DVHTOOL VERSION 1.0

PURPOSE

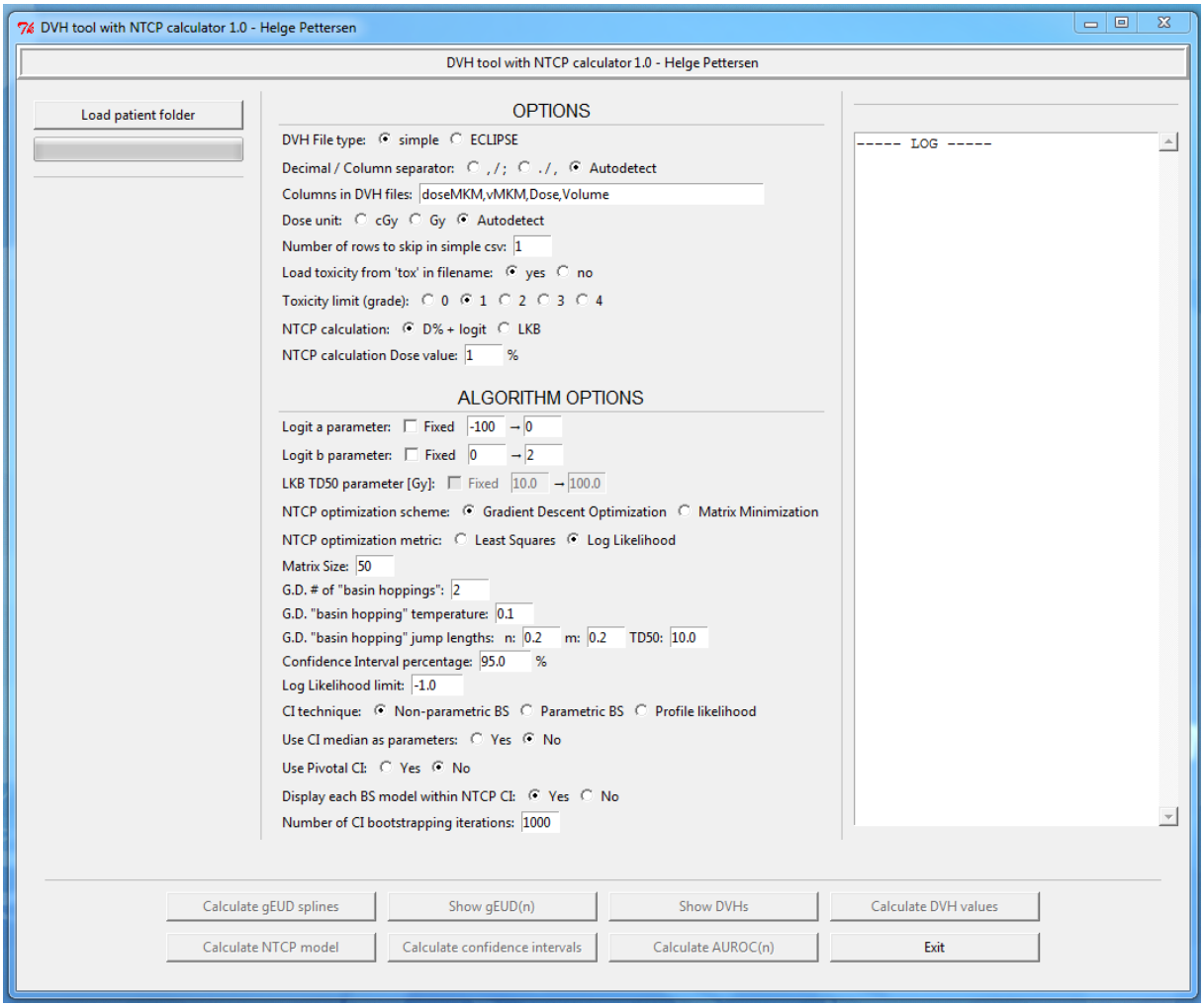
Extract information from Dose Volume Histogram files, calculate metrics such as D%, V% and fitting to NTCP models with methods of calculating the confidence intervals of the resulting parameters.

