	Photon	Lung	4	2.2
Meijer et al. DPBN [64]	Photon	Lung	4.9	-
			5.3	-
			4.2	-
			4	-
			5.9	-
			6.8	-
Papoutsis DPBN [69]	Photon	Lung	2.00	2.95
			1.99	1.70
			1.65	1.90
Barragán et al. [52]	Proton*	Head and Neck	1.03	-
			0.94	-
Current Study DPBN	Photon*	Head and Neck	1.8	-
	Proton*		1.2	-

Table 13: Comparison of different QFs from different works, * = robust treatment.

As one can see from table 13 the QF values found in this thesis are among the lower ones. The only ones which are better are the results found from Barragán et al. [52]. However, it should be noted that these values were obtained by treating the patient with seven subvolumes (a quasi-DPBN) instead of voxel by voxel treatment. A true DPBN approach will suffer more from the systems inability to deliver steep dose gradients between two neighboring voxel of significantly different dose value. This problem is diminished by the semi DPBN approach of Barragán et al. by creating isodose subvolumes. Although there will be steep dose gradients when using subvolumes as well, once inside the boundaries of a volume, the dose will remain the same. Barragán et al. found that increasing the number of subvolumes beyond seven produces worse QF values, in line with this reasoning. With the current state of TPS this might indicate that using subvolumes is a better approach than true DPBN, but more research should

be done in this area before anything can be concluded.

Looking at the $V_{Q=105\%}$ Barrigán et al. [52] found that for a PTV-based plan they got $V_{Q=105\%} = 35\% - 38\%$ for patient 1 and $V_{Q=105\%} = 29\% - 31\%$ for patient 2 of two patients considered in their study. For the robust plans, however, these values were $V_{Q=105\%} \simeq 7\%$ for both patients. This large difference indicates that proton DPBN plans will give a large overdose unless planned robust. Barragán et al. arguments that homogeneous PTV-based plans does not account for errors in proton therapy appropriately, and that this effect is even stronger when doing DPBN as the painted volumes become more sensitive to uncertainties in heterogenous doses with high in-field dose gradients [52].

By comparing Barragán et al.'s robust $V_{Q=105\%}$ with the ones presented in table 5, both the photon plans and proton plans of this thesis performs better. As Barragán et al. had better QF, this can indicate that even though Barragán et al.'s planned dose plans are closer to the prescribed dose plans, they overshoot more than the plans presented in this thesis. It should be noted that most of Barragán et al.'s $V_{Q=95\%} \simeq 100\%$. By forcing such high $V_{Q=95\%}$ Barragán et al. has strictly enforced a minimum dose to the GTV. These findings are similar to the ones presented in this thesis.

For photons, the QF found for this thesis are similar to those of Papoutsis [69] and Arnesen et al. [16]. This is as expected as these studies are based upon the DPBN implementation suggested by Arnesen et al. [16]. From the results presented above it seems as though this method is currently the best way to implement DPBN in a clinical TPS and still have good results. However, there are many factors which may influence the QF and no conclusions can be made without further research.

From table 13, a change from the planned dose to the delivered dose can be observed in the results by Papoutsis, where DPBN was delivered to a phantom. This is as expected as a linac is restricted by several physical restraints such as MLC movement speed, the thickness of each leaf and gantry rotation speed. Once a photon reacts with soft tissue it will start a cascade of interactions with origin in the interaction point. This cascade of reactions can be thought of as a point spread-kernel as described by Ahnesjö and Aspradakis [70]. The practical effect of the cascade is that even though one can estimate where a photon will interact, the cascade of follow up interactions will be randomly distributed around the interaction point and thus smearing out the dose. As DPBN aims for a precise voxel by voxel delivery of dose, this smearing out is unfortunate and will increase the QF of the delivery.

5.4.2 TCP

TCP values presented in table 7, one can see that the DPBN plans have a higher TCP than the conventional plans. This is as expected given that a higher dose will give a higher TCP and the dose in a DPBN is higher than that of a conventional plan [44]. Figure 25 tells us that the patients who had tumor recurrence post treatment have a lower TCP than the ones with successful treatment. Once again, this information would be more trustworthy if they were found in a larger group of patients, but the results found here are supported by those found by Malinen (personal communication, unpublished work). If one takes the difference between the TCP for DPBN and that for conventional treatment, the values presented in figure 35 are found.

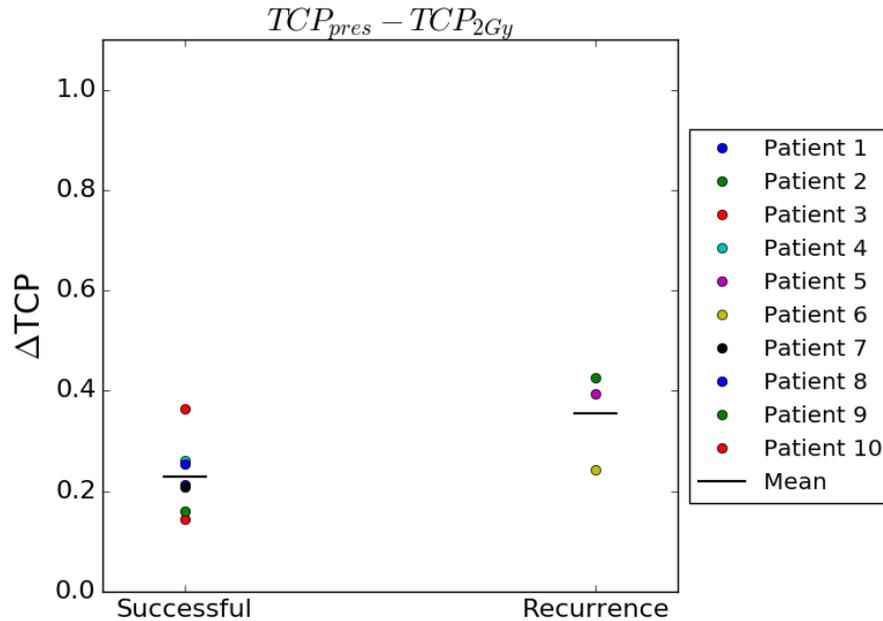


Figure 35: TCP values for 2 Gy per fraction plan subtracted from the TCP values for the prescribed DPBN plans for the patients with successful treatment and the ones who had recurrence.

looking at figure 35 it seems as the patients who would gain most from DPBN, i.e. the ones with the biggest difference, are the patients with recurrence. For future work one could test DPBN with the patients who had low TCP and see if there was an overall increase in successful tumor control compared to the 70% observed for this patient group overall.

5.5 Translation and Shrinkage

To the author's knowledge, there has been no previous study of DPBN where one compares the effect of shrinking and translation for proton and photon treatment. The discussion in this section is thus more speculative than previous sections as there is no current literature to compare the results to.

The translation plots in figure 27 are all translated by moving the tumor in one direction, i.e. moving along the main diagonal of a cube created by the x, y and z. Thus the information found is restricted to that movement and may not be applied generally. However, this was done to give an indication of what could be found if one were to move the tumor independently along each axis. As we can see from the line with shrinkage = 0 in figure 31, this assumption turned out to be correct. By uniformly translating the tumor, as in figure 27, we found that QF increased and TCP decreased. This is as expected given that the tumor was moved away from the high dose area of the DPBN dose distribution.

It should be noted that the proton plans seems to be more affected by the change than the photon plans. One possible cause of this can be found in the dose characteristics of protons and photons. While proton beams are aimed to give a high dose in a specific spot, photon beams have a more even dose deposition along the path of the beam. It is the same feature which causes the photon plans inability to create steep enough dose gradients required for DPBN as previously discussed. When the tumor is translated, the photon plan will be better suited for

the change than the proton plan due to the fact that the photon dose is more smeared out.

One should also note that there is a large standard deviation for TCP when one translates the tumor sufficiently. This effect is probably caused by the difference in tumor size where larger tumors are probably less susceptible to change than the smaller ones.

Looking at the relative and absolute shrinkage of the tumors one notices the same trends. The proton plans are more affected by the change than the photons. For TCP we observe an increase as the tumor shrinks. Since the tumor is scripted to shrink towards the center of mass it will in most cases shrink into a high dose area, as illustrated in figure 22, and thus increase TCP.

By combining shrinking and translation, the effects discussed above are found again. TCP decreases with translation and increases with shrinkage. It is interesting to note how much improvement there is for a tumor shrunk by 0.5 mm and translated an average of 4 mm compared to that of one translated the same, but which has not been shrunk. This is especially prominent for the protons plans, probably caused by the protons characteristics discussed above. However, once again we find that there is a large standard deviation when one translates some mm. What is surprising is that the standard deviation for an unshrunk tumor, seems to decrease and then increase while translating. This is not in agreement with the results in figure 25. The cause of this effect might be that as we move the tumor independently, different directions of movement may provide smaller differences of TCP.

For the QF in figure 30 we can see that QF increases with both movement and shrinkage. However, the increase seems to be less prominent for shrinking than translating. It should also be noted that the 0.5 mm shrink seems to be behaving different for both protons and photons than the rest of the color plot. In the plot for the standard deviation for protons it is the most prominent. The cause of this effect is unknown, but it might be caused by the scripts inability to shrink that specific amount or it might be that 0.5 mm shrink actually creates this effect.

5.6 Dose to Normal Tissue

Given the DVHs observed for the different parotid (figures 32a and 32b) one can not say anything certain given the large standard deviation. However, the data tends toward proton therapy giving a better dose to the ipsilateral parotid. This is further implied by the NTCP data in table 10 where the difference between the photon and proton plans are about one standard deviation.

For the spinal cord in figure 32c the matter is different. Although there is not quite two standard deviation difference between the proton and the photon plans, there is a clear indication that the protons give the lowest dose. From figure 32c the differences between each plan is minimized at the max dose. Generally, there seems to be little difference in DVHs for conventional radiotherapy vs DPBN, indicating that DPBN would not result in more toxicity.

It is likely that the authors inexperience at treatment planning may have affected these results as discussed above. For future works it would be wise to have an experience dose planner create DPBN plans for photons and protons to see if there is any real OAR dose sparing to be achieved with one method over the other. For conventional treatment planning there is observed a dose sparing to parotids [8, 9] and the Spinal Cord [8] when using protons instead of photons. However, it should be noted that Stutzer et al. emphasizes the importance of correct patient positioning and suggests frequent adaptation to ensure correct proton treatment [8].

The NTCP values for each parotid shows promise for the contralateral parotid although it is clear that the ipsilateral parotid might be damaged in some cases. In a clinical situation one should confer with the patient about these probabilities before commencing treatment. By plotting the NTCP as a function of the mean dose we find the graph in figure 36. Comparing this to a similar figure created by Roesink et al. [45, Figure 5] we find that the two figures are similar. This is as expected given that the m , n and $TD_{50}(1)$ values used in this thesis similar to those of Roesink et al, and thus serves as a quality check of the calculations done in the present study.

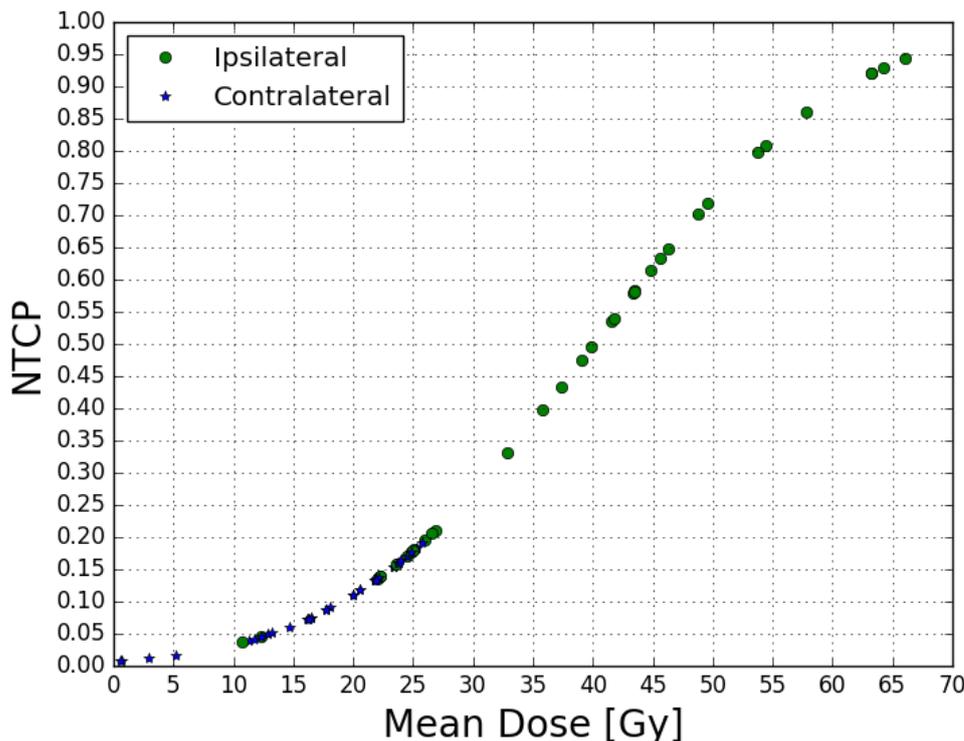


Figure 36: Complication probability as a function of mean parotid gland dose 12 months post treatment

5.7 Clinical Use

5.7.1 HPV

HPV is increasing as one of the causes of cancer in the head and neck area [2]. It is also found to increase the radiosensitivity of cancers in said area [71]. Although DPBN is not commonly used clinically in Norway, apart from a trial on recurrent head and neck cancer (<https://clinicaltrials.gov/ct2/show/NCT02921581>). Currently, patients in this trial who are HPV positive will not be treated with DPBN. As DPBN uses biological information to increase the treatment dose, it should also use the increased radiosensitivity to its advantage. Although more research is needed, one could explore if the increased radiosensitivity of HPV positive patients could be used to decrease the mean dose in the DPBN plan and still maintaining high TCP. Decreasing the dose will probably be positive, especially for photon treatment, as this will decrease the need for steep dose gradients. A more in depth discussion of this problem is beyond the scope of this thesis and is left for future works.

5.7.2 Recurring Cancer

A success rate of 70% witness that the treatment performed for these patients was quite good. However, there is always room for improvement. From this thesis it has been shown that the gain in TCP is bigger for the patients who had a recurring tumor, i.e. the 30% who had unsuccessful treatment. If one were to find a better identifier for which patients should receive DPBN one could give DPBN as the initial treatment and hopefully achieve an higher overall success rate. More research is needed also here, especially the SUV uncertainty provides a big challenge as it is the foundation of DPBN. However, there are some strong indications in this thesis towards the benefits of DPBN, and if treatment with this technique could increase tumor control for some patients, they should receive the treatment.

5.8 Future Works

Although future works have been mentioned in the text above, a brief summary will be given here. There have been done several studies on DPBN, but many of them does not implement a true voxel by voxel dose painting, but rather uses subcontours [52, 66–68]. In the future, one should do an in depth study of which of these methods give the best results. Probably, each method will have its advantages and disadvantage. One should also look into if the subcontours method requires less of the treatment machines and the TPS as this can be used as a compromise to improve efficiency.

One should also investigate further the linacs and proton treatment machines ability to deliver the planned dose. Some work has already been done for photons [69], but for proton treatment phantom studies should be undertaken to validate the DPBN..

As the robust DPBN dose plans were planned without a robust GTV because of constraints in the TPS, one could try to overcome this and create a truly robust dose plan. Thus, one could compare the results of a truly robust dose distribution with the ones found in this thesis. If one were to go even further one could also try giving these plans to phantoms and see if the robust treatment was successful even if one were to intentionally move the phantom.

For patients that have a recurring tumor, the TCP for the conventional plan seems to be lower, as found by this work. One should do a study on patients treated with conventional treatment compared to patients with DPBN to see if there can be any gain in successful treatment.

Lastly, the work done in this thesis should be repeated by a more experienced treatment planner. Especially the results regarding sparing of OARs would benefit from a more in depth study. One should also study the possibility of treating HPV positive patients as their increased radiosensitivity might be beneficial for DPBN treatment.

6 Conclusion

This work shows that it is possible to implement DPBN in a commercial TPS. Forty dose plans were created for ten patients. From these plans one could see that the proton plans conformed better than the photon plans to the prescribed DPBN dose map. TCP was about 0.7 for conventional plans, but with big differences within the patient group. The DPBN plans obtained highest TCPs of 0.96 and 0.97 for protons and photons respectively.

A tumor will shrink and move during treatment. By simulating this effect, one can see that the photon plans are less susceptible to change and may thus be the better option. Due to the amount of tumor shrinkage that may occur, one should not use DPBN through the whole treatment unless all cells in the wake of the shrinkage are necrotic. However, one could deliver DPBN for the first e.g. 10 fractions and treat conventionally afterwards. If one were to track tumor shrinkage during treatment, DPBN could be given until the tumor shrank past a set limit. Given that proton plans are more susceptible to change, if one were to give proton DPBN treatment clinically, one would have to heavily emphasize correct positioning and verification of tumor position before each treatment. As this is a general issue regarding proton therapy, there is much research done in this area.

It appears that there are great advantages to be gained from DPBN treatment, especially if one can track translation and shrinkage. The high TCP and low QF indicates that the treatment is both possible to give and will increase tumor control. This treatment technique could potentially be used more than it is today.

References

- [1] Cancer Registry of Norway. *Cancer in Norway 2016 - Cancer incidence, mortality, survival and prevalence in Norway*. Oslo: Cancer Registry of Norway, 2017.
- [2] Kreftforeningen. Hode- og halskreft - larynxcancer / pharynxcancer / oralcancer. <https://kreftforeningen.no/om-kreft/kreftformer/hode-og-halskreft/>. Last visited 17.04.2018.
- [3] Annika Jakobi, Anna Bandurska-Luque, Kristin Stützer, Robert Haase, Steffen Löck, Linda-Jacqueline Wack, David Mönnich, Daniela Thorwarth, Damien Perez, Armin Lühr, Daniel Zips, Mechthild Krause, Michael Baumann, Rosalind Perrin, and Christian Richter. Identification of patient benefit from proton therapy for advanced head and neck cancer patients based on individual and subgroup normal tissue complication probability analysis. *International Journal of Radiation Oncology*Biophysics*Physics*, 92(5):1165 – 1174, 2015.
- [4] Hans Paul van der Laan, Agata Gawryszuk, Miranda E.M.C. Christianen, Roel J.H.M. Steenbakkens, Erik W. Korevaar, Olga Chouvalova, Kim Wopken, Hendrik P. Bijl, and Johannes A. Langendijk. Swallowing-sparing intensity-modulated radiotherapy for head and neck cancer patients: Treatment planning optimization and clinical introduction. *Radiotherapy and Oncology*, 107(3):282 – 287, 2013.
- [5] Tara A. van de Water, Antony J. Lomax, Hendrik P. Bijl, Marije E. de Jong, Cornelis Schilstra, Eugen B. Hug, and Johannes A. Langendijk. Potential benefits of scanned intensity-modulated proton therapy versus advanced photon therapy with regard to sparing of the salivary glands in oropharyngeal cancer. *International Journal of Radiation Oncology*Biophysics*Physics*, 79(4):1216 – 1224, 2011.
- [6] Shravan Kandula, Xiaorong Zhu, Adam S. Garden, Michael Gillin, David I. Rosenthal, Kie-Kian Ang, Radhe Mohan, Mayankkumar V. Amin, John A. Garcia, Richard Wu, Narayan Sahoo, and Steven J. Frank. Spot-scanning beam proton therapy vs intensity-modulated radiation therapy for ipsilateral head and neck malignancies: A treatment planning comparison. *Medical Dosimetry*, 38(4):390 – 394, 2013.

- [7] Lamberto Widesott, Alessio Pierelli, Claudio Fiorino, Italo Dell'Oca, Sara Broggi, Giovanni Mauro Cattaneo, Nadia Di Muzio, Ferruccio Fazio, Riccardo Calandrino, and Marco Schwarz. Intensity-modulated proton therapy versus helical tomotherapy in nasopharynx cancer: Planning comparison and ntcp evaluation. *International Journal of Radiation Oncology*Biography*Physics*, 72(2):589 – 596, 2008.
- [8] Kristin Stützer, Annika Jakobi, Anna Bandurska-Luque, Steffen Barczyk, Carolin Arnsmeier, Steffen Löck, and Christian Richter. Potential proton and photon dose degradation in advanced head and neck cancer patients by intratherapy changes. 18, 09 2017.
- [9] Ingrid Kristensen, Kristina Nilsson, and Per Nilsson. Comparative proton and photon treatment planning in pediatric patients with various diagnoses. *International Journal of Particle Therapy*, 2(2):367–375, 2015.
- [10] Particle therapy facilities in operation.
<https://www.ptcog.ch/index.php/facilities-in-operation>, April 2018. Last visited 13.05.2018.
- [11] Particle therapy facilities under construction.
<https://www.ptcog.ch/index.php/facilities-under-construction>, January 2017. Last visited 13.05.2018.
- [12] Etablering av protonsentre i oslo og bergen.
<https://oushf.wordpress.com/2017/11/27/etablering-av-protonsentre-i-oslo-og-bergen/>. Last visited 13.05.2018.
- [13] Albin Fredriksson, Anders Forsgren, and Björn Hårdemark. Minimax optimization for handling range and setup uncertainties in proton therapy. *Medical Physics*, 38(3):1672–1684, 03 2011.
- [14] Søren M Bentzen. Theragnostic imaging for radiation oncology: dose-painting by numbers. *The Lancet Oncology*, 6(2):112 – 117, 2005.
- [15] M Alber, F Paulsen, S M Eschmann, and H J Machulla. On biologically conformal boost dose optimization. *Physics in Medicine and Biology*, 48(2):N31, 2003.
- [16] MR Arnesen et al. Dose painting by numbers in a standard treatment planning system using inverted dose prescription maps. *Acta Oncologica*, 2015.
- [17] Abdelkarim S Allal, Daniel O Slosman, Tayeb Kibdani, Mohamed Allaoua, Willy Lehmann, and Pavel Dulguerov. Prediction of outcome in head-and-neck cancer patients using the standardized uptake value of 2-[18f]fluoro-2-deoxy-d-glucose. *International Journal of Radiation Oncology*Biography*Physics*, 59(5):1295 – 1300, 2004.
- [18] Anne K Due et al. Recurrences after intensity modulated radiotherapy for head and neck squamous cell carcinoma more likely to originate from regions with high baseline [18f]-fdg uptake. *Radiother Oncol.*, 2014.
- [19] L.H. Sobin et al. (eds). *TNM classification of malignant tumours*. Wiley-Blackwell, seventh edition, 2010.
- [20] Dr. Jan Folkvard Evensen. Stadier ved kreft i svelg.
<http://oncolex.no/Hodehals/Diagnoser/Svelg/Bakgrunn/Stadier>. Last visited 30.04.2018.

