DeuteRater v6 ReadMe

Table of Contents

[Introduction 1](#_Toc183194535)

[Version Choice 2](#_Toc183194536)

[Prerequisites 2](#_Toc183194537)

[Installation 2](#_Toc183194538)

[Exe version 2](#_Toc183194539)

[Python Source Code Version 2](#_Toc183194540)

[Using DeuteRater v6 3](#_Toc183194541)

[Activating the Software 3](#_Toc183194542)

[Create Guide File 4](#_Toc183194543)

[Rate Calculation – General 7](#_Toc183194544)

[Rate Calculation – Extract 9](#_Toc183194545)

[Rate Calculation – Provide Time and Enrichment 11](#_Toc183194546)

[Rate Calculation – Combine Extracted Files 13](#_Toc183194547)

[Rate Calculation – Calculate Baseline Enrichment 13](#_Toc183194548)

[Rate Calculation – Rate Calculation 13](#_Toc183194549)

[Settings 15](#_Toc183194550)

[Settings Menu 15](#_Toc183194551)

[Adjusting Defaults or non-Menu Settings 20](#_Toc183194552)

[Guide\_settings.yaml 21](#_Toc183194553)

[Settings.yaml 22](#_Toc183194554)

[Adjusting Resource Files 27](#_Toc183194555)

[Adding new elements 28](#_Toc183194556)

[Adjusting Amino Acids 28](#_Toc183194557)

[Adjusting Post Translational Modifications 29](#_Toc183194558)

[Adding new Labeling Schemes 31](#_Toc183194559)

[Citations 31](#_Toc183194560)

# Introduction

DeuteRater is a software tool designed to allow scientists to perform kinetic proteomic and lipidomic experiments on small mammals or cell culture. It is related to our previously published DeuteRater software package which does the same for only kinetic proteomics. The DeuteRater v6 source code is available at [] with an .exe available at []. This readme will walk through the use of the software. For proper experimental design please reference the paper [].

## Version Choice

It is recommended that unless you need to alter the source code for some reason that you download the .exe as it is easy to use and should be resistant to bugs caused by the underlying python modules updating.

## Prerequisites

Both versions require a Windows operating system (they have only been tested on Windows 10, but other Windows versions may work as well).

Python source code was created in Python version 3.8.1, therefore that is the minimum python version recommended. Most imports are included in the basic python install, however you must ensure the follow packages are installed:

* PyQt5
* tqdm

# Installation

## Exe version

Download the zip folder from []. Extract the folder to a location you can find easily on your hard drive. Inside the unzipped folder will be an .exe file:

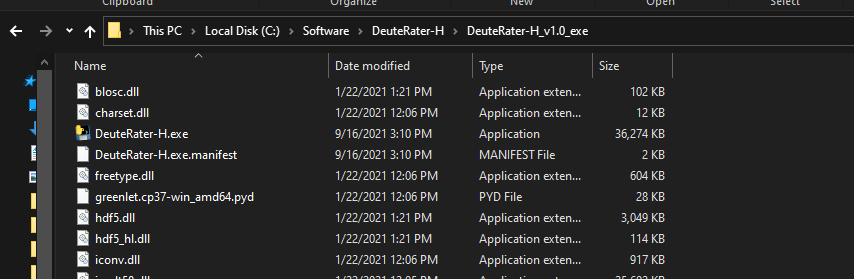
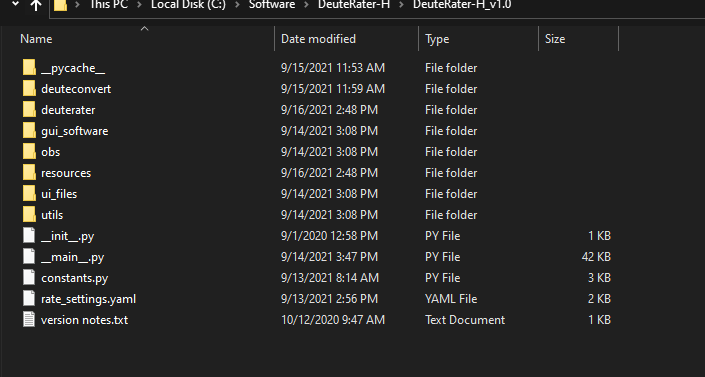


Figure 1 UPDATE PICTURE

Double click this .exe (or call it from the command prompt) to open DeuteRater v6. Nothing can be removed from this folder without risk of causing an error. If the position of the .exe is inconvenient you can right click on the .exe file and select “Create shortcut”. The shortcut can be moved anywhere you desire without causing an error.

## Python Source Code Version

Download the code from []. All folders must be in the same directory for the code to work. Use a command prompt to call the \_\_main\_\_.py using python:



It doesn’t matter if you call the full path including \_\_main\_\_.py, just the parent folder path, or cd into the directory and call \_\_main\_\_.py from there.

# Using DeuteRater v6

## Activating the Software

With the .exe version, double click on the .exe file or call the .exe on the command prompt. If using the python version, call \_\_main\_\_.py from the command line. This will open two windows. The first is the command prompt if you did not use it to open DeuteRater v6. It will be used to display progress bars and any errors that cause DeuteRater v6 to fail:



Next is the main interface window that governs all of DeuteRater v6’s operations:

A screenshot of a computer

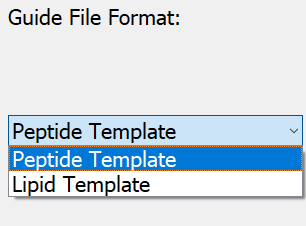
Description automatically generated

We will go through each step in the workflow with input and output requirements. The first step is to collect the information we need to start the analysis.

## Create Guide File

On the left of the interface is the “Create Guide File” button. A Guide File is a file containing all of the peptides or lipids you expect to see in your mass spectrometry files along with information to assist DeuteRater v6 in locating the appropriate features, such as retention time and mass, as well as data that will be needed later such as the peptide sequence.

How the guide file is created is governed by the pulldown menu below the “Create Guide File” button. There is different template guide file for peptide and lipid data respectively. The peptide template is the default option.



