User Manual
DVH Analytics 0.4.4

Dan Cutright

http://dvhanalytics.com

August 17, 2018

Contents

0.1 Introduction

DVH Analytics is a software application to help radiation oncology departments build an in-house database of treatment planning data for the purpose of historical comparisons and statistical analysis.

The application builds a SQL database of DVHs and various planning parameters from DICOM files (i.e., Plan, Structure, Dose). Since the data is extracted directly from DICOM files, we intend to accommodate an array of treatment planning system vendors.

In addition to viewing DVH data, this software provides methods to download queried data, view plan contours, create time-series plots of various planning and dosimetric variables, calculate correlations, and generate multi-variable linear regressions.

The code is built upon these core libraries:

- pydicom: https://github.com/pydicom/pydicom
  Read, modify and write DICOM files with python code

- dicompyler-core: https://github.com/dicompyler/dicompyler-core
  Extensible radiation therapy research platform and viewer for DICOM and DICOM RT

- Bokeh: https://bokeh.pydata.org/en/latest/
  Interactive Web Plotting for Python

0.2 Installation

There are multiple ways to install DVH Analytics, depending on the user's level of comfort with python and SQL. Users familiar with python will find the most flexibility with a source installation, while users new to python or unfamiliar with setting up a postgres SQL database will likely have more success installing via Docker in Section 0.2.2.

0.2.1 Source Code

DVH Analytics currently requires python 2.7 (python 3 has not been tested). Python is available for free here: https://www.python.org/. Alternatively, python installations via Anaconda are also supported: https://anaconda.org/. So not to turn this manual into a crash course for python, user's unfamiliar with installing python packages with pip are encouraged to use the Docker install. DVH Analytics source good can be installed with the following command:

```
$ pip install dvh-analytics
```

Depending on your operating system and user settings, you may need to prepend the above command with sudo to act as the root user:

```
$ sudo pip install dvh-analytics
```

Alternatively, users may use Git or checkout with SVN using https://github.com/cutright/DVH-Analytics.git or download the source code as a zip at https://github.com/cutright/DVH-Analytics/archive/master.zip

---

Note to Linux users
You may need FreeType (https://www.freetype.org/). For example:

```
$ sudo apt-get install libfreetype6-dev
```

Note to Mac users
Make sure Xcode command line tools are installed prior to DVH Analytics installation. If the full Xcode package is installed (from the Mac App Store), enter the following into a terminal window:

```
$ xcode-select –install
```

Note to Windows users
The python package Shapely frequently has issues installing on Windows. If your pip install dvh-analytics failed due to the Shapely package, consider installing Shapely from https://www.lfd.uci.edu/~gohlke/pythonlibs/#shapely.

0.2.2 Docker

Using Docker greatly simplifies the installation process of DVH Analytics. If you're famliar with Docker and prefer to work directly from the image, as opposed to using docker-compose below, it is hosted here: https://hub.docker.com/r/cutright/dvh-analytics/

1. Download our docker-compose.yml file to a location of your choosing. https://raw.githubusercontent.com/cutright/DVH-Analytics-Docker/master/docker-compose.yml

2. Create a folder in your home directory 'DVH-Analytics' or replace any reference to '~/DVH-Analytics/' in docker-compose.yml with any other folder you like. Be sure to keep directories to right of the colon the same.

3. From a terminal or command prompt, navigate to the directory with your docker-compose.yml, type 'docker-compose up'.

4. If you'd like to shutdown these services, simply press CTRL+C with the terminal or command prompt window activated.

0.2.3 Postgres SQL

DVH Analytics depends on psycopg2 (http://initd.org/psycopg/) to connect python to the SQL database, therefore a postgres SQL database is required. If you already have login credentials for a postgres SQL database, you likely only need to update these settings as described in Section 0.6. If you are running DVH Analytics with Docker as opposed to running from source code, you do not need to install postgres SQL.

If you need postgres SQL, here are some options for Mac OS, Ubuntu, and Windows. Installs of postgres SQL on other operating systems are also supported; any installation of postgres SQL should be sufficient with appropriate log in credentials.

Mac OS

The simplest choice is to download the postgres SQL app:
http://postgresapp.com/

1. Open the postgres SQL app.

2. Click Start.

3. Double-click "postgres" with the cylindrical database icon.

4. Type the following in the SQL terminal:

    - create database dvh;
    - \q

Ubuntu

You probably already have PostgreSQL installed, but if you don't, type the following in a terminal:

```
$ sudo apt-get install postgresql postgresql-client postgresql-contrib
$ sudo apt-get libpq-dev
```

From here, it's best to create a new role with same name as your user name for your OS, then create a database; DVH Analytics assumes a database name of 'dvh' by default, but if you choose to use another name, just update your login credentials with the Settings view as described in Section 0.6.

Alternatively, you can install a GUI for admin controls and follow the instructions below in Role and Database Initialization for pgAdmin3

```
$ sudo apt-get install pgadmin3
```

Windows

Download the installer for BigSQL: https://www.bigsql.org/postgresql/installers.jsp/. Be sure to include pgAdmin3 LTS.

Role and Database Initialization for pgAdmin3

1. Launch pgAdmin3 LTS.
2. Right-click localhost and then click connect.
3. Right-click Login Roles and then click New Login Role.
4. Fill in Role name (e.g., dvh), click OK.
5. Right-click Databases then click New Database.
6. Fill in Name (e.g., dvh), set owner to the Role name you just created.
7. Click OK.

0.2.4 Web Browser Requirements

The preferred web browser is Google Chrome. Mozilla Firefox should also work. Microsoft Internet Explorer is currently incompatible with Bokeh, and Apple Safari intermittently loses connection to Bokeh. These requirements are entirely dependent on Bokeh and possibly Tornado (a Bokeh dependency).

0.3 Initialization

There a few things that need to be done before using DVH Analytics. Depending on how it was installed, you may need to update the SQL connections settings. DVH analytics also needs to know where to find incoming DICOM files and where to move them to after import. And finally, there is some post-processing required regarding ROI names and PTV distance and overlap calculations.

0.3.1 Establishing a SQL Connection

If you're using Docker, the Settings web server is already running. Otherwise, launch the Settings web server with one of the following commands:

- From pip install:
  ```
  $ dvh settings
  ```
- From source code:
  ```
  $ bokeh serve dvh/settings.py --port=5008
  ```

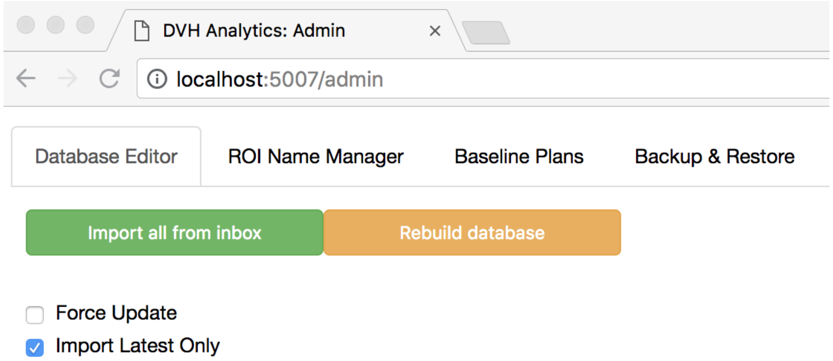
Launch the Settings view by navigating to http://localhost:5008. It's important to note that the values shown in this view are from the perspective of the web servers, so if you're running DVH Analytics with Docker, the directories and ports are Docker side. It is not recommended to edit the directories here if you're using Docker, instead edit the directories in the docker-compse.yml file as described in Section 0.2.2. The purpose of these directories are as follows:

- **inbox** - Place your DICOM files to be processed in this folder.
- **imported** - After DICOM files have been processed, they will be moved to this folder and organized by MRN (or Patient ID).
- **review** - DICOM files to be used for the DVH review feature, as described in Section 0.4.2, should live here.

0.3.2 Importing Data

If you're using Docker, the Admin web server is already running. Otherwise, launch the Admin web server using one of the the following commands, as appropriate:

- From pip install:
  ```
  $ dvh admin
  ```
- From source code:
  ```
  $ bokeh serve dvh/admin.py --port=5007
  ```



☐ Force Update

☑ Import Latest Only

Figure 1: A screen shot of the admin view used for importing data.

Clicking "Import all from inbox" will Process all DICOM files in the inbox directory specified in Section 0.3. Some important things to note:

- *Complete File Set: Plan, Dose, Structure*
  A complete set of RT Dose, RT Structure, and RT Plan connected by the same study instance UID is required by default. If Force Update is checked, importing without the RT Plan will be allowed.

- *Multiple File Sets*
  By default, only the latest DICOM file set will be parsed, based on the operating system's date modified timestamp on the file. If the user would like to import multiple DICOM file sets with the same study instance UID, uncheck 'Import Latest Only'.

- *Other DICOM Files*
  All DICOM files connected with the same Study Instance UID will be catalogued, even if the file is not used.

- *Study Instance UID*
  If the Study Instance UID of a DICOM file set in the inbox already exists in the database, this file set will not be imported unless "Force Update" is checked. This functionality is not recommended as one of the underlying assumptions about the database is that there is one Study Instance UID per course of treatment and the dose grid is the complete composite dose for the course.

- *CTV, GTV, PTV*
  Target volumes are best identified by labeling the ROI in the TPS prior to DICOM export. ROIs are typically labeled as ORGAN by default in many planning systems. DVH Analytics recognizes CTV, GTV, and PTV labels. If multiple PTVs are included in one file set, DVH Analytics will automatically label them as PTV1, PTV2, etc. in ascending order of $D_{95\%}$.

- *ITV*
  An ROI will be labeled as an ITV if the ROI name begins with “ITV”. If your TPS supports the ITV label, it is not necessary to begin your ITV ROI name with “ITV”.

WARNING: Clicking Rebuild database will wipe all data from the SQL data base and reprocess all DICOM files in the “imported” directory. If your database has not been backed-up, this can not be undone.

NOTE TO PINNACLE USERS:
If your treatment planning system is Pinnacle 9.10 and earlier (and perhaps later), prescription information is not stored in the exported DICOM files. However, DVH Analytics does provide a Pinnacle script which will create dummy POIs containing the prescription information. It is recommended that you run this script in Pinnacle for each patient prior to DICOM export. This script can be found here: https://github.com/cutright/DVH-Analytics/tree/master/Pinnacle%20Scripts

Both p3rtp files are needed, however, only the file DVH-Analytics_Create-POIs.Script.p3rtp is executed from user in Pinnacle; this script will automatically call the DVH-Analytics_Create-Rx-POI.Script.p3rtp script.

Alternatively, you can manually create POIs with names in the following format:

- “tx: <site>”
  will allow DVH-Analytics to add this site name to the database upon import.

