User Manual and Technical Reference for DVHA v0.7.2 (Draft)

Dan Cutright
http://dvhanalytics.com

February 11, 2020

Contents

1 Introduction

DVH Analytics (DVHA) 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). In addition to viewing DVH data, this software provides methods to download queried data, create time-series plots, calculate correlations, generate multi-variable linear & machine learning regressions, and generate control charts.

The code is built with these libraries:

- wxPython Phoenix: https://github.com/wxWidgets/Phoenix
  Build a native GUI on Windows, Mac, or Unix systems
- 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
- Shapely: https://github.com/Toblerity/Shapely
  Manipulation and analysis of geometric objects
- Scikit-Learn: https://github.com/scikit-learn/scikit-learn
  Machine Learning in Python
- SQLite3: https://www.sqlite.org/
  Light-weight file-based SQL databse
- PostgreSQL: https://www.postgresql.org
  Powerful, open-source network-based SQL database
  psycopg2 used for python/sql interaction http://initd.org/psycopg/
- PyInstaller: http://pyinstaller.org
  Freeze python packages
- Data science packages:
  NumPy: http://numpy.org
  SciPy: https://scipy.org
  Statsmodels: https://github.com/statsmodels/statsmodels
  regressors: https://pypi.org/project/regressors/

1.1 Database Type

Menu Navigation: Settings > Database Connection

DVH Analytics supports both SQLite and PostgreSQL. Most users will prefer SQLite due to ease. On first application launch, a SQLite database will be created in ~/Apps/dvh_analytics/data with the file name dvha.db. Deleting DVHA or running a different version of DVHA will not delete or remove your database.

1.1.1 SQLite

Advantages:

- No admin rights needed on your computer
- No need to figure out how to make user logins and databases in SQL
- Easier to share your database - just zip (and encrypt), send to colleague

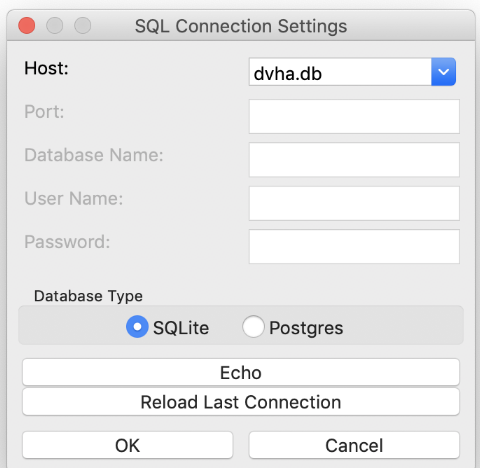


Figure 1: SQLite database connetion settings.

The dropdown box for host will list any files in ~/Apps/dvh_analytics/data using a .db extension. You may type in any absolute address you like if you'd prefer your database live on another drive. You can delete this database simply by deleting the .db file in Windows Explorer.

1.1.2 PostgreSQL

Advantages:

- Supports multiple instances of DVHA accessing the same database at once
- Database may be housed remotely (just need the accessible IP address)
- Supports user login and password

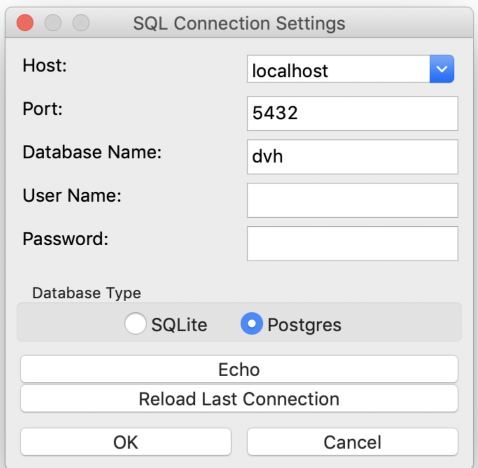


Figure 2: PostgreSQL figure.

The user name provided must have read and write privileges to a database that is already created. DVHA will create all of the necessary tables for you. PostgreSQL is not interchangeable with other SQL flavors, such as MySQL or Microsoft SQL. If your PostgreSQL database supports peer authentication, leave the user name and password blank.

⚠ **SECURITY NOTE**:
Although the user interface does not display your password, your SQL login credentials are stored in a text (pickle) file in ~/Apps/dvh_analytics/preferences/.options.

1.2 User Folder Data

Opening DVHA will initialize the following directories and files if they do not exist.

```
~/Apps/dvh_analytics
 ├── data
 │    ├── backup ...  Use is not yet implemented.
 │    ├── imported ...  Imported DICOM files will be sent here.
 │    ├── inbox ...  DICOM files to be imported.
 │    ├── models ...  Storage for multi-variable linear (.mvr)
 │    │               and machine learning (.mlr) model files .
 │    ├── review ...  Use is not yet implemented.              .
 │    ├── temp ...  wxPython in MS Windows requires that html
 │    │             content be loaded from a file.  Bokeh
 │    │             plots are stored here.
 │    └── dvha.db ...  Default SQLite database file.
 └── preferences
      ├── import_settings.txt ...  Simple text file containing user
      │                            preferences for alternate locations of
      │                            inbox, imported, and review directories.
      ├── institutional.roi ...  Simple text file containing institutional
      │                          ROI names separates by new lines.
      ├── physician_BBM.roi ...  A default physician roi file.
      ├── sql_connection.cnf ...  Deprecated as of v0.6.7.  SQL connection
      │                           information now lives in .options.
      ├── .options ...  A pickle file of the Options class from
      │                 dvha.options.
      └── .options_checksum ...  A checksum of .options preventing manual
                                 edits of .options.
```

1.3 DICOM Files

DVHA requires the following DICOM-RT files:

- Plan
- Structure
- Dose

Any other DICOM files that share the same StudyInstanceUID will be cataloged for potential use in later versions of DVHA. Importing of DICOM files from the following treatment planning systems have been reported to work as of the DVHA version for this document:

- BrainLab iPlan 4.5
- Elekta Monaco 5.0
- Elekta XiO (version unknown)
- Philips Pinnacle^3 8.0m $\rightarrow$ 9.10, 16.2
- RaySearch RayStation 5.0 $\rightarrow$ 6.1
- Varian Eclipse 11, 15

1.4 Troubleshooting & Bug Reporting

Menu Navigation: Help > Report an Issue
If you believe you have identified a bug, please visit our GitHub page to see if it has already been reported or submit a new issue.

2 User Settings

Menu Navigation: Settings ⟩ Preferences

Users may change default inbox and imported DICOM directories and various visual features of plots: Colors of lines and shaded regions, sizes of points, line widths, line styles, and alpha (opacity). The DVHs stored in the SQL database are always with 1cGy bin widths, however, users may increase the bin width of the queried DVHs.

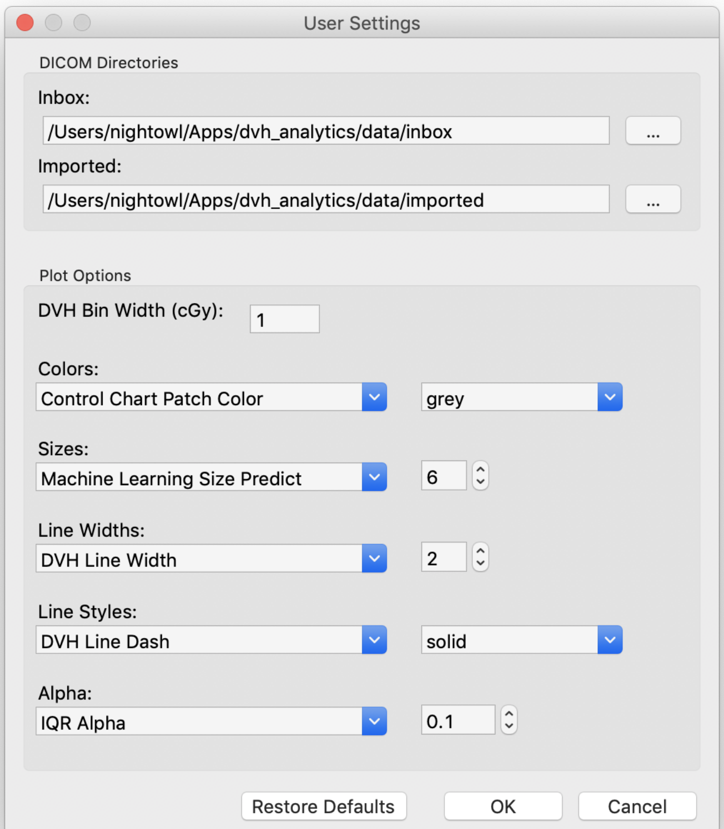


Figure 3: User settings window.