- [21] Frank Herbert Attix. *Introduction to Radiological Physics and Radiation Dosimetry*. Wiley-VCH, 2004.
- [22] M A Flower. *Webb's physics of Medical Imaging*. Taylor and Francis Group, 2015.
- [23] Dr Yuranga Weerakkody and Kyle Greenway et al. Hounsfield units. <https://radiopaedia.org/articles/hounsfield-unit>. Last visited 12.04.2018.
- [24] Annelies van der Plas. Ct picture. <http://www.startradiology.com/the-basics/x-rayct-technique/>. Last visited 16.04.2018.
- [25] Wikipedia. Beta decay. https://en.wikipedia.org/wiki/Beta_decay, 2018.
- [26] CEA. Pet picture. http://irfu.cea.fr/Spp/en/Phocea/Vie_des_labos/Ast/ast.php?t=fait_marquant&id_ast=4100. Last visited 05.04.2018.
- [27] 18f-fdg: [18]f-fluorodeoxyglucose (fdg) imaging. <http://www.med.harvard.edu/JPNM/chetan/basics/basics.html>. Last visited 13.05.2018.
- [28] Paul E. Kinahan and James W. Fletcher. Positron emission tomography-computed tomography standardized uptake values in clinical practice and assessing response to therapy. *Seminars in Ultrasound, CT and MRI*, 31(6):496 – 505, 2010. PET-CT in RT Planning and Assessment.
- [29] T. Landberg, J. Chavaudra, J. Dobbs, J. P. Gerard, G. Hanks, J. C. Horiot, K. A. Johansson, T. Möller, J. Purdy, N. Suntharalingam, and H. Svensson. Report 62. *Journal of the International Commission on Radiation Units and Measurements*, os32(1):NP, 1999.
- [30] Eric J. Hall and Amato J. Giaccia. *Radiobiology for the Radiologist*. Lippincott Williams and Wilkins, a Woters Kluwer business, 2012.
- [31] Nobuyoshi Fukumitsu. Particle beam therapy for cancer of the skull base, nasal cavity, and paranasal sinus. *ISRN otolaryngology*, 2012.
- [32] Saad Saeed. *Dynamic Log Files Analysis For Different Dose Rate IMRT Using DVH and Gamma Index*. PhD thesis, Aga Khan University, 2015.
- [33] Cecilia Borges, Miriam Zarza-Moreno, E Heath, Nuno Teixeira, and Pedro Vaz. Monte carlo modeling and simulations of the high definition (hd120) micro mlc and validation against measurements for a 6 mv beam. 39:415–23, 01 2012.
- [34] Albin Fredriksson. *Robust optimization of radiation therapy accounting for geometric uncertainty*. PhD thesis, Kungl Tekniska Högskolan, 2013.
- [35] Jong Hoon Kim. Linac-based high-precision radiotherapy: Radiosurgery, image-guided radiotherapy, and respiratory-gated radiotherapy. *J Korean Med Assoc*, 2008. Last visited 23.04.2018.
- [36] Wikipedia. Bethe formula. https://en.wikipedia.org/wiki/Bethe_formula, 2017.
- [37] Alfred R. Smith. Vision : Proton therapy. *Medical Physics*, 36(2):556–568.
- [38] Unknown. Nagoya proton treatment center. http://www.nptc.city.nagoya.jp/e_proton/e_equipment.html. Last visited 24.04.2018.

- [39] M. Böck, A. Forsgren, K. Eriksson, and B. Hårdemark. Towards Robust Adaptive Radiation Therapy Strategies. *ArXiv e-prints*, November 2016.
- [40] RayStation Laboratories AB. Robust optimization in raystation. <https://www.raysearchlabs.com/globalassets/about-overview/media-center/wp-re-ev-n-pdfs/white-papers/white-paper-8—robust-omptimization-aug-2015.pdf>. Last visited 23.04.2018.
- [41] Francisco Cutanda Henríquez and Silvia Vargas Castrillón. A quality index for equivalent uniform dose. 36:126–32, 07 2011.
- [42] Brian E. Staveley. Dna illustration. https://www.mun.ca/biology/desmid/brian/BIOL2060/BIOL2060-18/18_04.jpg. Last visited 16.04.2018.
- [43] Jean-Pierre Pouget, Isabelle Navarro-Teulon, Manuel Bardiès, Nicolas Chouin, Guillaume Cartron, André Pèlegri, and David Azria. Clinical radioimmunotherapy—the role of radiobiology. *Nature reviews. Clinical oncology*, 8:720–34, 11 2011.
- [44] Yong Yang and Lei Xing. Towards biologically conformal radiation therapy (bcrt): selective imrt dose escalation under the guidance of spatial biology distribution. *Medical physics*, 32 6:1473–84, 2005.
- [45] Judith M. Roesink, Marinus A. Moerland, Jan J. Battermann, Gerrit Jan Hordijk, and Chris H.J. Terhaard. Quantitative dose-volume response analysis of changes in parotid gland function after radiotherapy in the head-and-neck region. *International Journal of Radiation Oncology*Biology*Physics*, 51(4):938 – 946, 2001.
- [46] John T. Lyman. Complication probability as assessed from dose-volume histograms. 8:S13–9, 02 1985.
- [47] Gerald J. Kutcher and C. Burman. Calculation of complication probability factors for non-uniform normal tissue irradiation: The effective volume method gerald. *International Journal of Radiation Oncology*Biology*Physics*, 16(6):1623 – 1630, 1989.
- [48] Sarah L. Gulliford, Mike Partridge, Matthew R. Sydes, Steve Webb, Philip M. Evans, and David P. Dearnaley. Parameters for the lyman kutcher burman (lkb) model of normal tissue complication probability (ntcp) for specific rectal complications observed in clinical practise. *Radiotherapy and Oncology*, 102(3):347 – 351, 2012.
- [49] M Alber et al. On biologically conformal boost dose optimization. *Phys. Med. Biol*, 48, 2003.
- [50] SM Bentsen. Theragnostic imaging for radiation oncology: dose-painting by numbers. *Lancet Oncol*, 2005.
- [51] Barbara Vanderstraeten, Wim Duthoy, Werner De Gerssem, Wilfried De Neve, and Hubert Thierens. [18f]fluoro-deoxy-glucose positron emission tomography ([18f]fdg-pet) voxel intensity-based intensity-modulated radiation therapy (imrt) for head and neck cancer. *Radiotherapy and Oncology*, 79(3):249 – 258, 2006.
- [52] A. M. Barragán, S. Differding, G. Janssens, J. A. Lee, and E. Sterpin. Feasibility and robustness of dose painting by numbers in proton therapy with contour-driven plan optimization. *Medical Physics*, 42:2006–2017, April 2015.

- [53] Einar Dale, Jon M. Moan, Terje A. Osnes, and Trond V. Bogsrud. Cervical lymph node metastases of squamous cell carcinoma of unknown origin: the diagnostic value of fdg pet/ct and clinical outcome. *European Archives of Oto-Rhino-Laryngology*, 274(2):1015–1019, Feb 2017.
- [54] Adrianus Johannes de Langen, Andrew Vincent, Linda M. Velasquez, Harm van Tinteren, Ronald Boellaard, Lalitha K. Shankar, Maarten Boers, Egbert F. Smit, Sigrid Stroobants, Wolfgang Andreas Weber, and Otto S. Hoekstra. Repeatability of 18f-fdg uptake measurements in tumors: a metaanalysis. *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, 53 5:701–8, 2012.
- [55] Paul Keall and S Webb. Optimum parameters in a model for tumour control probability, including interpatient heterogeneity: Evaluation of the log-normal distribution. 52:291–302, 02 2007.
- [56] Tejpal Gupta, Chandni Hotwani, Sadhana Kannan, Zubin Master, Venkatesh Rangarajan, Vedang Murthy, Ashwini Budrukkar, Sarbani Ghosh-Laskar, and Jai Prakash Agarwal. Prospective longitudinal assessment of parotid gland function using dynamic quantitative pertechnate scintigraphy and estimation of dose-response relationship of parotid-sparing radiotherapy in head-neck cancers. *Radiation oncology (London, England)*, 10:67, 2015.
- [57] Tim Dijkema, Cornelis P.J. Raaijmakers, Randall K. Ten Haken, Judith M. Roesink, Pètra M. Braam, Anette C. Houweling, Marinus A. Moerland, Avraham Eisbruch, and Chris H.J. Terhaard. Parotid gland function after radiotherapy: The combined michigan and utrecht experience. *International Journal of Radiation Oncology*Biophysics*, 78(2):449 – 453, 2010.
- [58] Aisha B. Miah, Sarah L. Gulliford, Catharine H. Clark, Shreerang A. Bhide, Shane H. Zaidi, Kate L. Newbold, Kevin J. Harrington, and Christopher M. Nutting. Dose–response analysis of parotid gland function: What is the best measure of xerostomia? *Radiotherapy and Oncology*, 106(3):341 – 345, 2013.
- [59] Avraham Eisbruch, Randall K. Ten Haken, Hyungjin M. Kim, Lon H. Marsh, and Jonathan A. Ship. Dose, volume, and function relationships in parotid salivary glands following conformal and intensity-modulated irradiation of head and neck cancer. *International Journal of Radiation Oncology*Biophysics*, 45(3):577 – 587, 1999.
- [60] Eun-Tae Park and Sung Kwang Park. Setup uncertainties for inter-fractional head and neck cancer in radiotherapy. *Oncotarget*, 7, 05 2016.
- [61] Inderjit Kaur, Sheh Rawat, Parveen Ahlawat, Anjali Kakria, Gourav Gupta, Upasna Saxena, and Maninder Bhushan Mishra. Dosimetric impact of setup errors in head and neck cancer patients treated by image-guided radiotherapy. *Journal of Medical Physics*, 41:144–148, 04 2016.
- [62] Jerry L. Barker, Adam S. Garden, K.Kian Ang, Jennifer C. O’Daniel, He Wang, Laurence E. Court, William H. Morrison, David I. Rosenthal, K.S.Clifford Chao, Susan L. Tucker, Radhe Mohan, and Lei Dong. Quantification of volumetric and geometric changes occurring during fractionated radiotherapy for head-and-neck cancer using an integrated ct/linear accelerator system. *International Journal of Radiation Oncology*Biophysics*, 59(4):960 – 970, 2004.

- [63] Barbara Vanderstraeten, Werner De Gerssem, Wim Duthoy, Wilfried De Neve, and Hubert Thierens. Implementation of biologically conformal radiation therapy (bcrt) in an algorithmic segmentation-based inverse planning approach. *Physics in Medicine & Biology*, 51(16):N277, 2006.
- [64] Gert Meijer, Jacco Steenhuijsen, Matthieu Bal, Katrien De Jaeger, Danny Schuring, and Jacqueline Theuws. Dose painting by contours versus dose painting by numbers for stage ii/iii lung cancer: Practical implications of using a broad or sharp brush. *Radiotherapy and Oncology*, 100(3):396 – 401, 2011. Special Issue on Medical Physics and Brachytherapy.
- [65] Daniela Thorwarth, Susanne-Martina Eschmann, Frank Paulsen, and Markus Alber. Hypoxia dose painting by numbers: A planning study. *International Journal of Radiation Oncology*Biophysics*, 68(1):291 – 300, 2007.
- [66] Stine S. Korreman, Silke Ulrich, Steve Bowen, Michael Deveau, Søren M. Bentzen, and Robert Jeraj. Feasibility of dose painting using volumetric modulated arc optimization and delivery. *Acta Oncologica*, 49(7):964–971, 2010.
- [67] Michael A. Deveau, Stephen R. Bowen, David C. Westerly, and Robert Jeraj. Feasibility and sensitivity study of helical tomotherapy for dose painting plans. *Acta Oncologica*, 49(7):991–996, 2010.
- [68] Malgorzata Skorska, Tomasz Piotrowski, Adam Ryczkowski, and Joanna Kaźmierska. Comparison of treatment planning parameters for dose painting head and neck plans delivered with tomotherapy. 89:20150970, 02 2016.
- [69] Iosif Papoutsis. From dose prescription to dose delivery - can dose painting by numbers be accurately delivered? Master’s thesis, Biophysics and Medical Physics, Department of Physics, The Faculty of Mathematics and Natural Sciences, University of Oslo, May 2017.
- [70] A Ahnesjö and MM Aspradakis. Dose calculations for external photon beams in radiotherapy. *Physics in medicine and biology*, 44(11):R99–155, November 1999.
- [71] Haïtham Mirghani, Furrat Amen, Yungan Tao, Eric Deutsch, and Antonin Levy. Increased radiosensitivity of hpv-positive head and neck cancers: Molecular basis and therapeutic perspectives. *Cancer Treatment Reviews*, 41(10):844 – 852, 2015.

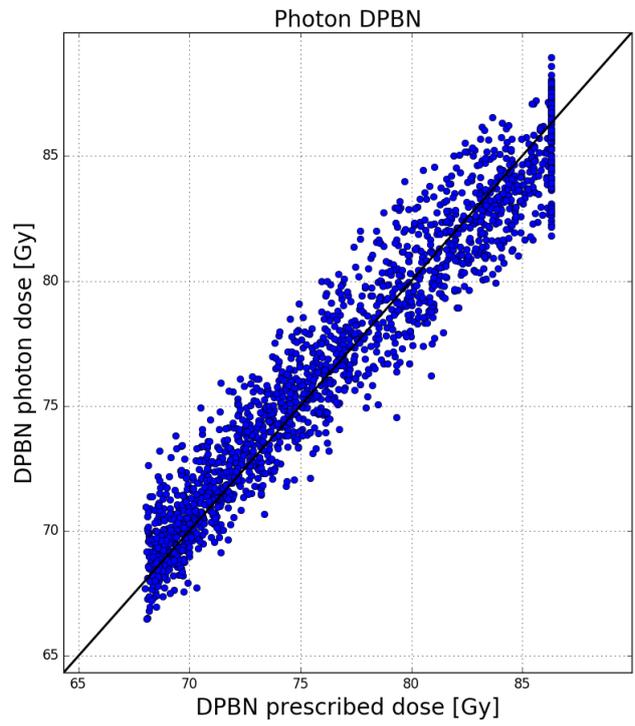
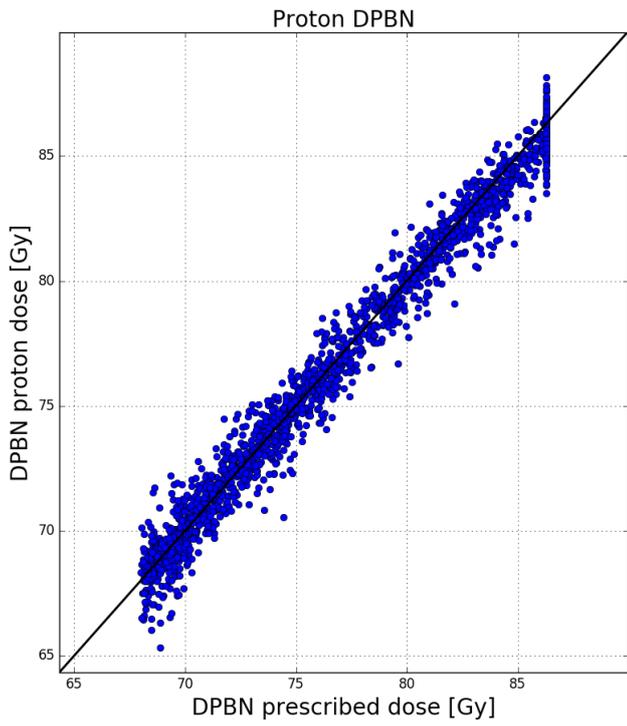
7 Appendix A

Table 14 contains the QF, TCP and D_{\max} for GTV. We can see that the conventional dose plans have a QF of about 0.10 and that the TCP varies.

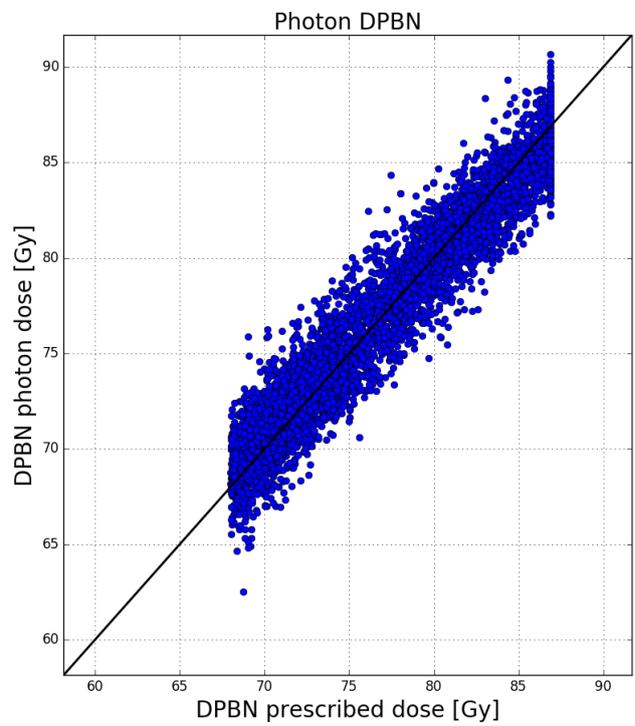
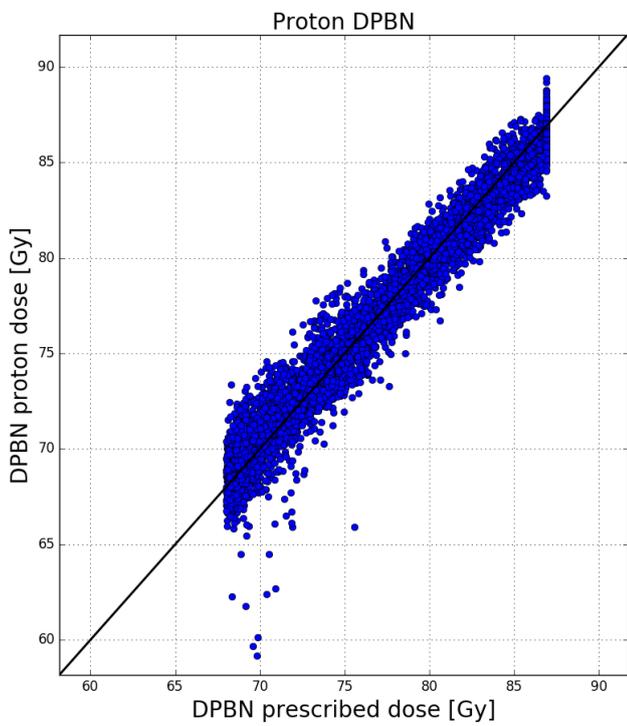
Patient	Plan	QF	TCP	D _{mean}
Patient 1	Photon	0.105	0.755	67.999
	Photon DPBN	0.016	0.982	76.635
	Proton	0.107	0.758	67.862
	Proton DPBN	0.008	0.978	76.340
Patient 2	Photon	0.106	0.497	67.968
	Photon DPBN	0.017	0.938	76.548
	Proton	0.105	0.403	68.076
	Proton DPBN	0.012	0.922	76.521
Patient 3	Photon	0.108	0.825	67.848
	Photon DPBN	0.015	0.983	76.233
	Proton	0.108	0.832	67.866
	Proton DPBN	0.009	0.979	76.180
Patient 4	Photon	0.106	0.699	67.955
	Photon DPBN	0.025	0.967	76.790
	Proton	0.105	0.713	68.044
	Proton DPBN	0.013	0.962	76.209
Patient 5	Photon	0.105	0.548	67.935
	Photon DPBN	0.022	0.957	76.658
	Proton	0.105	0.553	67.916
	Proton DPBN	0.016	0.942	76.159
Patient 6	Photon	0.106	0.732	68.065
	Photon DPBN	0.014	0.979	76.405
	Proton	0.107	0.726	68.006
	Proton DPBN	0.009	0.977	76.520
Patient 7	Photon	0.107	0.748	67.916
	Photon DPBN	0.022	0.981	76.478
	Proton	0.109	0.752	67.786
	Proton DPBN	0.015	0.972	75.941
Patient 8	Photon	0.107	0.694	67.784
	Photon DPBN	0.021	0.967	76.580
	Proton	0.107	0.701	67.820
	Proton DPBN	0.017	0.958	76.006
Patient 9	Photon	0.104	0.825	68.073
	Photon DPBN	0.013	0.983	76.464
	Proton	0.106	0.814	67.849
	Proton DPBN	0.008	0.978	76.208
Patient 11	Photon	0.106	0.583	67.912
	Photon DPBN	0.017	0.955	76.415
	Proton	0.107	0.582	67.885
	Proton DPBN	0.013	0.939	76.141

Table 14: The QF, TCP and D_{mean} for each GTV

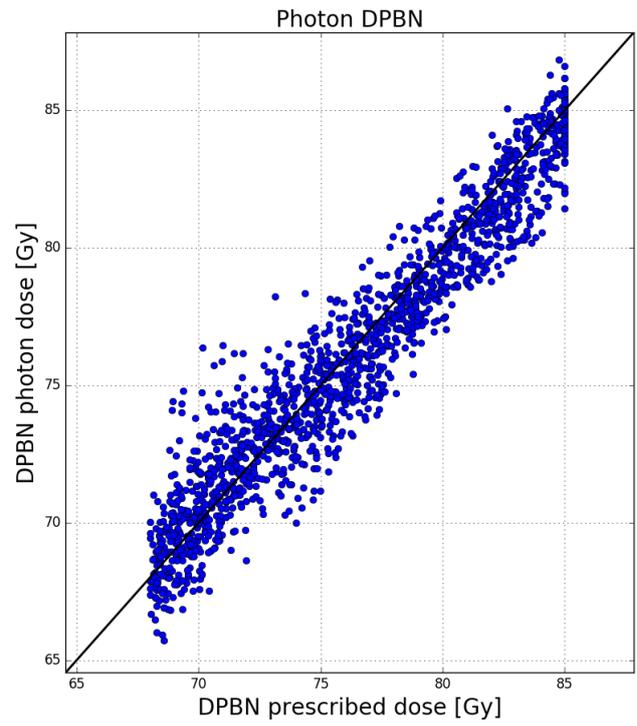
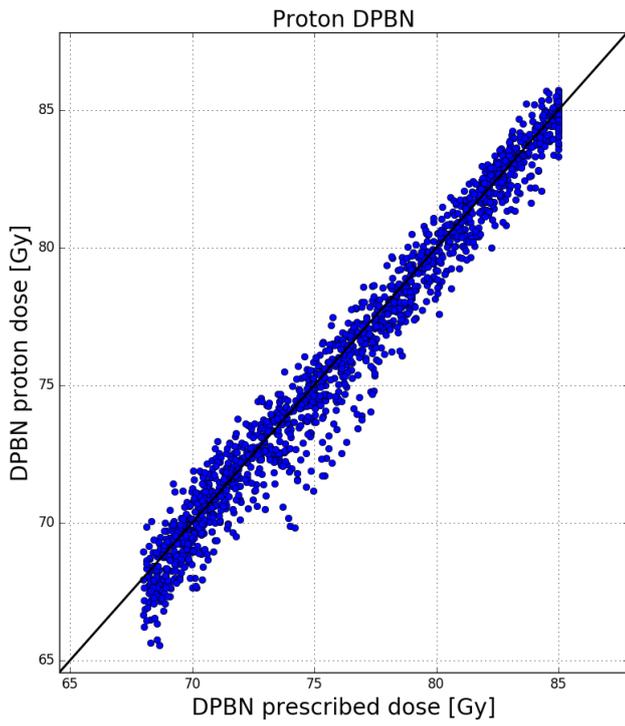
As figure 23 is combined of all patients, the individual patients are plotted in figure 37. Similar behavior is observed.