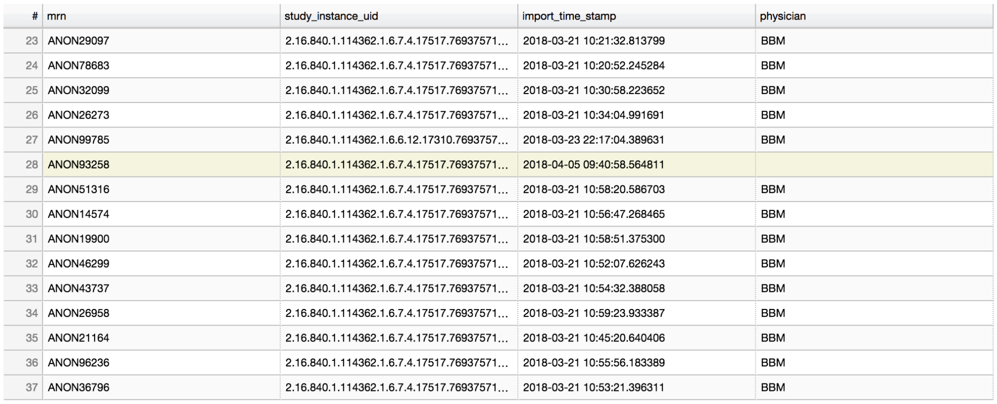
- “rx#: <rx name>: <rx dose in cGy> cGy x <fxs> to <normalization %>%: <normalization method>: <normalization object>”
  where # is rx number starting from 1, this will allow DVH-Analytics to retrieve rx information.

0.3.3 Post-Processing of Data

After some data has been imported, there is some additional mapping and processing needed. A physician ROI map needs to be generated and applied. The PTVs also need to be labeled as such for correct PTV distance and overlap calculations.

ROI Name Mapping

The ROI name mapping process requires the ‘ReferringPhysicianName’ or ‘PhysiciansOfRecord’ DICOM tag to match the physician tag shown in the ROI Map tab of the Admin view. Note that treatment planning systems may use different labels in their user interface (e.g., Attending Physician or Radiation Oncologist). You might need to do some investigating for your particular TPS. If the Physician is not set in the DICOM file, this can be easily edited in the Admin view (localhost:5007). Performing a query with the Plans table, selecting import_time_stamp and physician (with CTRL or COMMAND clicks to select both), and leaving the condition text field empty will yield results similar to those in Figure 2.



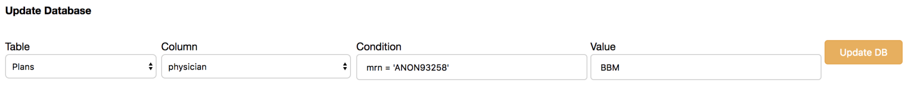
#	mrn	study_instance_uid	import_time_stamp	physician
23	ANON29097	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:21:32.813799	BBM
24	ANON78683	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:20:52.245284	BBM
25	ANON32099	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:30:58.223652	BBM
26	ANON26273	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:34:04.991691	BBM
27	ANON99785	2.16.840.1.114362.1.6.6.12.17310.7693757...	2018-03-23 22:17:04.389631	BBM
28	ANON93258	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-04-05 09:40:58.564811	
29	ANON51316	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:58:20.586703	BBM
30	ANON14574	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:56:47.268465	BBM
31	ANON19900	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:58:51.375300	BBM
32	ANON46299	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:52:07.626243	BBM
33	ANON43737	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:54:32.388058	BBM
34	ANON26958	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:59:23.933387	BBM
35	ANON21164	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:45:20.640406	BBM
36	ANON96236	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:55:56.183389	BBM
37	ANON36796	2.16.840.1.114362.1.6.7.4.17517.76937571...	2018-03-21 10:53:21.396311	BBM

Figure 2: Example case with a missing physician label.

In this example, we'd like to make sure the plan missing a physician is set to 'BBM'. This can be done in the Update Database section of the Admin view as shown in Figure 3. Note that you can double click a field in the table in Figure 2 to copy it's value to the clip board. If this particular patient had multiple plans, each with a different physician, you could define the condition based on study_instance_uid instead. Alternatively, the condition could be:

import_time_stamp > '2018-04-01'

causing all plans imported since April 1, 2018 to be updated. Additional details about this functionality can be found in Section 0.5.1.

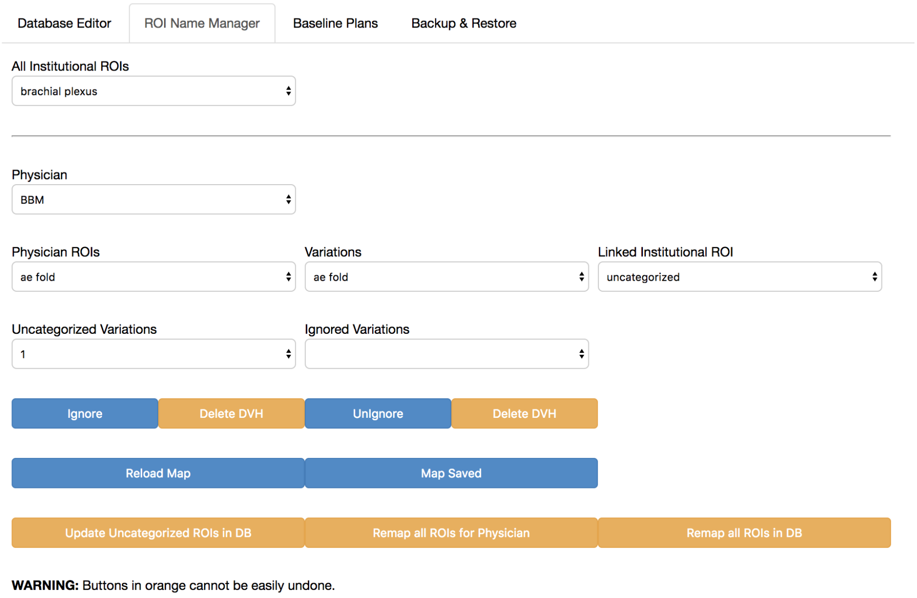


Update Database

Table	Column	Condition	Value
Plans	physician	mrn = 'ANON93258'	BBM

Figure 3: Example case to update physician.

Now that we have ensured all of our plans have a physician assigned, navigate to the ROI Name Mapping tab, as shown in Figure 4. From here, select the correct physician and click "Remap all ROIs for Physician" or simply "Remap all ROIs in DB". Additional details about the features in this tab are explained in Section 0.5.2. Alternatively, you can add a new physician. Note that the physician name will be automatically capitalized; the intent is to use initials here.



All Institutional ROIs: brachial plexus

Physician: BBM

Physician ROIs: ae fold

Variations: ae fold

Linked Institutional ROI: uncategorized

Uncategorized Variations: 1

Ignored Variations:

WARNING: Buttons in orange cannot be easily undone.

Figure 4: ROI Name Manager tab with in the Admin view.

PTV Distance and Overlap Calculations

Assuming that all imported plans have been assigned at least one PTV, the PTV distance and overlap calculations can be performed by clicking "Calc PTV Distances", and upon completion, "Calc PTV Overlap" (the order doesn't matter). The total time to calculate both the distances and overlaps is approximately as long as the initial import time.

If these values are not calculated for a given plan, then this plan will not be included in the data for the Correlations or Regressions tab since the statistical tests used require each category to have the same number of data points.

0.4 Main View: http://localhost:5006

This is the main view for a user after data has been successfully imported and organized. Data cannot be edited or added from this view. There are two persistent text fields labeled Group 1 (Blue) Custom Title and Group 2 (Red) Custom Title. The sole purpose is to let the user notate their own label for Group 1 and 2 that will be displayed on all tabs. Automating this is difficult due to the potential complexity of the query.

0.4.1 Query

You can query by categorical or numerical data, using the categories in Table 1. Note that querying by mrn or study instance uid (UID) is typically not useful for the statistical modules. Also, the Total Plan MU is total MU to be delivered to the patient per the plan (i.e., planned MU times number of fractions, summed across all prescriptions), all other categories are fairly self-explanatory.

Table 1: Searchable Categories

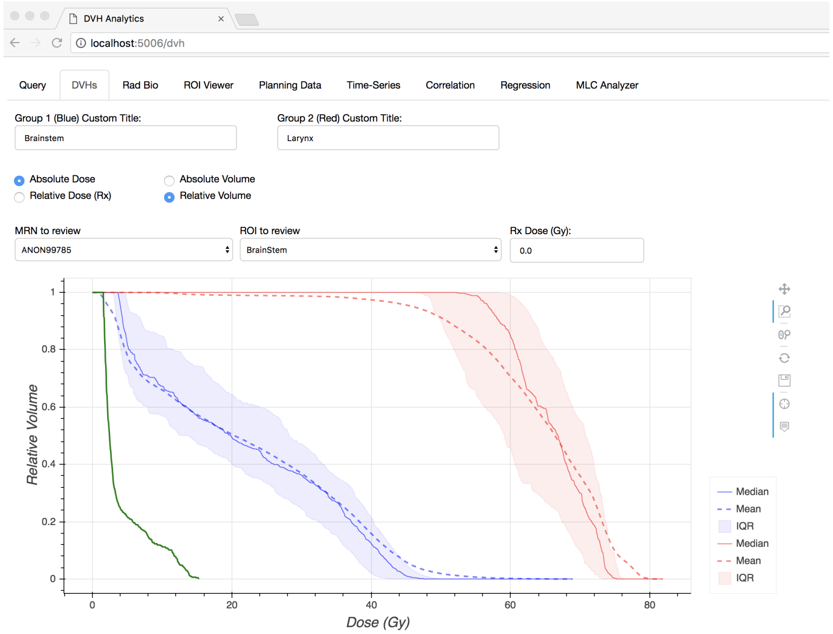
Categorical	**Numerical**
Baseline	Age (at study date)
Beam Type	Beam Dose
Collimator Rotation	Beam Energy
Couch Rotation	Beam MU
Dose Grid Resolution	Birthdate
Gantry Rotation	Collimator Angle
Heterogeneity Correction	Couch Angle
MRN	Distance to PTV
Norm. Method	Fraction Dose
Patient Orientation	Gantry Angle
Patient Sex	Planned Fractions
Physician	ROI Min/Mean/Max Dose
ROI Institutional Category	ROI Volume
ROI Physician Category	Rx Dose
ROI Type	Rx Isodose
Radiation Type	Scan Spots (Protons)
Scan Mode (Protons)	Simulation Date
Treatment Machine Name	SSD (Linac)
Treatment Modality	Total Plan MU
Treatment Site	Treatment time (Brachy)
UID	

The dropdown list for Category 2 will populate based on data in your database. Likewise, the titles of the Min and Max fields in the numerical data query definition will update based on your database, not necessarily correlating to the min and max values of the currently defined query.

The download dropdown button in the top-right applies to the retrieved data once the Query button has returned back to its original green color. From here you can download all data, all data sans DVHs, only DVHs, or anonymized DVHs; all as csv files.

0.4.2 DVHs

DVH plots and selected metrics are shown in this tab. The plot is interactive; you can hover over DVHs for information, you can zoom and pan, and toggle the median, mean, and IQR DVHs. The 'review' DVH is parsed and calculated on the fly from the DICOM files, which are selected by the mrn (PatientID DICOM tag). By default, only the statistical DVHs are displayed (sans min and max). To display retrieved DVHs, simply select them in the table below the plot; you may shift or CTRL click to highlight additional. The statistical DVHs are built dose bin-wise. The IQR spread represents the middle 50% of the retrieved data, as calculated with numpy.