By Helge Egil Seime Pettersen (helge.pettersen@helse-bergen.no), Haukeland University Hospital, 2019.

PREREQUISITES

Python 2 or 3 with the following packages: Pandas [1], Numpy, Matplotlib, Scipy, Tkinter

PROGRAM OVERVIEW



The program consists of a single window, with some dialog options. Most of the buttons and options are documented with a tooltip, accessed by holding the pointer above the wanted option. All settings are saved upon exiting.

FILE OPTIONS

Before loading any patient / cohort folders containing CVS or ECLIPSE DVH files, the files to be opened has to be configured using the «OPTIONS» part of the window. First, the file type has to be selected:

- CSV files: A simple CSV file is a DVH file that is either comma or semicolon separated, containing the columns «Dose» and «Volume». It is configured by choosing whether it is comma separated with period decimals (x.y, x.y), or semicolon separated with comma decimals (x,y; x,y): the autodetect option chooses semicolon if any are detected in the first line. Then, the columns has to be chosen. Use a comma-separated list without whitespace. The two columns «Dose» and «Volume» are read out from the CSV file, so if they are the 3. and 4. columns you can write «,,Dose,Volume» or «DoseWrong,VolumeWrong,Dose,Volume» etc. The dose unit is chosen below, either cGy or Gy. If the autodetect options is used, then cGy is chosen if any of the entries in the dose column exceeds «1000». If the CSV file contains a header row, skip this by choosing the correct options (skip = 1).
- ECLIPSE files: Most of the options here are similar, but in addition the program supports different organ structures. The file is read out, and all found structures are listed in a dialog when choosing the **Load patient folder** button. Write a string that matches the wanted structures («Brain» matches «Brain», «Whole Brain», «Brain L» etc.).

The patient toxicity can either be given as another CVS file (to be called «cohort_tox.csv» in the folder above the cohort folder): This file should contain each patient by their filename and toxicity grade. If the patient contained in 1234.csv has grade 3, write a line «1234,3» in the file. Then, choose the wanted toxicity threshold in the program. (2 -> all patients with grade ≥ 2 are defined as having complications). It is also possible to load the toxicity from the filenames, so that any file with «tox» in the filename is defined as having toxicity. In that case, use toxicity limit 1.

After these settings have been configured, load one or more patient cohorts using the **Load patient folder** button. This will open a file dialog, where you choose the folder containing the CSV files. The cohort name will be the name of the folder. All the added cohorts will be pooled to a single dataset. The validity of the loaded data may be controlled with the **Show DVHs** button: patients with tox are shown in red. In addition, certain % Dose or % Volume values can be calculated for each patient and stored in output CSV files for further use with the **Calculate DVH values** button. If the LKB model is to be used for calculations, a set of look-up-tables for the gEUD values must be calculated after loading the patient cohorts. This is only performed once per patient (per structure), and is done by clicking the **Calculate gEUD splines** button. The look-up-tables are stored as cubic splines in the cohort folder (subfolder gEUD), ensuring high interpolation accuracy for all values of n. The **Show gEUD(n)** button can subsequently be used to display the different gEUD values as functions of n: Patients with tox are shown in red.

In order to identify the strength of different n values for the dataset, it is possible to calculate the AUROC: For each n value, check how many patients are correctly modelled with toxicity using a dose threshold based on their gEUD values (gEUD < threshold -> no tox; gEUD > threshold -> tox). The AUROC for this sensitivity and specificity for this n value calculated, and the n values are varied according to the limits chosen [2].