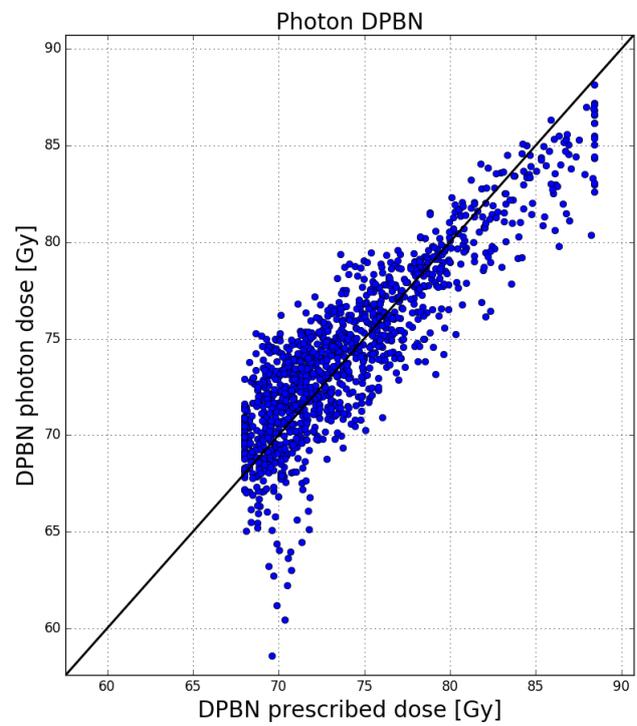
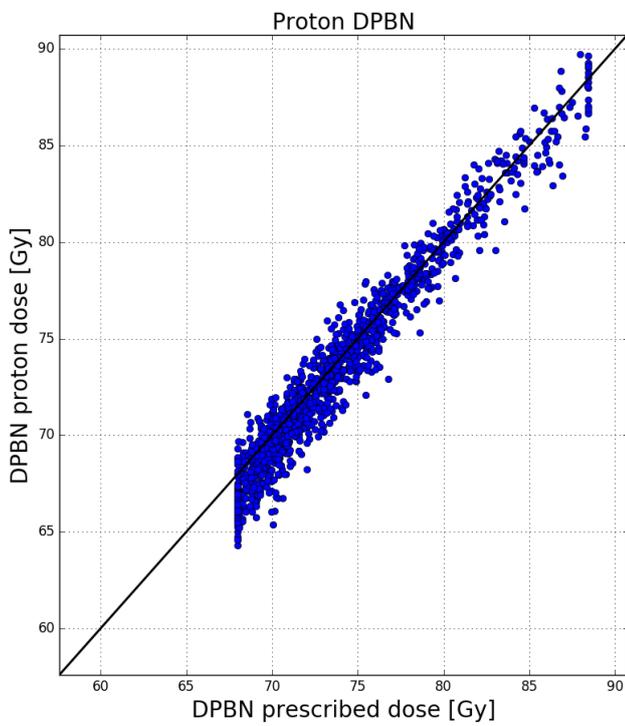
(a) Patient 1



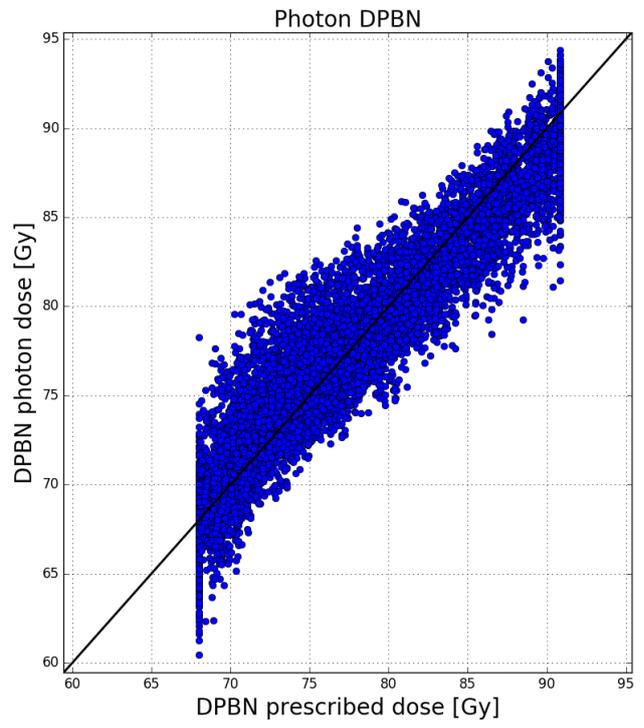
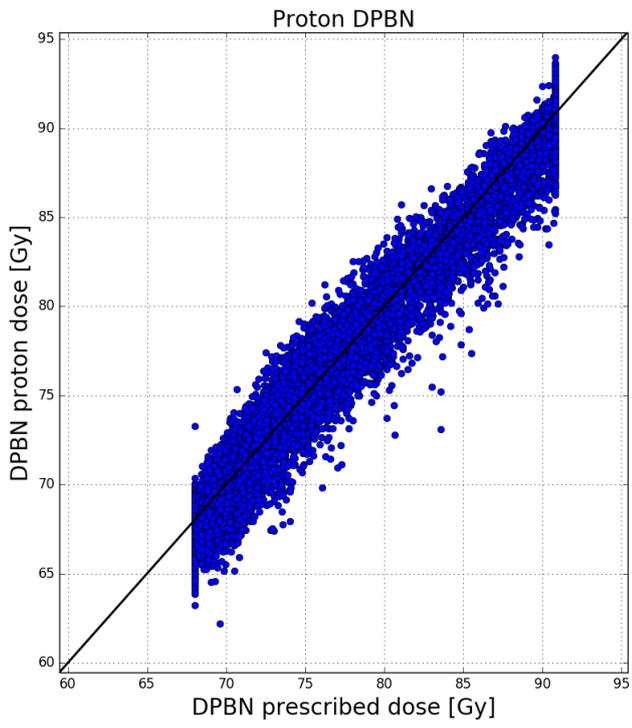
(b) Patient 2



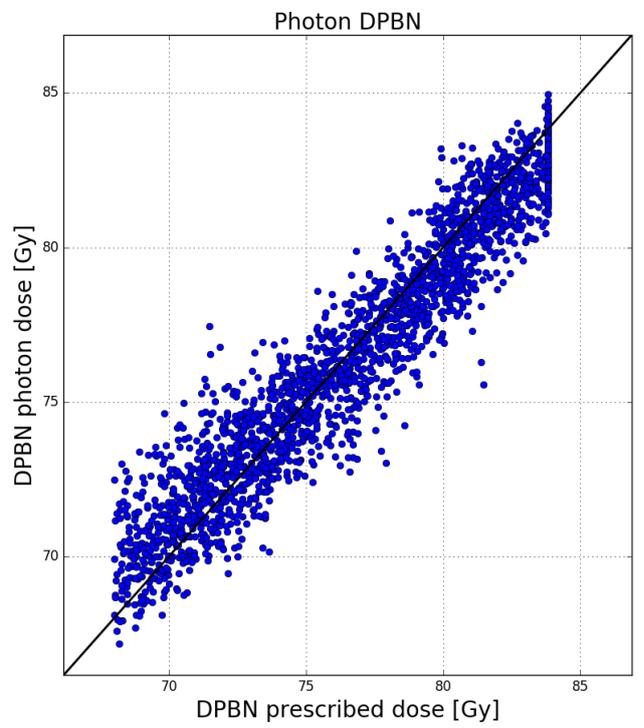
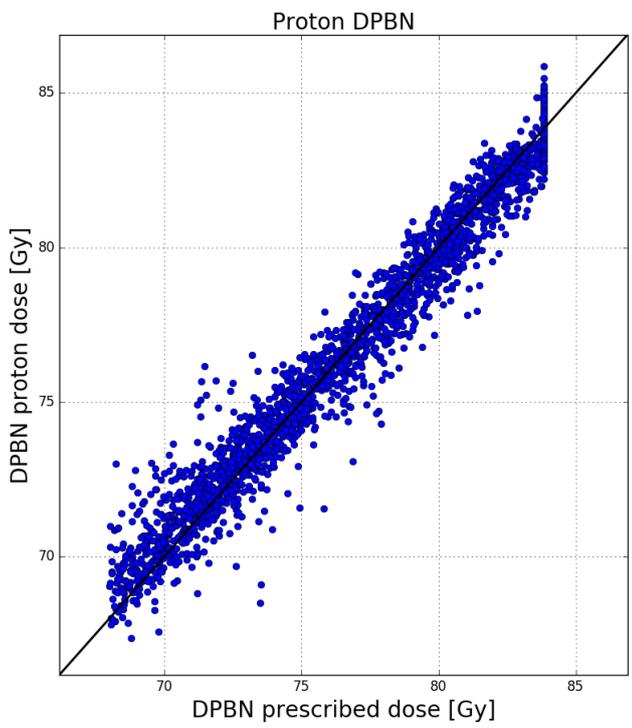
(c) Patient 3



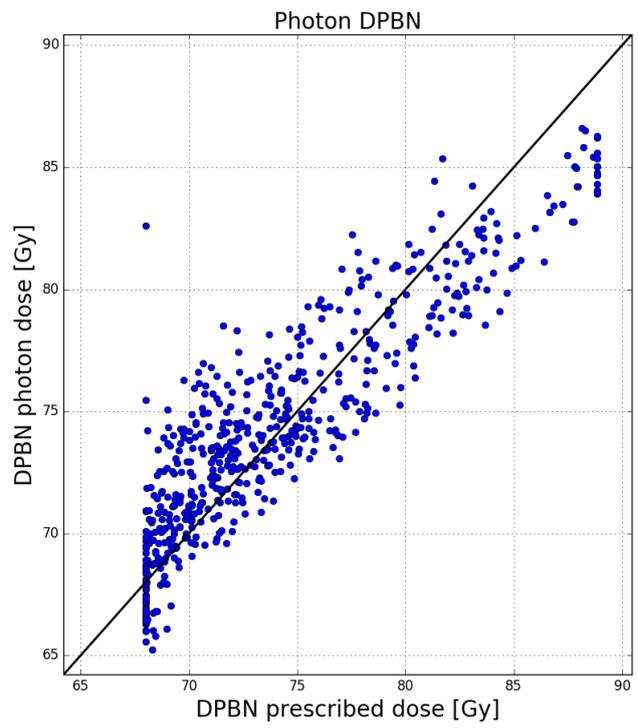
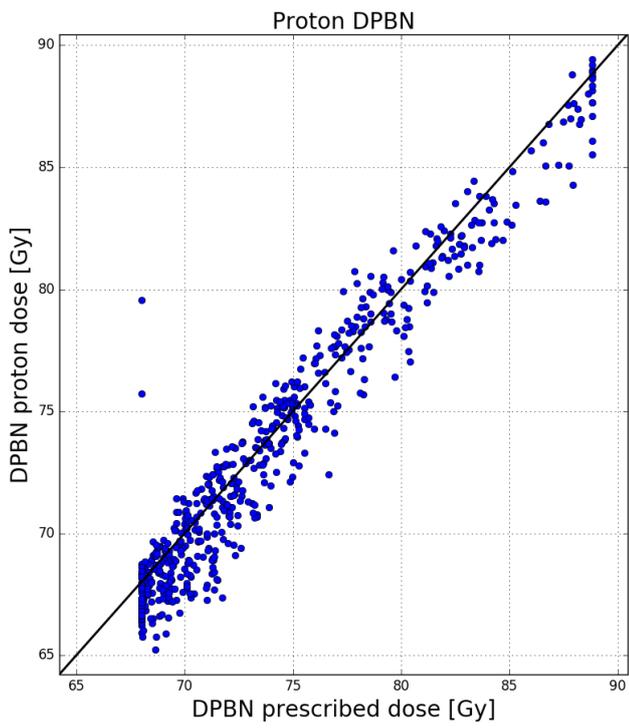
(d) Patient 4



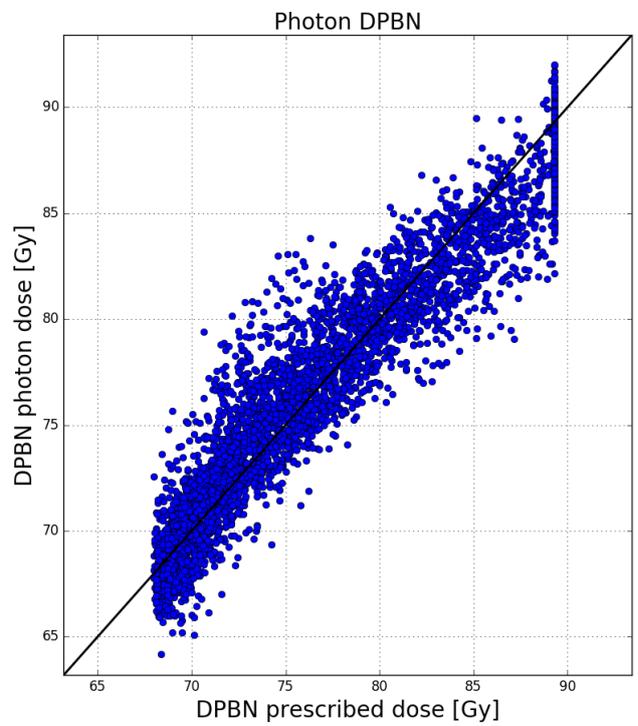
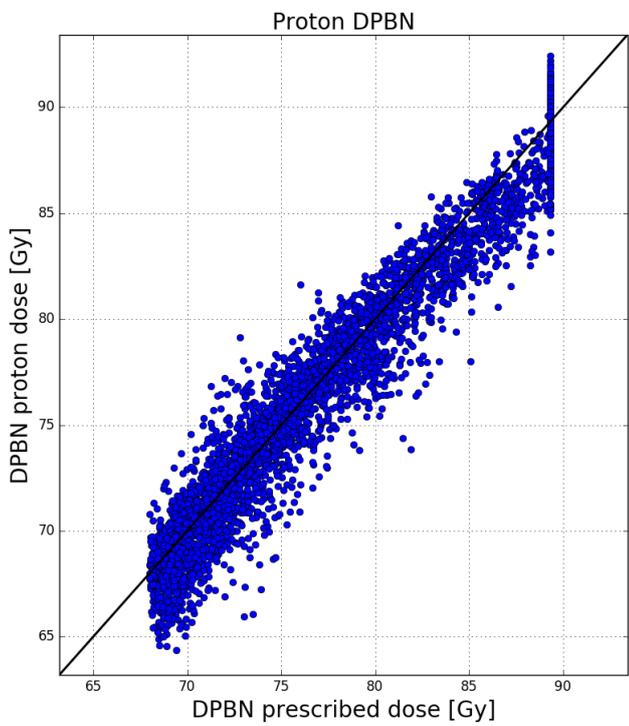
(e) Patient 5



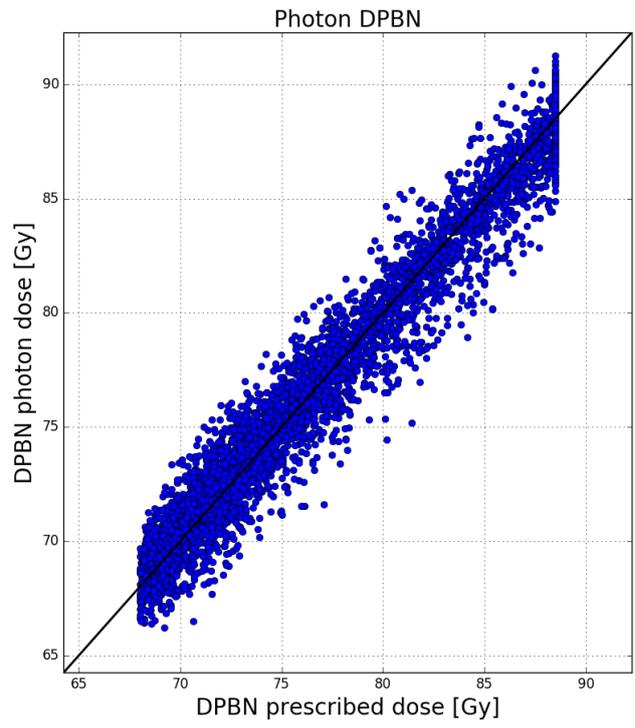
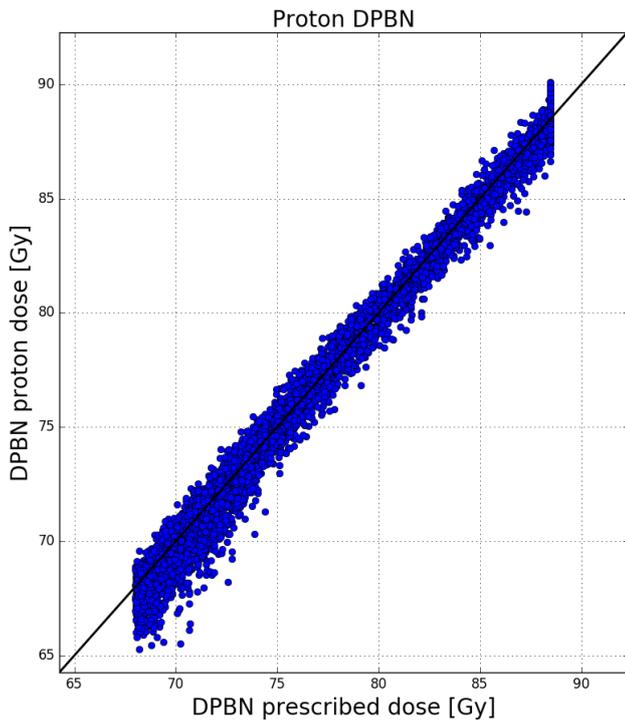
(f) Patient 6



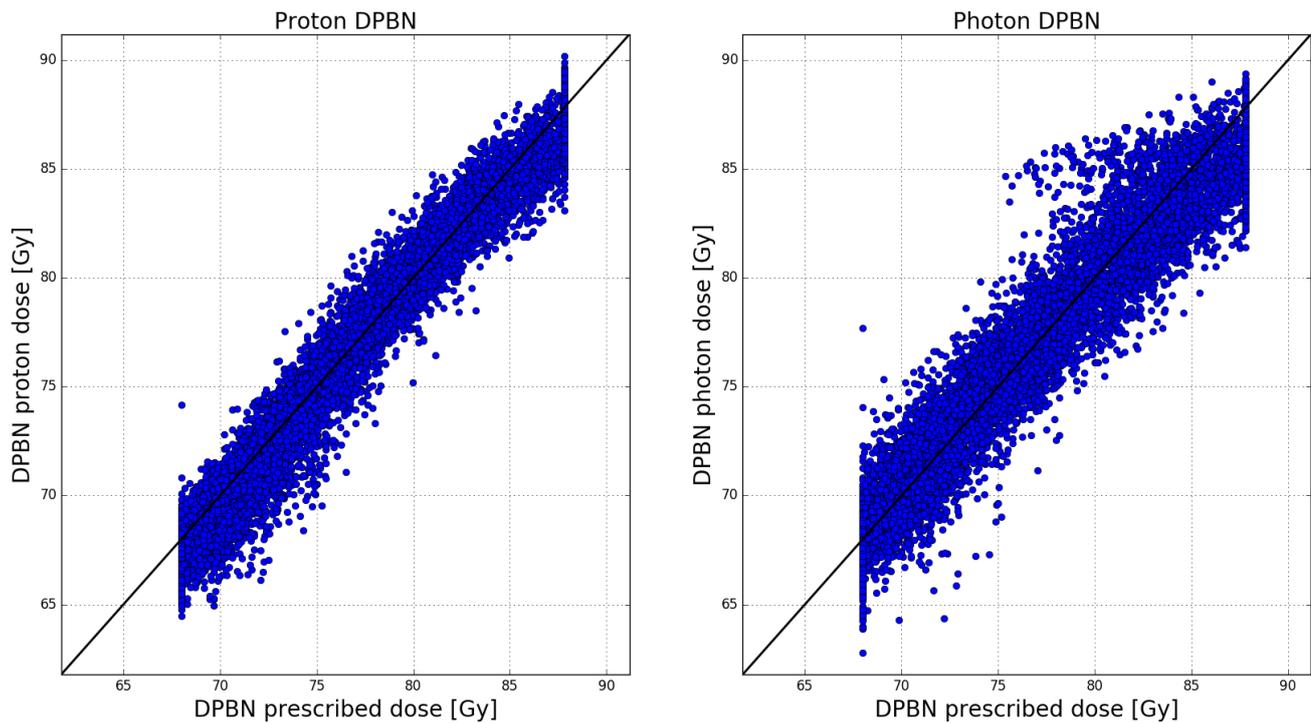
(g) Patient 7



(h) Patient 8



(i) Patient 9



(j) Patient 10

Figure 37: Comparison between prescribed plan, photon plan and proton plan for each patient's tumor. Each dot represents a voxel and its prescribed dose gives its x-position, while the photon/proton dose gives its y-position. The black line represents the identity line, i.e. where the planned plan is equal to the prescribed one.

Given the mean of mean and max dose in table 8, the individual data is displayed in table 15 and 16. Similar behavior is observed.

Patient	ROI	Mean/Max Dose	Photon	Photon DPBN	Proton	Proton DPBN
Patient 1	Ipsilateral	Max	68.17	68.99	67.43	69.13
		Mean	48.81	49.62	25.94	25.08
	Contralateral	Max	57.47	59.15	55.32	57.33
		Mean	23.38	23.33	11.35	16.12
	Spinal Cord	Max	43.68	44.57	37.14	38.18
		Mean	27.59	29.32	7.81	4.91
Patient 2	Ipsilateral	Max	62.91	63.46	63.86	77.61
		Mean	22.18	23.56	12.32	10.71
	Contralateral	Max	58.59	58.33	57.40	62.89
		Mean	21.84	25.25	14.66	12.93
	Spinal Cord	Max	32.04	25.41	19.08	23.93
		Mean	26.21	21.63	6.54	8.45
Patient 3	Ipsilateral	Max	66.71	66.15	65.37	66.14
		Mean	24.50	22.03	22.28	22.25
	Contralateral	Max	55.49	56.23	56.24	56.70
		Mean	21.95	20.60	16.23	20.02
	Spinal Cord	Max	42.81	42.12	39.03	36.31
		Mean	31.78	30.20	5.09	5.23
Patient 4	Ipsilateral	Max	68.43	81.18	69.15	81.55
		Mean	41.50	44.81	37.34	43.49
	Contralateral	Max	19.05	27.63	7.84	8.29
		Mean	2.98	5.19	0.63	0.61
	Spinal Cord	Max	29.69	16.25	2.23	4.77
		Mean	16.85	10.20	0.66	1.41

Table 15: Mean and Max Dose for Ipsilateral parotid, Contralateral parotid and Spinal Cord.

Patient	ROI	Mean/Max Dose	Photon	Photon DPBN	Proton	Proton DPBN
Patient 6	Ipsilateral	Max	70.06	83.25	68.84	85.10
		Mean	54.37	57.83	53.76	43.38
	Contralateral	Max	57.33	57.47	58.98	58.95
		Mean	23.73	24.48	18.11	22.15
	Spinal Cord	Max	45.81	42.98	39.37	45.82
		Mean	40.01	33.29	13.35	22.62
Patient 7	Ipsilateral	Max	68.37	78.78	68.10	77.49
		Mean	43.46	41.72	46.33	45.65
	Contralateral	Max	60.90	62.81	56.96	56.91
		Mean	21.77	19.94	17.77	17.70
	Spinal Cord	Max	45.02	45.61	38.00	39.65
		Mean	35.03	34.67	14.78	13.60
Patient 8	Ipsilateral	Max	68.91	73.84	68.13	74.18
		Mean	35.84	32.90	26.86	26.57
	Contralateral	Max	55.24	56.77	58.50	57.72
		Mean	16.47	16.43	11.77	12.32
	Spinal Cord	Max	41.58	40.42	38.34	37.27
		Mean	28.14	27.99	11.11	10.37
Patient 9	Ipsilateral	Max	67.72	69.92	65.22	66.61
		Mean	39.88	39.03	24.86	24.98
	Contralateral	Max	56.09	56.72	58.95	57.79
		Mean	25.71	24.82	23.82	23.99
	Spinal Cord	Max	38.93	38.70	36.93	37.15
		Mean	28.96	27.85	10.73	10.27
Patient 10	Ipsilateral	Max	68.78	81.56	68.70	83.31
		Mean	63.24	66.05	63.20	64.20
	Contralateral	Max	58.08	56.79	57.47	56.53
		Mean	13.26	21.86	20.59	16.20
	Spinal Cord	Max	44.35	42.11	36.12	39.94
		Mean	25.82	26.59	15.13	15.39

Table 16: Mean and Max Dose for Ipsilateral parotid, Contralateral parotid and Spinal Cord.

8 Appendix B: Scripts

More than 100 scripts were used for the work in this thesis and all can be found at the authors GitHub: <https://github.com/eirikraha/MasterScripts2>. The most important scripts are presented below, but for running the scripts or testing them out, I would recommend using GitHub. As a quick reminder, for IDL comments are initiated with ";", while for python they are initiated by "#".