Group 1 (Blue) Custom Title: Brainstem

Group 2 (Red) Custom Title: Larynx

● Absolute Dose
○ Relative Dose (Rx)

○ Absolute Volume
● Relative Volume

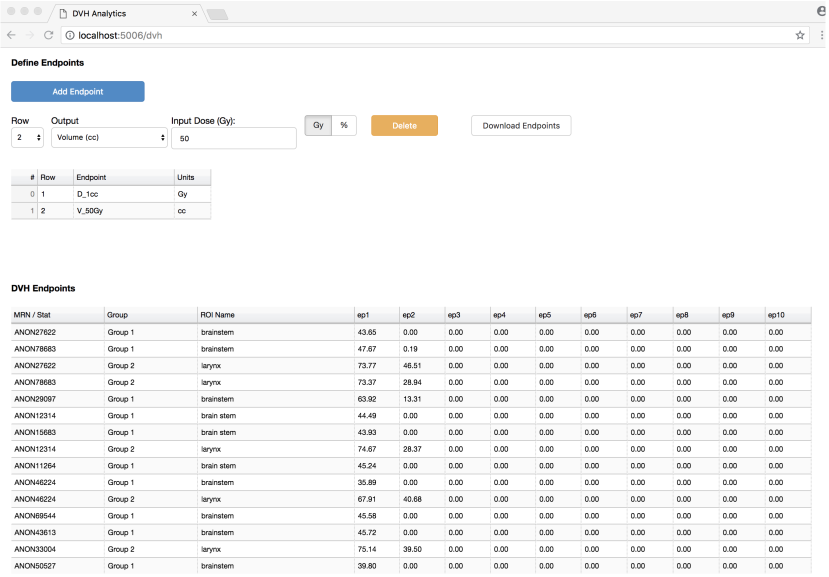
MRN to review: ANON99785

ROI to review: BrainStem

Rx Dose (Gy): 0.0

Figure 5: An example query of brainstem and larynx DVHs with a 'review' DVH of a brainstem plotted from the review directory (not from the database).

Below the DVH table, you can define volumetric and dosimetric endpoints. Note that only the first 10 endpoints are displayed, however, you may define as many as you'd like. All endpoints can be evaluated in other modules and all endpoints will be included in the csv file retrieved when clicking ‘Download Endpoints’.



Define Endpoints

Row: 2
Output: Volume (cc)
Input Dose (Gy): 50

#	Row	Endpoint	Units
0	1	D_1cc	Gy
1	2	V_50Gy	cc

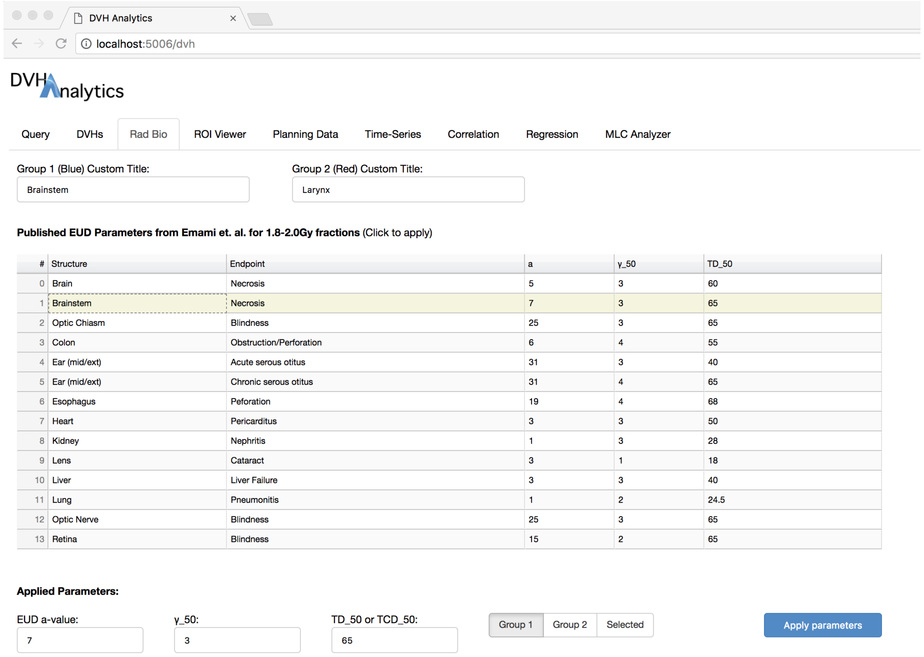
DVH Endpoints

MRN / Stat	Group	ROI Name	ep1	ep2	ep3	ep4	ep5	ep6	ep7	ep8	ep9	ep10
ANON27622	Group 1	brainstem	43.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON78683	Group 1	brainstem	47.67	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON27622	Group 2	larynx	73.77	46.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON78683	Group 2	larynx	73.37	28.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON29097	Group 1	brainstem	63.92	13.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON12314	Group 1	brain stem	44.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON15683	Group 1	brain stem	43.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON12314	Group 2	larynx	74.67	28.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON11264	Group 1	brain stem	45.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON46224	Group 1	brainstem	35.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON46224	Group 2	larynx	67.91	40.68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON69544	Group 1	brainstem	45.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON43613	Group 1	brainstem	45.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON33004	Group 2	larynx	75.14	39.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ANON50527	Group 1	brainstem	39.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 6: Endpoint calculations for the query from Figure 5.

0.4.3 Rad Bio

Equivalent uniform dose (EUD), tumor control probability (TCP), and normal tissue complication probability (NCTP) can be calculated in this module. For convenience, a sample of parameters are provided based on Emami et. al. for 1.8 to 2.0 Gy fractions (Figure 7); selecting a row from this table simply populates the fields below the table. Calculated values are tabulated in the table shown in Figure 8. Note that you can manually select rows with shift or CTRL clicks so that you can apply the parameters to this row with the ‘Selected’ radio button is depressed. Although TCP and NTCP represent different phenomena, they are computed with the same equation when replacing TD_{50} with TCD_{50}, or visa versa. The equation used is based on formalism described by Niemierko.



Group 1 (Blue) Custom Title: Brainstem

Group 2 (Red) Custom Title: Larynx

Published EUD Parameters from Emami et. al. for 1.8-2.0Gy fractions (Click to apply)

#	Structure	Endpoint	a	y_50	TD_50
0	Brain	Necrosis	5	3	60
1	Brainstem	Necrosis	7	3	65
2	Optic Chiasm	Blindness	25	3	65
3	Colon	Obstruction/Perforation	6	4	55
4	Ear (mid/ext)	Acute serous otitus	31	3	40
5	Ear (mid/ext)	Chronic serous otitus	31	4	65
6	Esophagus	Peforation	19	4	68
7	Heart	Pericarditus	3	3	50
8	Kidney	Nephritis	1	3	28
9	Lens	Cataract	3	1	18
10	Liver	Liver Failure	3	3	40
11	Lung	Pneumonitis	1	2	24.5
12	Optic Nerve	Blindness	25	3	65
13	Retina	Blindness	15	2	65

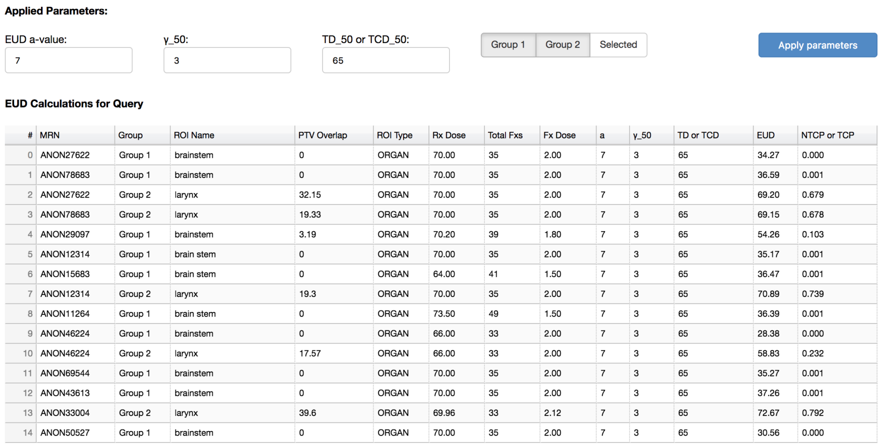
Applied Parameters:

EUD a-value: 7

y_50: 3

TD_50 or TCD_50: 65

Figure 7: Published EUD parameters provided for convenience.



Applied Parameters:

EUD a-value: 7

y_50: 3

TD_50 or TCD_50: 65

EUD Calculations for Query

#	MRN	Group	ROI Name	PTV Overlap	ROI Type	Rx Dose	Total Fxs	Fx Dose	a	y_50	TD or TCD	EUD	NTCP or TCP
0	ANON27622	Group 1	brainstem	0	ORGAN	70.00	35	2.00	7	3	65	34.27	0.000
1	ANON78683	Group 1	brainstem	0	ORGAN	70.00	35	2.00	7	3	65	36.59	0.001
2	ANON27622	Group 2	larynx	32.15	ORGAN	70.00	35	2.00	7	3	65	69.20	0.679
3	ANON78683	Group 2	larynx	19.33	ORGAN	70.00	35	2.00	7	3	65	69.15	0.678
4	ANON29097	Group 1	brainstem	3.19	ORGAN	70.20	39	1.80	7	3	65	54.26	0.103
5	ANON12314	Group 1	brain stem	0	ORGAN	70.00	35	2.00	7	3	65	35.17	0.001
6	ANON15683	Group 1	brain stem	0	ORGAN	64.00	41	1.50	7	3	65	36.47	0.001
7	ANON12314	Group 2	larynx	19.3	ORGAN	70.00	35	2.00	7	3	65	70.89	0.739
8	ANON11264	Group 1	brain stem	0	ORGAN	73.50	49	1.50	7	3	65	36.39	0.001
9	ANON46224	Group 1	brainstem	0	ORGAN	66.00	33	2.00	7	3	65	28.38	0.000
10	ANON46224	Group 2	larynx	17.57	ORGAN	66.00	33	2.00	7	3	65	58.83	0.232
11	ANON69544	Group 1	brainstem	0	ORGAN	70.00	35	2.00	7	3	65	35.27	0.001
12	ANON43613	Group 1	brainstem	0	ORGAN	70.00	35	2.00	7	3	65	37.26	0.001
13	ANON33004	Group 2	larynx	39.6	ORGAN	69.96	33	2.12	7	3	65	72.67	0.792
14	ANON50527	Group 1	brainstem	0	ORGAN	70.00	35	2.00	7	3	65	30.56	0.000

Figure 8: Calculated radbio metrics for the query performed for Figure 5.