When you click “Create Guide File”, you will be prompted for a location and a save file name. What you name this file and where you put it do not matter, so long as you can find it again. The file will be a .csv (comma separated values) file which can be opened in any spreadsheet program such as Microsoft Excel.

The following columns must be filled out or analysis will not proceed:

Peptide data: Sequence, Protein ID, Precursor Retention Time (sec), Precursor m/z, and Identification Charge

Lipid data: Lipid Name, Lipid Unique Identifier, Precursor Retention Time (m/z), Precursor m/z, cf

For peptide data, the following columns are needed for analysis but will be filled in by DeuteRater v6 if left blank (do note that if any blanks are present in any of these columns, all data in any of these columns will be overwritten): Peptide Theoretical Mass, cf, neutromers\_to\_extract, and literature\_n

All other columns, including any other columns you want to add, are for the user’s information and may be left blank or filled with data as you wish. A description of all columns included in the guide file templates are as follows:

Sequence – the amino acid sequence. It should be in the form of single letter amino acid code. Post translational modifications (PTMs) should be included here. The recognized PTMs are found in the file “aa\_elem\_compositions.tsv” in the resources file of your DeuteRater v6 download. This will be detailed in sections “Adjusting Amino Acids” and “Adjusting Post Translational Modifications”.

**Peptide columns:**

Protein ID – a unique identification code for the protein this peptide belongs to.

Protein Name – common name of the parent protein

Precursor Retention Time (sec) – the middle of the elution peak for this peptide. Time unit is set in the settings menu (default is seconds)

rt\_start – the start of the elution peak.

rt\_end – the end of the elution peak.

rt\_width – the width of the elution peak.

Precursor m/z – observed m/z of the peptide

Peptide Theoretical Mass – the theoretical neutral mass of this peptide

Identification Charge – observed charge state

ptm – post translational modifications observed

avg\_ppm – the average ppm mass error in the observed peptide

start\_loc – the location the peptide begins in the protein

end\_loc - the location the peptide ends in the protein

num\_peptides – number of peptides observed for this protein in this analysis

num\_unique – number of peptides observed for this protein in this analysis

accessions -

species – the species this protein is from

gene\_name – the identifier for the gene associated with this protein

protein\_existence – a code to indicate the confidence that this protein is biologically relevant

sequence\_version – a legacy setting from a previous version. Leave it blank.

cf – chemical formula in standard format (e.g. C34H60N10O10)

neutromers\_to\_extract – number of isotopes to consider

literature\_n – the number of deuteriums that can be incorporated

**Lipid columns:**

Lipid Name – common name of the parent lipid

Lipid Unique Identifier – the unique name for the specific lipid species

Precursor m/z – observed m/z of the lipid

Precursor Retention Time (sec) – the middle of the elution peak for this lipid. Time unit is set in the settings menu (default is seconds)

Identification Charge – observed charge state

cf – chemical formula in standard format (e.g. C7H16NO2)

neutromers\_to\_extract – number of isotopes to consider

Adduct – the molecular-ion complexes used for mass spectrometry analysis. For instance, a lipid molecule can create adducts with sodium ions, ammonium ions, or hydrogen ions, each yielding distinct m/z values while representing the same molecular species.

Adduct\_cf – the chemical formula of the lipid including the adduct

literature\_n – the number of deuteriums that can be incorporated. You likely won’t have literature n-values for lipids, so leave this column blank and DeuteRater will calculate empirical n-values to populate the column

Only molecules you put into this document will be included in the search, so if you wish to search for multiple post translational modifications, or multiple charge states each must be on a separate line.

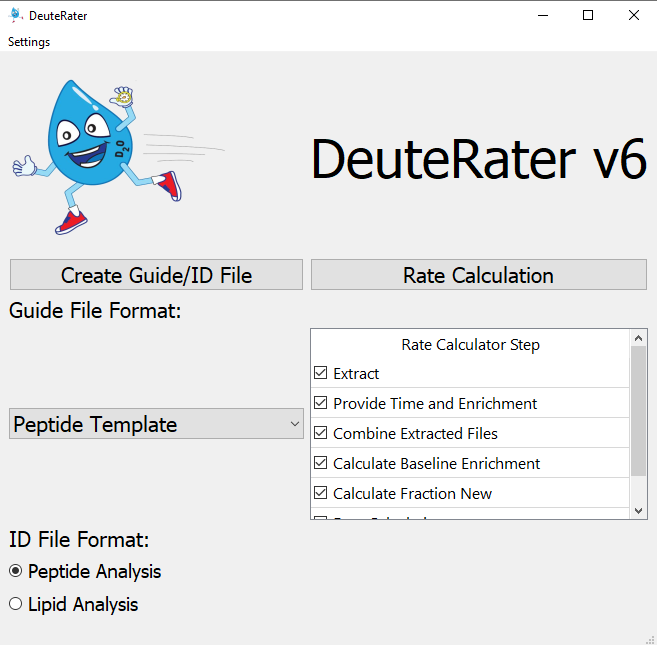
If the necessary columns are not completely filled or there is a problem auto-filling the necessary columns, an error message will appear as shown below. Correct the problem, save the file, and proceed.

A screenshot of a cell phone

Description automatically generated

## Rate Calculation – General

The main portion of DeuteRater v6 is the Rate Calculation button (red box below) which starts the analysis by going through each step checked in the “Rate Calculator Step” table (blue box below):



We will go over each step individually but first, a few general rules about it. All steps that are checked must be next to each other. If there is a gap, you will get an error message:

A screenshot of a message

Description automatically generated

No matter which step is checked, you will be asked to select a folder at first:

A screenshot of a computer

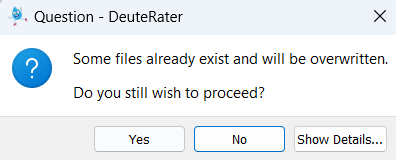
Description automatically generated

The names of the output files are always the same. If you have used the folder you select, you will get two messages. The first will notify you that there is a settings output (settings.yaml) present. You will be given the option to use the settings in the folder (“Use Settings”) or overwrite this file and use the settings you have currently set in DeuteRater v6 (“Overwrite”):

A screenshot of a computer

Description automatically generated

Second, if there are output files for a step that is currently checked present in your folder, you will be given a warning that you are going to overwrite output files. “Yes” overwrites the files, “No” returns you to the main DeuteRater v6 interface, and “Show Details” will list the files that will be overwritten so you may make an informed decision. You can rename your files to avoid them being overwritten, but that must be done manually. DeuteRater v6 will not do it for you:



In addition to output spreadsheet files, two graph folders will be produced during the Rate Calculation step. If this step is redone in the same parent folders, the graph subfolders will be deleted with all their contents.