CHOICE OF NTCP MODEL

Two methods for calculating the NTCP are supported:

- **The logit model** $NTCP = 1 - (1 + \exp(a + bD))^{-1}$, where D is calculated as the dose at a given volume fraction. In that case, choose the % value below, e.g. D20% or D1%. For each patient, this value is calculated from the DVH files.
- **The Lyman-Kutcher-Burman (LKB) model** [3], [4], in which a the DVH is aggregated to a gEUD value $gEUD = \left[{D_i}^{1/n} \Delta V_i\right]^n$ as calculated from a (here created) differential DVH. The n value

describes whether the organ is serial or parallel. Then, the two parameters m (population variance) and TD50 (dose giving a 50% probability of toxicity) are used in a cumulative density function (CDF)

$$NTCP = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t} \exp(-x^2/2)\,\mathrm{d}x\,, \qquad t = \frac{\mathrm{gEUD} - \mathrm{TD50}}{m \cdot \mathrm{TD50}}.$$

In this program, this CDF is substituted to increase the computational efficiency [5]:

$$\mathrm{NTCP} \simeq \frac{1}{1 + \exp\left(-\frac{238}{23}t + 111\,\mathrm{atan}\left(\frac{37}{294}t\right)\right)}, \qquad t = \frac{\mathrm{gEUD} - \mathrm{TD50}}{m \cdot \mathrm{TD50}}.$$

FINDING THE MODEL PARAMETERS OF THE NTCP MODEL

The main part of the program is to find the model parameters of the chosen NTCP model: a,b for the Logit model, and n,m,TD50 for the LKB model. The bounds of the parameters are defined (or the default values can be kept). Two methods are supplied for the optimization procedure:

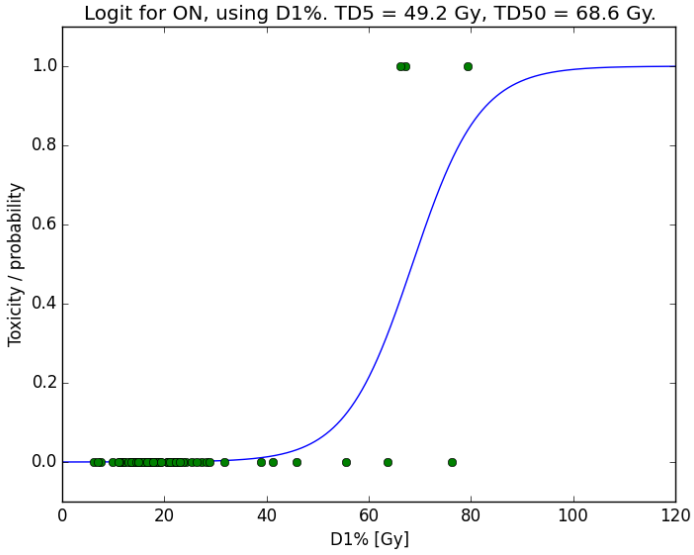
- Matrix Minimization: A grid of a given size (matrix size option) is made in 2 (logit) or 3 (LKB) dimensions. For each cell, the error between the calculated NTCP model and the truth (toxicity) is summed over all patient, and the parameters corresponding to the cell containing the lowest error are stored. This is time demanding, especially for the LKB model with large matrix sizes (high granularity of the output parameters).
- Gradient Descent Optimization: Here, an algorithm applying the truncated Newton algorithm [6] for unconstrained optimization of the parameters, together with a basin jumping procedure for locating the global minimum from any identified local minima [7]. The methods are implemented using the `scipy.optimize.basinhopping` routine [8]. It is substantially quicker compared to the matrix minimization, and can also locate minima with a higher accuracy. This method is configured with a certain number of «basin hoppings», or perturbations from a located minimum. The «temperature» is the expected change in error between minima and the «jump lengths» define the limits of a random variable defining the perturbation for a given parameter.