0.4.4 ROI Viewer

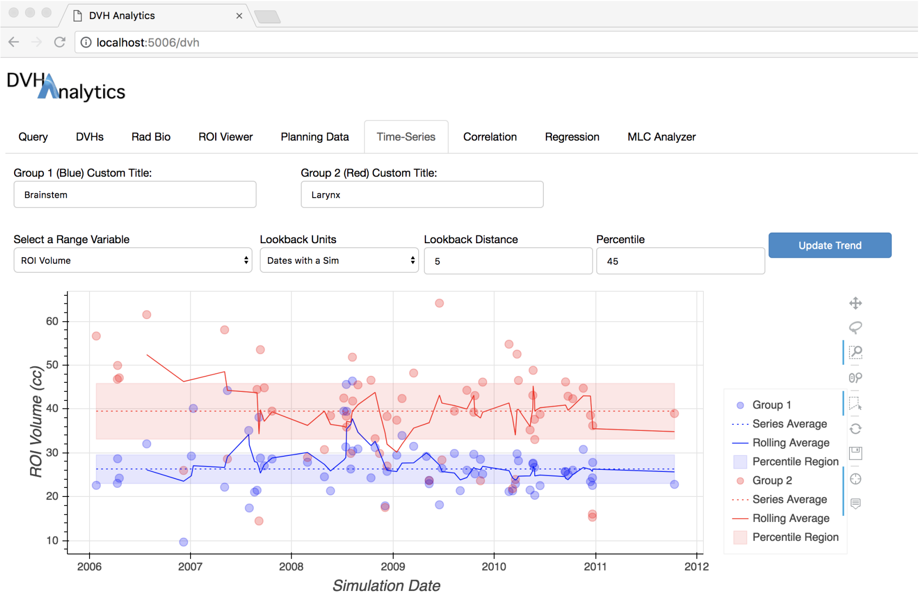
The purpose of the ROI Viewer module is to provide quick transverse visuals of ROIs stored in the SQL database. There are no values calculated in this module. An external or skin ROI will default to ROI 1 if found. Currently, your web browser will still scroll when checking 'Enable Slice Scrolling with Mouse Wheel'. A custom version of Bokeh will need to be developed fix this.

0.4.5 Planning Data

This section is string tabulated data, which can be particular useful if suspcious data is noted from visual in other modules. Clicking a row in any of these tables will call all rows with the same study instance UID, in all of the tables in this tab, to be highlighted.

0.4.6 Time-Series

Any of the quantitative data retrieved in the SQL query can be plotted vs simulation date. Additionally, any defined DVH endpoints or calculated radbio parameters can be plotted. Above the histogram plot in Figure 10, a few p-values are displayed; these are calculated using the normaltest, ttest_ind, and ranksums functions from scipy.stats.



Group 1 (Blue) Custom Title: Brainstem

Group 2 (Red) Custom Title: Larynx

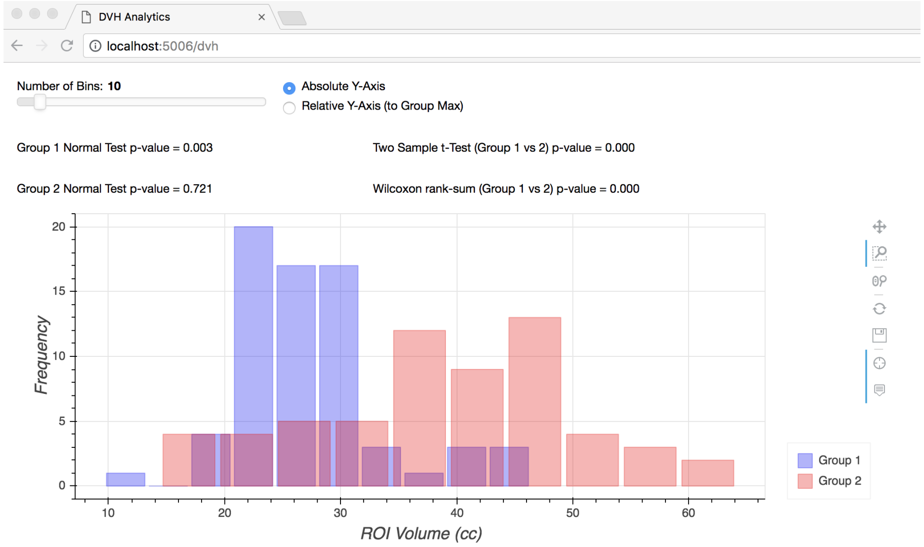
Select a Range Variable: ROI Volume

Lookback Units: Dates with a Sim

Lookback Distance: 5

Percentile: 45

Figure 9: ROI volume vs simulation date for the query performed for Figure 5.



Number of Bins: **10**

● Absolute Y-Axis
○ Relative Y-Axis (to Group Max)

Group 1 Normal Test p-value = 0.003

Two Sample t-Test (Group 1 vs 2) p-value = 0.000

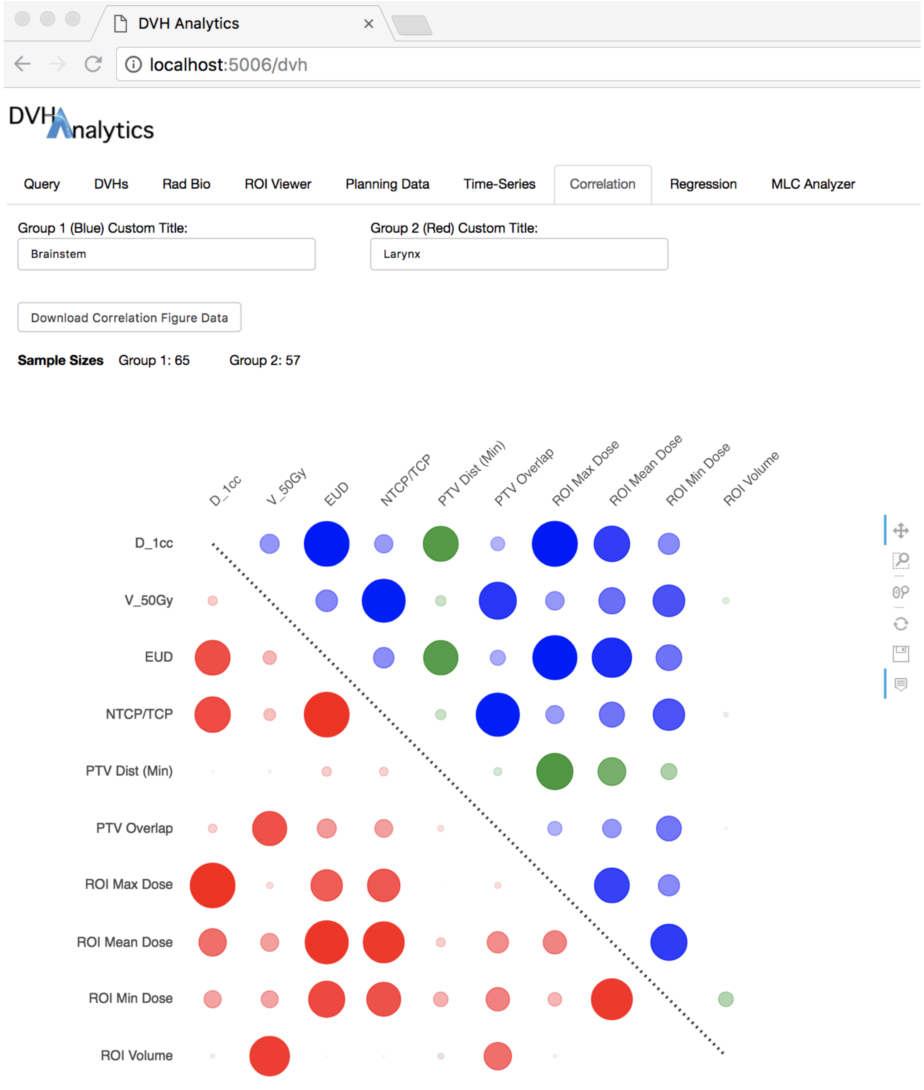
Group 2 Normal Test p-value = 0.721

Wilcoxon rank-sum (Group 1 vs 2) p-value = 0.000

Figure 10: Histograms for the data in Figure 9.

0.4.7 Correlation

This module provides a visual method to explore Pearson R correlations between selected numerical data. It is required that each variable for a given query group have the same number of data points, so be sure you've calculated PTV distances and overlaps from the admin view (see Section 0.3.3). The top-right of this matrix refers to Group 1, the bottom-left referring to Group 2. The radius and opacity of each dot is directly proportional to the magnitude of the Pearson-R value; negative correlations are displayed in green/puprle for blue/red groups (these colors can be changed in the Settings view). Hovering over a dot reveals additional statistical details (p-value for the Pearson test and p-values for normal tests on each variable).



Group 1 (Blue) Custom Title: Brainstem

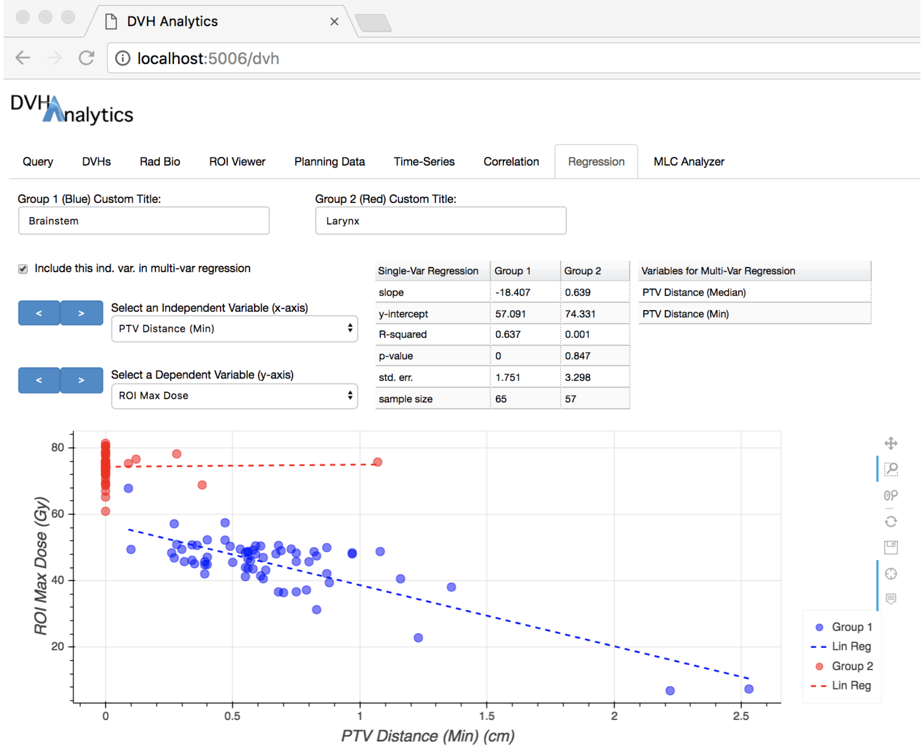
Group 2 (Red) Custom Title: Larynx

Sample Sizes Group 1: 65 Group 2: 57

Figure 11: Correlation matrix for the query performed for Figure 5.