3 Importing

Menu Navigation: File ❯ Import DICOM

After selecting a directory, DVHA will search for any DICOM files, match them with appropriate UIDs, and build a file tree including plan, structure, and dose files; other matching DICOM files will be cataloged. There are three warning levels:

- **RED**: Must resolve to import.
  Occurs if StudyInstanceUID exists in database or file set does not contain dose, structure, and dose.
- **ORANGE**: Strongly suggested to resolve.
  Occurs if missing physician or PTV assignments.
- **YELLOW**: Suggested to resolve.
  Occurs if Sim Study Date is missing. Data will be unavailable for Time Series and Control Charts if left unresolved.

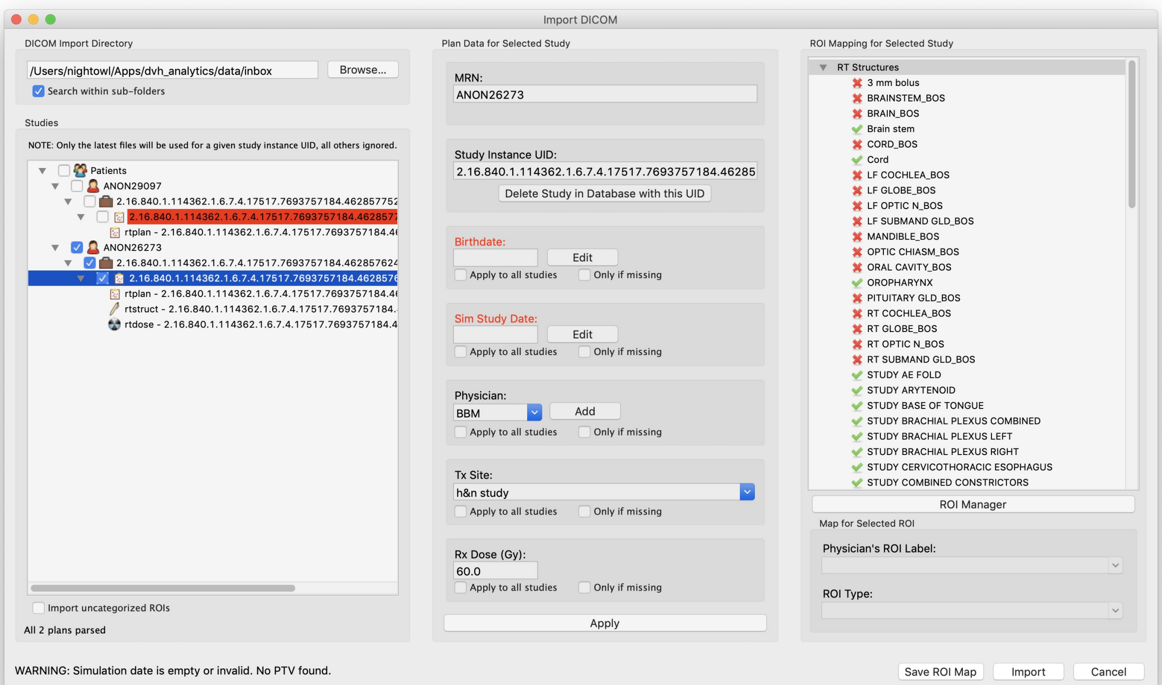


Figure 4: DVHA DICOM Import screen.

3.1 Assumptions and Requirements

Automatic dose summations

DVHA will automatically sum all dose grids attached to a StudyInstanceUID. This is a direct summation (with interpolation as necessary); no fusion is performed. This feature is intended for treatment planning systems that cannot export a composite dose grid of all plans on a single simulation image.

Editable tags

Default values are populated primarily based on DICOM tags. You may edit some of these values on the import screen, or through the Database Administrator in §5.

DICOM File matching

The file tree is built based on StudyInstanceUID (0020, 000d) and ReferencedSOPInstanceUID (0008, 1155), assuming a hierarchy similar to that used in Pinnacle: Planning Image → Plan → Structure and Dose. Other planning systems behave differently in that they may assign only one structure set to multiple plans (*e.g.*, Eclipse). If a structure set is not provided per plan, DVHA will still assign the correct structure file matched using its ReferencedSOPInstanceUID.

Duplicate StudyInstanceUIDs

We encourage you to import only one DICOM dose per planning image file set. This is accomplished by preventing users from importing a dicom file set with a StudyInstanceUID that already exists in your database. This may be overridden by manually changing the StudyInstanceUID to be imported on the import screen (or the previously imported plan in Database Administrator).

Tumor/Target ROIs

If an ROI name points to a physician ROI name of GTV, CTV, ITV, or PTV, the ROI Type will be matched. If the ROI name is already one of these types or one of these types appended by a number, the ROI Type will be set based on this name (e.g. PTV, PTV2, or PTV 1 are all auto-set to PTV ROI Type). Otherwise the ROI type assignment defaults to RTROIInterpretedType (3006, 00a4). While the ROI name will be preserved, ROI type will always number PTVs in increasing order of $D_{95\%}$.

Tx Site

The Tx Site parameter auto populates from the RTPlanLabel (300a, 0002), unless a POI exists in your structure file in the format "tx site: [custom site name]". The drop down is populated with all Tx Site names currently in your database. You may also type in a new Tx Site name.

Rx Dose

The Rx Dose is populated from TargetPrescriptionDose (300a, 0026), unless a POI exists in your structure file in the format "rx#: [rx name]: [rx dose] cGy x [fxs] to [normalization]%: [normalization method]: [normalization object]" where # is rx number starting from 1.

Special POIs

The special POIs referenced in the previous two paragraphs were originally built to accommodate versions of Pinnacle that did not store prescription information in its exported DICOM files. Pinnacle scripts to automatically create these POIs are provided here: https://github.com/cutright/DVH-Analytics/tree/master/resources/Pinnacle%20Scripts

3.2 ROI Map

Menu Navigation: Settings 〉 ROI Map

DVHA employs a map of ROI synonyms with two categories: Institutional and Physician. The map is case-insensitive and underscores are interpreted as spaces. The purpose of having two categories is to allow physicians to create their own ROI map using Physician ROIs. Institutional ROIs are intended to map all Physician ROIs into a common nomenclature for the entire database.

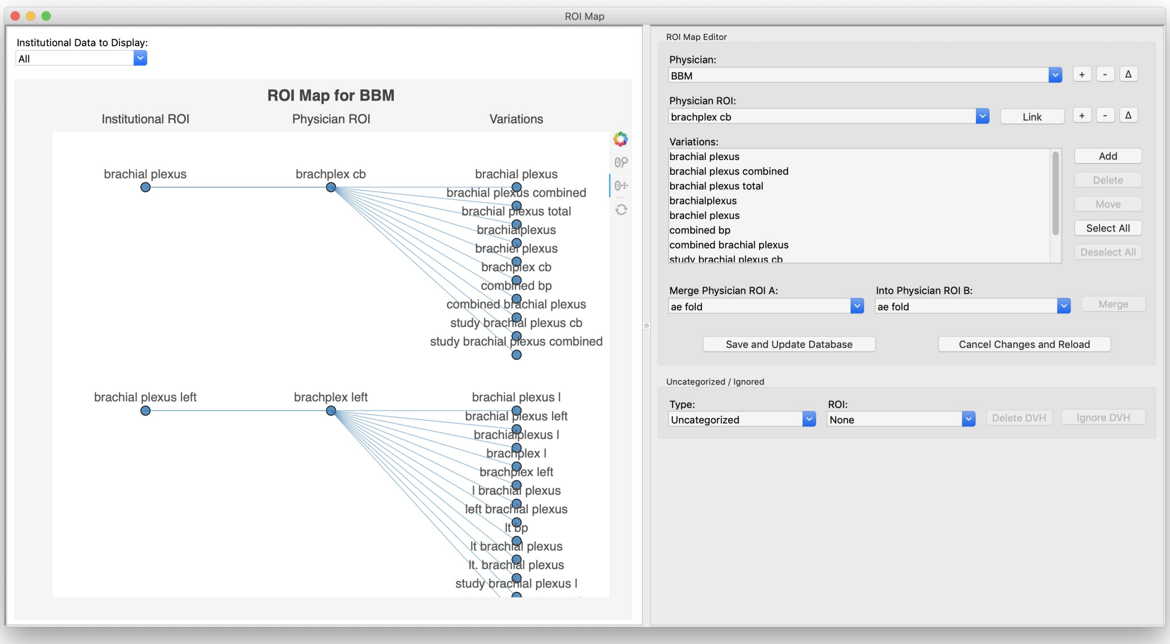


Figure 5: ROI Map editor.