The script `dpbn_max_norm_erh181017.pro` DICOM PET, CT and structure files and creates the prescribed and inverse DPBN dose maps:

```
1 Device , decomposed=0
2 loadct , 0
3 ;tag 001850 er snitt oppl dose
4 vol=52.5
5 d_low=68.
6 d_high=90.
7 p_low= 5.0;5.0
8 p_high= 100.0-p_low
9
10
11 set_base=1.
12 d_base=50.
13 m_r=2;
14 z_skal=2.0;funnet manuelt, m endres hvis annen oppl sning, tror den alltid
    vil v re lik snittykkelsen p ct, ev bruke det
15 ; Henter inn RT-strukturer og lagrer
16
17 figur_lokasjon = 'C:\Users\Eirik\Dropbox\Universitet\Master\Figurer '
18 txt_lokasjon = 'C:\Users\Eirik\Dropbox\Universitet\Master\Txt '
19 lokasjon='C:\Users\Eirik\Dropbox\Universitet\Master\pats\Patient 7 '
20 patient = (strsplit(lokasjon, '\', /EXTRACT))[-1]
21
22 dat_lokasjon = 'C:\Users\Eirik\Dropbox\Universitet\Master\dat_data\ '
23
24 cd, lokasjon
25 print , lokasjon
26 ;leser inn original dosefil
27 file_RD=(file_SEARCH('RD*'))[0]
28 obj = OBJ_NEW('IDLffDICOM')
29 read = obj->Read(file_RD)
30 ;pikseldata , zdim lang , pikseldata[i]=array (uin), [xdim, ydim]
31 pikseldata = obj->GetValue('7fe0'x, '0010'x);
32 xdim = obj->GetValue('0028'x, '0011'x)
33 ydim = obj->GetValue('0028'x, '0010'x)
34 zdim = obj->GetValue('0028'x, '0008'x)
35 dose_skal = obj->GetValue('3004'x, '000E'x)
36 posi = obj->GetValue('0020'x, '0032'x);x,y,z verste venstre hj rne?
37 res_xy=obj->GetValue('0028'x, '0030'x);pixel_spacing
38 bits_alloc=obj->GetValue('0028'x, '0100'x);
39 pixel_rep=obj->GetValue('0028'x, '0103'x);
40 sample_per=obj->GetValue('0028'x, '002'x);
41 high_b=obj->GetValue('0028'x, '0102'x);
42 obj_destroy , obj
43
44 xdim = float(*xdim[0])
45 ydim = float(*ydim[0])
46 zdim=float(*zdim[0])
47 dose_skal = float(strcompress(*dose_skal[0], /REMOVE_ALL))
48 posi=strsplit(*posi[0], '\', /EXTRACT)
49 xpos=float(strcompress(posi(0), /REMOVE_ALL))
50 ypos=float(strcompress(posi(1), /REMOVE_ALL))
51 zpos=float(strcompress(posi(2), /REMOVE_ALL))
52 res_xy=strsplit(*res_xy[0], '\', /EXTRACT)
53 res_x=float(strcompress(res_xy(0), /REMOVE_ALL))
54 res_y=float(strcompress(res_xy(1), /REMOVE_ALL))
55 bits_alloc=uint(*bits_alloc[0])
```

```

56 pixel_rep=uint(* pixel_rep [0])
57 sample_per=uint(* sample_per [0])
58 high_b=uint(* high_b [0])
59
60 ;for invers dosematrise til dicom
61 dp_inv=fltarr(xdim, ydim, zdim) ;res dosematrise
62 dim=size(dp_inv)
63
64 ;finner pet-filer
65 ;cd, lokasjon_pet
66 ;print, lokasjon_pet
67 ;file=file_search('f*')
68 file=file_search('PT*')
69 obj = OBJ_NEW('IDLffDICOM')
70 read = obj->Read(file(0))
71 ;finner div tagger i pet-fil
72 xdim_p = obj->GetValue('0028'x, '0011'x)
73 ydim_p = obj->GetValue('0028'x, '0010'x)
74 posi_p = obj->GetValue('0020'x, '0032'x);hj rne
75 res_xy_p=obj->GetValue('0028'x, '0030'x);pixel-spacing
76 z_skal_p=obj->GetValue('0018'x, '0050'x)
77
78 obj_destroy, obj
79
80 xdim_p = float(* xdim_p [0])
81 ydim_p = float(* ydim_p [0])
82 z_skal_p=float(* z_skal_p [0]); nb b r endres til avstand mellom snitt, ikke
    snittykkelse
83
84 posi_p=strsplit(* posi_p [0], '\', /EXTRACT)
85 xpos_p=float(strcompress(posi_p(0), /REMOVE_ALL))
86 ypos_p=float(strcompress(posi_p(1), /REMOVE_ALL))
87
88 res_xy_p=strsplit(* res_xy_p [0], '\', /EXTRACT)
89 res_x_p=float(strcompress(res_xy_p(0), /REMOVE_ALL))
90 res_y_p=float(strcompress(res_xy_p(1), /REMOVE_ALL))
91
92 x_shift=round((xpos-xpos_p)/res_x)
93 y_shift=round((ypos-ypos_p)/res_y)
94 x_skal=round(xdim_p*res_x_p/res_x); nb, round gj r ting litt feil her...
95 y_skal=round(ydim_p*res_y_p/res_y)
96
97 pet_matrix=fltarr(xdim, ydim, round(n_elements(file)*z_skal_p/z_skal));pet i
    dose ref
98
99
100 adhoc_just=[0,1] ;P1 = [0, 1], P2 = []
101 zpos_orig=zpos;ta vare p original zpos dosefil, h rer til strukt osv
102 ;leser inn alle billed-filer til matrise
103 diff_ref=0
104 if diff_ref then begin
105     zpos=1000
106     x_shift=0;??
107     y_shift=0;??
108     ;finner minste z verdi pet opptak
109     for i=0, n_elements(file)-1 do begin
110         obj = OBJ_NEW('IDLffDICOM')
111         read = obj->Read(file(i))
112         zpos_d=obj->GetValue('0020'x, '1041'x)

```

```

113     zpos_d=float(*zpos_d[0])
114     if zpos_d lt zpos then zpos=zpos_d
115     obj_destroy , obj
116   endfor
117   x_shift=round((xpos-xpos_p)/res_x);??
118   y_shift=round((ypos-ypos_p)/res_y);??
119 endif
120
121
122 zpos_dd = 0
123
124 for i=0, n_elements(file)-1 do begin
125
126   obj = OBJ_NEW('IDLFFDICOM')
127   read = obj->Read(file(i))
128   pikseldata = read_dicom(file(i))
129   conv_kern=obj->GetValue('0018'x, '1210'x)
130   xd = obj->GetValue('0028'x, '0011'x)
131   yd = obj->GetValue('0028'x, '0010'x)
132   posi_d = obj->GetValue('0020'x, '0032'x);x,y,z verste venstre hj rne?
133   res_xy_d=obj->GetValue('0028'x, '0030'x);pixel_spacing
134   pet_skal= obj->GetValue('0028'x, '1053'x)
135   pet_skal=float(*pet_skal[0]);OBS kan v re ulik for hvert snitt
136   ;print, 'conv-kern ', *conv_kern[0]
137   xd = *xd[0]
138   yd = *yd[0]
139   posi_d=strsplit(*posi_d[0], '\', /EXTRACT)
140   xpos_d=float(strcompress(posi_d(0), /REMOVE_ALL))
141   ypos_d=float(strcompress(posi_d(1), /REMOVE_ALL))
142   zpos_d=float(strcompress(posi_d(2), /REMOVE_ALL))
143   res_xy_d=strsplit(*res_xy_d[0], '\', /EXTRACT)
144   res_x_d=float(strcompress(res_xy_d(0), /REMOVE_ALL))
145   res_y_d=float(strcompress(res_xy_d(1), /REMOVE_ALL))
146   ;kontr at alle geom-param er like, kan slettes
147   if xdim_p ne xd or ydim_p ne yd or xpos_d ne xpos_p or ypos_d ne ypos_p or
148     res_y_d ne res_y_p or res_x_d ne res_x_p then stop
149   ;zpos_d=obj->GetValue('0020'x, '1041'x)
150   ;zpos_d=float(*zpos_d[0])
151
152   z_thickness = zpos_dd - zpos_d ;Ikke sikkert dette fungerer for alle PET-
153     filer , men fungerer for test pasient 1
154   zpos_dd = zpos_d
155
156   temp=congrid(float(pikseldata), x_skal, y_skal) ; sikre samme oppl sing i
157     dose og pet-bilde
158   ;innlesing til matrise i dosereferanse robust for ulik st rrelse og
159     plassering av pet-matrise
160   if zpos_d ge zpos then begin
161     for j=0, xdim-1 do begin
162       j_p=j+x_shift+adhoc_just(0)
163       if j_p ge 0 and j_p lt x_skal then begin
164         for k=0, ydim-1 do begin
165           k_p=k+y_shift+adhoc_just(1)
166           if k_p ge 0 and k_p lt y_skal then pet_matrix(j,k,round((zpos_d-
167             zpos)/z_skal))=pet_skal*temp(j_p, k_p)
168         endfor
169       endif
170     endfor
171   endif
172 endfor
173 endif

```

```

167 endfor
168
169
170 obj_destroy , obj
171 new_pet_dim=size(pet_matrix);
172 zpos=zpos_orig; tilbake til zpos dosematrise , i tilfelle ulik ref
173
174 ;fyller inn manglende pet_snitt (hvis st rre snittavstand pet og dosematrise
    );
175 for i=1,new_pet_dim(3)-2 do begin ;antar at det ikke er noe interessant i ev
    manglene f rste eller siste snit
176     if max(pet_matrix(* ,* ,i)) eq 0 then pet_matrix(* ,* ,i)=.5*(pet_matrix(* ,* ,i
        -1)+pet_matrix(* ,* ,i+1));
177 endfor
178
179 ;—— Skriv PET matrise til fil ——
180 ;openw , 4 , 'C:\Users\Eirik\Dropbox\Universitet\Master\pet_matrix.txt '
181 ;printf , 4 , pet_matrix
182 ;close , 4
183 ;—————
184
185 ;—kun aktuelt hvis pet og ct ikke er fra samme sesjon
186 ;z_diff_reg=70
187 ;x_diff_reg=2
188 ;y_diff_reg=-10
189 ;
190 ;if diff_ref then begin
191 ;   temp_pet_matrix=pet_matrix
192 ;   pet_matrix=fltarr(xdim , ydim , zdim)
193 ;   for i=0 , n_elements(file)-1 do begin
194 ;     pet_matrix(* ,* ,i+z_diff_reg)=shift(reform(temp_pet_matrix(* ,* ,i)) ,
        x_diff_reg , y_diff_reg)
195 ;   endfor
196 ;endif
197 ;—————
198
199 ;finner struktur-fil
200 ; NB
201 cd , lokasjon
202
203 file=file_search('RS*')
204 cont_pointr = use_rtss_test(file) ;Egentlig bare use_rtss
205
206 numbers = n_elements(cont_pointr[* , 0])
207 cont_mat= cont_pointr[* , 0]
208 cont_sort_mat= cont_pointr[* , 1]
209 punkt_indeks_mat= cont_pointr[* , 2]
210
211
212 ;test just——
213 dim_error=fltarr(4)
214 dim_error(0)=x_shift*res_x-(-xpos_p+xpos)
215 dim_error(1)=y_shift*res_y-(-ypos_p+ypos)
216 dim_error(2)=x_skal/(xdim_p*res_x_p/res_x)
217 dim_error(3)=y_skal/(ydim_p*res_y_p/res_y)
218 print , dim_error
219
220 store_matrix = ptrarr(numbers , 11)
221

```

```

222
223 for numb=0, numbers - 1 do begin
224   cont = *(cont_mat[numb])
225   cont_sort = *(cont_sort_mat[numb])
226   punkt_indeks = *(punkt_indeks_mat[numb])
227
228   cont(0,*)=(cont(0,*)-xpos)/res_x
229   cont(1,*)=(cont(1,*)-ypos)/res_y
230
231   ;f r orden p   indeksering av snittskifte i punktliste
232   for i=n_elements(punkt_indeks)-1,0, -1 do punkt_indeks(i)=total(
      punkt_indeks(0:i))-punkt_indeks(i)
233
234
235   ;sorterer punktliste etter stigende z
236   s_i=sort(cont(2,punkt_indeks))
237
238   punkt_indeks_sort=intarr(n_elements(punkt_indeks))
239   z_value = fltarr(n_elements(punkt_indeks))
240   count=0
241   for i=0, n_elements(s_i)-1 do begin
242     if s_i(i) eq n_elements(punkt_indeks) -1 then begin
243       ant=n_elements(cont(2,*))-punkt_indeks(s_i(i))
244     endif else begin
245       ant=punkt_indeks(s_i(i)+1)-punkt_indeks(s_i(i))
246     endelse
247     punkt_indeks_sort(i)=count
248     cont_sort(*,punkt_indeks_sort(i):punkt_indeks_sort(i)+ant-1)=cont(*,
      punkt_indeks(s_i(i)):punkt_indeks(s_i(i))+ant-1)
249     z_value(i) = cont(2,punkt_indeks(s_i(i)))
250     (* cont_sort_mat[numb])[*,punkt_indeks_sort(i):punkt_indeks_sort(i)+ant-1]
      = cont(*,punkt_indeks(s_i(i)):punkt_indeks(s_i(i))+ant-1)
251     count=count+ant
252   endfor
253
254   ;lage mask for roi til pet-matrise
255   gtv_pet_mask=intarr(dim(1),dim(2), dim(3))
256   for i=0,n_elements(punkt_indeks_sort)-1 do begin
257     snitt=((cont_sort(2,punkt_indeks_sort(i))-zpos)/z_skal)
258     if i ne n_elements(punkt_indeks_sort)-1 then points=reform(cont_sort(0:1,
      punkt_indeks_sort(i):punkt_indeks_sort(i+1)-1))
259     if i eq n_elements(punkt_indeks_sort)-1 then points=reform(cont_sort(0:1,
      punkt_indeks_sort(i):n_elements(cont_sort(2,*))-1))
260     obj_midl=OBJ_NEW('IDLanRoi', points(0,*),points(1,*))
261     mask_midl=obj_midl->IDLanRoi::ComputeMask(dimensions=[dim(1),dim(2)],
      mask_rule=m_r)
262     gtv_pet_mask(*,*, snitt)=gtv_pet_mask(*,*, snitt)+mask_midl; kan ha to
      lukkede konturer i samme snitt
263     obj_destroy, obj_midl
264   endfor
265
266
267   ;marise med nsket dp dosefordeling i gtv_pet, i samme ref-system som hele
      pet-matrisen for enkelhets skyld
268   dp_matr=fltarr(dim(1), dim(2), dim(3))
269   gtv_pet_ind=where(gtv_pet_mask);indekser til gtv_pet [som er over 0], [] =
      Lagt til av Eirik
270
271   store_matrix[numb, 0] = ptr_new(gtv_pet_mask)

```

```

272 store_matrix[numb, 2] = ptr_new(gtv_pet_ind)
273 store_matrix[numb, 9] = ptr_new(punkt_indeks_sort)
274 store_matrix[numb, 10] = ptr_new(z_value)
275 endfor
276
277 counter = 0
278 for i=0, numbers-1 do begin
279   for j = i+1, numbers-1 do begin
280     for k=0, n_elements((*store_matrix[i, 2]))-1 do begin
281       for l=0, n_elements((*store_matrix[j, 2]))-1 do begin
282         if (*store_matrix[i, 2])[k] eq (*store_matrix[j, 2])[l] then begin
283           ;print, 'You have overlapping doseplans at the index:', (*
store_matrix[i, 2])[k], (*store_matrix[j, 2])[l]
284             counter++
285           endif
286         endfor
287       endfor
288     endfor
289   endfor
290
291   print, 'You have ', counter, ' overlapping doseplan indeicies'
292
293
294   gtv_pet_ind = []
295
296   for i=0, numb-1 do begin
297     gtv_pet_ind = [gtv_pet_ind, (*store_matrix[i, 2])] ;Legger
alle indekser p samme matrise slik at jeg kan bruke alt til lage n
dosematrise.
298   endfor
299
300   pet_matrix_orig = pet_matrix
301   dp_matr_orig = dp_matr
302
303   n_tot=n_elements(gtv_pet_ind)
304   max_pet=max(pet_matrix(gtv_pet_ind))
305   min_pet=min(pet_matrix(gtv_pet_ind))
306
307   ;fast andel som f r maks og min
308   n_low=p_low
309   n_high=1.-p_high
310   temp_gtv=pet_matrix(gtv_pet_ind)
311   sort_gtv=temporary(temp_gtv(sort(temp_gtv)))
312
313   ;setter alt utenfor gtv_pet til mindose pga mer robust mhp scoring av dose
langs kanten av roi i doseplansyst
314   if set_base then begin
315     dp_matr(*,*,*)=d_low
316     dp_matr(gtv_pet_ind)=0.0
317   endif
318   print, min_pet, max_pet, mean(pet_matrix(gtv_pet_ind))
319
320   ;prank trenger prosent i tall fra 0 til 100, ikke 0 til 1.
321   if p_low gt 0.0 then min_pet=prank(pet_matrix(gtv_pet_ind), p_low) ;
sort_gtv(round(n_tot*p_low)-1)
322   if p_high lt 100.0 then max_pet=prank(pet_matrix(gtv_pet_ind), p_high)
323   print, min_pet, max_pet, mean(pet_matrix(gtv_pet_ind))
324
325

```

```

326 ;Finner D_high for YX
327 rho_ref = max_pet
328
329 sigma2 = total(aolog(rho_ref/pet_matrix[gtv_pet_ind]))
330 N = n_elements(pet_matrix[gtv_pet_ind])
331 n_tot = n_elements(gtv_pet_ind)
332
333 p_high = 0.01*p_high
334 p_low = 0.01*p_low ;Endres her fordi n m jeg g fra 0 til 100 for prank
    til 0 til 1 intervall
335
336 n_low=p_low
337 n_high=1.-p_high
338 n_mean=p_high-p_low
339 temp_gtv=pet_matrix(gtv_pet_ind)
340 sort_gtv=temporary(temp_gtv(sort(temp_gtv)))
341
342 d_avg = 76.0
343 d_mean = d_avg
344
345 n_tot = N
346
347 dp_skal=total((sort_gtv(round(n_tot*p_low):round(n_tot*p_high)-1)-min_pet)/(
    max_pet-min_pet))/(n_tot*n_mean);litt feil med avrunding her, men neppe
    relevant mange tusen piksler totalt
348 d_high_l=(d_mean+d_low*(dp_skal*n_mean-n_mean-n_low))/(dp_skal*n_mean+n_high)
    ;d_high found for a set d_mean
349
350
351 dp_matr(gtv_pet_ind)=d_low+(pet_matrix(gtv_pet_ind)-min_pet)*(d_high_l-d_low)
    /(max_pet-min_pet)
352 for i=long(0), n_tot-1 do begin
353     if dp_matr(gtv_pet_ind(i)) lt d_low then dp_matr(gtv_pet_ind(i))=d_low
354     if dp_matr(gtv_pet_ind(i)) gt d_high_l then dp_matr(gtv_pet_ind(i))=
        d_high_l
355
356 endfor
357
358 ;lager inv matrise for summert optimalisering i tps
359 ;setter d=d_baae utenfor gtv_pet og som maxdose til gtv_pet i invers matrise,
    minstedose i invers matrise blir d_base-(d_high-d_low)
360 if set_base then begin
361     dp_inv=d_low+d_base-dp_matr;
362 endif else begin
363     dp_inv(gtv_pet_ind)=d_low+d_base-dp_matr(gtv_pet_ind)
364 endelse
365
366
367
368 ;div sjekk av verdier
369 d_mean=mean(dp_matr(gtv_pet_ind))
370 print, 'Total presc: ', max(dp_matr(gtv_pet_ind)), min(dp_matr(gtv_pet_ind)),
    mean(dp_matr(gtv_pet_ind))
371 print, 'Total inv: ', max(dp_inv(gtv_pet_ind)), min(dp_inv(gtv_pet_ind))
372 for i=0, numb-1 do begin
373     print, 'Kontur ', *(cont_ptr[* , 3])[i], ' presc: ', max(dp_matr[( *
        store_matrix[i, 2])]), min(dp_matr[( * store_matrix[i, 2])]), mean(dp_matr
        [( * store_matrix[i, 2])])
374     print, 'Kontur ', *(cont_ptr[* , 3])[i], ' inv: ', max(dp_inv[( *

```

```

    store_matrix[i, 2]))), min(dp_inv [( * store_matrix [i, 2] ]))
375 endfor
376
377 A = 'Total presc: ' + strcompress(max(dp_matr(gtv_pet_ind))) + strcompress(
    min(dp_matr(gtv_pet_ind))) + strcompress(mean(dp_matr(gtv_pet_ind)))
378
379 txt_filename = txt_lokasjon + '/' + patient + '_total_dose_' + strcompress(
    numbers) + '_ROIs' + '.txt'
380 openw, 1, txt_filename
381 printf, 1, FORMAT = '(%"Total presc: %f %f %f")', max(dp_matr(gtv_pet_ind)),
    min(dp_matr(gtv_pet_ind)), mean(dp_matr(gtv_pet_ind))
382 printf, 1, FORMAT = '(%"Total inv: %f %f")', max(dp_inv(gtv_pet_ind)), min(
    dp_inv(gtv_pet_ind))
383 for i=0, numb-1 do begin
384     printf, 1, FORMAT = '(%"Kontur: %s, presc: %f %f %f")', *(cont_pointr[* ,
        3])[i], max(dp_matr [( * store_matrix [i, 2] ])), min(dp_matr [( * store_matrix [i
        , 2] ])), mean(dp_matr [( * store_matrix [i, 2] ])) ; 'Kontur ', *(cont_pointr
        [* , 3])[i], ' presc: ', max(dp_matr [( * store_matrix [i, 2] ])), min(dp_matr
        [( * store_matrix [i, 2] ])), mean(dp_matr [( * store_matrix [i, 2] ]))
385     printf, 1, FORMAT = '(%"Kontur: %s, inv: %f %f")', *(cont_pointr[* , 3])[i],
        max(dp_inv [( * store_matrix [i, 2] ])), min(dp_inv [( * store_matrix [i, 2] ])) ;
        'Kontur ', *(cont_pointr[* , 3])[i], ' inv: ', max(dp_inv [( * store_matrix [i
        , 2] ])), min(dp_inv [( * store_matrix [i, 2] ]))
386 endfor
387
388 close, 1
389
390 ;—— Skriv dose matrise og ROI indeks til fil ——
391 ;openw, 4, 'C:\Users\Eirik\Dropbox\Universitet\Master\presc_matrix.txt'
392 ;printf, 4, dp_matr
393 ;close, 4
394 ;
395 ;openw, 5, 'C:\Users\Eirik\Dropbox\Universitet\Master\gtv_pet_ind.txt'
396 ;printf, 5, gtv_pet_ind
397 ;close, 5
398 ;
399
400 ;—————Sammenlign med Researchs doseprofil—————
401 ;Skriver egentlig ut informasjonen funnet til .dat filer
402 ans = ''
403 READ, ans, prompt = 'Har du en dosefil sammenligne mot? (Y/n) '
404 if ans eq 'Y' then begin
405     a = write_gtv_pet_ind((dat_lokasjon + patient + '_gtv_pet_ind.dat'),
        gtv_pet_ind)
406     b = write_dat((dat_lokasjon + patient + '_dp_matr.dat'), dp_matr,
        n_elements(dp_matr), [xdim, ydim, zdim], [res_x, res_y, z_skal], posi,
        34)
407     c = write_dat((dat_lokasjon + patient + '_pet_matrix.dat'), pet_matrix,
        n_elements(pet_matrix), [xdim, ydim, zdim + 1], [res_x, res_y, z_skal],
        posi, 34)
408 endif
409
410
411
412
413 ;—————div for visuell sjekk (kan slettes)—————
414 ans = ''
415 READ, ans, prompt = 'Vil du se histogram? (Y/n) '
416 if ans eq 'Y' then begin