0.4.8 Regression

Multi-variable regression models can be performed using the python package statsmodels. First, select a dependent variable, then increment through independent variables. Results from a single variable linear regression are displayed for the currently selected dependent and independent variables. Check off any independent variables you'd like to include in a multi-variable regression model, then scroll down and click “Perform Multi-Variable Regression”, the results of this regression are tabulated just below this button.



Group 1 (Blue) Custom Title: Brainstem

Group 2 (Red) Custom Title: Larynx

☑ Include this ind. var. in multi-var regression

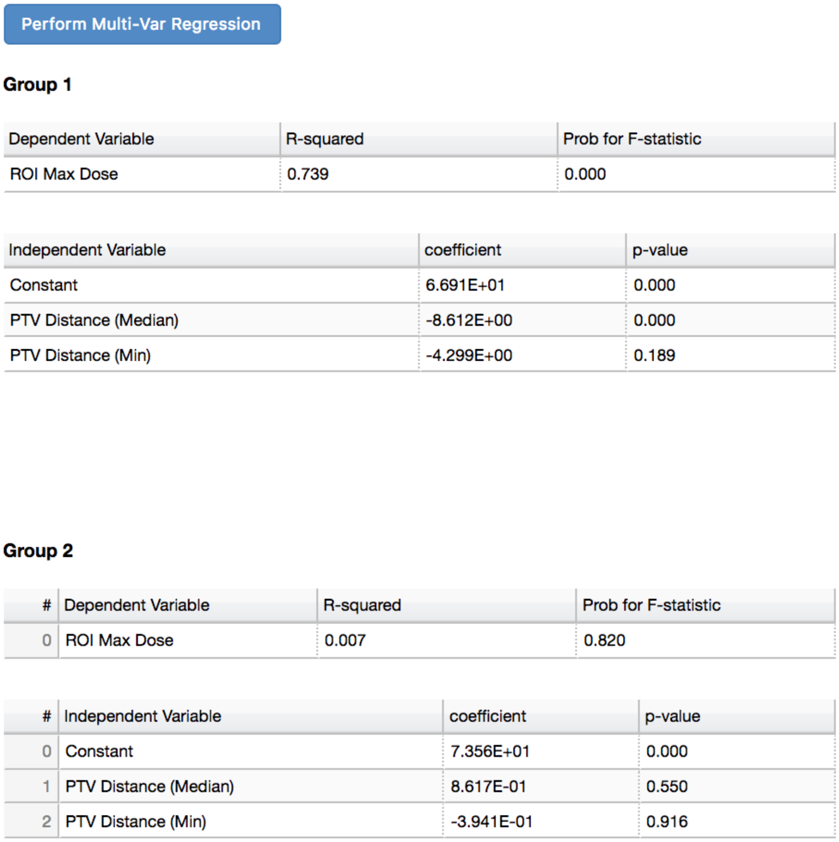
Select an Independent Variable (x-axis): PTV Distance (Min)

Select a Dependent Variable (y-axis): ROI Max Dose

Single-Var Regression	Group 1	Group 2
slope	-18.407	0.639
y-intercept	57.091	74.331
R-squared	0.637	0.001
p-value	0	0.847
std. err.	1.751	3.298
sample size	65	57

Variables for Multi-Var Regression
PTV Distance (Median)
PTV Distance (Min)

Figure 12: Example plot of ROI Max Dose vs Min PTV Distance for the query performed for Figure 5.



Group 1

Dependent Variable	R-squared	Prob for F-statistic
ROI Max Dose	0.739	0.000

Independent Variable	coefficient	p-value
Constant	6.691E+01	0.000
PTV Distance (Median)	-8.612E+00	0.000
PTV Distance (Min)	-4.299E+00	0.189

Group 2

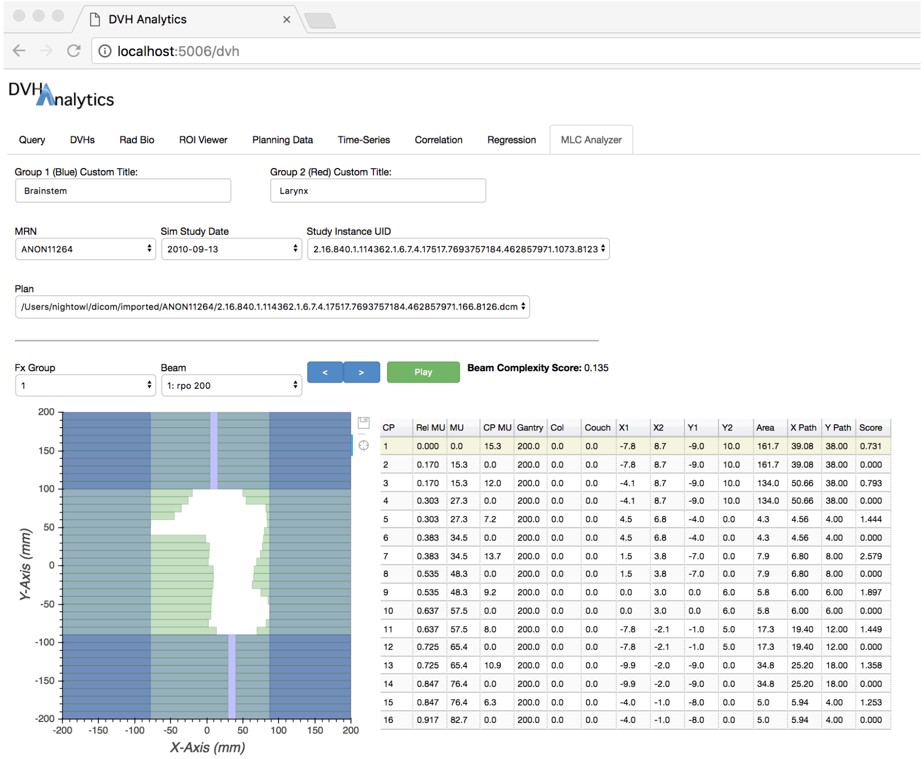
#	Dependent Variable	R-squared	Prob for F-statistic
0	ROI Max Dose	0.007	0.820

#	Independent Variable	coefficient	p-value
0	Constant	7.356E+01	0.000
1	PTV Distance (Median)	8.617E-01	0.550
2	PTV Distance (Min)	-3.941E-01	0.916

Figure 13: Example multi-variable regression of max ROI dose with median and minimum PTV distances for the query performed for Figure 5.

0.4.9 MLC Analyzer

The MLC and jaw positions are displayed here. You may select any mrn included in the query and subsequently the sim study date, study instance UID, and plan file (if there is more than one for a given mrn). Selecting a row in table displayed in Figure 14 will update the aperture visualization. A complexity score is calculated for each control point and the total beam complexity score is displayed above the table. Future versions of DVH Analytics will incorporate these scores into the Correlation and Regression modules. The complexity score is based on Younge et al. Penalization of aperture complexity in inversely planned volumetric modulated arc therapy. Med Phys. 2012;39(11):7160-70.



Group 1 (Blue) Custom Title: Brainstem

Group 2 (Red) Custom Title: Larynx

MRN: ANON11264

Sim Study Date: 2010-09-13

Study Instance UID: 2.16.840.1.114362.1.6.7.4.17517.7693757184.462857971.1073.8123

Plan: /Users/nightowl/dicom/imported/ANON11264/2.16.840.1.114362.1.6.7.4.17517.7693757184.462857971.166.8126.dcm

Fx Group: 1

Beam: 1: rpo 200

Beam Complexity Score: 0.135

CP	Rel MU	MU	CP MU	Gantry	Col	Couch	X1	X2	Y1	Y2	Area	X Path	Y Path	Score
1	0.000	0.0	15.3	200.0	0.0	0.0	-7.8	8.7	-9.0	10.0	161.7	39.08	38.00	0.731
2	0.170	15.3	0.0	200.0	0.0	0.0	-7.8	8.7	-9.0	10.0	161.7	39.08	38.00	0.000
3	0.170	15.3	12.0	200.0	0.0	0.0	-4.1	8.7	-9.0	10.0	134.0	50.66	38.00	0.793
4	0.303	27.3	0.0	200.0	0.0	0.0	-4.1	8.7	-9.0	10.0	134.0	50.66	38.00	0.000
5	0.303	27.3	7.2	200.0	0.0	0.0	4.5	6.8	-4.0	0.0	4.3	4.56	4.00	1.444
6	0.383	34.5	0.0	200.0	0.0	0.0	4.5	6.8	-4.0	0.0	4.3	4.56	4.00	0.000
7	0.383	34.5	13.7	200.0	0.0	0.0	1.5	3.8	-7.0	0.0	7.9	6.80	8.00	2.579
8	0.535	48.3	0.0	200.0	0.0	0.0	1.5	3.8	-7.0	0.0	7.9	6.80	8.00	0.000
9	0.535	48.3	9.2	200.0	0.0	0.0	0.0	3.0	0.0	6.0	5.8	6.00	6.00	1.897
10	0.637	57.5	0.0	200.0	0.0	0.0	0.0	3.0	0.0	6.0	5.8	6.00	6.00	0.000
11	0.637	57.5	8.0	200.0	0.0	0.0	-7.8	-2.1	-1.0	5.0	17.3	19.40	12.00	1.449
12	0.725	65.4	0.0	200.0	0.0	0.0	-7.8	-2.1	-1.0	5.0	17.3	19.40	12.00	0.000
13	0.725	65.4	10.9	200.0	0.0	0.0	-9.9	-2.0	-9.0	0.0	34.8	25.20	18.00	1.358
14	0.847	76.4	0.0	200.0	0.0	0.0	-9.9	-2.0	-9.0	0.0	34.8	25.20	18.00	0.000
15	0.847	76.4	6.3	200.0	0.0	0.0	-4.0	-1.0	-8.0	0.0	5.0	5.94	4.00	1.253
16	0.917	82.7	0.0	200.0	0.0	0.0	-4.0	-1.0	-8.0	0.0	5.0	5.94	4.00	0.000

Figure 14: Example view of MLC and Jaw positions of a plan.

0.5 Admin View: http://localhost:5007

The admin view is used for managing the database. Raw queries may be performed here, but there are no graphics or statistics modules.

0.5.1 Database Editor

The primary function of the Database editor is to import new data into the SQL database. Additionally, there methods provided to query, update/edit, reimport, and delete.

Quick Tip
You can double-click any of the cells in the table at the bottom of the page, after a query, then copy that value to your OS clip board (great for copying a mrn or study instance uid). Although this table is editable, manual changes to this table will not be reflected in the SQL database.

Query Database

This module allows you to directly query the SQL database, the results are populated in the table at the bottom of the page. The SQL database is spit into 5 tables: DVHs, Plans, Rxs, Beams, and DICOM_Files. These tables are connected by the Patient_ID DICOM tag (referred to as mrn here) and the study_instance_uid DICOM tag. The condition needs to be formatted per postgres SQL syntax. If the condition is blank, all data will be retrieved.