3.3 Import Calculations

3.3.1 DVH Calculations

All DVHs are calculated using dicompyler-core. Min, mean, and max doses, as well as ROI volume, are also stored in the database so they may be included in query design.

3.3.2 Geometric Calculations

Custom code using SciPy and Shapely libraries was written to calculate the following features.

ROI Spread: the max separation of surface points in each dimension,

$$S_i = |max(P_i) - min(P_i)|\,, \tag{1}$$

where is i is the x, y, or z dimension and P is all surface points of the ROI.

ROI Surface Area: calculated using perimeter and area calculations from Shapely for each slice of the ROI. These calculations have not been validated.

ROI Centroid: the weighted-average of ROI slice centroids, weighted by slice area. Slice centroids and areas per Shapely.

ROI Cross-Section: the median and max of all ROI slice areas per Shapely. These values refer to cross-sections in the axial plane.

PTV Union: if multiple PTVs are provided, DVHA will generate a composite PTV using Shapely one slice at a time. This composite PTV structure is used for cross-section, spread, surface area, volume, min dose, and max dose values in the Plans SQL table, as well as PTV overlap, PTV distances, and DTH described below.

PTV Overlap Volume: with the composite PTV structure, a boolean-AND operator is used per slice of the ROI. The volume is calculated by the weighted-sum of each slice area, weighted by the slice thickness, where the slice thickness is defined by the z-distance between adjacent slices.

PTV Distances and DTH: all point-to-point distances between two sets of points are calculated using `scipy.spatial.distance.cdist`. In this case, set 1 contains all surface points of the ROI and set 2 contains all surface points of the composite PTV. The min, mean, median, and max of these values are stored. The DTH is a histogram of all values. All distances are positive, rounded to the nearest $0.01cm$. If memory cannot be allocated for this calculation, then the set of points will be iteratively and uniformly reduced by a factor of 10 until there are no more than 3,000 points.

3.3.3 Plan and Beam Complexity

DVHA will calculate complexity scores based on Younge et. al. A complexity score C_i, for the i^{th} control point is calculated as

$$C_i = MU\frac{c_1 x + c_2 y}{A}, \tag{2}$$

where MU is the monitor units for the control point, x & y are aperture perimeters in their respective dimensions, A is aperture area, and c_1 & c_2 are weighting coefficients . The beam complexity is the sum of all control point complexity scores. The plan complexity is the weighted-sum of beam complexities, weighted by beam MU. The weighting coefficients are currently set to unity.

4 Main View

Query design is split into categorical and numerical filters. If multiple instances of the same filter are added, an OR operator is assumed between all instances of this filter. Users may construct a second query (*i.e.*, Group 2). This feature is ideally suited for studies such as photon vs proton, IMRT vs VMAT, small bladder vs large bladder, etc.

Categorical filters:
Users select one of the values from Table 1 for Category 1; the available values in Category 2 are based on the current database.

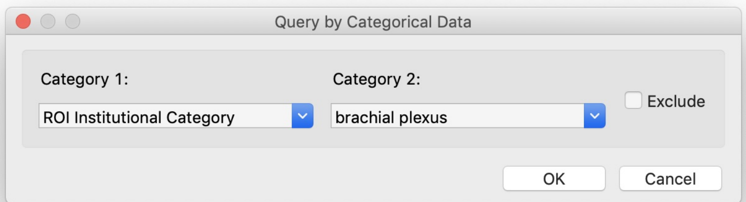


Figure 6: Categorical filter dialog window

Numerical filters:
Users select one of the values from Table 2; min and max values auto-populate based on the entire database. That is, these values do not "know" about other filters in your query design. Individual numerical filters assume an inclusive between operator for the given min and max values.

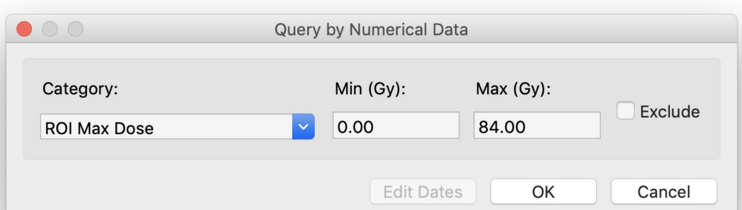


Figure 7: Numerical filter dialog window

Table 1: Categorical filters

Baseline	Normalization	ROI Physician Category
Beam Type	Patient Orientation	ROI Type
Collimator Rotation Direction	Patient Sex	Scan Mode
Couch Rotation Direction	Physician	Treatment Machine
Dose Grid Resolution	Protocol	Tx Modality
Gantry Rotation Direction	Radiation Type	Tx Site
Heterogeneity Correction	ROI Institutional Category	UID
MRN	ROI Name	

Table 2: Numerical filters

Age	Beam Perimeter Y (Max)	PTV Spread X
Beam Area (Max)	Beam Perimeter Y (Mean)	PTV Spread Y
Beam Area (Mean)	Beam Perimeter Y (Median)	PTV Spread Z
Beam Area (Median)	Beam Perimeter Y (Min)	PTV Surface Area
Beam Area (Min)	Birth Date	PTV Volume
Beam Complexity (Max)	Control Point Count	ROI Cross-Section Max
Beam Complexity (Mean)	Control Point MU (Max)	ROI Cross-Section Median
Beam Complexity (Median)	Control Point MU (Mean)	ROI Max Dose
Beam Complexity (Min)	Control Point MU (Median)	ROI Mean Dose
Beam Dose	Control Point MU (Min)	ROI Min Dose
Beam Energy Max	Fraction Dose	ROI Spread X
Beam Energy Min	Plan Complexity	ROI Spread Y
Beam MU	Planned Fractions	ROI Spread Z
Beam MU per control point	PTV Cross-Section Max	ROI Surface Area
Beam MU per deg	PTV Cross-Section Median	ROI Volume
Beam Perimeter (Max)	PTV Distance (Centroids)	Rx Dose
Beam Perimeter (Mean)	PTV Distance (Max)	Rx Isodose
Beam Perimeter (Median)	PTV Distance (Mean)	Scan Spots
Beam Perimeter (Min)	PTV Distance (Median)	Simulation Date
Beam Perimeter X (Max)	PTV Distance (Min)	SSD
Beam Perimeter X (Mean)	PTV Max Dose	Total Plan MU
Beam Perimeter X (Median)	PTV Min Dose	Toxicity Grade
Beam Perimeter X (Min)	PTV Overlap	

4.1 DVHs

This view displays the queried DVH data, including statistical DVHs calculated bin-wise. Selecting one or many DVHs in the table below will toggle the visibility of the selected DVH in the plot. Users may export raw DVH data from the menu bar. Data > Export > Data to csv