```

```

417 histo=histogram(dp_matr(gtv_pet_ind),locations=1, binsize=1, min=d_low, max
    =round(d_high))
418 histo_inv=histogram(dp_inv(gtv_pet_ind),locations=l2, binsize=1, min=d_low+
    d_base-round(d_high), max=d_base)
419 ;window, 1
420 cd, figur_lokasjon
421 hist1 = plot(1, histo, /stairstep, title = 'Preskribert dose til tumor',
    xtitle = 'Dose [Gy]', ytitle = '# voxler')
422 hist2 = plot(l2, histo_inv, /stairstep, title = 'Invers av prekskribert
    dose til tumor', xtitle = 'Dose [Gy]', ytitle = '# voxler')
423 hist1.Save, "presc_hist.png", BORDER=10, $
424     RESOLUTION=300, /TRANSPARENT
425 hist2.Save, "inv_hist.png", BORDER=10, $
426     RESOLUTION=300, /TRANSPARENT
427 window, 1
428 plot, 1, histo, psym=10
429 window, 2
430 plot, l2, histo_inv, psym=10
431 endif
432 ;-----
433
434 ;formaterer til uint med ny dose grid scaling
435 new_dose_skal=max(dp_inv)/65500.
436 new_dose_skal_o=max(dp_matr)/65500.
437 if pixel_rep eq 0 then begin
438     dp_inv=uint(dp_inv/new_dose_skal)
439     dp_matr=uint(dp_matr/new_dose_skal_o)
440 endif
441
442
443
444
445
446 ;Illustrering av konturer p PETbilde, kan sl yfes.
447 ans = 'Y'
448 READ, ans, prompt = 'Vil du se konturplot snitt for snitt? (Y/n) '
449 if ans eq 'Y' then begin
450
451
452     n = 0
453
454     for i=0, numbers-1 do begin
455         if n_elements(*punkt_indeks_mat[i]) ge n then begin
456             n = n_elements(*punkt_indeks_mat[i])
457         endif
458     endfor
459
460
461
462     punkt_indeks_min = 10000
463     punkt_indeks_maks = -punkt_indeks_min
464
465     for i=0, n_elements(store_matrix[*, 10])-1 do begin
466         if min(*store_matrix[i, 10]) le punkt_indeks_min then begin
467             punkt_indeks_min = min(*store_matrix[i, 10])
468         endif
469
470         if max(*store_matrix[i, 10]) ge punkt_indeks_maks then begin
471             punkt_indeks_maks = max(*store_matrix[i, 10])

```

```

472     endif
473   endfor
474
475   new_punkt_indeks_sort = make_array(2*numbers+1, (punkt_indeks_maks-
    punkt_indeks_min+z_skal_p)/z_skal_p, /INT) ;+z_skal_p pga 0 m telles
    med
476
477   for i=0, n_elements(new_punkt_indeks_sort[0,*])-1 do begin
478     new_punkt_indeks_sort[0,i] = punkt_indeks_min
479     punkt_indeks_min = punkt_indeks_min + 2
480   endfor
481
482   for i=0, numbers - 1 do begin
483     for j=0, n_elements(*store_matrix[i, 10])-1 do begin
484       for k=0, n_elements(new_punkt_indeks_sort[0,*])-1 do begin
485         if (*store_matrix[i, 10])[j] eq new_punkt_indeks_sort[0,k] then begin
486           new_punkt_indeks_sort[i+1, k] = 1
487           new_punkt_indeks_sort[i+numbers+1, k] = (*(store_matrix[i, 9]))[j]
488         endif
489       endfor
490     endfor
491   endfor
492
493
494   counter = 0
495
496   zoomf=3
497   window, 0,xsize=xdim*zoomf, ysize=ydim*zoomf
498   for i=0,n_elements(new_punkt_indeks_sort[0, *])-1 do begin
499     switcher = 1
500     ;print, 'i = ', i
501
502     for j=0, n_elements(new_punkt_indeks_sort[1:numbers, 0])-1 do begin
503       ;print, 'j = ', j
504       ;print, 'Supposed to be 0 or 1 = ',(new_punkt_indeks_sort[1:3, i])[j]
505       if (new_punkt_indeks_sort[1:numbers, i])[j] eq 1 then begin
506         cont_sort = *(cont_sort_mat[j])
507         ;punkt_indeks_sort = *(store_matrix[j, 9])
508
509         snitt = ((cont_sort(2, new_punkt_indeks_sort[j+numb+1, i]) - zpos)/
    z_skal)
510         if switcher eq 1 then begin
511           counter += 1
512           tvscl, rebin(reform(pet_matrix(*,*, snitt)), zoomf*xdim, zoomf*ydim)
513           switcher = 0
514         endif
515         if i ne n_elements((new_punkt_indeks_sort[j+numb+1, *])-1) then plots
    , zoomf*cont_sort(0,(new_punkt_indeks_sort[j+numb+1, *])(i):(
    new_punkt_indeks_sort[j+numb+1, *])(i+1)-1), zoomf*cont_sort(1,(
    new_punkt_indeks_sort[j+numb+1, *])(i):(new_punkt_indeks_sort[j+numb+1,
    *])(i+1)-1), /device
516         if i eq n_elements((new_punkt_indeks_sort[j+numb+1, *])-1) then plots
    , zoomf*cont_sort(0,(new_punkt_indeks_sort[j+numb+1, *])(i):n_elements(
    cont_sort(2,*)-1), zoomf*cont_sort(1,(new_punkt_indeks_sort[j+numb+1,
    *])(i):n_elements(cont_sort(2,*)-1), /device
517       endif
518     endfor
519     wait, .5
520   endfor

```

```

521 endif
522
523 end ;Kommenter ut end her og kommenter inn linjene nedenfor for
      eksportere til DICOM. OBS! Dicom lisens trengs.
524 ;for i=0, n_elements(store_matrix[* , 10])-1 do begin
525 ;   for j=i+1, n_elements(store_matrix[* , 10])-1 do begin
526 ;     for k=0, n_elements(*store_matrix[i , 10])-1 do begin
527 ;       for l=0, n_elements(*store_matrix[j , 10])-1 do begin
528 ;         if (*store_matrix[i , 10])[k] eq (*store_matrix[j , 10])[l] then begin
529 ;           ;print , i , j , k , l , (*store_matrix[i , 10])[k] , (*store_matrix[j ,
      10])[l]
530 ;         endif
531 ;       endfor
532 ;     endfor
533 ;   endfor
534 ;endfor
535
536 ;stop
537 ;
538 ;cd , lokasjon
539 ;print , lokasjon
540 ;inv_name=strcompress(round(d_low*100))+ '_' +strcompress(round(d_high*100))+
      strcompress(round(d_mean*100))+ 'Gy_inv_base'+strcompress(round(set_base))
      + '.dcm'
541 ;presc_name=strcompress(round(d_low*100))+ '_' +strcompress(round(d_high*100))+
      strcompress(round(d_mean*100))+ 'Gy_presc_base'+strcompress(round(set_base
      ))+ '.dcm'
542 ;s_file=DIALOG_PICKFILE(PATH=FILEPATH(' ', root_dir=lokasjon),TITLE='Select
      original RT-Dose file to be cloned ', FILTER='rd*', GET_PATH=path)
543 ;
544 ;dose_exp =getenv(s_file)+path+inv_name
545 ;dose_exp_o =getenv(s_file)+path+presc_name
546 ;dose_inv_obj = OBJ_NEW('IDLffDicomEx',dose_exp , CLONE=s_file)
547 ;dose_o_obj = OBJ_NEW('IDLffDicomEx',dose_exp_o , CLONE=s_file)
548 ;
549 ;if bits_alloc ne 16 then begin;problemer med 32 bit i pixel matrix for
      eclipse filer , endrer til 16 bit
550 ;   dose_inv_obj->setvalue , '0028,0100' , 'US' , uint(16)
551 ;   dose_o_obj->setvalue , '0028,0100' , 'US' , uint(16)
552 ;   dose_inv_obj->setvalue , '0028,0101' , 'US' , uint(16)
553 ;   dose_o_obj->setvalue , '0028,0101' , 'US' , uint(16)
554 ;   dose_inv_obj->setvalue , '0028,0102' , 'US' , uint(15)
555 ;   dose_o_obj->setvalue , '0028,0102' , 'US' , uint(15)
556 ;   bits_alloc=uint(16)
557 ;endif
558 ;
559 ;;sette ny dose grid scaling og skriver inn pixeldata
560 ;dose_inv_obj ->SetValue , '3004,000E' , 'DS' ,string(new_dose_skal , format='(
      e11.4)')
561 ;dose_inv_obj ->SetPixelData , dp_inv , BITS_ALLOCATED=bits_alloc , COLUMNS=
      uint(xdim) , /order , ROWS=uint(ydim) , NUMBER_OF_FRAMES=uint(zdim) ,
      PIXEL_REPRESENTATION=piksel_rep , SAMPLES_PER_PIXEL=sample_per
562 ;dose_o_obj ->SetValue , '3004,000E' , 'DS' ,string(new_dose_skal_o , format='(
      e11.4)')
563 ;dose_o_obj ->SetPixelData , dp_matr , BITS_ALLOCATED=bits_alloc , COLUMNS=
      uint(xdim) , /order , ROWS=uint(ydim) , NUMBER_OF_FRAMES=uint(zdim) ,
      PIXEL_REPRESENTATION=piksel_rep , SAMPLES_PER_PIXEL=sample_per
564 ;dose_inv_obj ->commit
565 ;dose_o_obj ->commit

```

```

566 ;OBJ_DESTROY, dose_inv_obj
567 ;OBJ_DESTROY, dose_o_obj
568 ;print, 'done'
569 ;
570 ;end

```

The script `read_dat.pro` reads the dose .dat files from RayStation

```

1 function read_dat, filename
2   openr, lun, filename, /GET_LUN
3
4   nr_voxels = make_array(3, 1, /ulong, value = 0)
5   voxel_size = make_array(3, 1, /double, value = 0)
6   corner = make_array(3, 1, /double, value = 0)
7   fractions = ulong(0)
8   total_voxels = ulong(0)
9
10
11  readu, lun, nr_voxels
12  readu, lun, voxel_size
13  readu, lun, corner
14  readu, lun, fractions
15  readu, lun, total_voxels
16
17  d = make_array(nr_voxels[0], nr_voxels[1], nr_voxels[2], /double, value=0)
18
19  readu, lun, d
20
21  free_lun, lun
22
23  store_matrix = ptrarr(6)
24
25  store_matrix[0] = ptr_new(d)
26  store_matrix[1] = ptr_new(nr_voxels)
27  store_matrix[2] = ptr_new(voxel_size)
28  store_matrix[3] = ptr_new(corner)
29  store_matrix[4] = ptr_new(fractions)
30  store_matrix[5] = ptr_new(total_voxels)
31
32  return, store_matrix
33 end

```

The script `dat_treater2.pro` uses the dose files to find among others, TCP and QF.

```

1 dat_lokasjon = 'C:\Users\Eirik\Dropbox\Universitet\Master\dat_data\'
2 ind_lokasjon = 'C:\Users\Eirik\Dropbox\Universitet\Master\IndData\'
3 ;P_list = [strmid(patient, 0, 1) + strmid(patient, 8, 1)]
4 patients = ['Patient 1', 'Patient 2', 'Patient 3', 'Patient 4', 'Patient 5',
5            'Patient 6', 'Patient 7', 'Patient 8', 'Patient 9', 'Patient 11']
6 P_list = ['p1', 'P2', 'P3', 'P4', 'P5', 'P6', 'P7', 'P8', 'P9', 'P11'] ;P7 og
7            P9 gir problemer i read_dat.pro
8
9 info_matrix = ptrarr(n_elements(P_list))

```

```

9 for i=0, n_elements(patients) - 1 do begin
10 ;a = read_dat_ind((ind_lokasjon + "ROI_indices_patient_Test~
    P9_Doseplan_rob_p_BeamSet_inv.dat"))
11
12
13 patient = patients[i]
14 print, patient, P_list[i]
15
16 ;retrieving data from IDL
17 gtv_pet_ind = read_gtv_pet_ind((dat_lokasjon + patient + '_gtv_pet_ind.dat'
    ))
18 doseIDL_infomatrix = read_dat((dat_lokasjon + patient + '_dp_matr.dat'))
19 pet_infomatrix = read_dat((dat_lokasjon + patient + '_pet_matrix.dat'))
20
21 ;Using the read_dat function will give you a list of pointers
22 ;The index are as follows:
23 ;[0] = Dose matrix
24 ;[1] = Number of voxels along each axis [#x, #y, #z]
25 ;[2] = Voxel size array [dx, dy, dz]
26 ;[3] = Upper left corner array [x, y, z]
27 ;[4] = Number of fractions
28 ;[5] = Total number of voxels
29
30 ;retrieving data from Raystation Research
31 if patient eq 'Patient 1' then begin
32     RSlokasjon = 'C:
    \Users\Eirik\Dropbox\Universitet\Master\DoseData\fraction_dose_patient_test_hn_dpbn_p1
    ^p1_Doseplan_'
33     presc_doseRS_infomatrix = read_dat((RSlokasjon + 'presc_BeamSet_presc.dat
    '))
34     inv_doseRS_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_inv.dat'))
35
36     p_infomatrix = read_dat((RSlokasjon + 'robust_pr_BeamSet_robust_pr.dat'))
37     p_dpbn_infomatrix = read_dat((RSlokasjon + '
    robust_pr_BeamSet_robust_dpbn_pr.dat'))
38     ph_infomatrix = read_dat((RSlokasjon + 'robust_ph_BeamSet_robust_ph.dat')
    )
39     ph_dpbn_infomatrix = read_dat((RSlokasjon + '
    robust_ph_BeamSet_robust_dpbn_ph.dat'))
40 endif
41
42 if patient ne 'Patient 1' then begin
43     RSlokasjon = 'C:
    \Users\Eirik\Dropbox\Universitet\Master\DoseData\fraction_dose_patient_Test
    ^' + P_list[i] + '_Doseplan_'
44
45
46     inv_doseRS_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_inv.dat'))
47
48     p_infomatrix = read_dat((RSlokasjon + 'rob_p_BeamSet_rob_p.dat'))
49     p_dpbn_infomatrix = read_dat((RSlokasjon + 'rob_p_BeamSet_rob_p_dpbn.dat'
    ))
50     ph_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_rob_ph.dat'))
51     ph_dpbn_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_rob_ph_dpbn.
    dat'))
52 endif
53
54
55 p_dose = *(p_infomatrix[0])/100. * 34. ;M deles p 100 for g fra

```

```

cGy til Gy
56 p_dpbn_dose = *(p_dpbn_infomatrix[0])/100. * 34. ;Ganges med 34
   fraksjoner
57 ph_dose = *(ph_infomatrix[0])/100. * 34.
58 ph_dpbn_dose = *(ph_dpbn_infomatrix[0])/100. * 34.
59 RSpresc = *(presc_doseRS_infomatrix[0])/100. ;NB! Bare n fraksjon p
   preskribert dose
60 RSinv = *(inv_doseRS_infomatrix[0])/100.
61 pet_matrix = *(pet_infomatrix[0]) ;NB! Although I use read_dat here as
   well, don't use any information beyond "Dose", which in this case is the
   pet matrix, and "number of voxels" as these are just fillers
62 IDLpresc = *(doseIDL_infomatrix[0])
63
64 gtv_RS_ind = where(RSinv lt 50.0)
65 D.mean = [mean(ph_dose[gtv_RS_ind]), mean(ph_dpbn_dose[gtv_RS_ind]), mean(
   p_dose[gtv_RS_ind]), mean(p_dpbn_dose[gtv_RS_ind])]
66
67
68 gtv_RS_ind1 = read_dat_ind(patient, 'GTV68')
69
70
71 ; print, patient, '&', mean(RSpresc[gtv_pet_ind]), '\pm ', stddev(RSpresc[
   gtv_pet_ind]), '&', mean(RSpresc[gtv_RS_ind]), '\pm ', stddev(RSpresc[
   gtv_RS_ind]), '&', mean(RSpresc[gtv_RS_ind1]), '\pm ', stddev(RSpresc[
   gtv_RS_ind1])
72 ;for i = 0, n_elements(gtv_RS_ind) do begin
73 ; print, index_inv[i], gtv_RS_ind[i], inv_dose[index_inv[i]], inv_dose[
   gtv_RS_ind[i]]
74 ;endfor
75 a = dose_difference(IDLpresc, ph_dose, ph_dpbn_dose, p_dose, p_dpbn_dose,
   RSpresc, gtv_RS_ind, *(p_infomatrix[2]))
76 ; ;b = DoseofPet(pet_matrix, IDLpresc, gtv_RS_ind)
77 c = tcpmodel(ph_dose, ph_dpbn_dose, p_dose, p_dpbn_dose, RSpresc,
   pet_matrix, *(p_infomatrix[2]), *(p_infomatrix[4]), gtv_RS_ind,
   Constant = 22974.8)
78 ; ;c = convolver(pet_matrix, gtv_pet_ind, IDLpresc, 2)
79 d = wcsv((*a[0]), D.mean, c, patient)
80 ;e = yxtcp(pet_matrix, RSpresc, p_dose, *(p_infomatrix[2])[0], *(
   p_infomatrix[2])[1], *(p_infomatrix[2])[2])
81 ;f = translate_volume(p_dose, p_dpbn_dose, ph_dose, ph_dpbn_dose, *(
   p_infomatrix[2]), RSpresc, gtv_RS_ind, IDLpresc, patient, pet_matrix)
82 ;g = shrink_volume2(p_dpbn_dose, ph_dpbn_dose, *(p_infomatrix[2]),
   RSpresc, gtv_RS_ind, IDLpresc, patient, pet_matrix, ph_dose, p_dose)
83 h = ntcp(p_dose, p_dpbn_dose, ph_dose, ph_dpbn_dose, 'Parotid_L', patient)
84 stop
85
86 ;
87 ;
88 ; QFnTCP = translate_volume2(p_dose, p_dpbn_dose, ph_dose, ph_dpbn_dose, *(
   p_infomatrix[2]), RSpresc, gtv_RS_ind, IDLpresc, patient, pet_matrix)
89 ;
90 ; info_matrix[i] = ptr_new(QFnTCP)
91 ;
92 ; avg_mov = make_array((size(QFnTCP))[2])
93 ; shrink = avg_mov
94 ; phQF = avg_mov
95 ; pQF = avg_mov
96 ; phTCP = avg_mov
97 ; pTCP = avg_mov

```

```

98 ;
99 ; for j = 0, (size(QFnTCP))[2] - 1 do begin
100 ;     avg_mov[j] = *(QFnTCP[0, *])[j]
101 ;     shrink[j]  = *(QFnTCP[1, *])[j]
102 ;     phQF[j]   = *(QFnTCP[2, *])[j]
103 ;     pQF[j]    = *(QFnTCP[3, *])[j]
104 ;     phTCP[j]  = *(QFnTCP[4, *])[j]
105 ;     pTCP[j]   = *(QFnTCP[5, *])[j]
106 ; endfor
107 ;
108 ; avg_mov_elements = sort_one(avg_mov)
109 ; shrink_elements = sort_one(shrink)
110 ;
111 ; test = make_array(n_elements(avg_mov_elements), n_elements(shrink_elements))
112 ;
113 ; for j = 0, n_elements(shrink_elements) - 1 do begin
114 ;     for k = 0, n_elements(avg_mov_elements) - 1 do begin
115 ;         test[k, j] = mean(pQF[intersect(where(avg_mov eq avg_mov_elements[k]),
116 ;         where(shrink eq shrink_elements[j]))])
117 ;         print, avg_mov_elements[k], shrink_elements[j], test[k, j]
118 ;     endfor
119 ; endfor
120 ;
121 ; ptr_free, h
122 ptr_free, p_infomatrix, p_dpbn_infomatrix, ph_infomatrix,
123     ph_dpbn_infomatrix, presc_doseRS_infomatrix, pet_infomatrix,
124     doseIDL_infomatrix, inv_doseRS_infomatrix
125 endfor
end

```

The script `translate_volume2.pro` translates the tumor and initiates shrinking.

```

1 function translate_volume2, p_dose, p_dpbn_dose, ph_dose, ph_dpbn_dose,
2     voxelsize, RSpresc, gtv_pet_ind, idl_presc, patient, petmatrix
3
4     gtv_RS_ind = gtv_pet_ind
5
6     x = [-0.1, 0, 0.1, 0.2, 0.3] ;cm
7     ;x = [-0.3, 0, 0.3] ;cm
8     y = x
9     z = x
10
11     z_index = 0;round(n_elements(x)/2. + 0.4)
12 ; p_QF = fltarr(n_elements(x), n_elements(y), n_elements(z))
13 ; p_dpbn_QF = p_QF
14 ; ph_QF = p_QF
15 ; ph_dpbn_QF = p_QF
16 ; p_dpbn_TCP = p_QF
17 ; ph_dpbn_TCP = p_QF
18
19 p_ones = p_dpbn_dose
20 p_ones[gtv_pet_ind] = 1
21 p_ones[where(p_ones ne 1)] = 0

```

```

22
23 shrinking = [2];[0, 0.5, 1, 1.5, 2, 2.5]
24
25 store_matrix = ptrarr(6, n_elements(x)^3*n_elements(shrinking))
26
27 TIC, /PROFILER
28 prevPerc = 0
29 timetaker = 0
30 ;CGprogressbar = Obj_new('progressbar', /Start, Percent = 0)
31 counter = 0
32 for i = 0, n_elements(x) - 1 do begin
33     for j = 0, n_elements(x) - 1 do begin
34         for k = 0, n_elements(x) - 1 do begin
35             ;OBS! CHECK out the translate difference. Why does one have - and the
36             other not?
37             ph_dpbn_trans = transform_volume(ph_dpbn_dose, translate = [x[i]/
38             voxelsize[0], y[j]/ voxelsize[1], z[k]/ voxelsize[2]])
39
40             ;Satt inn denne istedenfor den nedenfor 11.04
41             p_dpbn_trans = transform_volume(p_dpbn_dose, translate = [(x[i]/
42             voxelsize[0]), (y[j]/ voxelsize[1]), (z[k]/ voxelsize[2])])
43
44             ;RSpresc_trans = transform_volume(RSpresc, translate = [(x[i]/
45             voxelsize[0]), (y[j]/ voxelsize[1]), (z[k]/ voxelsize[2])])
46             for l = 0, n_elements(shrinking) - 1 do begin
47                 clock = TIC()
48                 ;print, round(x[i]/ voxelsize[0]), round(y[j]/ voxelsize[1]), round(z
49                 [k]/ voxelsize[2])
50
51                 QFnTCP_dpbn_shrunk = shrink_volume2(p_dpbn_trans, ph_dpbn_trans,
52                 voxelsize, RSpresc, gtv_RS_ind, idl_presc, patient, petmatrix, ph_dose,
53                 p_dose, mmshrink = shrinking[l])
54
55                 store_matrix[0, counter] = ptr_new(abs(x[i] + y[j] + z[k])/3.)
56                 store_matrix[1, counter] = ptr_new(shrinking[l])
57                 store_matrix[2, counter] = ptr_new(*(QFnTCP_dpbn_shrunk[0])) ;
58                 ph_dpbn_QF
59                 store_matrix[3, counter] = ptr_new(*(QFnTCP_dpbn_shrunk[1])) ;
60                 p_dpbn_QF
61                 store_matrix[4, counter] = ptr_new(*(QFnTCP_dpbn_shrunk[2])) ;
62                 ph_dpbn_TCP
63                 store_matrix[5, counter] = ptr_new(*(QFnTCP_dpbn_shrunk[3])) ;
64                 p_dpbn_TCP
65
66                 ptr_free, QFnTCP_dpbn_shrunk
67                 counter = counter + 1
68
69                 time = TOC(clock)
70
71                 timetaker = (timetaker + time)
72                 averagetime = timetaker/float(counter)
73                 curPerc = float(counter)/(n_elements(x)^3*n_elements(shrinking))
74
75                 print, 'Estimated time left: ', time2string(((1.0 - curPerc) / (
76                 curPerc-prevPerc) * averagetime)), '. Mean dt = ', averagetime
77
78                 ;print, float(counter)/(n_elements(x)^3*n_elements(shrinking))