Update Database

There are some instances where the database needs to be edited directly. Common examples are:

1. Ensuring proper PTV tags (roi_type)
2. Correcting physician name/intials (physician)
3. Adding simulation date to anonymized data (sim_study_date)

Like querying described above, the condition should be formatted per postgres SQL syntax. DVH Analytics will automatically add single quotes around the value if calling float(value) throws a ValueError in python. Additionally, dates should be entered at YYYY-MM-DD::date; DVH Analytics will appropriately modify this for postgres syntax.

Reimport from DICOM

If data has been accidentally or incorrectly edited, you can reimport directly from DICOM here. Entering a mrn will update the subsequent dropdown menus.

Delete all data with mrn or study_instance_uid

This module will delete all data for a specified mrn or study instance uid across all tables.

Change mrn or study_instance_uid in all tables

This module was implemented to accommodate studies that may want multiple plans for a single patient (e.g., photon vs proton comparision). In this case, one could manually edit the study_instance_uid to 'study_name_photon_patient_N'. So to avoid having multiple plans for one study_instance_uid at any point, it would be recommended to import all photon plans, then edit all study_instance_uid fields; then import all proton plans and edit.

Post Import Calculations

There are a few calculations that need to be performed after proper roi_type assignment for PTVs: PTV distance, PTV overlap volumes, and OAR-PTV centroid distances. For each PTV calculation below, DVH Analytics will generate a combined PTV contour based on all PTVs in a plan. Currently, uncategorized, external, and skin ROIs will not be included in the PTV distance calculations since very large contours could crash DVH Analytics due to memory issues. Finally, recalculating ages was added for ease in the case of editing sim_study_date. All functions in this section can be found in utilities.py

Default Post-Import Calculations
OAR-PTV distance, overlap, and centroids are calculated when this option is selected. These are not calculated at the time of import as proper ROI mapping should be verified prior to these calculations. ROI centroid, spread, and cross-section are automatically calculated at the time of DICOM import as of DVH Analytics 0.4.4. Data imported prior to this version can use the Admin view to calculate these values without the need to reimport from DICOM files.

PTV Distance (*cm*)
The cdist function in scipy.spatial.distance is used to calculate all point-to-point distances between the combined PTV and the OAR. The minimum, median, mean, and maximum values are recorded into the SQL database.

Source code: get_min_distances_to_target(), then take the minimum value

PTV Overlap (cm^3)
The python package, Shapely, is used to calculate the area of the intersection of the combined PTV and OAR, considering only slices involving the PTV. These areas are then each multiplied by the distance between adjacent slices, while assuming the most superior slice thickness is equal the minimum slice thickness for this PTV-OAR pair. This effectively, and albeit crudely, interpolates the delineated PTV, however, it's recommended that all intended slices are contoured or interpolated by your treatment planning system.

Source code: calc_overlap()

ROI Centroid (*mm*)
The geometrical centroid is calculated per slice with Shapely, then the average of the x, y, and z coordinates is calculated, weighted by the respective slice area.
Source code: calc_centroid()

ROI Spread (*cm*)
The spread of a ROI is essentially the dimensions of a rectangular prism that encompasses the ROI, with its sides parallel/perpendicular to the DICOM coordinate axes. This is calculated by difference between the max and min coordinates for each dimension
Source code: calc_spread()

ROI Cross-Section (cm^2)
The cross-sectional area of each slice is calculated with Shapely. The median and maximum values are stored.
Source code: calc_cross_section()

OAR-PTV Centroid Distance (*cm*)
The distance between an OAR and PTV is simply the distance between the stored centroid of the OAR and the centroid of the union of all PTVs. This is calculated in one line with the update function below.
Source code: update_dist_to_ptv_centroids_in_db()

Ages (*years*)
This function is included for convenience if you've imported anonymized data and then add simulation dates afterwards.
Source code: recalculate_ages()

Distance-to-Target Histogram (DTH)
The DTH is calculated and stored in the DVHs SQL table automatically when PTV distances are calculate and updated since this calculation uses the array of all point-to-point distances calculated in PTV distance calculation. This array is converted into a histogram with 0.1mm bin sizes using numpy. As of DVH Analytics 0.4.4, this data is not yet used in any visuals.
Source code: calc_dth()

0.5.2 ROI Name Manager

To address the issue of varying contour/ROI names, a mapping system has been implemented in DVH Analytics with two ROI categories: Institutional and Physician. This provides the flexibility for physicians to have their own list of ROIs, while still allowing for institutional-wide queries. When adding a new physician to the map, the default list of institutional ROIs are automatically populated as a starting point, but may be edited or deleted. Physicians ROIs that do not correspond to any particular ROI in the Institutional ROI list may be labeled as uncategorized.

The example in Figure 15 illustrates a large variation in possible labels in cervicothoracic esophagus. As plans are imported, any ROI that is not recognized will be assigned the ROI type of uncategorized. From ROI Name Manager, you may select an uncategorized ROI in the database for a given physician, and then choose to delete the ROI, label it as ignored (to keep it out of the uncategorized list so that it's not in your uncategorized list in a later review), or add this variation to the ROI map so that future instances will be caught. Note that you must remap the the ROIs for the physician or database in order to apply the change of the ROI map; editing the ROI map does not apply any changes in the database until then.

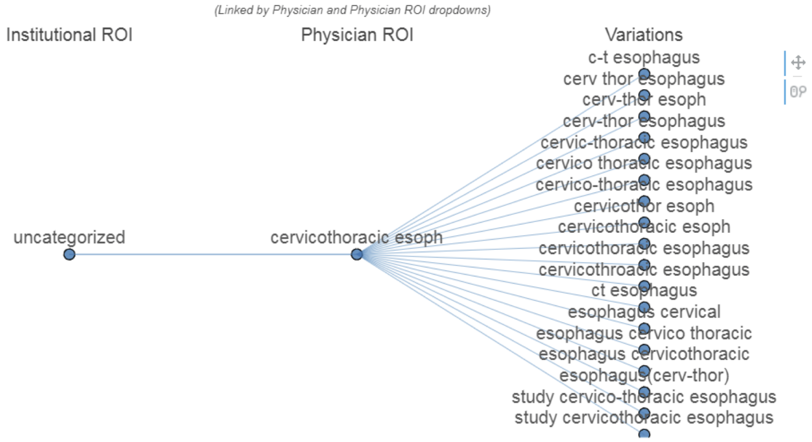

(Linked by Physician and Physician ROI dropdowns)
Institutional ROI
Physician ROI
Variations
uncategorized
cervicothoracic esoph
c-t esophagus
cerv thor esophagus
cerv-thor esoph
cerv-thor esophagus
cervic-thoracic esophagus
cervico thoracic esophagus
cervico-thoracic esophagus
cervicothor esoph
cervicothoracic esoph
cervicothoracic esophagus
cervicothroacic esophagus
ct esophagus
esophagus cervical
esophagus cervico thoracic
esophagus cervicothoracic
esophagus(cerv-thor)
study cervico-thoracic esophagus
study cervicothoracic esophagus

Figure 15: An example ROI map for cervicothoracic esophagus.

0.5.3 Baseline Plans

The intent of this module is to allow the user to label plans as baseline or 'good' plans. Ideally, all plans will be pushed to this database, but perhaps only a subset of them may deemed helpful for statistical analysis. It is left entirely to the user to decide how to use this label. An alternative use could be to consider baseline line status as an approval after the imported data has been validated.

0.5.4 Backup & Restore

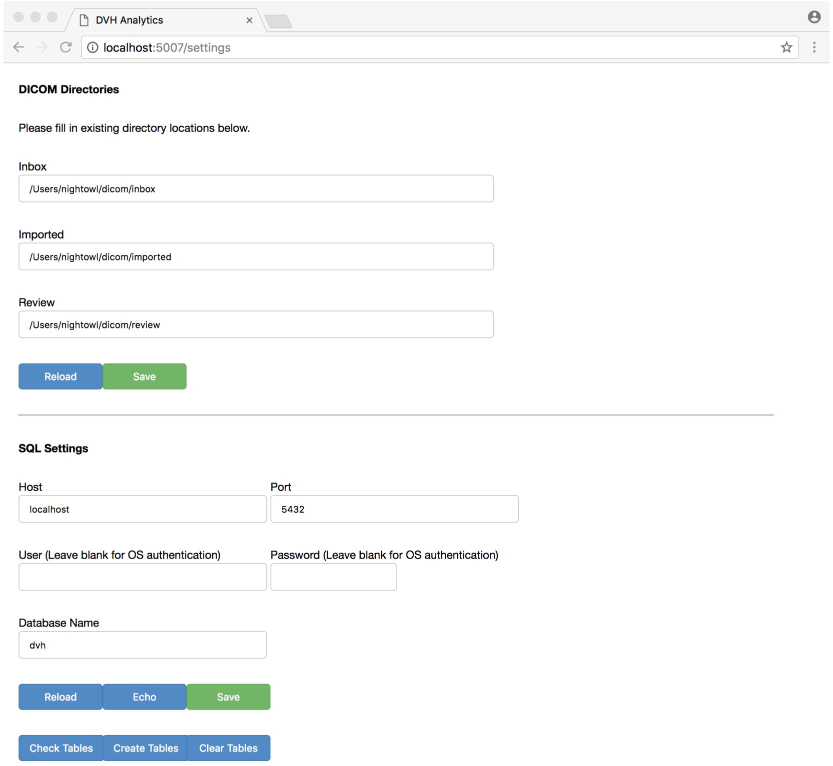
The database backup feature in DVH Analytics is dependent on the pg_dump command from the psql tools. This may not be installed if you're using the posgres app for Mac and have not installed the command line tools. Separately, user preferences, such as import settings, SQL connection settings, and ROI maps can be backed up.

0.6 Settings View: http://localhost:5008

The purpose of this interface is to:

- Define directories for storing and accessing DICOM files
- Define login credentials for the SQL database
- Customize a variety of visual options

This view does not explicitly require a valid SQL connection and may be the only available view until valid credentials are saved within this view. This page allows the user to test the SQL connection with an 'echo' as well as delete or create tables in the database, however, launching the admin view with valid SQL credentials will automatically create the necessary tables if they don't already exist.



DICOM Directories

Please fill in existing directory locations below.