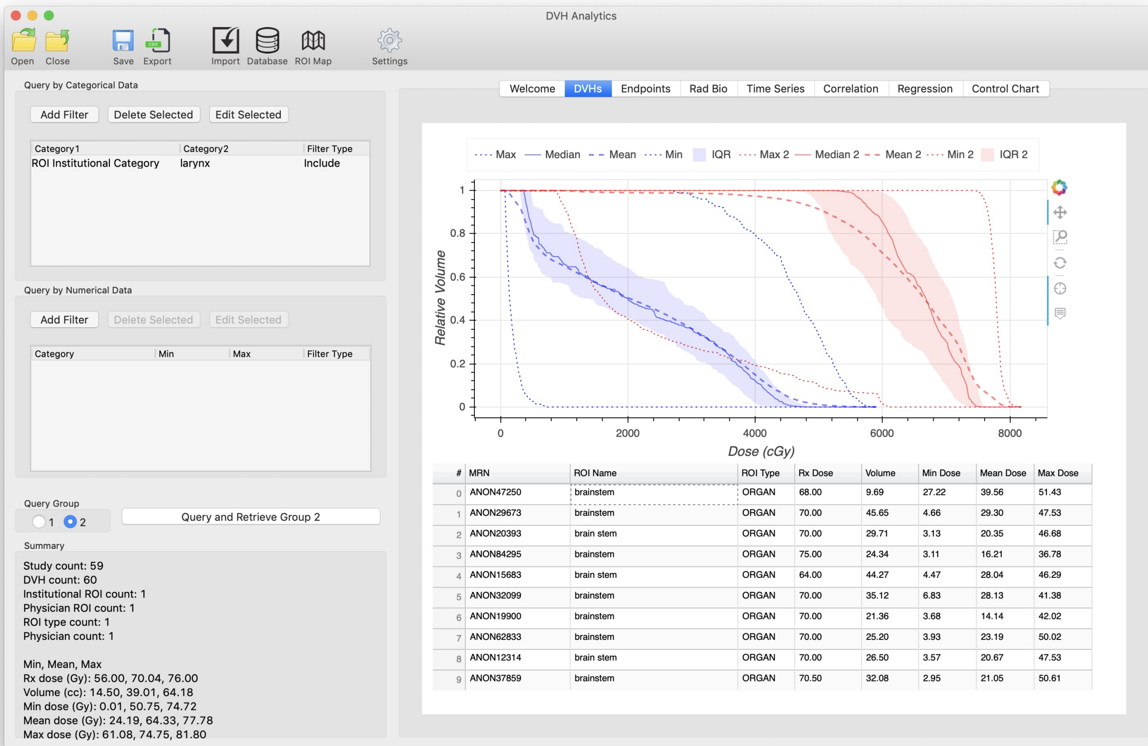


Figure 8: DVHs view.

4.2 Endpoints and RadBio

From these two tabs, users can calculate individual DVH points, EUD, or $TCP/NTCP$ for the entire queried data set. Once values are calculated, they propagate to the Time Series, Correlation, Regression, and Control Chart tabs. EUD is calculated as indicated in Eq 3, $NTCP$ in Eq 4. TCP is calculated with the same formula by replacing TD_{50} with TCP_{50}.

$$EUD = \left(\sum_i v_i \cdot D_i^a\right)^{1/a}, \tag{3}$$

$$NTCP = \frac{1}{1 + \left(\frac{TD_{50}}{EUD}\right)^{4\gamma_{50}}} \tag{4}$$

4.3 Time Series

This module displays numerical data vs Simulation Date. A rolling average is plotted based on the user specified lookback distance. Normality tests, two sample t-test, and Wilcoxon ranksum are calculated (if applicable) using `ttest_ind`, `ranksums`, and `normaltest` from `scipy.stats`.

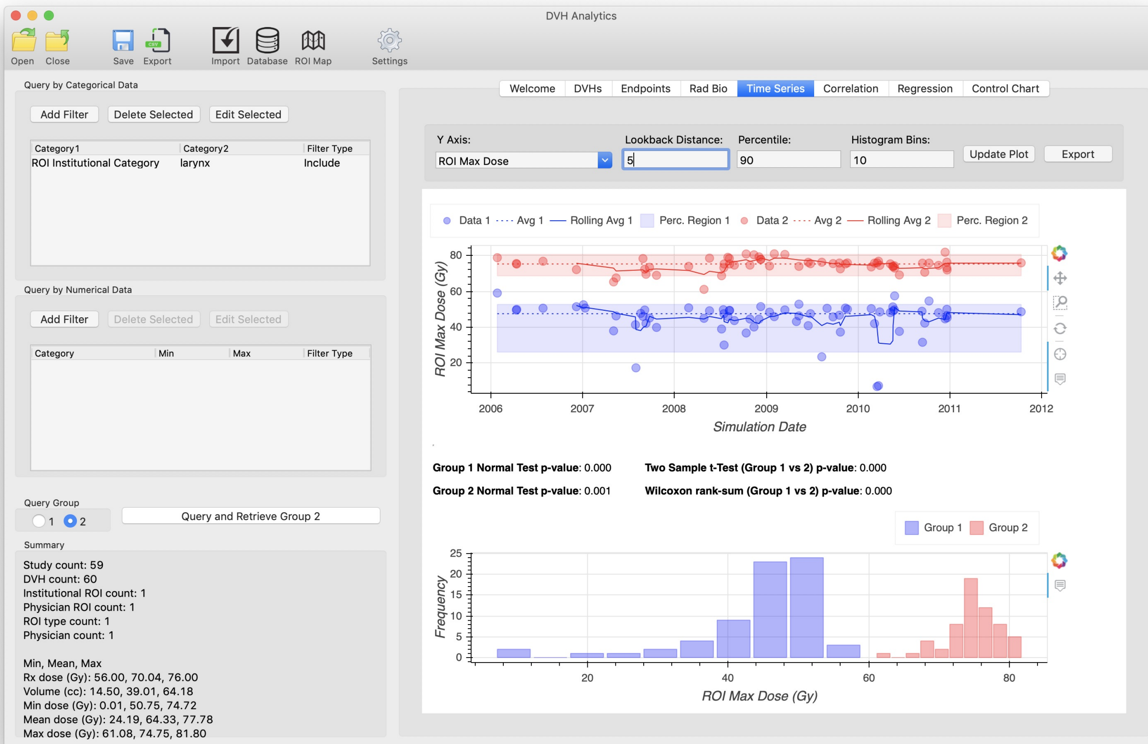


Figure 9: Time Series view.

4.4 Correlation

This module displays a Pearson-R correlation matrix, values calculated using `scipy.stats.pearsonr`. The circle diameter and opacity scale linearly with the absolute value of correlation – negative correlations using an alternate color. Normality test p-values are shown in pop-up windows on mouse-over, calculated with `scipy.stats.normaltest`.

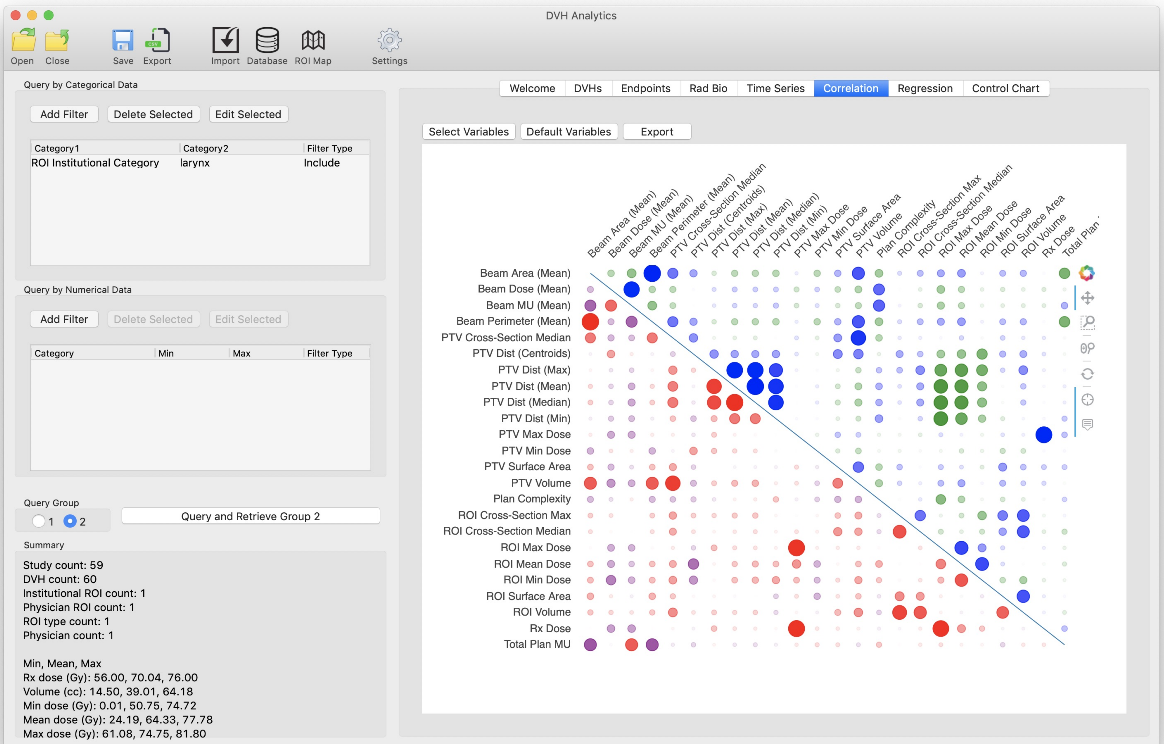


Figure 10: Correlation view.