If a file is open in another program while an overwrite is occurring, it cannot be overwritten. You will be given an error message and returned to the main interface. This will also occur if you are attempting to write to a folder which DeuteRater v6 does not have permission to write to for whatever reason.

Each step will produce a .tsv output file. These are tab separated text documents which can be opened in any text editor or spreadsheet program like Microsoft Excel. The .tsv step from a one step can be used as the input file for the next step in the Rate Calculator, allowing you to start and stop if necessary (e.g. to redo part of the analysis with different settings). Be warned, that except for the guide file, DeuteRater v6 will only check your input file for proper filetype and column names. It will not check that the data makes sense. Be careful changing intermediate files before re-using them as an input file for a later step, because changes may cause DeuteRater v6 to error out or give nonsensical results.

When running a worklist, rate\_settings.yaml will be saved in your output file. This should allow you to track what settings were used for each analysis should that be necessary.

## Rate Calculation – Extract

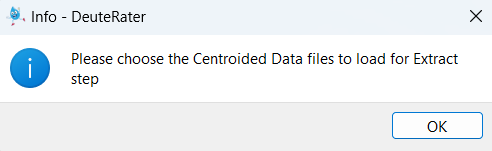
The first step of Rate Calculation is “Extract”. The purpose of the Extract step is to find neutromers in centroided MS1 mass spectrometry data. This requires two things. The first is the Guide file that you created earlier (see “Create Guide File” section) and the second is one or more .mzml files. The .mzML files need to be centroided and ms level 1 only. If you require a utility to transform your mass spectrometry data, the msconvert tool from proteowizard can do this1. Us the peakpicking setting to centroid.

First you will be asked to provide the guide file:

A screenshot of a computer

Description automatically generated

Navigate to wherever you have the guide file and select it. You will then be prompted to select your .mzML files. You may select one or several at once:



This may prompt another window warning about overwriting files if you selected files that have already been extracted. This is because DeuteRater v6 does not know the filenames of the extracted outputs until you choose the .mzML files to extract, so cannot determine which files may be overwritten.

Each .mzML you provided to DeuteRater v6 will have a corresponding output file. The file will have the same name as the original with a .tsv extension instead of an .mzML extension. The file will be the same as the guide file with several extra columns:

Id\_index – an internal index of the file

mz – calculated m/z value

Z – charge state

rt – retention time in minutes

n\_isos – neutromers to extract

mass – neutral mass

mzs – the actual mzs found in the .mzML

abundances – the actual abundances found in the .mzML

lookback\_mzs – mz of the isotope peak in front of the first neutromer extracted if the extractor is extracting on an isotope other than the first

lookback\_abundances – abundance of the isotope peak in front of the first neutromer extracted if the extractor is extracting on an isotope other than the first

lookahead\_mzs – mz of the isotope peak behind the last neutromer extracted if the extractor is extracting insufficient neutromers.

lookback\_abundances - abundance of the isotope peak behind the last neutromer extracted if the extractor is extracting insufficient neutromers.

rt\_min – minimum retention time observed

rt\_max – maximum retention time observed

baseline\_signal – abundance of the baseline in the area around this neutromer envelope

signal\_noise – irrelevant

mads – an error metric

mzs\_list – only filled for chromatography division setting.

intensities\_list – only filled for chromatography division setting.

rt\_list – only filled for chromatography division setting.

baseline\_list – only filled for chromatography division setting.

num\_scans\_combined – the number of scans used to generate relevant values.

mzml\_path – the path to the mzml file used for this data

file\_name –

mzml\_name –

name\_check –

Extraction\_Updated –

Extraction\_Error –

row\_num –

Columns mzs to num\_scans\_combined are only filled in if data was found for this row. If data was not found, and all of these columns are blank, the rows will be discarded in step “Combine Extracted Files”.

You will see progress bars on the black command prompt that appeared with the interface (or the normal command prompt if you called the program from there). Do note that if you have “Provide Time and Enrichment” checked, you will be prompted to fill out the tables associated with that step after selecting .mzMLs but before the actual extraction happens. This is so you can walk away from the analysis without it stalling to ask for your input halfway through.

## Rate Calculation – Provide Time and Enrichment

The purpose of the “Provide Time and Enrichment” is to provide information on the .mzml files you are using. If you are starting with this step you will be asked to select the output .tsvs from an extractor step, otherwise it will continue with the data that was used in the extractor. If step is checked you will be provided with a table like the following:

A screenshot of a computer

Description automatically generated

Each extracted .tsv file will be present in the “Filename” column. You must fill out the “Time” column and the “Subject ID” column.

“Time” must be the time since the beginning of deuterium labeling in the subject. Days is the best unit for time. DeuteRater v6 can use any time unit so long as it is consistent, but several later filters are calibrated for time data being in units of days.

“Subject ID” is a label that will applied to your data. Different Subject IDs will not be mixed at later steps in the analysis. This can be useful for separating data from different subjects in the same analysis, separating different tissues for the same subject, or for simply providing an identifier to this data which will be preserved as a column in every output and as part of the title of every graph. Do keep in mind that to a computer “A” and “a” are different and a space “ “ is considered a character so be careful to make each Subject ID for the same subject (or whatever you wish to mark) exactly the same.

Ctrl+c for copy, Ctrl+v for paste and del and backspace for delete are enabled for this table and the next one.