```

```

69     prevPerc = curPerc
70
71     ;           ;QFs = dose_difference_dpbn(idl_presc , ph_dpbn_trans , p_dpbn_trans
, RSpresc , gtv_pet_ind , voxelsize )
72     ;           ;TCPs = tcpmodel(ph_dose , ph_dpbn_trans , p_dose , p_dpbn_trans ,
RSpresc , petmatrix , voxelsize , 34.0 , gtv_pet_ind , Constant = 22974.8)
73     ;
74     ;           ph_dpbn_QF[i , j , k] = (*(QFs[0]))[1]
75     ;           p_dpbn_QF[i , j , k] = (*(QFs[0]))[2]
76     ;           ph_dpbn_TCP[i , j , k] = *(TCPs[1])
77     ;           p_dpbn_TCP[i , j , k] = *(TCPs[3])
78     ;           endfor
79     ;           endfor
80     ;           endfor
81     ;           endfor
82     TOC
83     return , store_matrix
84 end

```

The script `transform_volume.pro` was used by `translate_volume2.pro` to translate .

```

1 ;+
2 ; NAME:
3 ;     TRANSFORM_VOLUME
4 ;
5 ; PURPOSE:
6 ;
7 ;     The purpose of this program is to transform (e.g., rotate ,
8 ;     scale , and translate) a 3D array or volume.
9 ;
10 ; AUTHOR:
11 ;
12 ;     Martin Downing ,
13 ;     Clinical Research Physicist ,
14 ;     Grampian Orthopaedic RSA Research Centre ,
15 ;     Woodend Hospital , Aberdeen , AB15 6LS.
16 ;     Pphone: 01224 556055 / 07903901612
17 ;     Fa: 01224 556662
18 ;     E-mail: m.downing@abdn.ac.uk
19 ;
20 ; CATEGORY:
21 ;
22 ;     Mathematics , graphics .
23 ;
24 ; CALLING SEQUENCE:
25 ;
26 ;     result = TRANSFORM_VOLUME( volume )
27 ;
28 ; INPUTS:
29 ;
30 ;     volume:     The 3D array or volume to be transformed.
31 ;
32 ; OPTIONAL KEYWORDS:
33 ;
34 ;     BUFFER_SIZE: To reduce memory overhead the routine processes the job
in chunks , the number

```

```

35 ;         of elements of which can be set using the BUFFER_SIZE keyword, set
        this keyword to
36 ;         0 to force the whole array to be processed at one time. The default
        value is 128.
37 ;
38 ;     MISSING: The value to return for transformed values outside the bounds
        of
39 ;         the volume. (Passed to the INTERPOLATE function.) Default is 0.
40 ;
41 ;     T3DMAT: The homogeneous transformation matrix. If this keyword is not
        present,
42 ;         the following keywords can be used to create a homogeneous
        transformation matrix:
43 ;
44 ;     ROTATION – The rotation vector [rx,ry,rz]. The order of rotation is
        ZYX.
45 ;     TRANSLATE – The translation vector [tx,ty,tz].
46 ;     SCALE – The scale vector [sx,sy,sz].
47 ;     CENTRE_ROTATION – The centre of rotation [cx,cy,cz].
48 ;
49 ; OUTPUTS:
50 ;
51 ;     result:     The transformed array or volume.
52 ;
53 ; COMMON BLOCKS:
54 ;
55 ;     None.
56 ;
57 ; DEPENDENCIES:
58 ;
59 ;     The program uses the library INTERPLOLATE routine, which currently (
        IDL 5.4)
60 ;     uses linear interpolation. Note that the operation is performed in
        chunks,
61 ;     each of which is independant of the result of the others, so the
        operation
62 ;     could easiliy be parallelised.
63 ;
64 ; MODIFICATION HISTORY:
65 ;
66 ;     Written by: Martin Downing, 16 September 2001.
67 ;     Added MISSING keyword. Removed INPLACE keyword. 25 Nov 2001. MD
68 ;—
69 ;
        *****
70 ; Copyright (c) 2008, by Fanning Software Consulting, Inc.
        ;
71 ; All rights reserved.
        ;
72 ;
        ;
73 ; Redistribution and use in source and binary forms, with or without
        ;
74 ; modification, are permitted provided that the following conditions are met
        :
75 ;
        ;
76 ; * Redistributions of source code must retain the above copyright

```

```

77 ;         ;
77 ;         notice , this list of conditions and the following disclaimer .
78 ;         ;
78 ;         * Redistributions in binary form must reproduce the above copyright
79 ;         ;
79 ;         notice , this list of conditions and the following disclaimer in the
80 ;         ;
80 ;         documentation and/or other materials provided with the distribution .
81 ;         ;
81 ;         * Neither the name of Fanning Software Consulting , Inc . nor the names
of its
82 ;         ;
82 ;         contributors may be used to endorse or promote products derived from
this
83 ;         ;
83 ;         software without specific prior written permission .
84 ;         ;
85 ;         ;
85 ;         THIS SOFTWARE IS PROVIDED BY FANNING SOFTWARE CONSULTING, INC. 'AS IS'
AND ANY
86 ;         ;
86 ;         EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES
87 ;         ;
87 ;         OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT
88 ;         ;
88 ;         SHALL FANNING SOFTWARE CONSULTING, INC. BE LIABLE FOR ANY DIRECT, INDIRECT
,
89 ;         ;
89 ;         INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED
90 ;         ;
90 ;         TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS;
91 ;         ;
91 ;         LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND
92 ;         ;
92 ;         ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
;
93 ;         ;
93 ;         (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS
94 ;         ;
94 ;         SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
95 ;         ;
*****
96 FUNCTION Transform_Volume , volume , Rotation=rotation , $
97     Scale=scale , Translate=translate , Centre_Rotation=centre_rotation , $
98     T3Dmat=t3dmat , Buffer_Size=buffer_size , Missing=missing
99
100 ; Error handling.
101
102 Catch , theError
103 IF theError NE 0 THEN BEGIN
104     Catch , /Cancel
105     ok = Dialog_Message (!Error.State.Msg)
106     RETURN , -1
107 ENDIF
108
109 ; Find the dimensions of the volume.
110
111 ;!P.T3d = 1
112 s = Size(volume)
113 sx=s[1] & sy=s[2] & sz=s[3]
114 st = sx*sy*sz

```

```

115
116 vol_t = volume
117 IF N_Elements(missing) THEN missing = 0
118
119 ; Create a transform matrix , if one is not provided.3
120
121 IF N_Elements(t3dmat) EQ 0 THEN begin
122
123     IF N_Elements(rotation) EQ 0 THEN rotation =[0,0,0]
124     IF N_Elements(centre_rotation) EQ 0 THEN centre_rotation =[(sx-1)/2.0 ,(sy
125     -1)/2.0 ,(sz-1)/2.0]
126     IF N_Elements(translate) EQ 0 THEN translate =[0,0,0]
127     IF N_Elements(scale) EQ 0 THEN scale =[1,1,1]
128
129     T3D, /Reset , Translate = -centre_rotation
130     T3D, Rotate=rotation
131     T3D, Translate= centre_rotation + translate , Scale=scale
132     t3dmat = !P.T
133
134 ENDIF
135
136
137 ; Check buffer size . The size 128 is optimim on my system , You may
138 ; want to try other values .
139
140 IF N_Elements(buffer_size) EQ 0 THEN buffer_size = 128
141 IF buffer_size LE 0 THEN buffer_size = st
142
143 ; Perform the transformations .
144
145 FOR j=0L,(st-1),buffer_size DO BEGIN
146
147     ; Account for possible odd last chunk .
148
149     bufsize = buffer_size < (st-j)
150
151     ; Generate volume coordinates by interpolating temporary array of
152     volume indices .
153
154     i = j + Lindgen(bufsize)
155     coords = [ [(i MOD sx)],[((i / sx) MOD (sy))], [(i / (sx * sy))], [
156     Replicate(1b, bufsize)]]
157     coords = Temporary(coords) # t3dmat
158     vol_t[j:j+bufsize-1] = Interpolate(volume , coords[* ,0] , coords[* ,1] ,
159     coords[* ,2] , Missing=missing)
160
161 ENDFOR
162
163 ; Return the transformed volume .
164
165 RETURN, vol_t
166 END

```

The script shrink_volume2.pro shrinks the tumor

```

1 function shrink_volume2, p_dpbn_dose, ph_dpbn_dose, voxelsize, RSpresc,
   gtv_RS_ind, idl_presc, patient, petmatrix, ph_dose, p_dose, mmshrink =
   mmshrink
2
3 if n_elements(mmshrink) eq 0 then mmshrink = 1
4
5 gtv_pet_ind = gtv_RS_ind
6
7 orig_RS_zeros = make_array(value = 0, size = Size(RSpresc))
8 orig_ones_zeros = orig_RS_zeros
9 orig_p_zeros = orig_RS_zeros
10 orig_ph_zeros = orig_RS_zeros
11
12 RS_ones = RSpresc
13 RS_ones[gtv_RS_ind] = 1
14 RS_ones[where(RS_ones ne 1)] = 0
15
16
17 RS_mc = make_array(3, n_elements(gtv_RS_ind))
18 for i = 0, n_elements(gtv_RS_ind) - 1 do begin
19     RS_mc[0:2, i] = array_indices(RSpresc, gtv_RS_ind[i])
20 endfor
21
22 orig_RS_mc = round(total(RS_mc, 2)/n_elements(gtv_RS_ind))
23 orig_RS_size = Size(RSpresc)
24
25 max_distance = 0
26 max_index = -1
27 for i = 0, n_elements(RS_mc[0, *, *]) - 1 do begin
28     if sqrt((RS_mc[0, i] - orig_RS_mc[0])^2 + ((RS_mc[1, i] - orig_RS_mc[1])
   ^2) + (RS_mc[2, i] - orig_RS_mc[2])^2) gt max_distance then begin
29         max_distance = sqrt((RS_mc[0, i] - orig_RS_mc[0])^2 + ((RS_mc[1, i] -
   orig_RS_mc[1])^2) + (RS_mc[2, i] - orig_RS_mc[2])^2)
30         max_index = i
31     endif
32 endfor
33
34 radius = ceil(max_distance)
35
36
37 scale_factor = (radius*voxelsize[0]*10 - mmshrink)^3/(radius*voxelsize
   [0]*10)^3
38
39
40
41 print, scale_factor;, orig_RS_mc
42
43 shrunk_RS_dose = congrid(RSpresc, scale_factor*orig_RS_size[1],
   scale_factor*orig_RS_size[2], scale_factor*orig_RS_size[3])
44 shrunk_RS_ones = congrid(RS_ones, scale_factor*orig_RS_size[1],
   scale_factor*orig_RS_size[2], scale_factor*orig_RS_size[3])
45
46
47 ;Find center of Ones-matrix
48 ones_RS_mc = make_array(3, n_elements(where(shrunk_RS_ones gt 0)))
49 for i = 0, n_elements(where(shrunk_RS_ones gt 0)) - 1 do begin
50     ones_RS_mc[0:2, i] = array_indices(shrunk_RS_ones, (where(shrunk_RS_ones
   gt 0))[i])
51 endfor

```

```

52
53 ones_RS_mc = round(total(ones_RS_mc, 2)/n_elements(where(shrunk_RS_ones gt
    0)))
54
55
56
57 orig_ones_zeros [0:(size(shrunk_RS_dose))[1] - 1, 0:(size(shrunk_RS_dose))
    [2] - 1, 0:(size(shrunk_RS_dose))[3] - 1] = shrunk_RS_ones
58 new_ones = shift(orig_ones_zeros, [abs(orig_RS_mc[0] - ones_RS_mc[0]), abs(
    orig_RS_mc[1] - ones_RS_mc[1]), abs(orig_RS_mc[2] - ones_RS_mc[2])])
59
60 orig_RS_zeros [0:(size(shrunk_RS_dose))[1] - 1, 0:(size(shrunk_RS_dose))[2]
    - 1, 0:(size(shrunk_RS_dose))[3] - 1] = shrunk_RS_dose
61 new_RS = shift(orig_RS_zeros, [abs(orig_RS_mc[0] - ones_RS_mc[0]), abs(
    orig_RS_mc[1] - ones_RS_mc[1]), abs(orig_RS_mc[2] - ones_RS_mc[2])])
62
63
64 QFs = dose_difference_dpbn(idl_presc, ph_dpbn_dose, p_dpbn_dose, new_RS,
    where(new_ones eq 1), voxelsize)
65
66 TCPs = tcpmodel(ph_dose, ph_dpbn_dose, p_dose, p_dpbn_dose, new_RS,
    petmatrix, voxelsize, 34.0, where(new_ones eq 1), Constant = 22974.8)
67
68
69 p_dpbn_QF = (*(QFs[0]))[2]
70 ph_dpbn_QF = (*(QFs[0]))[1]
71
72
73
74 p_dpbn_TCP = *(TCPs[3])
75 ph_dpbn_TCP = *(TCPs[1])
76
77
78
79 store_matrix = ptrarr(4)
80 store_matrix[0] = ptr_new (*(QFs[0])[1]) ; ph_dpbn_QF
81 store_matrix[1] = ptr_new (*(QFs[0])[2]) ; p_dpbn_QF
82 store_matrix[2] = ptr_new (*(TCPs[1])) ; ph_dpbn_TCP
83 store_matrix[3] = ptr_new (*(TCPs[3])) ; p_dpbn_TCP
84
85 return, store_matrix
86 end

```

The script `tcpmodel.pro` finds the TCP

```

1 function tcpmodel, ph, ph_dpbn, p, p_dpbn, presc, petmatrix, voxelsize,
    fractions, gtv_pet_ind, Constant = constant
2
3 if n_elements(Constant) eq 0 then constant = 1
4
5 pet_matrix = petmatrix * constant
6 alpha = (-5)*alog(0.48)/(12.0)
7 voxelvol = voxelsize[0]*voxelsize[1]*voxelsize[2]
8
9 Array_68 = presc
10 Array_68[where(Array_68 ne 68)] = 68.0
11

```

```

12 TCPi_presc = exp( - pet_matrix[gtv_pet_ind]* voxelvol*exp(-(alpha*(1.0 + (
    presc[gtv_pet_ind]/fractions)/10.0))*(presc[gtv_pet_ind])))
13 TCPi_68 = exp( - pet_matrix[gtv_pet_ind]* voxelvol*exp(-(alpha*(1.0 + (
    Array_68[gtv_pet_ind]/fractions)/10.0))*(Array_68[gtv_pet_ind])))
14 ;Skal den preskriberte dosen deles p fractions n r den ikke har blitt
    gitt i fractions? Ja, tror det ettersom den skal v re helt lik en gitt
    dose.
15
16
17
18 TCPi_ph = exp(- pet_matrix[gtv_pet_ind] * voxelvol * exp(-(alpha * (1.0 + (
    ph[gtv_pet_ind]/fractions)/10.0)) * (ph[gtv_pet_ind])))
19 TCPi_ph_dpbn = exp(- pet_matrix[gtv_pet_ind]* voxelvol*exp(-(alpha*(1.0 + (
    ph_dpbn[gtv_pet_ind]/fractions)/10.0))*(ph_dpbn[gtv_pet_ind])))
20 TCPi_p = exp(- pet_matrix[gtv_pet_ind]* voxelvol*exp(-(alpha*(1.0 + (p[
    gtv_pet_ind]/fractions)/10.0))*(p[gtv_pet_ind])))
21 TCPi_p_dpbn = exp(- pet_matrix[gtv_pet_ind]* voxelvol*exp(-(alpha*(1.0 + (
    p_dpbn[gtv_pet_ind]/fractions)/10.0))*(p_dpbn[gtv_pet_ind])))
22
23 ; print , min(p_dpbn[gtv_pet_ind]) , mean(p_dpbn[gtv_pet_ind]) , max(p_dpbn[
    gtv_pet_ind])
24 ; print , min(TCPi_p_dpbn[gtv_pet_ind]) , mean(TCPi_p_dpbn[gtv_pet_ind]) , max(
    TCPi_p_dpbn[gtv_pet_ind])
25 ; print , mean(pet_matrix[gtv_pet_ind])
26
27 TCP_ph = product(TCPi_ph)
28 TCP_ph_dpbn = product(TCPi_ph_dpbn)
29 TCP_p = product(TCPi_p)
30 TCP_p_dpbn = product(TCPi_p_dpbn)
31
32 TCP_presc = product(TCPi_presc)
33 TCP_68 = product(TCPi_68)
34
35 store_matrix = ptrarr(6)
36
37 store_matrix[0] = ptr_new(TCP_ph)
38 store_matrix[1] = ptr_new(TCP_ph_dpbn)
39 store_matrix[2] = ptr_new(TCP_p)
40 store_matrix[3] = ptr_new(TCP_p_dpbn)
41 store_matrix[4] = ptr_new(TCP_presc)
42 store_matrix[5] = ptr_new(TCP_68)
43
44 return , store_matrix
45 end

```

The script `dose_difference.pro` finds the QF

```

1 function dose_difference , d , d_rob_ph , d_rob_ph_dpbn , d_rob_p , d_rob_p_dpbn ,
    dose_presc , gtv_pet_ind , voxel_size
2 N_T = n_elements(gtv_pet_ind)
3
4 Q_i = d[gtv_pet_ind]/dose_presc[gtv_pet_ind]
5 Q_i_rob_ph = d_rob_ph[gtv_pet_ind]/dose_presc[gtv_pet_ind]
6 Q_i_rob_ph_dpbn = d_rob_ph_dpbn[gtv_pet_ind]/dose_presc[gtv_pet_ind]
7 Q_i_rob_p = d_rob_p[gtv_pet_ind]/dose_presc[gtv_pet_ind]
8 Q_i_rob_p_dpbn = d_rob_p_dpbn[gtv_pet_ind]/dose_presc[gtv_pet_ind]
9

```

```

10 QF = 1.0/N.T * total(abs(Q_i - 1))
11 QF_rob_ph = 1.0/N.T * total(abs(Q_i_rob_ph - 1))
12 QF_rob_ph_dpbn = 1.0/N.T * total(abs(Q_i_rob_ph_dpbn - 1))
13 QF_rob_p = 1.0/N.T * total(abs(Q_i_rob_p - 1))
14 QF_rob_p_dpbn = 1.0/N.T * total(abs(Q_i_rob_p_dpbn - 1))
15
16
17 store_matrix = ptrarr(1)
18 store_matrix[0] = ptr_new([QF, QF_rob_ph, QF_rob_ph_dpbn, QF_rob_p,
19     QF_rob_p_dpbn])
20 return , store_matrix
21 end

```

The script `ntcp.pro` finds the NTCP

```

1 function dose_difference , d, d_rob_ph, d_rob_ph_dpbn, d_rob_p, d_rob_p_dpbn,
2     dose_presc, gtv_pet_ind, voxel_size
3
4     N.T = n_elements(gtv_pet_ind)
5
6     Q_i = d[gtv_pet_ind]/dose_presc[gtv_pet_ind]
7     Q_i_rob_ph = d_rob_ph[gtv_pet_ind]/dose_presc[gtv_pet_ind]
8     Q_i_rob_ph_dpbn = d_rob_ph_dpbn[gtv_pet_ind]/dose_presc[gtv_pet_ind]
9     Q_i_rob_p = d_rob_p[gtv_pet_ind]/dose_presc[gtv_pet_ind]
10    Q_i_rob_p_dpbn = d_rob_p_dpbn[gtv_pet_ind]/dose_presc[gtv_pet_ind]
11
12    QF = 1.0/N.T * total(abs(Q_i - 1))
13    QF_rob_ph = 1.0/N.T * total(abs(Q_i_rob_ph - 1))
14    QF_rob_ph_dpbn = 1.0/N.T * total(abs(Q_i_rob_ph_dpbn - 1))
15    QF_rob_p = 1.0/N.T * total(abs(Q_i_rob_p - 1))
16    QF_rob_p_dpbn = 1.0/N.T * total(abs(Q_i_rob_p_dpbn - 1))
17
18    store_matrix = ptrarr(1)
19    store_matrix[0] = ptr_new([QF, QF_rob_ph, QF_rob_ph_dpbn, QF_rob_p,
20     QF_rob_p_dpbn])
21 return , store_matrix
22 end

```

The script `dvh.Eirik.pro` finds the NTCP

```

1 function DVH_Eirik , p_dose , p_dpbn_dose , ph_dose , ph_dpbn_dose , ROI , patient
2
3     ;Gjennomsnitt for alle organ, men ett plot per doseplan. S   alts   et 2x2
4     grid med plots for f.eks. Parotid 1 og 2, Medulla og GTV?
5
6     index = read_dat_ind(patient , ROI)
7     pD = p_dose[index]
8     pDPD = p_dpbn_dose[index]
9     phD = ph_dose[index]
10    phDPD = ph_dpbn_dose[index]
11
12    maximum = 100 ; Gy

```

```

12 minimum = 1
13
14 pDhist = histogram(pD, max = maximum, min = minimum, locations = xbinP)
15 pDPDhist = histogram(pDPD, max = maximum, min = minimum, locations =
16   xbinPDP)
17 phDhist = histogram(phD, max = maximum, min = minimum, locations = xbinPHD)
18 phDPDhist = histogram(phDPD, max = maximum, min = minimum, locations =
19   xbinPHDPD)
20
21 tot_vox = make_array(size = size(xbinP*4))
22 tot_voxpluss = make_array(size = size(tot_vox))
23 tot_voxminus = make_array(size = size(tot_vox))
24
25 for i = 0, n_elements(pDhist)-1 do begin
26   ;tot_vox[i] = (total(pDhist[i:-1]) + total(pDPDhist[i:-1]) + total(
27     phDhist[i:-1]) + total(phDPDhist[i:-1])) / float(total(pDhist) + total(
28     pDPDhist) + total(phDhist) + total(phDPDhist))
29   tot_vox[i] = mean([total(pDhist[i:-1]), total(pDPDhist[i:-1]), total(
30     phDhist[i:-1]), total(phDPDhist[i:-1])]) / float(mean([total(pDhist), total(
31     pDPDhist), total(phDhist), total(phDPDhist)]))
32   tot_voxpluss[i] = tot_vox[i] + stddev([total(pDhist[i:-1]), total(
33     pDPDhist[i:-1]), total(phDhist[i:-1]), total(phDPDhist[i:-1])]) / float(mean(
34     [total(pDhist), total(pDPDhist), total(phDhist), total(phDPDhist)]))
35   tot_voxminus[i] = tot_vox[i] - stddev([total(pDhist[i:-1]), total(
36     pDPDhist[i:-1]), total(phDhist[i:-1]), total(phDPDhist[i:-1])]) / float(mean(
37     [total(pDhist), total(pDPDhist), total(phDhist), total(phDPDhist)]))
38 endfor
39 stop
40 end
41
42 function DVH_Eirik2, ROIlst, patientlist
43   maximum = 100; Gy
44   minimum = 0.1
45
46
47   pD = ptrarr(n_elements(patientlist))
48   pDP = ptrarr(n_elements(patientlist))
49   phD = ptrarr(n_elements(patientlist))
50   phDP = ptrarr(n_elements(patientlist))
51
52   ipsi = ['L', 'R', 'L', 'L', 'L', 'R', 'L', 'R', 'R']
53   cont = ['R', 'L', 'R', 'R', 'R', 'L', 'R', 'L', 'L']
54
55
56   ROIindex = ptrarr(n_elements(patientlist))
57
58
59   for i = 0, n_elements(patientlist)-1 do begin
60     patient = patientlist[i]
61     print, patient
62
63     if patient eq 'Patient 1' then begin
64       P_list = 'p' + (strsplit(patient, /extract))[-1]
65       RSlokasjon = 'C:
66       \Users\Eirik\Dropbox\Universitet\Master\DoseData\fraction_dose_patient_test_hn_dpbn_p1
67       ^p1_Doseplan_'
68       inv_doseRS_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_inv.dat'
69       ))