4.5 Regression

DVHA provides modules to generate single & multi-variable linear and machine learning regressions with any of the numerical data from a queried data set. Categorical data is currently not supported.

4.5.1 Linear Regressions

The main Regression view displays one linear regression (per group) at a time. Residual and Quantile plots are also shown for the currently selected Query Group (via the radio button just above the Summary in the bottom-left). Users may build a collection of independent variables per dependent variable by checking the Include check-box or by using the Variable Quick Select dialog. Linear regressions are calculated using `sklearn.linear_model`.

Figure 11: Regression view.

After selecting a set of independent variables, a multi-variable regression using `sklearn.linear_model` may generated for the currently selected Query Group. The f-stat is calculated using `regressors.stats.f_stat`.

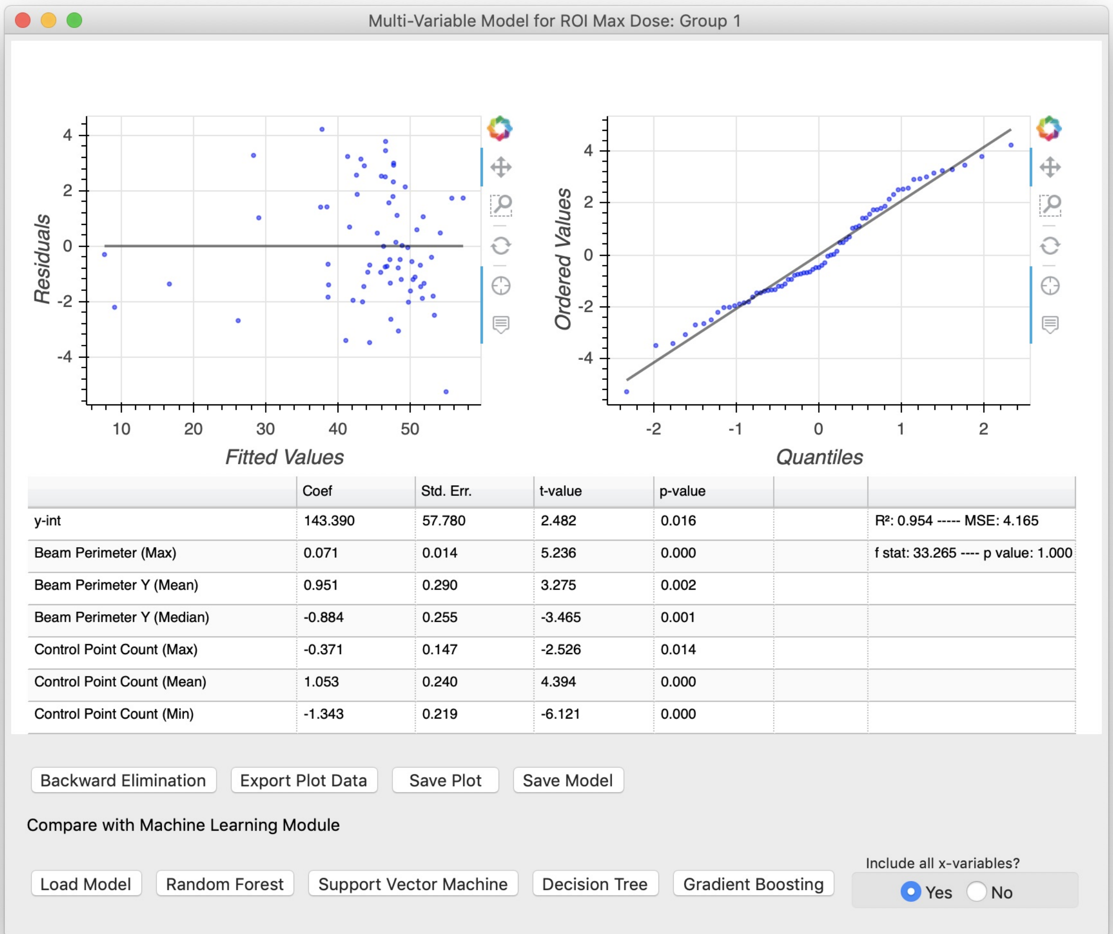


Figure 12: Multi-variable regression view.

Backward Elimination: recursively remove independent variables with the highest p-value one at a time until all variables have a p-value < 0.05 or only one independent variable is left; performing a new regression in each iteration.

Save MVR Model: save the multi-variable regression model to file. This file may be loaded using a different query or even with a different DVHA database. Advanced users may inspect this file using python's `pickle` library. To load a MVR model with the currently queried data, navigate to the menu to Data > Load Model > Multi-Variable Regression

4.5.2 Machine Learning Regression

A machine learning modeling view may be launched from the Multi-Variable Regression window. For comparison, the current multi-variable regression predictions are passed into this module. All machine learning algorithms are from Scikit-Learn:

- Random Forest: `sklearn.ensemble.RandomForestRegressor`
- Support Vector Machine: `sklearn.svm.SVR`
- Decision Tree: `sklearn.tree.DecisionTreeRegressor`
- Gradient Boosting: `sklearn.ensemble.GradientBoostingRegressor`

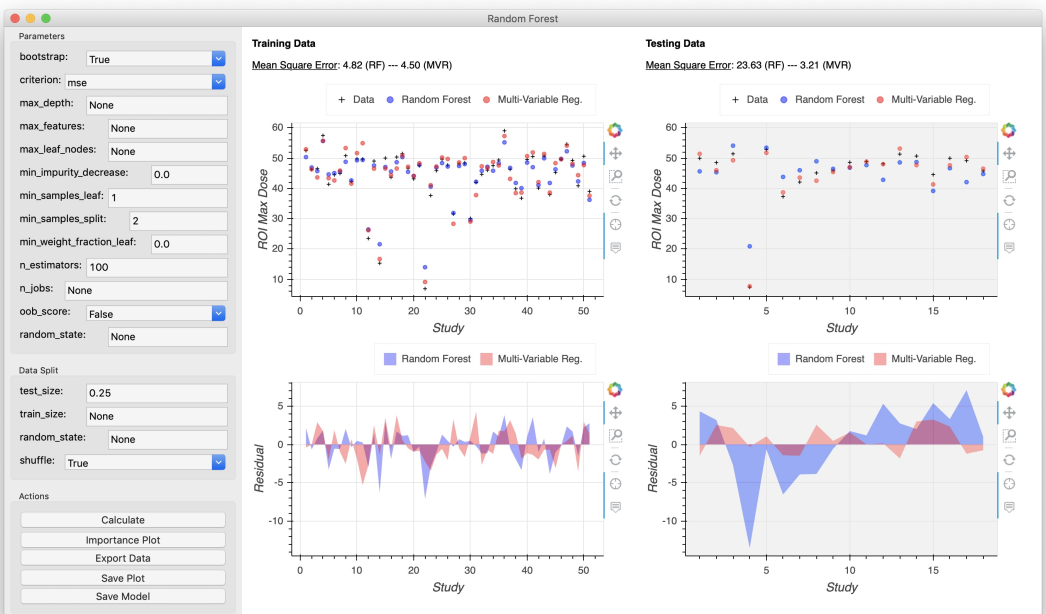


Figure 13: Random forest modeling view.

Nearly all of the hyper-parameters available to a command-line user are available in this module. Hover over an input with your mouse to see details, or find the algorithm at https://scikit-learn.org/ to learn more.

Load ML Model: load a machine learning model from file with the currently queried data. Loading a model does not split your data nor allow you to change hyper-parameters, it is simply for testing a model on a new data set. Users may load a model from the Multi-Variable Regression window or by navigating the menu bar: Data › Load Model › Machine Learning

4.6 Control Charts

Control charts are meant to identify data points that may be behaving based on unidentified causes. Users unfamiliar with control charts should refer to Montgomery DC. Statistical Quality Control. 7th ed. New York, NY: Wiley; 2012. Some paraphrased quotes:

> "A value outside of the control limits should be interpreted as evidence that the process is out of control."