Fill out the table and hit the green “Proceed” button. The red “Cancel” button will prevent further analysis and return you to the main interface. If there is a problem with the data you entered, such as a missing value or a non-numeric or negative time, a warning message will appear. You must correct the problem

After you hit proceed, another window will appear:

(old picture deleted)

DeuteRater v6 needs to know how the deuterium enrichment changes over time for each subject. To start, adjust the “Number of Enrichment Measurements per Subject” box to the maximum number of enrichment measurements any subject has and press the “Update Table” button. This will ensure each subject has enough row to fill in. If subjects have different numbers of measurements, for example if S2 in the table above has 10 measurements, but S3 has only 5, set the “Number of Enrichment Measurements per Subject” to the higher value (10) and leave the extra rows for the subject with less blank. So long as both “Time” and “Enrichment (in Decimal)” are blank, DeuteRater v6 will discard the empty column with no problem.

“Time” is the Time using whatever units you used in the last table (days are the preferred unit). “Enrichment (in Decimal)” is the amount of deuterium in the body water of the subject, with natural enrichment being 0. Use decimal (so .05 not 5%). These do not have to correspond in any way to the times of the .mzML files you provided.

You must have at least 4 measurements per subject or the fitting spline will not be able to be fit. Also every row must be completely filled or blank except for Subject Id.

As with the previous table, Ctrl+c, Ctrl+v, del, and backspace are enabled. If there is an error filling out the table you will be told to correct before proceeding.

The output file is called “time\_enrichment\_data.tsv”. This file contains the exact contents of each table you filled out with a spacing column in between them and the second table’s Subject ID column called “Subject ID Enrichment” to differentiate the two columns.

Also created is a folder called “Enrichment\_Graphs”. Inside will be a graph for every subject, with red dots indicating the data you provided and the black dotted line representing the spline fit that DeuteRater v6 created. Names of graphs are subject names. Filetype is determined in the settings menu.

Warning: the next step uses the filenames in the “time\_enrichment\_data.tsv” file to find the extracted data. if these files have been renamed or moved since the “time\_enrichment\_data.tsv” you must alter the filenames to match the current position if you want to use “time\_enrichment\_data.tsv” as an input.

## Rate Calculation – Combine Extracted Files

This button merely combines the outputs from the last two steps. It takes “time\_enrichment\_data.tsv” as an input file if starting from this step.

The output file is called “combined\_extracted\_files\_output.tsv”. This output contains the “Precursor Retention Time (sec)”, “Protein ID”, “Protein Name”, “Precursor m/z”, “Identification Charge”, “Homologous Proteins”, “n\_isos”, “literature\_n”, “Sequence”, “cf”, “abundances”, and “mzs” from the Extract output files. It also has the “time” and “sample\_id” which are the values from the “Time” and “Subject ID” column from the first Provide Time and Enrichment table. Finally, it has the “Time Enrichment” and “Enrichment Values” columns which contain the data from the “Time” and “Enrichment (in Decimal)” columns of the second Provide Time and Enrichment table.

## Rate Calculation – Calculate Baseline Enrichment

This button merely calculates the baseline isotopic abundances of each sequence. It takes “combined\_extracted\_files\_output.tsv” as an input file if starting from this step.

Output file is “delta\_by\_enrichment.tsv”. It is identical to “combined\_extracted\_files\_output.tsv” except for the “Theoretical Unlabeled Normalized Abundances”. The Theoretical Unlabeled Normalized Abundances column contains the baseline isotopic abundance of a given sequence. It will also sometimes contain text saying “less than () labeling sites” where () is some number. There is a setting in the settings menu which sets a minimum number of possible deuteriums a peptide sequence must have in order to be considered further. Rows with this text will be dropped in the next step.

## Rate Calculation – Rate Calculation

This step performs the actual rate calculation on the peptides. The different rows with the same peptide sequence (including modifications) and same subject will be combined, so different charge states and different times. Nothing else will be combined at this stage.

It takes “delta\_by\_enrichment.tsv” as an input file if starting from this step.

There are three outputs, “rate\_by\_sequence.csv” which has numerical data, and two graph folders, “Graph\_Folder\_Isotopes” and “Graph\_Folder\_Optimization”. Graph\_Folder\_Optimization is optional based on the settings.

rate\_by\_sequence.csv is a .csv file which should be easy to open in any spreadsheet program. It has the following columns:

Subject ID – the Subject ID for the data analyzed in this row.

Protein ID – the Protein ID for the data analyzed in this row.

Protein Name – the Protein Name for the data analyzed in this row.

Sequence – the amino acid sequence for the data analyzed in this row.

Abundance rate – the actual rate calculated for this peptide. Will be blank if rate could not be calculated for some reason.

Unique Timepoints – the number of unique times this peptide was observed in based on the times provided for your extracted .mzml files.

Number of Measurements – number of measurements (isotope data for a measured time) included in the graph. Will include different timepoints, different charge states, and replicates. Will not include any points rejected for running afoul of any of DeuteRater v6’s filters.

Approximate Variance – an error metric for the rate calculation. Smaller is better. Will be blank if rate could not be calculated for some reason.

mean of all absolute residuals – an error metric describing how far the points are from the fitted line. Smaller is better. Will be blank if rate could not be calculated for some reason.

num times – number of different times used. Will include any duplicates of unique times, but not any times that were rejected for not passing filters.

n\_isos – number of isotopes used in the calculations, as in previous files

num measurements – n\_isos multiplied by num times or Number of Measurements since both are the same. This is done since each isotope is a separate line, and therefore each isotope of each measurement is a point on the resulting graph.

time values – a semicolon delimited list of the times that passed the filters.

dropped points – the number of points dropped for failing any of the filters.

M0 constantly decreasing – a True or False value indicating whether M0, the isotope composed of only the lightest elements, is constantly decreasing over time. In practice this means does not go over a certain amount over its previous value (there is a setting that governs this cutoff). Since heavy water is being given to the subject, M0 should be constantly stable or decreasing. This is for the user’s information, peptides with False are not discarded unless all timepoints have only a single measurement. Will be blank if rate could not be calculated for some reason.