With each optimization method, the error calculation can be as least squares or a Maximum Log Likelihood

$$LS = \min(\sum(\mathrm{tox}_i - \mathrm{NTCP}_i)^2)\,,$$

$$LLH = \max\left(\sum_{\mathrm{tox}} \ln \mathrm{NTCP}_i + \sum_{\mathrm{not\ tox}} \ln(1 - \mathrm{NTCP}_i)\right),$$

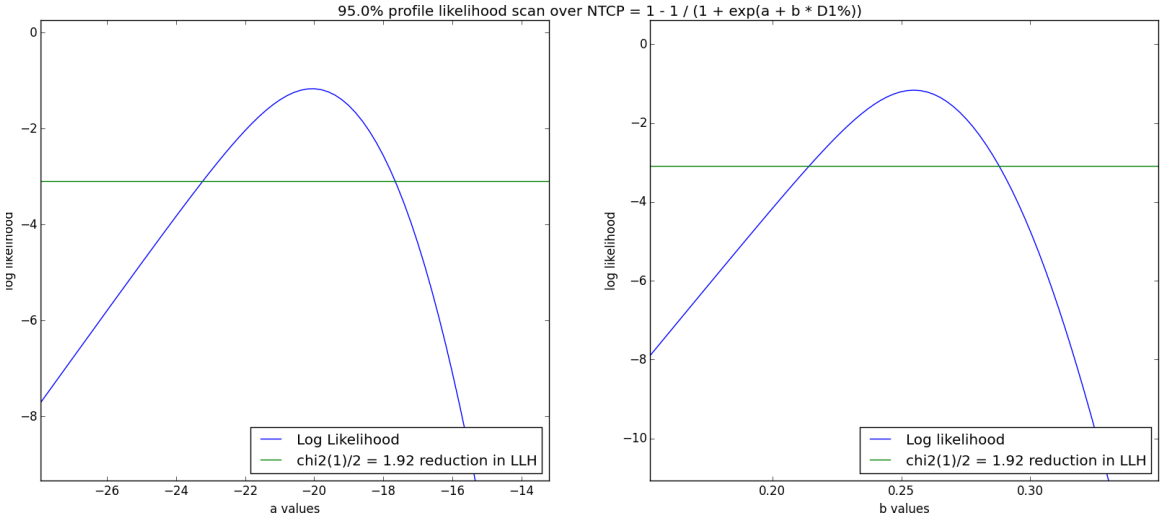
the latter being more adopted [3]. The chosen method is applied by using the **Calculate NTCP** button, in which a plot as shown below is displayed (here with D1% and logit):



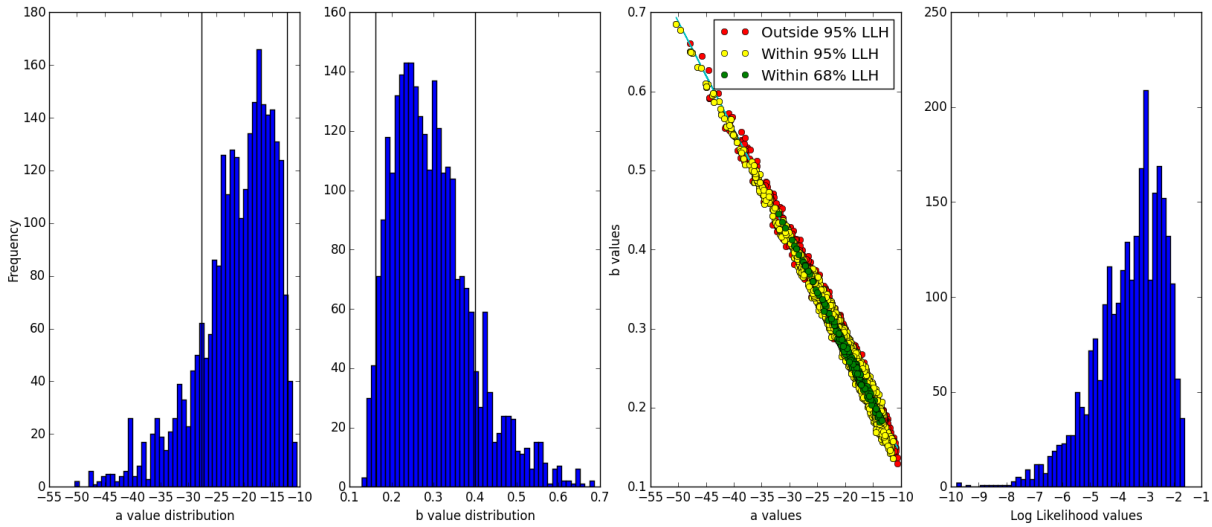
CALCULATING THE CONFIDENCE LIMIT OF THE MODELS