> "The most important use of a control chart is to improve the process and it should be noted that most processes do not operate in a state of statistical control."

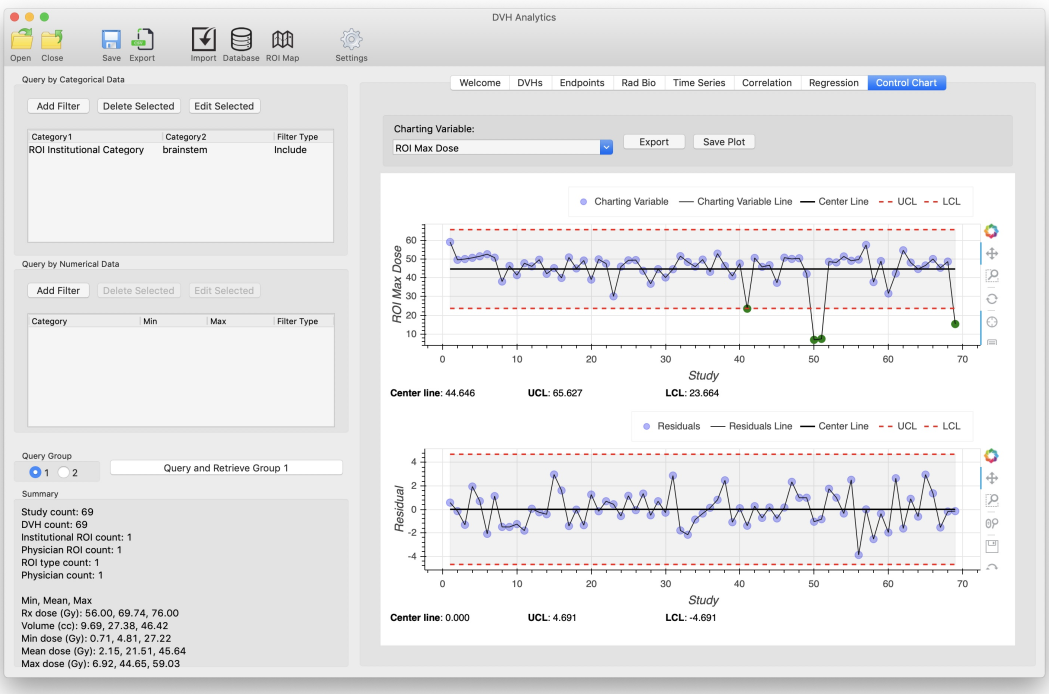


Figure 14: Control Chart view.

A control chart is simply a plot of chronological data with a center line and control limits (upper and lower). DVHA calculates a 2-point moving-range based on Eq. 5. The center line is the mean value of all points, $\overline{x}$. Control limits (CL) are calculated using Eq. 6 assuming the standard $\pm 3\sigma$.

$$\overline{mR} = \frac{1}{n-1}\sum_{i=2}^{n} |x_i - x_{i-1}|, \tag{5}$$

$$CL = \overline{x} \pm 3\frac{\overline{mR}}{1.128}, \tag{6}$$

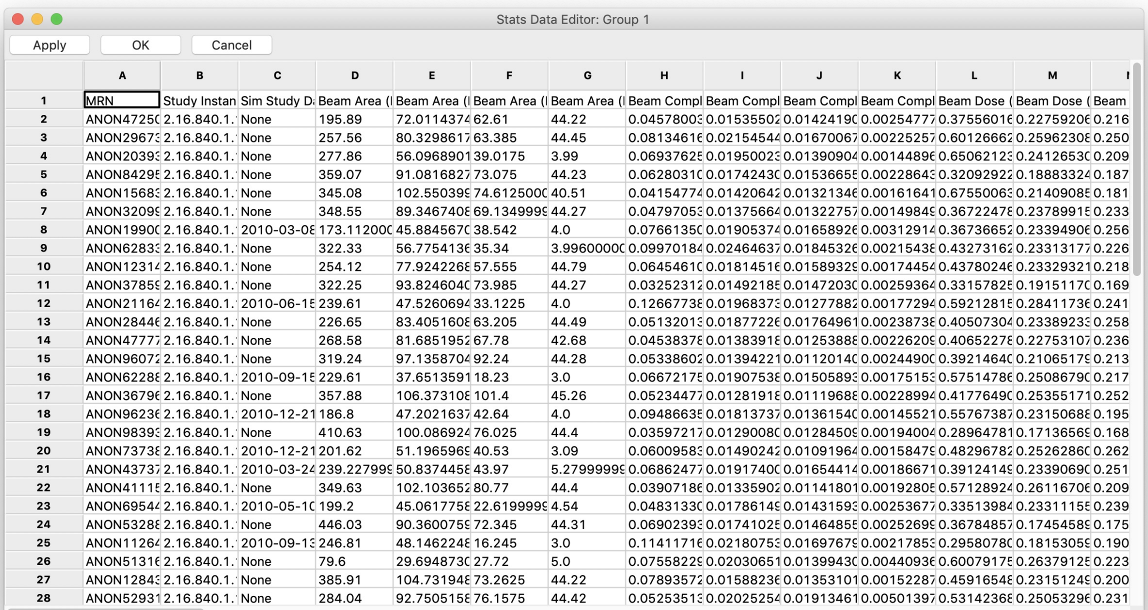
If a multi-variable linear regression has been generated for the charting variable, a risk-adjusted control is created. This is essentially a control chart of residuals. Additional details can be found in "A Risk-Adjusted Control Chart to Evaluate IMRT Plan Quality" by Roy *et. al* (DOI: https://doi.org/10.1016/j.adro.2019.11.006).

4.7 User Provided Data

Menu Navigation: Data ❯ Show Stats Data

A basic spreadsheet view is available for the purpose of adding additional numerical variables for statistical analysis. Users can shift- or ctrl-click items and copy to the clipboard for pasting into other applications (*e.g.*, Microsoft Excel); ctrl+A or cmd+A (on Mac) will select all cells. It will be easier to ensure the user provided data is in the correct order using an external spread sheet application before copying and pasting back into the DVHA spreadsheet.