Error Column – contains a description of any error that occurred in processing this row. Usually insufficient timepoints, the filters dropped too many timepoints or the mean of absolute residuals was above a cutoff. Any row with text in this column will be kept out of further analysis.

The folder named Graph\_Folder\_Isotopes contains graphs of all the fits, with lines being the fits, diamonds being the baselines and circles being the actual points used. The legend indicates which color corresponds to which isotope peak. Graph files are named [Subject\_ID]\_[peptide\_sequence]\_isotopes.

The folder named Graph\_Folder\_Optimization has graphs showing the error minimization to determine k. This folder is not created by default. The blue line with points along it indicates the amount of error (y-axis) for each attempted k value (x-axis) with k being the rate. The lighter blue vertical line indicates the k-value that was actually chosen. Graph files are named [Subject\_ID]\_[peptide\_sequence]\_optimization.

# Settings

DeuteRater v6 has several optional settings. The most common settings to adjust are accessible from the settings menu in the main interface. This is only useful for the current run of DeuteRater v6. If you close and then re-open the program, all settings will return to their default values. Default settings and any settings not in the settings menu can be altered but be warned that this has fewer protections than the settings menu, so it is recommended to be cautious and save a backup of the relevant files before alteration. Both the settings menu and altering default values will be discussed in this section.

When running a worklist, rate\_settings.yaml will be saved in your output file. This should allow you to track what settings were used for each analysis should that be necessary.

## Settings Menu

## The settings menu for is accessed through using the “Options” pull-down in the main interface and selecting “Rate Calculation Settings” which is currently the only option in the pull down:

## A screenshot of a computer Description automatically generated

This will open the following window:

A screenshot of a computer

Description automatically generated

To start we will discuss the buttons on the right of the screen:

**Save Settings –** will save the currently displayed settings.

**Load Peptide Settings –** this will load default settings for running peptide data. These default settings in DeuteRater are always the default peptide settings. Some important settings to keep in mind for this option: Use Empirical N Values will default to False, Asymptote Calculation will default to Fixed, and Fraction new calculation type will default to combined.

**Load Lipid Settings -** this will load default settings for running lipid data. Some important settings to keep in mind for this option: Use Empirical N Values will default to True, Asymptote Calculation will default to Variable, and Fraction new calculation type will default to abundance.

**Load Settings –** will prompt you to navigate to and select a Rate\_Settings.yaml file from a previous run of DeuteRater v6. The settings from that file will then be displayed. You still must save the settings for them to be applied.

**Exit –** Exit the settings menu. If the settings displayed in the menu are not the same as those currently in use by DeuteRater v6, you will be prompted to save. You may choose to save or reject your changes.

The settings are grouped by the process that uses them, though the names aren’t always an exact match; Both the N Value Calculations and Combine Extraction settings are in the Combine Extracted Files step. We will go through the settings one at a time:

# General

**Recognize available cores –** most steps of DeuteRater v6 use multiprocessing to increase speed. If this setting is “Yes”, DeuteRater will determine how many cores you have and attempt to use all of them.

**Default Cores to Use –** the number of processor cores to use for calculations if “Recognize available cores” is set to “No”.

**Graph file type –** the file type for the output graphs. Available options are png (default), svg, and None.

**Extract**

**Retention Time Unit –** if the provided retention times in the guide file are seconds or minutes.

**Extraction Time Window (min) –** the guide file provides a retention time for each feature to be searched for. This is the allowed error for that retention time while searching in the .mzML files for that feature. Value is one way, so allowed error is plus or minus this value. Time unit is minutes.

**Allowed m/z error (ppm) –** the guide file provides a Precursor m/z for each feature to be searched for. This is the allowed error for that m/z while searching in the .mzML files for that feature. Value is one way, so allowed error is plus or minus this value. The value is ppm error.

**Use Chromatography Division –** this setting attempts to find the best chromatography peak for extraction. It can be turned off (“No”) used to compare the chromatography of different charge states of the same peptide within each .mzML file (“Intrafile”) or to compare the same peptide and its charge states between files (“Interfile”) This will increase calculation time, generate extra intermediate files and may remove data.

**N Value Calculations**

**N-Value Confidence Value Limit –** A standard deviation-based filter for n-value calculations. Calculated by dividing the n-value standard deviation by the n-value. This is later used in the fraction new calculation step to ignore any rows with a confidence value (cv) above the set limit.

**Make N-Value Calculation Graphs –** If turned on, this will greatly decrease the speed and performance! It most cases, it should only be used for debug purposes or if you want to visualize how DeuteRater calculates empirical n-values. It will create a graph folder and for each n-value calculation, it will save a graph showing the number of Deuteriums on the x-axis and the standard deviation on the y-axis.

**Output N-Value Calculation Data –** Similar to the last setting, but it won’t impact the performance as much. Will output a .tsv file containing the following features for each peptide/lipid: Index, cf (chemical formula), sample\_group, n\_value, std\_dev (standard deviation), all\_std\_dev, and all\_n\_D\_values. The last two columns can be used to generate the same graphs the previous setting generates (all\_n\_D\_values on the x-axis and all\_std\_dev on the y-axis).

**Combine Extraction**

**mz proximity filter (ppm) –** there is a possibility that two different peptide sequences will have the same mass and retention time (especially if multiple charge states are considered). Because DeuteRater v6 does not consider fragmentation data, there is no way to give preference to one sequence over another. Therefore, if multiple sequences are too close in both m/z and retention time, all these sequences are deleted. M/z within this setting of another peptide are determined to be close in m/z.

**Retention Time proximity filter (min) –** there is a possibility that two different peptide sequences will have the same mass and retention time (especially if multiple charge states are considered). Because DeuteRater v6 does not consider fragmentation data, there is no way to give preference to one sequence over another. Therefore, if multiple sequences are too close in both m/z and retention time, all these sequences are deleted. Times within this setting of each other are considered close in time.

**Calculate Baseline Enrichment**

**Minimum Allowed N Value –** the minimum amount of potential deuterium sites a peptide/lipid must have to be considered for further analysis.