In this part of the program, several approaches have been attempted and compared. In [3], three methods are described:

- **Profile likelihood**: For each parameter, a 1D scan is performed. The Log Likelihood value is drawn as a function of the parameter, with its maximum at the fitted value. Then, the two locations where
$$LLH = LLH_{\max} - \chi^2(1)/2$$
are identified, where $\chi^2(1)/2$ is the critical χ^2 value with 1 DoF. A 95% confidence interval corresponds to $\chi^2(1)/2 = 1.92$.



- **Non-parametric bootstrap**: This statistical method originates from [9], and is a Monte Carlo-type method to increase the statistical power of a sample. With n patients in the cohort, n patients are chosen at random (with replacement). For each synthetic cohort, a parameter optimization is performed to obtain a new set of parameters a, b or $n, m, TD50$. The resulting distribution of parameters is used to calculate the statistics such as 95% confidence interval, median parameters etc. This process is repeated 1000-2000 times (chosen as the **Number of CI bootstrapping iterations**). If the «Use CI median as parameters» is chosen, the median value of this new distribution substitutes the former found optimal parameter set in future NTCP plots (**calculate NTCP model** after **Calculate Confidence Intervals**)



- **Parametric bootstrap:** Similar to the non-parametric bootstrap. Instead of choosing a synthetic cohort, now the cohort it kept but for each patient a random number r_N is generated. If the NTCP calculated from the model fitted paramateters is $\leq r_N$, no complication is assumed, and if $r_N < NTCP$ complication is assumed. Then, a parameter set is found from this new toxicity cohort--to be repeated 1000-2000 times.

Using the bootstrap methods, a «pivot» interval can be used as improved CI [10, p. 11], and it is calculated as

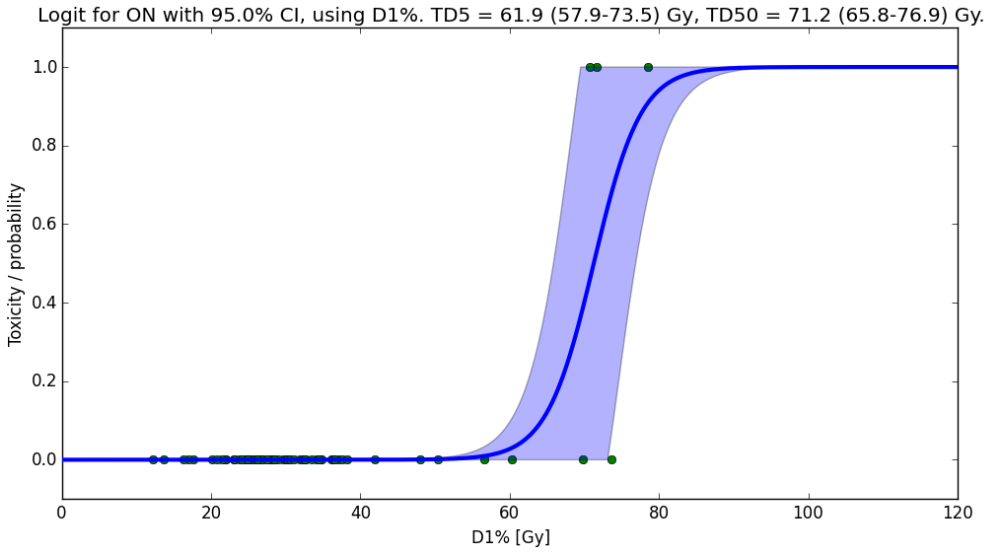
$$[\hat{\theta}_L, \hat{\theta}_U] \rightarrow [2\hat{\theta} - \hat{\theta}_U, 2\hat{\theta} - \hat{\theta}_L],$$

Where $\hat{\theta}_{L,U}$ are the upper and lower limits, respectively, and $\hat{\theta}$ is the best estimate value from the original distribution.