Inbox: /Users/nightowl/dicom/inbox

Imported: /Users/nightowl/dicom/imported

Review: /Users/nightowl/dicom/review

SQL Settings

Host: localhost

Port: 5432

User (Leave blank for OS authentication):

Password (Leave blank for OS authentication):

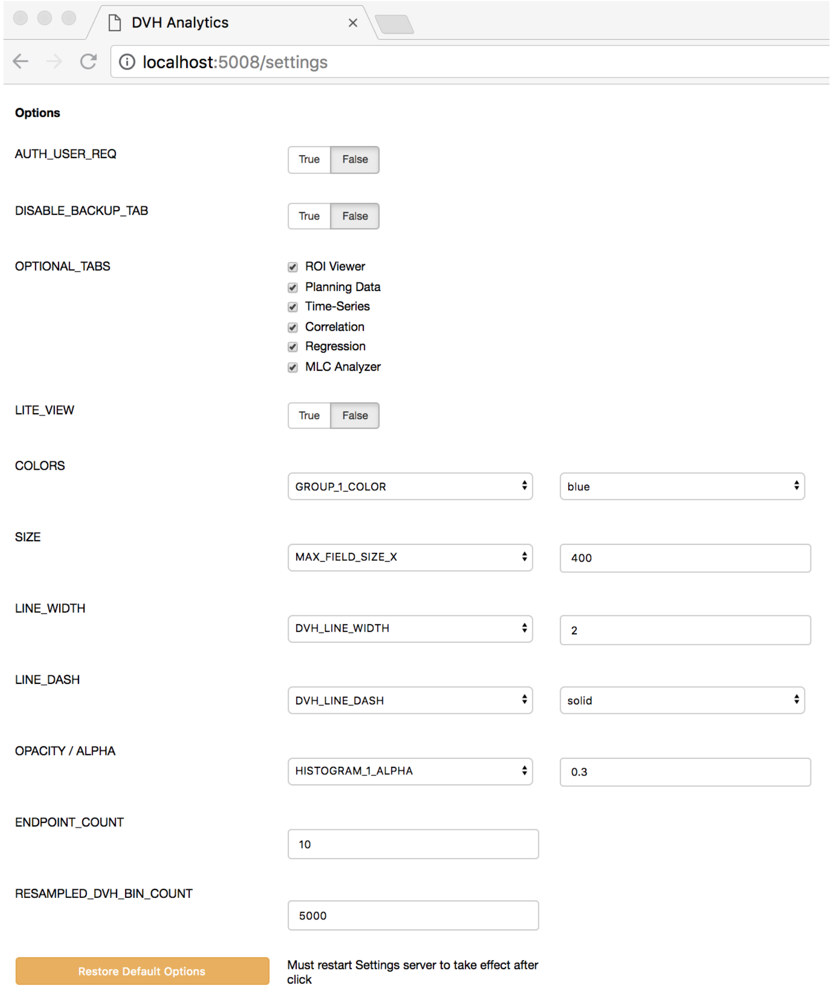
Database Name: dvh

Figure 16: Example directory and SQL connection settings.

A variety of visual options are available:

- AUTH_USER_REQ
  Turning on this feature will require web-users to login. By default, any login credentials will gain access. You'll need to edit the file auth.py in the source directory. This file is already populated with code that may get you started with LDAP, however, it is entirely up to the user to design a secure log in method.
- DISABLE_BACKUP_TAB
  Set this to true if you'd like to hide the Backup Tab in the Admin view.
- OPTIONAL_TABS
  If you find there are tabs you do not use, disabling them will speed up the launch speed of the Main view.
- LITE_VIEW
  Enabling this option will strip all graphics and statistical modules from the Main view. Users with VERY large datasets may find this useful if all they are interested in is querying and downloading data to a csv file. Endpoints may still be calculted as well, but you'll need to download the csv file to view them.
- COLORS
  Group, MLC, and jaw colors can be set here.
- SIZE
  Maximum linac field size, font sizes, and circle sizes in Time-Series and Regression tabs can be edited here.
- LINE_WIDTH
  Adjust the size of line widths of any plotted lines.
- LINE_DASH
  Change line styles on any of the plots.
- OPACITY / ALPHA
  Adjust the opacity of various semi-transparent features in any of the plots (except Correlation).
- ENDPOINT_COUNT
  Editing the number of columns in a Bokeh table requires refreshing the entire layout, which can be fairly slow. To avoid this, the number of columns displayed for the DVH endpoints is static but editable. Note that this is simply capping the number of DVH endpoints displayed in the table, if more endpoints are calculated, they will still be available in Time-Series, Correlation, Regression, and downloaded data.

- RESAMPLED_DVH_BIN_COUNT
  Displaying DVHs with dose relative to prescription dose requires resampling the stored DVH data (which is binned by 1cGy intervals). Plotting DVHs with relative dose will set the number of bins between 0 to prescription dose with the bin count defined here.



Options

AUTH_USER_REQ: True [✓ False]

DISABLE_BACKUP_TAB: True [✓ False]

OPTIONAL_TABS:
☑ ROI Viewer
☑ Planning Data
☑ Time-Series
☑ Correlation
☑ Regression
☑ MLC Analyzer

LITE_VIEW: True [✓ False]

COLORS: GROUP_1_COLOR blue

SIZE: MAX_FIELD_SIZE_X 400

LINE_WIDTH: DVH_LINE_WIDTH 2

LINE_DASH: DVH_LINE_DASH solid

OPACITY / ALPHA: HISTOGRAM_1_ALPHA 0.3

ENDPOINT_COUNT: 10

RESAMPLED_DVH_BIN_COUNT: 5000

Must restart Settings server to take effect after click

Figure 17: Default visual options.

0.7 Docker

The docker image is hosted on docker.com here:
https://hub.docker.com/r/cutright/dvh-analytics/

The code for this docker image, along with the docker-compose file, is hosted on GitHub here:
http://docker.dvhanalytics.com

Docker essentially creates some isolated space on your computer to run a premade image; somewhat like a virtual machine. In order for your local computer to interact with this container, you need to let Docker know how this information may flow. Since DVH Analytics depends on multiple services, we can use docker-compose to spin-up multiple images as well as map ports and directories; this is accomplishedin the docker-compose.yml file.

If you'd like to customize your environment, simply edit the left side of the colon under any of the port or volume sections in docker-compose.yml. For example, if you are already running a different postgres service on port 5432 and you'd like to switch to port 5433, and separately you'd also like to map your postgres database to another location on your local drive, you can change the docker-compose.yml as inidcated with the bolded text below. This process of editing items to left of the colon extends to the remaining services in docker-compose.yml.

```
db:
    image: postgres
    restart: always
    ports:
        - 5433:5432
    volumes:
        - /some-new-directory/:/var/lib/postgresql/data
```

0.8 SQL Tools

Users who wish to use DVH Analytics simple for the database, and build their own analysis tools, can find some useful functionality in sql_connector.py. Below is a list of functions available with some examples.

0.8.1 Opening a connection

Simply create a DVH_SQL object, and a SQL connection is created using the stored login credentials (as shown in the Settings view).

```
$ cnx = DVH_SQL()
```

0.8.2 Closing a connection

Though not explicitly needed, as the connection will automatically close when the script or python console is closed, you can manually close the SQL connection with close().

```
$ cnx.close()
```

0.8.3 Generic Query

You can perform a generic query by providing a string with the text you would provide a SQL console. The return of this funtion will be a list of rows, each row being a list of columns.

```
$ query_return = cnx.query_generic(command_string)
```

0.8.4 Running a SQL file

If you create a plain text file with SQL commands, simply provide the absolute file path to execute_file(), and the file will be executed line-by-line.

```
$ cnx.execute_file(abs_file_path)
```

0.8.5 Query

For ease, this function can perform a query with minimal SQL syntax. The table name may be either 'DVHs', 'Plans', 'Beams', 'Rxs', or 'DICOM_Files'. The return column string is the column name, or a comma-separated list of column names (a string, not a python list). The list of available columns is exhaustive, but are listed in a drop-down menu in the admin view. These columns names can also be found in /preferences/create_sql_tables.sql. The optional condition string should follow postgresql syntax. You can also use the get_column_names function().

```
$ query_return = cnx.query(table_name, return_col_str)
$ query_return = cnx.query(table_name, return_col_str, *condition_str)
```

Example:

```
$ query_return = cnx.query(‘Plans’, ‘mrn, tx_site’, “physician = ‘BBM’”)
```

0.8.6 Get Column Names

This function will return a list of all of the column names for a given table.

```
$ column_names = cnx.get_column_names(table_name)
```

0.8.7 Update Data

The update() function will replace all data in a column with the provided value where the condition is met. The example below will set the physician to BBM for all plans imported after the given date.

```
$ cnx.update(table_name, column, value, condition)
```

Example:

```
$ cnx.update(‘Plans’, ‘physician’, ‘BBM’, “import_time_stamp > ‘8/1/2018’::date”)
```

0.8.8 Change MRN or Study Instance UID

All MRNs or study instance UIDs in all tables can be changed at once with change_mrn() or change_uid().

```
$ cnx.change_mrn(old, new)
```

0.8.9 Get Min or Max Value

The minimum or maximum values of a given column can be queried using either get_min_value() or get_max_value().

```
$ min_value = cnx.get_min_value(table_name, column_name)
```

0.8.10 Get Unique Values

To get a list of all unique values in a given column, call get_unique_values(), with an optional condition. The example below returns a list of all physicians.

```
$ values = cnx.get_unique_values(table_name, column, *condition)
```

Example:

```
$ physician_list = cnx.get_unique_values('Plans', 'physician')
```

0.8.11 Delete All Data

The following function will drop all tables, and then recreate empty tables. This cannot be undone, unless you've backed up your database.

```
$ cnx.reinitialize_database()
```

0.9 SQL to Python

The easiest way to pull data from all columns of a specific table, without explicitly using the SQL Tools, is with the QuerySQL object in sql_to_python.py. Simply provide the table name and condition.

```
$ my_data = QuerySQL(table_name='table_name', condition=condition_string)
```

Example:

```
$ BBM = QuerySQL(table_name='Plans', condition="physician = 'BBM'")
```

You can access the data by appending a period and the column name from the SQL table.

Examples:

```
$ mrns = BBM.mrn
$ tx_sites = BBM.tx_site
```

0.10 Analysis Tools

While sql_connector.py provides tools to directly query and interact with the SQL database, analysis_tools.py provides simple tools to query the DVHs table and return a python object with DVH data already parsed. There are also functions to calculate some common metrics.

0.10.1 Retrieving a set of DVHs

Simply create a DVH object with a list of study instance uids and an option condition string, and a SQL connection is created using the stored login credentials (as shown in the Settings view). All DVHs are are 1-D numpy arrays with 1 *cGy* bins, unless otherwise noted.

```
$ my_dvhs = DVH(uid=list_of_uids, condition=condition_string)
```

Example:

```
$ my_dvhs = DVH(uid=['uid1', 'uid2', 'uid3'], condition="roi_type like 'PTV%'")
```

0.10.2 Accessing Column Data

All column data from the DVHs SQL table is stored in the DVH() object. Simply append a period and column name of the SQL table to access. Each object in this fashion is a list of items of whatever type is returned from the SQL query. The order of these lists is consistent across all variables; that is, all the data for the patient associated with my_dvhs.mrn[i] is accessed with index *i* for all variables.

Examples:

```
$ mrns = my_dvhs.mrn
$ volumes = my_dvhs.volume
```

0.10.3 Accessing DVHs

The dvh_string column is automatically parsed into 1-D numpy arrays and stored in a 2-D array. This is designed this way for easy calculations across the entire DVH dataset.