```

```

58
59     p_infomatrix = read_dat((RSlokasjon + 'robust_pr_BeamSet_robust_pr.dat'
60 ))
61     p_dpbn_infomatrix = read_dat((RSlokasjon + '
robust_pr_BeamSet_robust_dpbn_pr.dat'))
62     ph_infomatrix = read_dat((RSlokasjon + 'robust_ph_BeamSet_robust_ph.dat
'))
63     ph_dpbn_infomatrix = read_dat((RSlokasjon + '
robust_ph_BeamSet_robust_dpbn_ph.dat'))
64
65     endif
66
67     if patient ne 'Patient 1' then begin
68         P_list = 'P' + (strsplit(patient, /extract))[-1]
69         RSlokasjon = 'C:
\Users\Eirik\Dropbox\Universitet\Master\DoseData\fraction_dose_patient_Test
^' + P_list + '_Doseplan_'
70         inv_doseRS_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_inv.dat'
))
71
72         p_infomatrix = read_dat((RSlokasjon + 'rob_p_BeamSet_rob_p.dat'))
73         p_dpbn_infomatrix = read_dat((RSlokasjon + 'rob_p_BeamSet_rob_p_dpbn.
dat'))
74         ph_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_rob_ph.dat'))
75         ph_dpbn_infomatrix = read_dat((RSlokasjon + 'rob_ph_BeamSet_rob_ph_dpbn
.dat'))
76         endif
77
78         pD[i] = ptr_new(*(p_infomatrix[0])/100. * 34.) ;M deles p 100 for
g fra cGy til Gy
79         pDP[i] = ptr_new(*(p_dpbn_infomatrix[0])/100. * 34.) ;Ganges med 34
fraksjoner
80         pH[i] = ptr_new(*(ph_infomatrix[0])/100. * 34.)
81         pHDP[i] = ptr_new(*(ph_dpbn_infomatrix[0])/100. * 34.)
82         RSinv = *(inv_doseRS_infomatrix[0])/100.
83
84         ROIindex_pat = ptrarr(n_elements(ROIlist))
85
86         for j = 0, n_elements(ROIlist)-1 do begin
87             if ROIlist[j] eq 'Ipsi' then begin
88                 ROIer = 'Parotid_' + ipsi[i]
89                 ROIindex_pat[j] = ptr_new(read_dat_ind(patient, ROIer))
90             endif
91
92             if ROIlist[j] eq 'Cont' then begin
93                 ROIer = 'Parotid_' + cont[i]
94                 ROIindex_pat[j] = ptr_new(read_dat_ind(patient, ROIer))
95             endif
96
97             if ROIlist[j] eq 'GTV68' then begin
98                 ROIindex_pat[j] = ptr_new(where(RSinv lt 50))
99             endif
100
101             if (ROIlist[j] ne 'Ipsi' && ROIlist[j] ne 'Cont' && ROIlist[j] ne '
GTV68') then begin
102                 ROIindex_pat[j] = ptr_new(read_dat_ind(patient, ROIlist[j]))
103             endif
104         endfor
105
106         ROIindex[i] = ptr_new(ROIindex_pat)

```

```

105 endfor
106
107 tot_vox_arr = ptrarr(n_elements(ROIlist)*4)
108 tot_vox_arr_std = ptrarr(n_elements(ROIlist)*4)
109
110 for k = 0, n_elements(ROIlist)-1 do begin
111     pDhistos = ptrarr(n_elements(patientlist))
112     pDPhistos = ptrarr(n_elements(patientlist))
113     pHhistos = ptrarr(n_elements(patientlist))
114     pHDPhistos = ptrarr(n_elements(patientlist))
115
116     for j = 0, n_elements(patientlist) - 1 do begin
117         index = (*ROIindex[j])[k]
118         pDhistos[j] = ptr_new(histogram((*pD[j])[index], min = minimum, max =
maximum, nbins = 1001, locations = xbin))
119         pDPhistos[j] = ptr_new(histogram((*pDP[j])[index], min = minimum, max =
maximum, nbins = 1001, locations = xbin))
120         pHhistos[j] = ptr_new(histogram((*pH[j])[index], min = minimum, max =
maximum, nbins = 1001, locations = xbin))
121         pHDPhistos[j] = ptr_new(histogram((*pHDP[j])[index], min = minimum, max
= maximum, nbins = 1001, locations = xbin))
122     endfor
123
124     tot_vox1 = make_array(size = size(xbin*4))
125     tot_vox_std1 = make_array(size = size(xbin*4))
126     tot_vox2 = make_array(size = size(xbin*4))
127     tot_vox_std2 = make_array(size = size(xbin*4))
128     tot_vox3 = make_array(size = size(xbin*4))
129     tot_vox_std3 = make_array(size = size(xbin*4))
130     tot_vox4 = make_array(size = size(xbin*4))
131     tot_vox_std4 = make_array(size = size(xbin*4))
132
133     norm1=fltarr(n_elements(patientlist))
134     norm2=fltarr(n_elements(patientlist))
135     norm3=fltarr(n_elements(patientlist))
136     norm4=fltarr(n_elements(patientlist))
137
138     for i = 0, n_elements(patientlist)-1 do begin ;Goes through every
patient
139         norm1[i] = total((*pDhistos[i])[0:-1]) ;Counts all contained in
this bin (j) and all after for plan pDhistos
140         norm2[i] = total((*pDPhistos[i])[0:-1])
141         norm3[i] = total((*pHhistos[i])[0:-1])
142         norm4[i] = total((*pHDPhistos[i])[0:-1])
143     endfor
144
145     for j = 0, n_elements(tot_vox1)-1 do begin ; Goes through every bin
146         temp_arr1 = fltarr(n_elements(patientlist))
147         temp_arr2 = fltarr(n_elements(patientlist))
148         temp_arr3 = fltarr(n_elements(patientlist))
149         temp_arr4 = fltarr(n_elements(patientlist)) ;make_array(n_elements(
patientlist), value = 5)
150
151         for i = 0, n_elements(patientlist)-1 do begin ;Goes through every
patient
152             temp_arr1[i] = total((*pDhistos[i])[j:-1])/norm1[i] ;Counts all
contained in this bin (j) and all after for plan pDhistos
153             temp_arr2[i] = total((*pDPhistos[i])[j:-1])/norm2[i]
154             temp_arr3[i] = total((*pHhistos[i])[j:-1])/norm3[i]

```

```

155     temp_arr4[i] = total ((* (phDPHistos[i])) [j:-1]) / norm4[i]
156     endfor
157
158     tot_vox1[j] = mean(temp_arr1)      ;Mean values of the total of all
patients for bin (j)
159     tot_vox_std1[j] = stddev(temp_arr1)      ;STD of total of all patients
for bin (j)
160     tot_vox2[j] = mean(temp_arr2)
161     tot_vox_std2[j] = stddev(temp_arr2)
162     tot_vox3[j] = mean(temp_arr3)
163     tot_vox_std3[j] = stddev(temp_arr3)
164     tot_vox4[j] = mean(temp_arr4)
165     tot_vox_std4[j] = stddev(temp_arr4)
166
167
168     endfor
169
170     tot_vox_arr[k + k*3: k + k*3 + 3] = [ptr_new(tot_vox1), ptr_new(tot_vox2)
, ptr_new(tot_vox3), ptr_new(tot_vox4)]
171     tot_vox_arr_std[k + k*3: k + k*3 + 3] = [ptr_new(tot_vox_std1), ptr_new(
tot_vox_std2), ptr_new(tot_vox_std3), ptr_new(tot_vox_std4)]
172 ;     hP = histogram(mean([ total (pDhist[i:-1]), total (pDPDhist[i:-1]), total (
phDhist[i:-1]), total (phDPDhist[i:-1])]) / float(mean([ total (pDhist), total
(pDPDhist), total (phDhist), total (phDPDhist) ])))
173 ;     hP = histogram(mean([ total ((* pD[0]) (* ROIindex[0]) [i] [i:-1] ])))
174 ;     endfor
175 endfor
176
177 colorlist = ['-r', '-b', '-g', '-k']
178
179
180 for i = 0, n_elements(ROIlist)-1 do begin
181     filename = 'C:\Users\Eirik\Dropbox\Universitet\Master\Txt\' + "DVHfor_" +
ROIlist[i] + "basedOn_" + strcompress(string(n_elements(patientlist)), /
remove_all) + "_patients.txt"
182     openw, 1, filename
183     printf, 1, xbin
184     printf, 1, 'Goodbye'
185     for j = 0, 3 do begin
186         printf, 1, *(tot_vox_arr[j + i*3])
187         printf, 1, 'Hello'
188         printf, 1, *(tot_vox_arr_std[j + i*4])
189         printf, 1, 'Goodbye'
190     endfor
191     close, 1
192 endfor
193
194
195 stop
196
197 end

```

The script `ROIcreator.py` asks for which ROIs represents the GTV68, the CTV64 and the CTV54 before creating and naming all relevant volumes.

```
1 #
```

```

#####
2 #
#####
3 ##### Created by Eirik Ramsli Hauge
#####
4 ##### Based on: www.voidspace.org.uk/ironpython/winforms/part6.
#####
5 ##### Contact: eirikhaug@hotmail.com
#####
6 ##### Feel free to ask if anythings wrong :)
#####
7 #
#####
8 #
#####
9
10
11
12 from connect import *
13
14 import clr
15
16 clr.AddReference('System.Windows.Forms')
17 clr.AddReference('System.Drawing')
18
19 import sys
20
21 from math import ceil
22 from System.Drawing import Color, Font, FontStyle, Point
23 from System.Windows.Forms import Application, BorderStyle, Button, CheckBox,
Form, Label, Panel, RadioButton
24
25 class SelectROIForm(Form):
26     def __init__(self, plan):
27         self.Text = "Choose the appropriate ROI"
28
29         structure_set = plan.GetStructureSet()
30         self.roi_names = sorted([rg.OfRoi.Name for rg in structure_set.
RoiGeometries if rg.PrimaryShape != None])
31         self.roi_colors = [rg.OfRoi.Color for rg in structure_set.RoiGeometries
if rg.PrimaryShape != None]
32
33         self.Width = 160*4 + 50
34         self.Height = 55*int(ceil(len(self.roi_names)/4.)) + 150
35
36         print self.roi_colors
37
38         self.setupCheckButtons()
39
40         # Add button to press OK and close the Form
41         button = Button()
42         button.Text = "OK"
43         button.AutoSize = True
44         button.Location = Point(self.Width - 105, self.Height - 100)
45         button.Click += self.ok_button_clicked

```

```

46
47 # Add button to press Stop and close the Form
48 button2 = Button()
49 button2.Text = "Stop"
50 button2.AutoSize = True
51 button2.Location = Point(self.Width - 210, self.Height - 100)
52 button2.Click += self.stop_button_clicked
53
54 self.Controls.Add(button)
55 self.Controls.Add(button2)
56
57 self.Controls.Add(self.checkPanel)
58
59 def newPanel(self, x, y):
60     panel = Panel()
61
62     panel.Width = 5120
63     panel.Height = 256*(len(self.roi_names)/2.)
64     panel.Location = Point(x, y)
65     panel.BorderStyle = BorderStyle.Fixed3D
66     return panel
67
68 def setupCheckButtons(self):
69     self.checkPanel = self.newPanel(0,0)
70
71     self.checkLabel = Label()
72     self.checkLabel.Text = "Choose the ROI(s) that represents GTV68"
73     self.checkLabel.Location = Point(25, 25)
74     self.checkLabel.AutoSize = True
75
76     self.checkBoxList = [CheckBox() for i in range(0, len(self.roi_names))]
77     checkBoxJumper = [i*4 for i in range(1, int(ceil(len(self.roi_names)/4.))]
78
79     xcounter = 0
80     ycounter = 1
81     for i in range(0, len(self.roi_names)):
82         for j in checkBoxJumper:
83             if i == j:
84                 xcounter = 0
85                 ycounter += 1
86
87                 self.checkBoxList[i].Text = self.roi_names[i]
88                 self.checkBoxList[i].Location = Point(xcounter*160 + 25, 55*ycounter)
89                 xcounter += 1
90
91
92                 self.checkBoxList[i].Width = 150
93                 self.checkPanel.Controls.Add(self.checkBoxList[i])
94
95     self.checkPanel.Controls.Add(self.checkLabel)
96
97 def ok_button_clicked(self, sender, event):
98     # Method invoked when the button is clicked
99     # Save the selected ROI name
100     self.roi_name_list = [i.Text for i in self.checkBoxList if str(i.
101     CheckState) == "Checked"]
102
103     #for i in self.checkBoxList:

```

```

103     # if i.CheckState == "Checked":
104     #     self.roi_name_list.Append(i.Text)
105     #self.roi_name = self.combobox.SelectedValeu
106     # Close the form
107     self.Close()
108
109     def stop_button_clicked(self, sender, event):
110
111         # Close the form
112         self.Close()
113
114
115     def redefine_text(self, labeltext):
116         self.Text = labeltext
117         self.checkLabel.Text = labeltext
118         #self.Controls.Add(self.Text) maybe this is needed?
119
120
121     def renameROI(ROIname, plan, case):
122         Popup = SelectROIForm(plan)
123         Popup.redefine_text('Select the roi that is %s' %ROIname)
124         Application.Run(Popup)
125
126         with CompositeAction('Apply ROI changes (%s)' %Popup.roi_name_list[0]):
127             case.PatientModelRegionsOfInterest['%s' %Popup.roi_name_list[0]].Name =
128                 ROIname
129
130         plan = get_current("Plan")
131
132         form = SelectROIForm(plan)
133         Application.Run(form)
134
135         case = get_current("Case")
136         examination = get_current("Examination")
137
138         #Try to copy ROI specified from UI
139
140         with CompositeAction('ROI Algebra (GTV68, Image set: CT 1)'):
141
142             retval_GTV68 = case.PatientModel.CreateRoi(Name = "GTV68", Color = "255,
143                 140, 0", Type = "Gtv", TissueName = None, RoiMaterial = None)
144
145             retval_GTV68.CreateAlgebraGeometry(Examination=examination, Algorithm="Auto
146                 ",
147                 ExpressionA={ 'Operation': "Union", 'SourceRoiNames': [i
148                     for i in form.roi_name_list], 'MarginSettings': { 'Type': "Expand", '
149                     Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, '
150                     Left': 0 } },
151                 ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [], '
152                     MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
153                     Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
154                 ResultOperation="None", ResultMarginSettings={ 'Type': "
155                     Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '
156                     Right': 0, 'Left': 0 })
157
158         with CompositeAction('ROI Algebra (PTV68, Image set: CT 1)'):
159
160             retval_PTV68 = case.PatientModel.CreateRoi(Name="PTV68", Color="Blue",

```

```

Type="Ptv", TissueName=None, RoiMaterial=None)
152
153     retval_PTV68.CreateAlgebraGeometry(Examination=examination, Algorithm="
Auto",
154         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
GTV68"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0.3, 'Inferior
': 0.3, 'Anterior': 0.3, 'Posterior': 0.3, 'Right': 0.3, 'Left': 0.3 } },
155         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [],
'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
156         ResultOperation="None", ResultMarginSettings={ 'Type': "
Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '
Right': 0, 'Left': 0 })
157
158 with CompositeAction('ROI Algebra (PTV68_eks, Image set: CT 1)'):
159
160     retval_PTV68_eks = case.PatientModel.CreateRoi(Name="PTV68_eks", Color="
Blue", Type="Ptv", TissueName=None, RoiMaterial=None)
161
162     retval_PTV68_eks.CreateAlgebraGeometry(Examination=examination, Algorithm
="Auto",
163         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
PTV68"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior':
0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
164         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': ["
GTV68"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior':
0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
165         ResultOperation="Subtraction", ResultMarginSettings={ '
Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
: 0, 'Right': 0, 'Left': 0 })
166
167 ### Need to insert a UI for selecting ITVs ###
168 ### Add this later, until then: Make the ROI before you start the script. ###
169
170 form2 = SelectROIForm(plan)
171 form2.redefine_text("Select the ROI that is CTV64")
172 Application.Run(form2)
173
174 with CompositeAction('ROI Algebra (CTV64, Image set: CT 1)'):
175
176     retval_CTV64 = case.PatientModel.CreateRoi(Name = "CTV64", Color = "Red",
Type = "Ctv", TissueName = None, RoiMaterial = None)
177
178     retval_CTV64.CreateAlgebraGeometry(Examination=examination, Algorithm="Auto
",
179         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': [i
for i in form2.roi_name_list], 'MarginSettings': { 'Type': "Expand", '
Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, '
Left': 0 } },
180         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [], '
MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
181         ResultOperation="None", ResultMarginSettings={ 'Type': "
Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '
Right': 0, 'Left': 0 })
182
183 ### From reording: ###
184 #with CompositeAction('ROI Algebra (CTV64_eks, Image set: CT 1)'):
185 #

```

```

186 # retval_1 = case.PatientModel.CreateRoi(Name="CTV64_eks", Color="Red", Type
    ="Ctv", TissueName=None, RoiMaterial=None)
187 #
188 # retval_1.CreateAlgebraGeometry(Examination=examination, Algorithm="Auto",
189 #                               ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
    ITVlkmet I/cecia", "ITVlkmet III uni", "ITVlkmetII", "ITV-PETlkmetII", "
    ITV-PETTumor", "ITVtumor CT/ cec"]},
190 #                               'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
    Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
191 #                               ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [],
    'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
    Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
192 #                               ResultOperation="None", ResultMarginSettings={ 'Type': "
    Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '
    Right': 0, 'Left': 0 })
193 #
194 #####
195
196 with CompositeAction('ROI Algebra (PTV64, Image set: CT 1)'):
197
198     retval_PTV64 = case.PatientModel.CreateRoi(Name="PTV64", Color="Blue",
    Type="Ptv", TissueName=None, RoiMaterial=None)
199
200     retval_PTV64.CreateAlgebraGeometry(Examination=examination, Algorithm="
    Auto",
201                                     ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
    CTV64"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0.3, 'Inferior
    ': 0.3, 'Anterior': 0.3, 'Posterior': 0.3, 'Right': 0.3, 'Left': 0.3 } },
202                                     ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [], '
    MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
    Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
203                                     ResultOperation="None", ResultMarginSettings={ 'Type': "
    Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '
    Right': 0, 'Left': 0 })
204
205 ### Need to insert a UI for selecting ITVs ###
206 ### Add this later, until then: Make the ROI before you start the script. ###
207
208 form3 = SelectROIForm(plan)
209 form3.redefine_text("Select the ROI that is CTV54 without anything subtracted
    ")
210 Application.Run(form3)
211
212 with CompositeAction('ROI Algebra (CTV54, Image set: CT 1)'):
213
214     retval_CTV54 = case.PatientModel.CreateRoi(Name = "CTV54", Color = "Red",
    Type = "Ctv", TissueName = None, RoiMaterial = None)
215
216     retval_CTV54.CreateAlgebraGeometry(Examination=examination, Algorithm="Auto
    ",
217                                     ExpressionA={ 'Operation': "Union", 'SourceRoiNames': [i
    for i in form3.roi_name_list], 'MarginSettings': { 'Type': "Expand", '
    Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, '
    Left': 0 } },
218                                     ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [], '
    MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
    Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
219                                     ResultOperation="None", ResultMarginSettings={ 'Type': "
    Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '

```

```

    Right': 0, 'Left': 0 })
220
221 with CompositeAction('ROI Algebra (CTV54_eks, Image set: CT 1)'):
222
223     retval_CTV54_eks = case.PatientModel.CreateRoi(Name = "CTV54_eks", Color =
224         "Red", Type = "Ctv", TissueName = None, RoiMaterial = None)
225
226     retval_CTV54_eks.CreateAlgebraGeometry(Examination=examination, Algorithm="
227         Auto",
228         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': [i
229             for i in form3.roi_name_list], 'MarginSettings': { 'Type': "Expand", '
230             Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, '
231             Left': 0 } }},
232         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': ["
233             PTV64"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior':
234             0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
235         ResultOperation="Subtraction", ResultMarginSettings={ 'Type
236             ': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0,
237             'Right': 0, 'Left': 0 })
238
239
240
241
242
243 ### From recording: ###
244 #
245 #with CompositeAction('ROI Algebra (CTV54_eks, Image set: CT 1)'):
246 #
247 #     retval_3 = case.PatientModel.CreateRoi(Name="CTV54_eks", Color="Red", Type
248 #         ="Ctv", TissueName=None, RoiMaterial=None)
249 #
250 #     retval_3.CreateAlgebraGeometry(Examination=examination, Algorithm="Auto",
251 #         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
252 #             ITVelektive lk s", "ITVelektive lk.d"],
253 #             'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
254 #             Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
255 #         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': ["
256 #             PTV64_eks"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
257 #             Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
258 #         ResultOperation="Subtraction", ResultMarginSettings={ 'Type
259 #             ': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0,
260 #             'Right': 0, 'Left': 0 })
261 #
262 #
263 #####
264
265 with CompositeAction('ROI Algebra (PTV54_eks, Image set: CT 1)'):
266
267     retval_PTV54_eks = case.PatientModel.CreateRoi(Name="PTV54_eks", Color="
268         Blue", Type="Ptv", TissueName=None, RoiMaterial=None)
269
270     retval_PTV54_eks.CreateAlgebraGeometry(Examination=examination, Algorithm
271         ="Auto",
272         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
273             CTV54_eks"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0.3, '
274             Inferior': 0.3, 'Anterior': 0.3, 'Posterior': 0.3, 'Right': 0.3, 'Left':
275             0.3 } }},
276         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': ["
277             PTV64"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior':
278             0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
279         ResultOperation="Subtraction", ResultMarginSettings={ '
280             Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
281             ': 0, 'Right': 0, 'Left': 0 })

```

```

253
254 with CompositeAction('ROI Algebra (CTV64_eks, Image set: CT 1)'):
255
256     retval_CTV64_eks = case.PatientModel.CreateRoi(Name="CTV64_eks", Color="Red
257         ", Type="Ctv", TissueName=None, RoiMaterial=None)
258
259     retval_CTV64_eks.CreateAlgebraGeometry(Examination=examination, Algorithm
260         ="Auto",
261         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["CTV64"], '
262             MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
263             Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
264         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': ["PTV68"], '
265             MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
266             Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
267         ResultOperation="Subtraction", ResultMarginSettings={ 'Type': "Expand",
268             'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0,
269             'Left': 0 })
270
271 with CompositeAction('ROI Algebra (PTV64, Image set: CT 1)'):
272
273     retval_PTV64_eks = case.PatientModel.CreateRoi(Name="PTV64_eks", Color="
274         Blue", Type="Ptv", TissueName=None, RoiMaterial=None)
275
276     retval_PTV64_eks.CreateAlgebraGeometry(Examination=examination, Algorithm
277         ="Auto",
278         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
279             CTV64"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0.3, 'Inferior
280             ': 0.3, 'Anterior': 0.3, 'Posterior': 0.3, 'Right': 0.3, 'Left': 0.3 } },
281         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': ["
282             PTV68"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior':
283             0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
284         ResultOperation="Subtraction", ResultMarginSettings={ '
285             Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
286             ': 0, 'Right': 0, 'Left': 0 })
287
288 External_intersection_list = ["CTV64_eks", "CTV54_eks", "PTV64_eks", "
289     PTV54_eks"]
290
291 form4 = SelectROIForm(plan)
292 form4.redefine_text("Select the ROI that is the external. Choose only one!")
293 Application.Run(form4)
294
295 margin = 0.5
296 ExternalName = form4.roi_name_list #Don't use an index here
297 print ExternalName
298
299 with CompositeAction('ROI Algebra (PTV64_eks_5mm, Image set: CT 1)'):
300
301     retval_PTV64_eks_5mm = case.PatientModel.CreateRoi(Name="PTV64_eks_5mm",
302         Color="Blue", Type="Ptv", TissueName=None, RoiMaterial=None)
303
304     retval_PTV64_eks_5mm.CreateAlgebraGeometry(Examination=examination,
305         Algorithm="Auto",
306         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
307             PTV64_eks"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
308             Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
309         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [
310             ExternalName[0]], 'MarginSettings': { 'Type': "Contract", 'Superior':