Coincidentally, users may edit the currently queried numerical data. Although this does not get stored in the database, it is saved in a DVHA session: File ❯ Save.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	MRN	Study Instan	Sim Study Da	Beam Area (l	Beam Area (l	Beam Area (l	Beam Area (l	Beam Compl	Beam Compl	Beam Compl	Beam Compl	Beam Dose (	Beam Dose (	Beam
2	ANON47250	2.16.840.1.	None	195.89	72.0114374	62.61	44.22	0.04578003	0.01535502	0.01424190	0.00254777	0.37556016	0.22759206	0.216
3	ANON29673	2.16.840.1.	None	257.56	80.3298617	63.385	44.45	0.08134616	0.02154544	0.01670067	0.00225257	0.60126662	0.25962308	0.250
4	ANON20393	2.16.840.1.	None	277.86	56.0968901	39.0175	3.99	0.06937625	0.01950023	0.01390904	0.00144896	0.65062123	0.24126530	0.209
5	ANON84295	2.16.840.1.	None	359.07	91.0816827	73.075	44.23	0.06280310	0.01742430	0.01536655	0.00228643	0.32092922	0.18883324	0.187
6	ANON15683	2.16.840.1.	None	345.08	102.550399	74.6125000	40.51	0.04154774	0.01420642	0.01321346	0.00161641	0.67550063	0.21409085	0.181
7	ANON32099	2.16.840.1.	None	348.55	89.3467408	69.1349999	44.27	0.04797053	0.01375664	0.01322757	0.00149849	0.36722478	0.23789915	0.233
8	ANON19900	2.16.840.1.	2010-03-08	173.112000	45.8845670	38.542	4.0	0.07661350	0.01905374	0.01658926	0.00312914	0.36736652	0.23394906	0.256
9	ANON62833	2.16.840.1.	None	322.33	56.7754136	35.34	3.99600000	0.09970184	0.02464637	0.01845326	0.00215438	0.43273162	0.23313177	0.226
10	ANON12314	2.16.840.1.	None	254.12	77.9242268	57.555	44.79	0.06454610	0.01814516	0.01589329	0.00174454	0.43780246	0.23329321	0.218
11	ANON37859	2.16.840.1.	None	322.25	93.8246040	73.985	44.27	0.03252312	0.01492185	0.01472030	0.00259364	0.33157825	0.19151170	0.169
12	ANON21164	2.16.840.1.	2010-06-15	239.61	47.5260694	33.1225	4.0	0.12667738	0.01968373	0.01277882	0.00177294	0.59212815	0.28411736	0.241
13	ANON28446	2.16.840.1.	None	226.65	83.4051608	63.205	44.49	0.05132013	0.01877226	0.01764961	0.00238738	0.40507304	0.23389233	0.258
14	ANON47777	2.16.840.1.	None	268.58	81.6851952	67.78	42.68	0.04538378	0.01383918	0.01253888	0.00226209	0.40652278	0.22753107	0.236
15	ANON96072	2.16.840.1.	None	319.24	97.1358704	92.24	44.28	0.05338602	0.01394221	0.01120140	0.00244900	0.39214640	0.21065179	0.213
16	ANON62288	2.16.840.1.	2010-09-15	229.61	37.6513591	18.23	3.0	0.06672175	0.01907538	0.01505893	0.00175153	0.57514786	0.25086790	0.217
17	ANON36796	2.16.840.1.	None	357.88	106.373108	101.4	45.26	0.05234477	0.01281918	0.01119688	0.00228994	0.41776490	0.25355171	0.252
18	ANON96236	2.16.840.1.	2010-12-21	186.8	47.2021637	42.64	4.0	0.09486635	0.01813737	0.01361540	0.00145521	0.55767387	0.23150688	0.195
19	ANON98393	2.16.840.1.	None	410.63	100.086924	76.025	44.4	0.03597217	0.01290080	0.01284509	0.00194004	0.28964781	0.17136569	0.168
20	ANON73738	2.16.840.1.	2010-12-21	201.62	51.1965969	40.53	3.09	0.06009583	0.01490242	0.01091964	0.00158479	0.48296782	0.25262860	0.262
21	ANON43737	2.16.840.1.	2010-03-24	239.227999	50.8374458	43.97	5.27999999	0.06862477	0.01917400	0.01654414	0.00186671	0.39124149	0.23390690	0.251
22	ANON41115	2.16.840.1.	None	349.63	102.103652	80.77	44.4	0.03907186	0.01335902	0.01141801	0.00192805	0.57128924	0.26116706	0.209
23	ANON69544	2.16.840.1.	2010-05-10	199.2	45.0617758	22.6199999	4.54	0.04831330	0.01786149	0.01431593	0.00253677	0.33513984	0.23311155	0.239
24	ANON53288	2.16.840.1.	None	446.03	90.3600759	72.345	44.31	0.06902393	0.01741025	0.01464855	0.00252699	0.36784857	0.17454589	0.175
25	ANON11264	2.16.840.1.	2010-09-13	246.81	48.1462248	16.245	3.0	0.11411716	0.02180753	0.01697679	0.00217853	0.29580780	0.18153059	0.190
26	ANON51316	2.16.840.1.	None	79.6	29.6948730	27.72	5.0	0.07558229	0.02030651	0.01399430	0.00440936	0.60079175	0.26379125	0.223
27	ANON12843	2.16.840.1.	None	385.91	104.731948	73.2625	44.22	0.07893572	0.01588236	0.01353101	0.00152287	0.45916548	0.23151249	0.200
28	ANON52931	2.16.840.1.	None	284.04	92.7505158	76.1575	44.42	0.05253513	0.02025254	0.01913461	0.00501397	0.53142368	0.25053296	0.231

Figure 15: Stats data editor window.

5 Database Administrator

Menu Navigation: Data ⟩ Database Administrator

DVHA provides a simple database viewer. Clicking query will return all data that meets the provided condition (if any) along with the selected columns on the left (or all columns if none are selected). The format of the condition (*i.e.* the `WHERE` block of a SQL query) must follow SQL syntax for the current database type (*i.e.*, PostgreSQL or SQLite3).

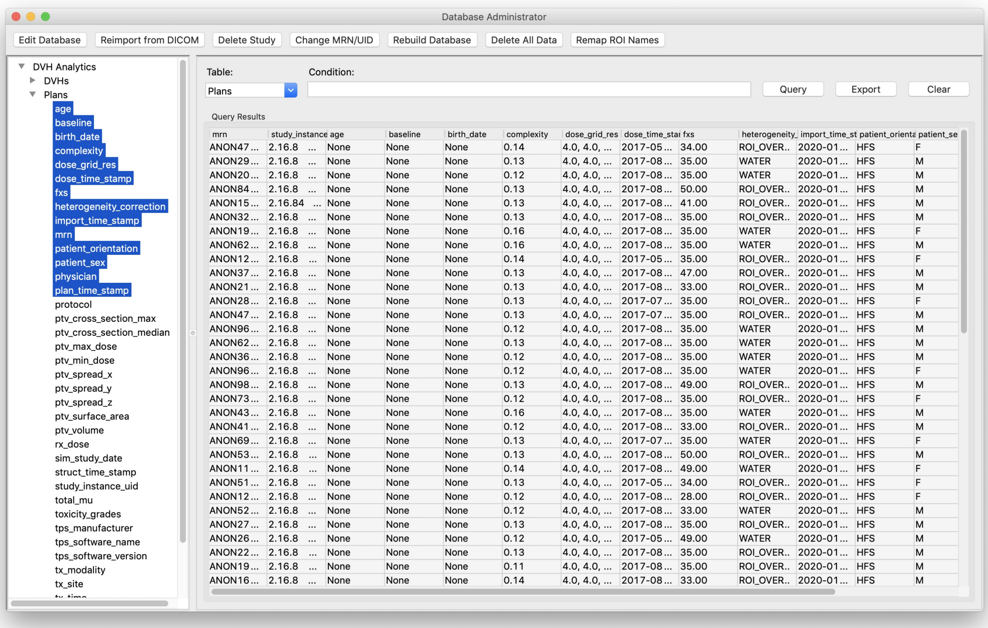


Figure 16: Database Administrator view.

Some additional features of this module include:

- Edit Database
- Reimport from DICOM
- Delete Study
- Change MRN/UID
- Rebuild Database
- Delete All Data
- Remap ROI Names

6 Data Dictionary

The following tables itemize each column of the SQL tables. Units and DICOM tags are noted where applicable.

Table 3: DVHs table columns.

Column	Data Type	Units	DICOM Tag	Description
mrn	text	-	(0010, 0020)	Medical Record Number (PatientID)
study_instance_uid	text	-	(0020, 000d)	Unique ID tied to planning image set
institutional_roi	varchar(50)	-	-	Standard ROI name for all physician
physician_roi	varchar(50)	-	-	Standard ROI name for patient's physician
roi_name	varchar(50)	-	(3006, 0026)	ROI name as in plan
roi_type	varchar(20)	-	(3006, 00a4)	ROI categegoy (e.g., ORGAN, PTV)
volume	real	cm^3	-	ROI volume per dicompyler
min_dose	real	Gy	-	Min ROI dose per dicompyler
mean_dose	real	Gy	-	Mean ROI dose per dicompyler
max_dose	real	Gy	-	Max ROI dose per dicompyler
dvh_string	text	cGy	-	CSV of DVH in 1 cGy bins
roi_coord_string	text	-	(3006, 0050)	Single string containing all ROI points
dist_to_ptv_min	real	cm	-	Calculated with scipy's cdist function
dist_to_ptv_mean	real	cm	-	"
dist_to_ptv_median	real	cm	-	"
dist_to_ptv_max	real	cm	-	"
surface_area	real	cm^2	-	DVHA custom function, needs validation
ptv_overlap	real	cm^3	-	DVHA custom function with Shapely
import_time_stamp	timestamp	-	-	Time per SQL at time of import
centroid	varchar(35)	-	-	DVHA custom function
dist_to_ptv_centroids	real	cm	-	DVHA custom function
dth_string	text	cm	-	numpy histogram of scipy cdist with PTV
spread_x	real	cm	(3006, 0050)	Max distance in x-dim of ROI
spread_y	real	cm	(3006, 0050)	Max distance in y-dim of ROI
spread_z	real	cm	(3006, 0050)	Max distance in z-dim of ROI
cross_section_max	real	cm^2	-	DVHA custom function with Shapely
cross_section_median	real	cm^2	-	DVHA custom function with Shapely
toxicity_grade	smallint	-	-	Not yet implemented

Table 4: Plans table columns.