**Minimum Peptide Length –** the minimum length of a peptide to be considered for further analysis.

**Rate Calculation**

**Label Key –** the amount of deuterium incorporated into any amino acid is determined by the biochemical pathways in an organism. This may vary by organism, time of life or disease state. This tells DeuteRater v6 what the potential deuterium incorporation of each amino acid is. The default “tissue” is standard adult mammalian tissue. If you wish to examine or change these values or add a new label key see the “Adding new Labeling Schemes” portion of this read me.

**Use Empirical N Values –** Default for peptides is “No” and the default for lipids is “Yes”. Unlike amino acids, there isn’t an established database for lipid constituent n values, so you’ll likely need to calculate them with empirical data. If you select “No”, DeuteRater will expect you to populate the literature\_n column in your ID/guide file, and then transfer those values over to the n\_value column for use in the workflow. If you select “Yes”, leave the literature\_n column empty, and DeuteRater will populate the n\_value column with empirical n values.

**Abundance Manual Bias –**

**Minimum allowed rate –** the lowest peptide/lipid turnover rate that is considered valid.

**Maximum allowed rate –** the highest peptide/lipid turnover rate that is considered valid.

**\*\*\*Error Graph Option –** Whether or not to generate the Graph\_Folder\_Optimization folder and its associated graphs of how the error was minimized to produce the reported rate value. “none” produces no graphs. “approximate” produces a parabolic approximation of the error minimization which is slightly easier to see and faster to produce. “exact” produces the exact error minimization curve used to calculate the rate value for each peptide sequence.

**Remove Outlier Data Points –**

**Proliferation Adjustment –**

**Minimum Non-zero points (rate) –** the number of unique times a peptide/lipid must be observed to allow a calculation of the peptide turnover rate.

**Combined Manual Bias –**

**Asymptote Calculation –**

**Neutromer Spacing Manual Bias –**

**Separate Lipid Adducts –**

**Abundance type -**

**Verbose Rate –**

**Bias Calculation –**

**Maximum fraction new standard deviation –**

**Fraction new calculation type –**

**Outlier Removal Z-Score Cutoff –**

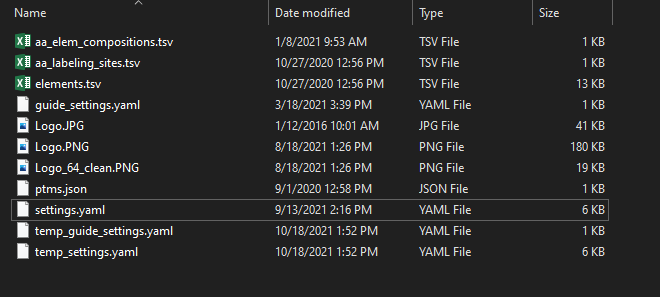
**R-Squared Threshold -**

## Adjusting Defaults or non-Menu Settings

In the folder containing your .exe (or your \_\_main\_\_.py for the python version) is a folder called “resources”:

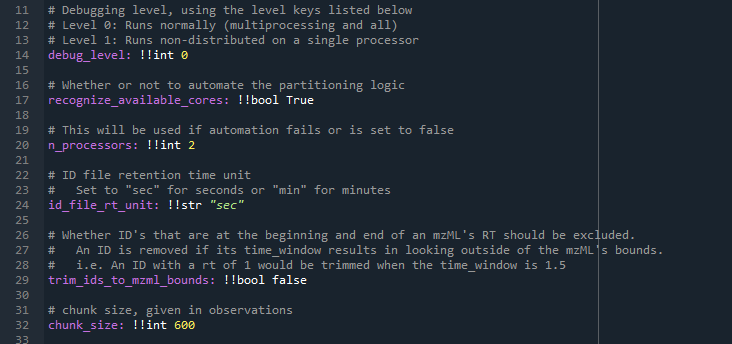


In this folder are several files that DeuteRater v6 uses to load various values. These must be altered to change default values or reach settings the settings menu does not provide access to. The ones we are concerned with are settings.yaml and guide\_settings.yaml:



Do note that DeuteRater v6 has limited checks on if these files are good, so be careful when altering them. Also in this folder you will often find temp\_guide\_settings.yaml and temp\_settings.yaml. These are temporary files created by DeuteRater v6 to store temporary changes to the settings made at run time (such as if the user adjusts a value in the settings menu) and so will be deleted and recreated every time you run DeuteRater v6. The files without the “temp\_” prefix are the ones we want here.

When you open the .yaml (any code or text editor should be able to do this) you will see something like this:



This is in a code editor to aid visualization of the different points of interest. Anything after a # (gray text in the image above) is a comment for those altering the file. DeuteRater v6 will ignore them and so they may be safely added, altered or deleted.

The variable names are shown in purple above, they are on the far left side of the screen and followed by a colon. They cannot have spaces. This is the name of the variable and they should not be altered. What each variable does will be detailed after all portions of the .yaml are described.

Separated from the colon by a space are two exclamation points followed by a few letters, like !!int or !!str. this tells DeuteRater v6 what kind of variable this is. !!int is and integer, !!float is a number with a decimal point in it, !!bool is either true or false, and !!str is text. These should not be altered.

Separated from the variable type by another space is the value. You may change this. You should know what the variable is in order to set reasonable values. Also the variable type must not change, so if you wish to change an integer it must still be and integer. Text must still have quotations.

If you wish to change a default value change it in these .yaml files. We will go through each setting now.

### **Guide\_settings.yaml**

Guide\_settings.yaml this is related to the Create Guide File button. Most are not useful if using the “Template” option, but will be here for completeness:

**Mass\_cuttoffs –** the mass thresholds for number of isotopes in Peaks template files. the format is [neutral mass, n\_isos].

**\*\*\*rt\_proximity\_tolerance –** there is a possibility that two different peptide sequences will have the same mass and retention time (especially if multiple charge states are considered). Because DeuteRater v6 does not consider fragmentation data, there is no way to give preference to one sequence over another. Therefore, if multiple sequences are too close in both m/z and retention time, all of these sequences are deleted. Times within this setting of each other are considered close in time. Relevant for the PEAKS settings, not Template.