```

```

0.5, 'Inferior': 0.5, 'Anterior': 0.5, 'Posterior': 0.5, 'Right': 0.5, '
Left': 0.5 } }},
290     ResultOperation="Intersection", ResultMarginSettings={ '
Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
: 0, 'Right': 0, 'Left': 0 })
291
292 with CompositeAction('ROI Algebra (PTV54_eks_5mm, Image set: CT 1)'):
293
294     retval_PTV54_eks_5mm = case.PatientModel.CreateRoi(Name="PTV54_eks_5mm",
Color="Blue", Type="Ptv", TissueName=None, RoiMaterial=None)
295
296     retval_PTV54_eks_5mm.CreateAlgebraGeometry(Examination=examination,
Algorithm="Auto",
297         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
PTV54_eks"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
298         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [
ExternalName[0]], 'MarginSettings': { 'Type': "Contract", 'Superior':
0.5, 'Inferior': 0.5, 'Anterior': 0.5, 'Posterior': 0.5, 'Right': 0.5, '
Left': 0.5 } }},
299         ResultOperation="Intersection", ResultMarginSettings={ '
Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
: 0, 'Right': 0, 'Left': 0 })
300
301 with CompositeAction('ROI Algebra (CTV64_eks_5mm, Image set: CT 1)'):
302
303     retval_CTV64_eks_5mm = case.PatientModel.CreateRoi(Name="CTV64_eks_5mm",
Color="Red", Type="Ctv", TissueName=None, RoiMaterial=None)
304
305     retval_CTV64_eks_5mm.CreateAlgebraGeometry(Examination=examination,
Algorithm="Auto",
306         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
CTV64_eks"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
307         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [
ExternalName[0]], 'MarginSettings': { 'Type': "Contract", 'Superior':
0.5, 'Inferior': 0.5, 'Anterior': 0.5, 'Posterior': 0.5, 'Right': 0.5, '
Left': 0.5 } }},
308         ResultOperation="Intersection", ResultMarginSettings={ '
Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
: 0, 'Right': 0, 'Left': 0 })
309
310 with CompositeAction('ROI Algebra (CTV54_eks_5mm, Image set: CT 1)'):
311
312     retval_CTV54_eks_5mm = case.PatientModel.CreateRoi(Name="CTV54_eks_5mm",
Color="Red", Type="Ctv", TissueName=None, RoiMaterial=None)
313
314     retval_CTV54_eks_5mm.CreateAlgebraGeometry(Examination=examination,
Algorithm="Auto",
315         ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
CTV54_eks"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0, '
Inferior': 0, 'Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } }},
316         ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [
ExternalName[0]], 'MarginSettings': { 'Type': "Contract", 'Superior':
0.5, 'Inferior': 0.5, 'Anterior': 0.5, 'Posterior': 0.5, 'Right': 0.5, '
Left': 0.5 } }},
317         ResultOperation="Intersection", ResultMarginSettings={ '
Type': "Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior'
: 0, 'Right': 0, 'Left': 0 })

```

```

318
319 if form4.roi_name_list[0] != "External":
320     with CompositeAction('Apply ROI changes (%s)' %form4.roi_name_list[0]):
321
322         case.PatientModel.RegionsOfInterest['%s' %form4.roi_name_list[0]].Name =
323             "External"
324 nothurtlist = ['Parotid_R', 'Parotid_L', 'Submandibular_L', 'Submandibular_R',
325               , 'SpinalCord']
326 #for i in nothurtlist:
327 # renameROI(i, plan, case)
328
329 structure_set = plan.GetStructureSet()
330 roi_names = [rg.OfRoi.Name for rg in structure_set.RoiGeometries if rg.
331               PrimaryShape != None]
332 for j in nothurtlist:
333     counter = 0
334     for i in roi_names:
335         if i == j:
336             counter = 1
337     if counter == 0:
338         form_loop = SelectROIForm(plan)
339         form_loop.redefine_text("Select the ROI that is %s. Choose only one!" %j)
340         Application.Run(form_loop)
341
342         with CompositeAction('Apply ROI changes (%s)' %form_loop.roi_name_list
343                               [0]):
344             case.PatientModel.RegionsOfInterest['%s' %form_loop.roi_name_list[0]].
345             Name = j
346
347             if j == "SpinalCord":
348                 with CompositeAction('ROI Algebra (SpinalCord_PRV, Image set: CT 1)'):
349
350                     retval_SpinalCord_PRV = case.PatientModel.CreateRoi(Name="
351                     SpinalCord_PRV", Color="Green", Type="Avoidance", TissueName=None,
352                     RoiMaterial=None)
353
354                     retval_SpinalCord_PRV.CreateAlgebraGeometry(Examination=examination,
355                     Algorithm="Auto",
356                     ExpressionA={ 'Operation': "Union", 'SourceRoiNames': ["
357                     SpinalCord"], 'MarginSettings': { 'Type': "Expand", 'Superior': 0.3, '
358                     Inferior': 0.3, 'Anterior': 0.3, 'Posterior': 0.3, 'Right': 0.3, 'Left':
359                     0.3 } },
360                     ExpressionB={ 'Operation': "Union", 'SourceRoiNames': [], '
361                     MarginSettings': { 'Type': "Expand", 'Superior': 0, 'Inferior': 0, '
362                     Anterior': 0, 'Posterior': 0, 'Right': 0, 'Left': 0 } },
363                     ResultOperation="None", ResultMarginSettings={ 'Type': "
364                     Expand", 'Superior': 0, 'Inferior': 0, 'Anterior': 0, 'Posterior': 0, '
365                     Right': 0, 'Left': 0 })
366
367 elif counter > 1 or counter < 0:
368     print "Your counter is wrong"
369
370

```

The script `export_dose_Eirik.py` exports the dose from RayStation

```

1 #
2 #####
3 #
4 #####
5 ##### Created by Eirik Ramsli Hauge
6 #####
7 ##### Based on: Program given by RayStation upon request
8 #####
9 ##### Contact: eirikhaug@hotmai.com
10 #####
11 ##### Feel free to ask if anythings wrong :)
12 #####
13 #
14 #####
15 #
16 #####
17 #####
18
19 from connect import *
20
21 import struct
22 from array import array
23 path = r'C:\Users\eiraha\Documents\DoseData'
24
25 #export the fraction dose
26 patient = get_current("Patient")
27 plan = get_current('Plan')
28 beam_set = get_current("BeamSet")
29 dose = beam_set.FractionDose
30 fractions = beam_set.FractionationPattern.NumberOfFractions
31 filename = path + r'\fraction_dose_patient_%s_Doseplan_%s_BeamSet_%s.dat' %(
32 patient.PatientName, plan.Name, str(str(beam_set).split(" ")[1]))
33 #
34
35
36 # write the dose file
37
38 #open file
39 file = open(filename, 'wb')
40
41 # write dose grid dimensions
42 dg = dose.InDoseGrid
43 # number of voxels (DICOM axes x, y and z)
44 file.write(struct.pack('III', dg.NrVoxels.x, dg.NrVoxels.y, dg.NrVoxels.z))
45 #voxel size
46 file.write(struct.pack('ddd', dg.VoxelSize.x, dg.VoxelSize.y, dg.VoxelSize.z)
47 )
48 #corner of the corner voxel (cm)
49 file.write(struct.pack('ddd', dg.Corner.x, dg.Corner.y, dg.Corner.z))
50
51 # write the number of fractions
52 file.write(struct.pack('I', fractions))

```

```

42
43 # write the length of the dose array and the array (cGy)
44 binary_dose = dose.DoseValues.DoseData
45 length = len(binary_dose)
46 file.write(struct.pack('I', length))
47 array('d', binary_dose).tofile(file)
48
49 # done
50 file.close()

```

The script DoseGoalssetup.py sets up the clinical goals and optimization settings for the proton plan.

```

1 from connect import *
2
3 plan = get_current("Plan")
4 db = get_current("PatientDB")
5
6
7
8 # Unscriptable Action 'Save' Completed : SaveAction(...)
9
10 plan.PlanOptimizations[2].ApplyOptimizationTemplate(Template=db.
    TemplateTreatmentOptimizations['Eirik_rob_pro'])
11
12 plan.TreatmentCourse.EvaluationSetup.ApplyClinicalGoalTemplate(Template=db.
    TemplateTreatmentOptimizations['Eirik_rob_del'])
13
14 # Unscriptable Action 'Save' Completed : SaveAction(...)
15
16 plan.PlanOptimizations[1].ApplyOptimizationTemplate(Template=db.
    TemplateTreatmentOptimizations['Eirik_rob_pro'])
17
18 structure_set = plan.GetStructureSet()
19 roi_names = sorted([rg.OfRoi.Name for rg in structure_set.RoiGeometries if rg
    .PrimaryShape != None])
20 name_list = ["Parotid_R", "Parotid_L", "Submandibular_R", "Submandibular_L",
    "SpinalCord_PRV"]
21 FunctionTypeList = ["MaxEud", "MaxEud", "MaxEud", "MaxEud", "MaxDose"]
22 index_list = [0, 1, 2, 3, 12]
23 retval_list = ["retval_%d" %i for i in index_list]
24 exsisting_list = []
25
26 counter2 = 0
27 for j in name_list:
28     counter = 0
29     for i in roi_names:
30         if j == i:
31             counter = 1
32         if counter == 1:
33             exsisting_list.append(counter2)
34             counter2 += 1
35
36
37 for i in range(0, len(exsisting_list)):
38     with CompositeAction('Edit Optimization Function'):

```

```

39     retval_list[i] = plan.PlanOptimizations[1].EditOptimizationFunction(
    FunctionType="%s" %FunctionTypeList[i], DoseBasedRoiFunction=plan.
    PlanOptimizations[1].Objective.ConstituentFunctions[index_list[i]],
    RoiName="%s" %name_list[i], IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
40
41
42 with CompositeAction('Edit Optimization Function'):
43
44     retval_21 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[4], RoiName="SpinalCord", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
45
46     # CompositeAction ends
47
48
49 with CompositeAction('Edit Optimization Function'):
50
51     retval_22 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="UniformDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[5], RoiName="GTV68", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet=None, UseRbeDose=False)
52
53     plan.PlanOptimizations[1].Objective.ConstituentFunctions[5].
    DoseFunctionParameters.DoseLevel = 11800
54
55     # CompositeAction ends
56
57
58 with CompositeAction('Edit Optimization Function'):
59
60     retval_23 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="UniformDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[6], RoiName="CTV54_eks", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
61
62     # CompositeAction ends
63
64
65 with CompositeAction('Edit Optimization Function'):
66
67     retval_24 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="UniformDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[7], RoiName="CTV64_eks", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
68
69     # CompositeAction ends
70
71
72 with CompositeAction('Edit Optimization Function'):
73
74     retval_25 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="DoseFallOff", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.

```

```

75     ConstituentFunctions [8], RoiName="External", IsConstraint=False,
76     RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
77     RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
78
79 # CompositeAction ends
80
81 with CompositeAction('Edit Optimization Function'):
82     retval_26 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType
83     ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations [1].Objective.
84     ConstituentFunctions [9], RoiName="External", IsConstraint=False,
85     RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
86     RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
87     plan.PlanOptimizations [1].Objective.ConstituentFunctions [9].
88     DoseFunctionParameters.DoseLevel = 9000
89 # CompositeAction ends
90
91 with CompositeAction('Edit Optimization Function'):
92
93     retval_27 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType
94     ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations [1].Objective.
95     ConstituentFunctions [10], RoiName="CTV64_eks", IsConstraint=False,
96     RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
97     RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
98 # CompositeAction ends
99
100 with CompositeAction('Edit Optimization Function'):
101
102     retval_28 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType
103     ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations [1].Objective.
104     ConstituentFunctions [11], RoiName="CTV54_eks", IsConstraint=False,
105     RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
106     RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
107 # CompositeAction ends
108
109 with CompositeAction('Edit Optimization Function'):
110
111     retval_30 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType
112     ="MinDvh", DoseBasedRoiFunction=plan.PlanOptimizations [1].Objective.
113     ConstituentFunctions [13], RoiName="GTV68", IsConstraint=False,
114     RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=True,
115     RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
116 # CompositeAction ends
117
118 with CompositeAction('Edit Optimization Function'):
119
120     retval_31 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType
121     ="MinDvh", DoseBasedRoiFunction=plan.PlanOptimizations [1].Objective.
122     ConstituentFunctions [14], RoiName="CTV64_eks_5mm", IsConstraint=False,
123     RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=True,
124     RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)

```

```

110 # CompositeAction ends
111
112
113 with CompositeAction('Edit Optimization Function'):
114
115     retval_32 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
116         ="MinDvh", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
117         ConstituentFunctions[15], RoiName="CTV54_eks_5mm", IsConstraint=False,
118         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=True,
119         RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
120
121 # CompositeAction ends
122
123
124 with CompositeAction('Edit Optimization Function'):
125
126     retval_33 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
127         ="DoseFallOff", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
128         ConstituentFunctions[16], RoiName="External", IsConstraint=False,
129         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
130         RestrictToBeamSet="rob_p_dpbn", UseRbeDose=False)
131
132 # CompositeAction ends
133
134 # Unscriptable Action 'Modify optimization settings' Completed :
135     SaveOptimizationSettingsAction(...)
136
137 plan.PlanOptimizations[1].OptimizationParameters.SaveRobustnessParameters(
138     PositionUncertaintyAnterior=0.6, PositionUncertaintyPosterior=0.6,
139     PositionUncertaintySuperior=0.6, PositionUncertaintyInferior=0.6,
140     PositionUncertaintyLeft=0.6, PositionUncertaintyRight=0.6,
141     DensityUncertainty=0.03, IndependentBeams=False,
142     ComputeExactScenarioDoses=False, NamesOfNonPlanningExaminations=[])
143 plan.PlanOptimizations[2].OptimizationParameters.SaveRobustnessParameters(
144     PositionUncertaintyAnterior=0.6, PositionUncertaintyPosterior=0.6,
145     PositionUncertaintySuperior=0.6, PositionUncertaintyInferior=0.6,
146     PositionUncertaintyLeft=0.6, PositionUncertaintyRight=0.6,
147     DensityUncertainty=0.03, IndependentBeams=False,
148     ComputeExactScenarioDoses=False, NamesOfNonPlanningExaminations=[])

```

The script `DoseGoalssetupPH.py` sets up the clinical goals and optimization settings for the photon plan.

```

1 # Script recorded 31 Jan 2018
2
3 # RayStation version: 4.99.1.12
4 # Selected patient: ...
5
6 from connect import *
7
8 plan = get_current("Plan")
9 db = get_current("PatientDB")
10
11 plan.PlanOptimizations[2].ApplyOptimizationTemplate(Template=db.
12     TemplateTreatmentOptimizations['Eirik_rob_ph_hn'])

```

```

12
13 plan.TreatmentCourse.EvaluationSetup.ApplyClinicalGoalTemplate( Template=db .
    TemplateTreatmentOptimizations [ 'Eirik_rob_del' ])
14
15
16 plan.PlanOptimizations [1].ApplyOptimizationTemplate( Template=db .
    TemplateTreatmentOptimizations [ 'Eirik_rob_ph_hn' ])
17
18
19 structure_set = plan.GetStructureSet ()
20 roi_names = sorted([rg.OfRoi.Name for rg in structure_set.RoiGeometries if rg
    .PrimaryShape != None])
21 name_list = ["Parotid_L", "Parotid_R", "Submandibular_L", "Submandibular_R",
    "SpinalCord-PRV"]
22 FunctionTypeList = ["MaxEud", "MaxEud", "MaxEud", "MaxEud", "MaxDose"]
23 retval_list = ["retval_%d" %i for i in range(0, len(name_list))]
24 index_list = [0, 1, 2, 3, 5]
25 exsisting_list = []
26
27 counter2 = 0
28 for j in name_list:
29     counter = 0
30     for i in roi_names:
31         if j == i:
32             counter = 1
33     if counter == 1:
34         exsisting_list.append(counter2)
35     counter2 += 1
36
37
38 for i in range(0, len(exsisting_list)):
39     with CompositeAction('Edit Optimization Function'):
40         retval_list[i] = plan.PlanOptimizations [1].EditOptimizationFunction (
            FunctionType="%s" %FunctionTypeList [i], DoseBasedRoiFunction=plan .
            PlanOptimizations [1].Objective . ConstituentFunctions [index_list [i]],
            RoiName="%s" %name_list [i], IsConstraint=False ,
            RestrictAllBeamsIndividually=False , RestrictToBeam=None, IsRobust=False ,
            RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False )
41
42 with CompositeAction('Edit Optimization Function'):
43
44     retval_4 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType=
        "MaxDose", DoseBasedRoiFunction=plan . PlanOptimizations [1].Objective .
        ConstituentFunctions [4], RoiName="SpinalCord", IsConstraint=False ,
        RestrictAllBeamsIndividually=False , RestrictToBeam=None, IsRobust=False ,
        RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False )
45
46 # CompositeAction ends
47
48
49 with CompositeAction('Edit Optimization Function'):
50
51     retval_6 = plan.PlanOptimizations [1].EditOptimizationFunction (FunctionType=
        "MinDvh", DoseBasedRoiFunction=plan . PlanOptimizations [1].Objective .
        ConstituentFunctions [6], RoiName="GTV68", IsConstraint=False ,
        RestrictAllBeamsIndividually=False , RestrictToBeam=None, IsRobust=False ,
        RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False )
52
53 # CompositeAction ends

```

```

54
55
56 with CompositeAction('Edit Optimization Function'):
57
58     retval_7 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType=
59         "UniformDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
60         ConstituentFunctions[7], RoiName="CTV54_eks", IsConstraint=False,
61         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
62         RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
63
64     # CompositeAction ends
65
66
67 with CompositeAction('Edit Optimization Function'):
68
69     retval_8 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType=
70         "UniformDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
71         ConstituentFunctions[8], RoiName="CTV64_eks", IsConstraint=False,
72         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
73         RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
74
75     # CompositeAction ends
76
77
78 with CompositeAction('Edit Optimization Function'):
79
80     retval_9 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType=
81         "DoseFallOff", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
82         ConstituentFunctions[9], RoiName="External", IsConstraint=False,
83         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
84         RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
85
86     # CompositeAction ends
87
88
89 with CompositeAction('Edit Optimization Function'):
90
91     retval_10 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
92         ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
93         ConstituentFunctions[10], RoiName="External", IsConstraint=False,
94         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
95         RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
96     plan.PlanOptimizations[1].Objective.ConstituentFunctions[10].
97     DoseFunctionParameters.DoseLevel = 9000
98
99     # CompositeAction ends
100
101
102 with CompositeAction('Edit Optimization Function'):
103
104     retval_11 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
105         ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
106         ConstituentFunctions[11], RoiName="CTV64_eks", IsConstraint=False,
107         RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
108         RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
109
110     # CompositeAction ends
111
112
113 with CompositeAction('Edit Optimization Function'):

```

```

92
93  retval_12 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="MaxDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[12], RoiName="CTV54_eks", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
94
95  # CompositeAction ends
96
97
98  with CompositeAction('Edit Optimization Function'):
99
100  retval_13 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="UniformDose", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[13], RoiName="GTV68", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=False,
    RestrictToBeamSet=None, UseRbeDose=False)
101
102  plan.PlanOptimizations[1].Objective.ConstituentFunctions[13].
    DoseFunctionParameters.DoseLevel = 11800
103
104  # CompositeAction ends
105
106
107  with CompositeAction('Edit Optimization Function'):
108
109  retval_14 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="MinDvh", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[14], RoiName="CTV64_eks_5mm", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=True,
    RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
110
111  # CompositeAction ends
112
113
114  with CompositeAction('Edit Optimization Function'):
115
116  retval_15 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="MinDvh", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[15], RoiName="CTV54_eks_5mm", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=True,
    RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
117
118  # CompositeAction ends
119
120
121  with CompositeAction('Edit Optimization Function'):
122
123  retval_16 = plan.PlanOptimizations[1].EditOptimizationFunction(FunctionType
    ="MinDvh", DoseBasedRoiFunction=plan.PlanOptimizations[1].Objective.
    ConstituentFunctions[16], RoiName="GTV68", IsConstraint=False,
    RestrictAllBeamsIndividually=False, RestrictToBeam=None, IsRobust=True,
    RestrictToBeamSet="rob_ph_dpbn", UseRbeDose=False)
124
125  # CompositeAction ends
126
127
128  plan.PlanOptimizations[1].OptimizationParameters.SaveRobustnessParameters(
    PositionUncertaintyAnterior=0.6, PositionUncertaintyPosterior=0.6,

```

```

    PositionUncertaintySuperior=0.6, PositionUncertaintyInferior=0.6,
    PositionUncertaintyLeft=0.6, PositionUncertaintyRight=0.6,
    DensityUncertainty=0, IndependentBeams=False, ComputeExactScenarioDoses=
    False, NamesOfNonPlanningExaminations=[])
129 plan.PlanOptimizations[2].OptimizationParameters.SaveRobustnessParameters(
    PositionUncertaintyAnterior=0.6, PositionUncertaintyPosterior=0.6,
    PositionUncertaintySuperior=0.6, PositionUncertaintyInferior=0.6,
    PositionUncertaintyLeft=0.6, PositionUncertaintyRight=0.6,
    DensityUncertainty=0, IndependentBeams=False, ComputeExactScenarioDoses=
    False, NamesOfNonPlanningExaminations=[])

```

The script `ROI_indices_Extractor.py` extrats the ROI indicies from RayStation.

```

1 from connect import *
2 import struct
3 from array import array
4
5 path = r'C:\Users\eiraha\Documents\DoseData'
6
7 patient = get_current("Patient")
8 plan = get_current('Plan')
9 beam_set = get_current("BeamSet")
10 case = get_current("Case")
11 fractions = beam_set.FractionationPattern.NumberOfFractions
12 #filename = path + r'\ROI_indices_patient-%s_Doseplan-%s_BeamSet-%s.dat' %(
13     patient.PatientName, plan.Name, str(str(beam_set).split(" ")[1]))
14
15 structure_set = plan.GetStructureSet()
16 roi_names = [rg.OfRoi.Name for rg in structure_set.RoiGeometries if rg.
17     PrimaryShape != None]
18
19 roi_really_wants = ['GTV68', 'PTV68', 'PTV68_eks', 'PTV64', 'PTV64_eks', '
20     PTV64_eks_5mm', 'PTV54_eks_5mm', 'CTV64', 'CTV64_eks', 'CTV54_eks', '
21     CTV64_eks_5mm', 'CTV54_eks_5mm', 'Parotid_R', 'Parotid_L', '
22     Submandibular_L', 'Submandibular_R', 'SpinalCord', 'SpinalCord_PRV', ]
23 #indices_ROIs = {'GTV68': 0, 'PTV68': 0, 'PTV64': 0, 'CTV64': 0, '
24     CTV64_eks': 0, 'CTV54_eks': 0, 'CTV64_eks_5mm': 0, 'CTV54_eks_5mm': :
25     0, 'Parotid_R': 0, 'Parotid_L': 0,
26     'Submandibular_L': 0, 'Submandibular_R': 0, 'SpinalCord': 0}
27
28 #roi_wants = ['Parotid_R']
29 #indices_ROIs = {'Parotid_R': 0}
30
31 roi_not_wants = ['External', 'ext_sub', 'CTV_64.temp', 'CTV_54.temp']
32
33 roi_names = [e for e in roi_names if e not in roi_not_wants]
34
35 for i in roi_names:
36     print i
37     name = list(i)
38     for j in range(0, len(name)):
39         if name[j] == '\\':
40             name[j] = '-'
41         elif name[j] == '/':
42             name[j] = '-'
43         elif name[j] == ' ':
44             name[j] = '-'

```

```

38     elif name[j] == '<':
39         name[j] = 'lt'
40     elif name[j] == '>':
41         name[j] = 'gt'
42 name = "".join(name)
43 print name
44 filename = path + r'\ROIindices\%s.txt' %name
45 file = open(filename, 'w')
46 indices_ROIs = beam_set.FractionDose.GetDoseGridRoi(RoiName=i).
    RoiVolumeDistribution.VoxelIndices
47 length = len(list(indices_ROIs))
48 length2 = len(indices_ROIs)
49 print length, length2
50 #print indices_ROIs
51 file.write('%d \r\n' %length)
52 file.write('%d \r\n' %int(list(indices_ROIs)[-1]))
53 file.write('\r\n')
54 for j in list(indices_ROIs):
55     file.write('%d \r\n' %j)
56 #file.write(struct.pack('%dI' %length, list(indices_ROIs)))
57 #s = beam_set.FractionDose.GetDoseGridRoi(RoiName='Parotid_L').
    RoiVolumeDistribution.VoxelIndices

```