Column	Data Type	Units	DICOM Tag	Description
mrn	text	-	(0010, 0020)	Medical Record Number (PatientID)
study_instance_uid	text	-	(0020, 000d)	Unique ID tied to planning image set
birth_date	date	-	(0010, 0030)	
age	smallint	*years*	-	Patient age on day of simulation
patient_sex	char(1)	-	(0010, 0040)	Patient's sex
sim_study_date	date	-	(0008, 0020)	Date of simulation imaging
physician	varchar(50)	-	(0010, 0048)	(1) PhysiciansOfRecord or
			(0008, 0090)	(2) ReferringPhysiciansName
tx_site	varchar(50)	-	(300a, 0002)	RTPlanLabel
rx_dose	real	Gy	(300a, 0026)	TargetPrescriptionDose
fxs	int	-	(300a, 0078)	NumberOfFractionsPlanned
patient_orientation	varchar(3)	-	(0018, 5100)	Acronym of patient's sim orientation
plan_time_stamp	timestamp	-	(300a, 0006)	Timestamp for plan
			(300a, 0007)	
struct_time_stamp	timestamp	-	(3006, 0008)	Timestamp for structure set
			(3006, 0009)	Timestamp for structure set
dose_time_stamp	timestamp	-	(3006, 00012)	Timestamp for dose file
			(3006, 00013)	Timestamp for structure set
tps_manufacturer	varchar(50)	-	(0008, 0070)	Manufacturer in RTPlan
tps_software_name	varchar(50)	-	(0008, 1090)	ManufacturerModelName in RTPlan
tps_software_version	varchar(30)	-	(0018, 1020)	CSV of SoftwareVersions in RTPlan
tx_modality	varchar(30)	-	(300a, 00c6)	Based on RadiationType, includes 3D or arc
			(300a, 011e)	
tx_time	time	-	(300a, 0286)	For brachy plans
total_mu	real	-	(300a, 0086)	Total MU to be delivered to the patient
dose_grid_res	varchar(16)	mm	(0028, 0030)	Resolution of dose grid
			(0018, 0050)	
heterogeneity_correction	varchar(30)	-	(3004, 0014)	CSV of heterogeneity correction
baseline	boolean	-	-	Not yet implemented
import_time_stamp	timestamp	-	-	Time per SQL at time of import
toxicity_grades	text	-	-	Not yet implemented
protocol	text	-	-	Not yet implemented
complexity	real	-	-	Plan complexity score
ptv_cross_section_max	real	cm^2	-	Area of largest PTV slice for plan
ptv_cross_section_median	real	cm^2	-	Median slice area of PTV for plan
ptv_spread_x	real	cm	-	Largest x-dim distance of PTV for plan
ptv_spread_y	real	cm	-	Largest y-dim distance of PTV for plan
ptv_spread_z	real	cm	-	Largest z-dim distance of PTV for plan
ptv_surface_area	real	cm^2	-	Surface area of PTV for plan
ptv_volume	real	cm^3	-	Volume of PTV for plan
ptv_max_dose	real	Gy	-	
ptv_min_dose	real	Gy	-	

Table 5: Rxs table columns.

Column	Data Type	Units	DICOM Tag	Description
mrn	text	-	(0010, 0020)	Medical Record Number (PatientID)
study_instance_uid	text	-	(0020, 000d)	Unique ID tied to planning image set
plan_name	varchar(50)	-	(300A, 0002)	
fx_grp_name	varchar(30)	-	(300A, 0071)	Primarily for Pinnacle with special POIs
fx_grp_number	smallint	-	(300A, 0071)	
fx_grp_count	smallint	-	-	Number of fraction groups in RTPlan
fx_dose	real	-	-	rx_dose / fxs
fxs	smallint	-	(300A, 0078)	
rx_dose	real	-	(300A, 0026)	Per dicompyler if not found
rx_percent	real	-	-	Currently only available with special POIs
normalization_method	varchar(30)	-	(300A, 0014)	
normalization_object	varchar(30)	-	-	Intended for special POIs
import_time_stamp	timestamp	-	-	Time per SQL at time of import

Table 6: Beams table columns.

Column	Data Type	Units	DICOM Tag	Description
mrn	text	-	(0010, 0020)	Medical Record Number (PatientID)
study_instance_uid	text	-	(0020, 000d)	Unique ID tied to planning image set
beam_number	int	-	(300A, 00C0)	
beam_name	varchar(30)	-	(300A, 00C3)	(1) Beam Description or
		-	(300A, 00C2)	(2) Beam Name
fx_grp_number	smallint	-	-	See Rxs table
fx_count	int	-	-	See Rxs table
fx_grp_beam_count	smallint	-	-	See Rxs table
beam_dose	real	-	(300A, 008B)	
beam_mu	real	-	(300A, 0086)	
radiation_type	varchar(30)	-	(300A, 00C6)	
beam_energy_min	real	-	(300A, 0114)	
beam_energy_max	real	-	”	
beam_type	varchar(30)	-	(300A, 00C4)	
control_point_count	int	-	-	
gantry_start	real	-	(300A, 011E)	
gantry_end	real	-	”	
gantry_range	real	-	”	
gantry_min	real	-	”	
gantry_max	real	-	”	
gantry_rot_dir	varchar(5)	-	(300A, 011F)	
collimator_start	real	-	(300A, 0120)	
collimator_end	real	-	”	
collimator_range	real	-	”	
collimator_min	real	-	”	
collimator_max	real	-	”	
collimator_rot_dir	varchar(5)	-	(300A, 0121)	
couch_start	real	-	(300A, 0122)	
couch_end	real	-	”	
couch_range	real	-	”	
couch_min	real	-	”	
couch_max	real	-	”	
couch_rot_dir	varchar(5)	-	(300A, 0123)	
beam_dose_pt	varchar(35)	-	(300A, 0082)	
isocenter	varchar(35)	-	(300A, 012C)	
ssd	real	-	(300A, 0130)	Average of these values
treatment_machine	varchar(30)	-	(300A, 00B2)	

Table 7: Beams table columns (continued).

Column	Data Type	Units	DICOM Tag	Description
scan_mode	varchar(30)	-	(300A, 0308)	
scan_spot_count	real	-	(300A, 0392)	
beam_mu_per_deg	real	-	-	
beam_mu_per_cp	real	-	-	
import_time_stamp	timestamp	-	-	Time per SQL at time of import
area_min	real	-	-	
area_mean	real	-	-	
area_median	real	-	-	
area_max	real	-	-	
x_perim_min	real	-	-	
x_perim_mean	real	-	-	
x_perim_median	real	-	-	
x_perim_max	real	-	-	
y_perim_min	real	-	-	
y_perim_mean	real	-	-	
y_perim_median	real	-	-	
y_perim_max	real	-	-	
complexity_min	real	-	-	
complexity_mean	real	-	-	
complexity_median	real	-	-	
complexity_max	real	-	-	
cp_mu_min	real	-	-	
cp_mu_mean	real	-	-	
cp_mu_median	real	-	-	
cp_mu_max	real	-	-	
complexity	real	-	-	
tx_modality	varchar(30)	-	-	
perim_min	real	-	-	
perim_mean	real	-	-	
perim_median	real	-	-	
perim_max	real	-	-	

Table 8: DICOM_Files table columns.

Column	Data Type	Units	DICOM Tag	Description
mrn	text	-	(0010, 0020)	Medical Record Number (PatientID)
study_instance_uid	text	-	(0020, 000d)	Unique ID tied to planning image set
folder_path	text	-	-	-
plan_file	text	-	-	-
structure_file	text	-	-	-
dose_file	text	-	-	-
import_time_stamp	timestamp	-	-	-

7 Acknowledgements

DVH Analytics was initially developed at Northwestern Memorial Hospital by Dan Cutright (coding, design, and UI development) and Mahesh Gopalakrishnan (design). Statistical methods for plan quality assessments were provided by Arkajyoti Roy (Bowling Green State University *at the time*). This software is under continuing development at the University of Chicago Medical Center.

We would like to extend our sincere gratitude to Dr. Bharat B. Mittal for providing a large database of patients used for the initial development.

Many thanks to Aditya Panchal for developing dicompyler and providing feedback about DVHA. The availability of dicompyler as an open-source project was extremely helpful, both in calculations and as an example for wxPython GUI development.

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." *J. App. Clin. Med. Phys.* (2018).