Examples:

```
$ dvh_1 = my_dvhs.dvh[:,0]
$ dvh_2 = my_dvhs.dvh[:,1]
$ dvh_3 = my_dvhs.dvh[:,2]
```

0.10.4 Calculate a Statistical DVH

Statistical DVHs are calculated bin-wise. The percentile, or stat method specified, of each bin is calculated independently using numpy. The returned object is a 1-D numpy array. The stat_type may be any of the following: 'min', 'mean', 'median', 'max', or 'std'. The default type is mean. For get_stat_dvh only: by default, values are absolute dose in units of *Gy* and relative volume, but volume_scale and relative_scale may be defined as either 'absolute' or 'relative'.

This functionality will be added to get_percentile_dvh in a later release.

```
$ percentile_dvh = my_dvhs.get_percentile_dvh(percentile)
$ stat_dvh = my_dvhs.get_stat_dvh(stat_type=type)
```

Examples:

```
$ upper_quartile_dvh = my_dvhs.get_percentile_dvh(75)
$ mean_dvh = my_dvhs.get_stat_dvh()
$ min_dvh = my_dvhs.get_stat_dvh(stat_type=‘min’)
```

0.10.5 Calculate a Standard Set of Statistical DVHs

To easily calculate the min, mean, median, max, q1, and q3 DVHs, you can simply call get_standard_stat_dvh(), with optional volume_scale and dose_scale specified as absolute or relative. This functions returns a dictionary of DVHs

```
$ dvh_set = my_dvhs.get_standard_stat_dvh()
```

Examples:

```
$ upper_quartile_dvh = dvh_set[‘q3’]
$ mean_dvh = dvh_set[‘mean’]
```

0.10.6 Get Dose to Volume for DVH Object

All dosimetric endpoints (*e.g.*, D_{2cc}) can be calculated using get_dose_to_volume(). By default, values are absolute in units of Gy or cm^3, but volume_scale and relative_scale may be defined as either ‘absolute’ or ‘relative’. A list of floats are returned.

```
$ dose_eps = my_dvhs.get_dose_to_volume(volume)
```

Example:

```
$ dose_eps = my_dvhs.get_dose_to_volume(2)
```

0.10.7 Get Volume of Dose for DVH Object

All volumetric endpoints (*e.g.*, V_{20Gy}) can be calculated using get_volume_of_dose(). By default, values are absolute in units of Gy or cm^3, but volume_scale and relative_scale may be defined as either ‘absolute’ or ‘relative’. A list of floats are returned.

```
$ volume_eps = my_dvhs.get_volume_of_dose(dose)
```

Example:

```
$ volume_eps = my_dvhs.get_volume_of_dose(20, volume_scale=‘relative’)
```

0.10.8 Absolute Volume DVHs

All DVHs can be turned into absolute volume DVHs with dvhs_to_abs_vol().

```
$ abs_volume_dvhs = my_dvhs.dvhs_to_abs_vol(dvhs)
```

Examples:

```
$ abs_volume_dvhs = my_dvhs.dvhs_to_abs_vol(my_dvhs.dvh)
```

0.11 Security

Obviously, data intended for an application like this may be sensitive and require HIPPA compliance. The end user is entirely liable for setting up an appropriately secure environment. Bokeh provides some help with regards to a reverse proxy, so that HTTPS may be implemented.
https://bokeh.pydata.org/en/latest/docs/user_guide/server.html

Your SQL DB password is stored as plain text in preferences/sql_connection.cnf. This is not ideal, but if OS user authentication is implemented, you don't need a password. Alternatively, you could change permissions on this file so only you or root can access it, just be sure to run your bokeh serve with sudo if needed.

0.11.1 Network/Web Access

By default, Bokeh servers restrict access to localhost. However, users may choose to open up external access. If you append '–allow-websocket-origin=¡IP:port¿' to the command launch the bokeh server, any user with access to this IP and port will have access to the bokeh server. Depending on your environment (either from pip install or running from source directory), one of the following will open up access outside of localhost:

```
$ dvh run --allow-websocket-origin=XX.XXX.X.X:PORT
$ bokeh serve dvh --allow-websocket-origin=XX.XXX.X.X:PORT
```

Given the complexity and variation of hospital network setups, we can't really streamline this process for other users. It's recommended that users work with their IT or IS team and start with this data flow diagram.

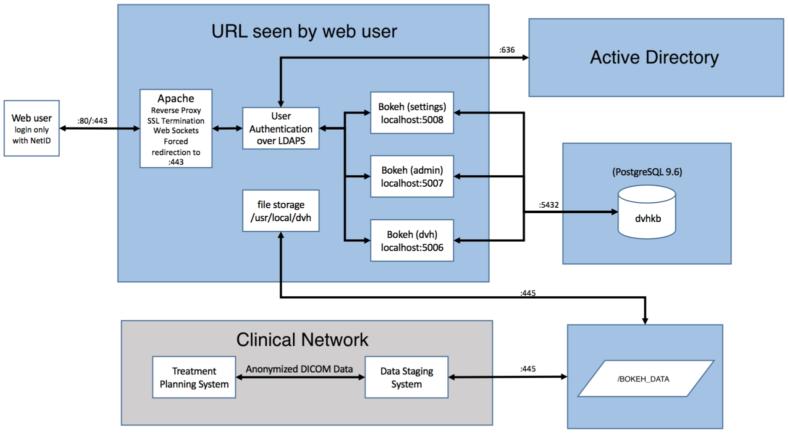


Figure 18: An example network setup of DVH Analytics for network access including LDAP authentication and HTTPS. This implementation was running source code on a CentOS virtual server. Credit: Marc Broxton

0.11.2 HTTPS

If you wish to allow DVH Analytics to be accessible outside of the computer/server running the software, and you have not anonymized your patient data, it's very important to implement HTTPS so that network activity is encrypted between the web user and the DVH Analytics server. DVH Analytics does not provide this service, but Bokeh does give some documentation of potential implementations. We have successfully implemented DVH Analytics behind an Apache server with a reverse proxy as illustrated in Figure 18.

0.11.3 LDAP

DVH Analytics does provide a loose framework for authentication using LDAP. This feature can be turned on in the Settings view. If set to True, user name and password fields will be provided to the user when accessing any of the Bokeh servers in this app. These credentials will be passed to the check_credentials function in auth.py. By default, this function simply returns True. The end user must supply their own code for authentication. An example code using python-ldap is provided in auth.py, but requires some editing for each end user's own implementation. And please remember, you need to setup HTTPS on your own for this authentication to make any sense, otherwise user name and passwords will not be encrypted in between the web-user and the Bokeh server.

0.12 License

BSD 3-Clause License

Copyright (c) 2017, Dan Cutright All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the dicompyler-core library:

Copyright (c) 2009-2017 Aditya Panchal and dicompyler-core contributors

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The name of Aditya Panchal may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the pydicom library:

License file for pydicom, a pure-python DICOM library

Copyright (c) 2008-2017 Darcy Mason and pydicom contributors

Except for portions outlined below, pydicom is released under an MIT license:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of pydicom (private dictionary file(s)) were generated from the private dictionary of the GDCM library, released under the following license:

Program: GDCM (Grassroots DICOM). A DICOM library Module: http://gdcm.sourceforge.net/Copyright

Copyright (c) 2006-2010 Mathieu Malaterre Copyright (c) 1993-2005 CREATIS (CREATIS = Centre de Recherche et d'Applications en Traitement de l'Image) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither name of Mathieu Malaterre, or CREATIS, nor the names of any contributors (CNRS, INSERM, UCB, Universite Lyon I), may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the NumPy library:

Copyright 2005-2017, NumPy Developers. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ?AS IS? AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the MATPLOTLIB library:

LICENSE AGREEMENT FOR MATPLOTLIB 1.1.0 ————————

1. This LICENSE AGREEMENT is between John D. Hunter ("JDH"), and the Individual or Organization ("Licensee") accessing and otherwise using matplotlib software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, JDH hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use matplotlib 1.1.0 alone or in any derivative version, provided, however, that JDH's License Agreement and JDH's notice of copyright, i.e., "Copyright (c) 2002-2009 John D. Hunter; All Rights Reserved" are retained in matplotlib 1.1.0 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates matplotlib 1.1.0 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to matplotlib 1.1.0.

4. JDH is making matplotlib 1.1.0 available to Licensee on an "AS IS" basis. JDH MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, JDH MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF MATPLOTLIB 1.1.0 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. JDH SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF MATPLOTLIB 1.1.0 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING MATPLOTLIB 1.1.0, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between JDH and Licensee. This License Agreement does not grant permission to use JDH trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using matplotlib 1.1.0, Licensee agrees to be bound by the terms and conditions of this License Agreement.

DVH Analytics uses the Six library:

Copyright (c) 2010-2017 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ”AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

DVH Analytics uses the Bokeh library:

Copyright (c) 2012, Anaconda, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Anaconda nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the Statsmodels library:

Copyright (C) 2006, Jonathan E. Taylor All rights reserved.

Copyright (c) 2006-2008 Scipy Developers. All rights reserved.

Copyright (c) 2009-2012 Statsmodels Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

a. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. b. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. c. Neither the name of Statsmodels nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL STATSMODELS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the SciPy library:

Copyright 2001, 2002 Enthought, Inc. All rights reserved.

Copyright 2003-2013 SciPy Developers. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Enthought nor the names of the SciPy Developers may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ?AS IS? AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

DVH Analytics uses the psycopgy2 library:

psycopg2 and the LGPL psycopg2 is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

psycopg2 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

In addition, as a special exception, the copyright holders give permission to link this program with the OpenSSL library (or with modified versions of OpenSSL that use the same license as OpenSSL), and distribute linked combinations including the two.

You must obey the GNU Lesser General Public License in all respects for all of the code used other than OpenSSL. If you modify file(s) with this exception, you may extend this exception to your version of the file(s), but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version. If you delete this exception statement from all source files in the program, then also delete it here.

You should have received a copy of the GNU Lesser General Public License along with psycopg2 (see the doc directory.) If not, see http://www.gnu.org/licenses/.

DVH Analytics uses the python-future library:

Copyright (c) 2013-2016 Python Charmers Pty Ltd, Australia

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ”Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ”AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS

IN THE SOFTWARE.

0.13 Acknowledgements

DVH Analytics was largely developed at Northwestern Memorial Hospital by Dan Cutright (coding, design, and UI development) and Mahesh Gopalakrishnan (design). Software and programming guidance was provided by Adit Panchal (Northwestern Medicine Chicago Proton Center). Statistical methods for plan quality assessments were provided by Arkajyoti Roy (Bowling Green State University). This software is under continuing development at Brown University, Northwestern Memorial Hospital, and University of Texas, San Antonio.

We would like to extend our sincere gratitude to:

- David Juliano for general coding and database consultation
- Marc Broxton from the Northwestern University IT team for designing and implementing a reliable and secure network environment for DVH Analytics
- Dr. Bharat B. Mittal for providing a large, consistent database of patients used for development as well as his clinical expertise and perspective

If you find this software useful as part of your research and you'd like to reference us, please refer to our manuscript in the Journal of Applied Clinical Medical Physics.

https://doi.org/10.1002/acm2.12401
D. Cutright, M. Gopalakrishnan, A. Roy, A. Panchal, B.B. Mittal. ”DVH Analytics: A DVH database for clinicians and researchers.” Journal of Applied Clinical Medical Physics (2018).