After calculation of the confidence intervals, a renewed push on the **calculate NTCP models** button will display the confidence limits in terms of the NTCP calculation. If the "display each BS model within NTCP CI" option is used, all the individual models (that are within the LLH CI) are drawn on top of the shaded NTCP CI. The error is propagated from the separate parameter limits to the NTCP using the error propagation function

$$\sigma_{NTCP} = \sqrt{\left[\sigma_a \frac{\partial NTCP}{\partial a}\right]^2 + \left[\sigma_b \frac{\partial NTCP}{\partial b}\right]^2 + \rho_{ab}\sigma_a\sigma_b \frac{\partial NTCP}{\partial a}\frac{\partial NTCP}{\partial b}}$$
$$= \sqrt{\sigma_a^2 \frac{e^{2a+2bD}}{(e^{a+bD}+1)^4} + \sigma_b^2 \frac{D^2 e^{2a+2bD}}{(e^{a+bD}+1)^4} + \rho_{ab}\sigma_a\sigma_b \frac{De^{2a+2bD}}{(e^{a+bD}+1)^4}},$$

and requiring that $0 \leq NTCP \leq 1$. This function has only been implemented for logit so far, due to difficulties with the propagation of the error on the n parameter through gEUD.



Note:

- The CI part of the program needs more work before being completely validated. Depending on the method used (Profile Likelihood, (Non) parametric bootstrap, with / without pivot methods etc.), how the final CI estimate is propagated to the NTCP, and whether the optimized parameter is the median value in the bootstrapping distribution (the "use CI median as parameters" button enforces this), we see different CIs. I would love to test the program on a larger dataset to see whether the methods converge. I am no statistician. ☺
- Though, I am fairly confident on the methods used for parameter optimization.

BIBLIOGRAPHY

[1] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51–56.

[2] T. Boulé, M. I. G. Fuentes, J. V. Roselló, R. A. Lara, J. L. Torrecilla, and A. L. Plaza, “Clinical comparative study of dose–volume and equivalent uniform dose based predictions in post radiotherapy acute complications,” *Acta Oncol.*, vol. 48, no. 7, pp. 1044–1053, Jan. 2009.

[3] M. Carolan, B. Oborn, K. Foo, A. Haworth, S. Gulliford, and M. Ebert, “An MLE method for finding LKB NTCP model parameters using Monte Carlo uncertainty estimates,” *J. Phys. Conf. Ser.*, vol. 489, p. 012087, Mar. 2014.

[4] S. L. Gulliford, M. Partridge, M. R. Sydes, S. Webb, P. M. Evans, and D. P. Dearnaley, “Parameters for the Lyman Kutcher Burman (LKB) model of Normal Tissue Complication Probability (NTCP) for specific rectal complications observed in clinical practise,” *Radiother. Oncol.*, vol. 102, no. 3, pp. 347–351, Mar. 2012.

[5] Hector Vazquez-Leal, Roberto Castaneda-Sheissa, Uriel Filobello-Nino, Arturo Sarmiento-Reyes, and Jesus Sanchez Orea, “High Accurate Simple Approximation of Normal Distribution Integral,” *Math. Probl. Eng.*, vol. 124029, 2012.

[6] L. Grippo, F. Lampariello, and S. I. L. Lucid, “A truncated Newton method with nonmonotone line search for unconstrained optimization,” *J. Optim. Theory Appl.*, pp. 401–419, 1989.

[7] D. J. Wales and J. P. K. Doye, “Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms,” *J. Phys. Chem. A*, vol. 101, no. 28, pp. 5111–5116, Jul. 1997.

[8] E. Jones, E. Oliphant, P. Petersen, and and others, *SciPy: Open Source Scientific Tools for Python*. 2001.

[9] B. Efron, “Bootstrap Methods: Another Look at the Jackknife,” *Ann. Stat.*, vol. 7, no. 1, pp. 1–26, 1979.

[10] D. Banks, “STAT110 lecture: Bootstrap Confidence Intervals.” http://www2.stat.duke.edu/~banks/111-lectures.dir/lect13.pdf