**mz proximity filter (ppm) –** there is a possibility that two different peptide sequences will have the same mass and retention time (especially if multiple charge states are considered). Because DeuteRater v6 does not consider fragmentation data, there is no way to give preference to one sequence over another. Therefore, if multiple sequences are too close in both m/z and retention time, all of these sequences are deleted. M/z within this setting of another peptide are determined to be close in m/z. Relevant for the PEAKS settings, not Template.

**\*\*\*Start\_time –** the minimum retention time allowed. Relevant for the PEAKS settings, not Template.

**study\_type -** the amount of deuterium incorporated into any amino acid is determined by the biochemical pathways in an organism. This may vary by organism, time of life or disease state. This tells DeuteRater v6 what the potential deuterium incorporation of each amino acid is. The default “tissue” is standard adult mammalian tissue. If you wish to examine or change these values, or add a new label key see the “Adding new Labeling Schemes” portion of this read me. Relevant for the PEAKS settings, not Template.

**aa\_elemental\_composition\_path –** the name of the file in resource file that has the amino acid elemental composition. This file is detailed in the “Adjusting Amino Acids” section of this read me.

**aa\_labeling\_sites\_path –** the name of the file in resource file that has number of hydrogens for each amino acid that can be labeled with deuterium. This file is detailed in the “Adding new Labeling Schemes” section of this read me.

**elements\_path –** the name of the file in resource file that has details on each element and their isotopes. This file is detailed in the “Adding new elements” section of this read me.

**\*\*\*post\_translational\_modifications\_path –** the name of the file in resource file that has details on post-translational modification. This file is detailed in the “Adjusting Post Translational Modifications” section of this read me.

**\*\*\*Min\_charge\_state –** when making a guide file from PEAKS, charge states for each peptide will be automatically generated. This sets the minimum charge state. No function for “Template” settings.

**\*\*\*Max\_charge\_state –** when making a guide file from PEAKS, charge states for each peptide will be automatically generated. This sets the maximum charge state. No function for “Template” settings.

### **Settings.yaml**

Settings.yaml contains the settings used in the “Rate Calculation” button. Some are present in the settings menu, some are not.

**debug\_level –** if this is at 0, DeuteRater v6 will run normally. If it is set at 1, it will deactivate multiprocessing for easier troubleshooting. Should only be changed from 0 if troubleshooting.

**recognize\_available\_cores –** most steps of DeuteRater v6 use multiprocessing to increase speed. If this setting is “Yes” DeuteRater v6 will determine how many cores you have and will attempt to use all of them. Default value for “Recognize available cores” in the settings menu.

**n\_processors –** the number of processor cores to use for calculations if “recognize\_available\_cores” is set to False. Default value for “Default Cores to Use” in the settings menu.

**id\_file\_rt\_unit –** if the provided retention times in the guide file are seconds or minutes. Allowed values are “sec” for seconds or "min" for minutes. Any other value will cause an error. Default value for “Retention Time Unit” in the settings menu.

**trim\_ids\_to\_mzml\_bounds –** used in the extractor. if this is true, any id in an .mzML that is closer than the “time\_window” setting to the start or end of the file, it will be discarded.

**chunk\_size –** used in the extractor. If a file is less than "chunking\_method\_threshold" times "chunk\_size" lines long, then split the identification file evenly among the processors. This is to preserve memory when analyzing very large .mzMLs.

**chunking\_method\_threshold –** used in the extractor. If a file is less than "chunking\_method\_threshold" times "chunk\_size" lines long, then split the identification file evenly among the processors. This is to preserve memory when analyzing very large .mzMLs.

**max\_valid\_angle –** used in the extractor. neutromer envelopes in adjacent scans are compared to each other as vectors. If the vector of the next scan in time and the current summed scans have an angle greater than this setting, the next scan will not be incorporated into the total.

**peak\_ratio\_denominator –** used in extractor. For a neutromer envelope to be added to an id, it’s m0 value must be greater than 1/peak\_ratio\_denominator of the maximum m0 currently included in that id.

**time\_window –** the guide file provides a retention time for each feature to be searched for. This is the allowed error for that retention time while searching in the .mzML files for that feature. Value is one way, so allowed error is plus or minus this value. Time unit is minutes. Default value for “Extraction Time Window (min)” in the settings menu.

**ppm\_window –** the guide file provides a Precursor m/z for each feature to be searched for. This is the allowed error for that m/z while searching in the .mzML files for that feature. Value is one way, so allowed error is plus or minus this value. Value is ppm error. Default value for “Allowed m/z error (ppm)” in the settings menu.

**label\_key –** the amount of deuterium incorporated into any amino acid is determined by the biochemical pathways in an organism. This may vary by organism, time of life or disease state. This tells DeuteRater v6 what the potential deuterium incorporation of each amino acid is. The default “tissue” is standard adult mammalian tissue. If you wish to examine or change these values, or add a new label key see the “Adding new Labeling Schemes” portion of this read me. Default value for “Label Key” in the settings menu.

**aa\_labeling\_sites\_path –** the name of the file in resource file that has number of hydrogens for each amino acid that can be labeled with deuterium. This file is detailed in the “Adding new Labeling Schemes” section of this read me.

**peak\_lookback –** Used in the extractor. The number of peaks to look for in front of the first neutromer for troubleshooting the extractor.

**peak\_lookahead** – Used in the extractor. The number of peaks to look for after of the last extracted neutromer for troubleshooting the extractor.

**baseline\_lookback –** Used in the extractor. The amount of m/z to examine for the baseline calculation both before and after the precursor m/z.

**min\_envelopes\_to\_combine –** Used in the extractor. If the extractor cannot find at least this number of scans to combine for a given id, no results are provided for that id.

**zscore\_cutoff –** Used in the extractor. A median absolute deviation test is used to identify outlier scans. this is done by getting the mean absolute deviation of an m/z value compared to the median m/z for that neutromer. This is the zscore used to determine if a value is an outlier, and so the scan should be dropped.

**\*\*\*rt\_proximity\_tolerance –** there is a possibility that two different peptide sequences will have the same mass and retention time (especially if multiple charge states are considered). Because DeuteRater v6 does not consider fragmentation data, there is no way to give preference to one sequence over another. Therefore, if multiple sequences are too close in both m/z and retention time, all of these sequences are deleted. Times within this setting of each other are considered close in time. Default value for “Retention Time proximity filter (min)” in the settings menu.

**\*\*\*mz\_proximity\_tolerance –** there is a possibility that two different peptide sequences will have the same mass and retention time (especially if multiple charge states are considered). Because DeuteRater v6 does not consider fragmentation data, there is no way to give preference to one sequence over another. Therefore, if multiple sequences are too close in both m/z and retention time, all of these sequences are deleted. M/z within this setting of another peptide are determined to be close in m/z. Default value for “Allowed m/z error (ppm)”.

**min\_aa\_sequence\_length –** the minimum length of a peptide to be considered for further analysis. Default value for “Minimum Peptide Length” in the settings menu.

**min\_allowed\_n\_values –** the minimum amount of potential deuterium sites a peptide must have to be considered for further analysis. Default value for “Minimum Allowed N Value” in the setting menu.

**\*\*\*max\_enrichment\_allowed –** the maximum allowed deuterium enrichment. This is mainly to ensure the values are in decimal not percent.

**\*\*\*error\_estimation –** Whether or not to generate the Graph\_Folder\_Optimization folder and its associated graphs of how the error was minimized to produce the reported rate value. “none” produces no graphs. “approximate” produces a parabolic approximation of the error minimization which is slightly easier to see and faster to produce. “exact” produces the exact error minimization curve used to calculate the rate value for each peptide sequence. Default value for “Error Graph Option” in the settings menu.

**min\_non\_zero\_timepoints\_rate –** the number of unique times a peptide must be observed at to allow a calculation of the peptide turnover rate. Default value for “Minimum Non-zero points (rate)” in the settings menu.

**min\_allowed\_timepoints\_enrichment –** the minimum number or timepoints that are required for an enrichment curve, and thus minimum entries on the second table in Provide Time and Enrichment. Default value for “Minimum timepoints (enrichment)” in the settings menu.

**minimum\_allowed\_sequence\_rate –** the lowest peptide turnover rate that is considered valid. Default value for “Minimum allowed rate” in the settings menu.

**maximum\_allowed\_sequence\_rate –** the highest peptide turnover rate that is considered valid. Default value for “Maximum allowed rate” in the settings menu.

**minimum\_sequences\_to\_combine\_for\_protein\_rate –** the minimum number of peptide rates that must be combined together for a protein rate to be considered valid. Default value for “Minimum number of peptide sequences per protein rate” in the settings menu.

**lowest\_allowed\_norm\_isotope –** Noise or mis-identification can lead to points that are theoretically impossible. All peaks after M0 should be increasing above their unlabeled baseline value. To ensure this is the case we multiply the baseline value of each isotope except M0 by this value, which should be less than 1. If any of the isotopes drop below this setting multiplied by their baseline value, all isotopes of that point are dropped. Default value for “Lowest allowed isotope value (fraction of unlabeled)” in the settings menu.

**highest\_allowed\_norm\_isotope –** Noise or mis-identification can lead to points that are theoretically impossible. M0 should be decreasing below its unlabeled baseline value. To ensure this is the case we multiply the baseline value of M0 by this value, which should be greater than 1. If M0 increases above this setting multiplied by its baseline value, all isotopse of that point are dropped. Default value for “Highest allowed isotope value (fraction of unlabeled)” in the settings menu.

**m0\_decreasing\_allowed\_noise –** M0 should be constantly decreasing, or at the very least staying steady. We wish to determine if that is the case. This value is multiplied by the baseline m0 value. If any point is higher in value than the previous point + baseline M0 x this setting, it is determined that M0 is not constantly decreasing. Default value for “M0 decreasing allowed noise (fraction of unlabeled M0)” in the settings menu.

**median\_absolute\_residuals\_cutoff\_single\_point –** the error metric for assessing the goodness of a peptide rate fit is determined by the distance of the points from their fit line. The median of the absolute value of these residuals is compared to a cutoff to determine if the rate curve is considered good or bad. Because single timepoints have fewer points to fit, the lines will be closer to the points in general and noise is more problematic so they need a special cutoff. This setting is that cutoff. If the median absolute residuals is above this value, the associated peptide rate will be discarded. Default value for “Median absolute residuals cutoff single point” in the settings menu.

**median\_absolute\_residuals\_cutoff\_two\_points –** the error metric for assessing the goodness of a peptide rate fit is determined by the distance of the points from their fit line. The median of the absolute value of these residuals is compared to a cutoff to determine if the rate curve is considered good or bad. Because two timepoints have fewer points to fit, the lines will be closer to the points in general and noise is more problematic so they need a special cutoff. This setting is that cutoff. If the median absolute residuals is above this value, the associated peptide rate will be discarded. Default value for “Median absolute residuals cutoff two points” in the settings menu.

**median\_absolute\_residuals\_cutoff\_general –** the error metric for assessing the goodness of a peptide rate fit is determined by the distance of the points from their fit line. The median of the absolute value of these residuals is compared to a cutoff to determine if the rate curve is considered good or bad. This is the cutoff for lines fit with many points. If the median absolute residuals is above this value, the associated peptide rate will be discarded. Default value for “Median absolute residuals cutoff general” in the settings menu.

**desired\_points\_for\_optimization\_graph –** if optimization graphs are created because of the error\_estimation in this .yaml or “Error Graph Option” in the settings menu, this determines the number of points in the graphs.

**intensity\_filter –** Used in the extractor. A filter used in many areas for scan combining. If a neutromer peak is below the intensity filter, it is not considered a valid scan and is discarded.

**ms\_level –** the ms level to use for the extraction. Note that this has not been tested with values other than 1 and may lead to unexpected results if altered.

**graph\_output\_format –** the file type for the output graphs. Default value for “Graph file type” in the settings menu.

**protein\_combination\_method –** whether to use average or median to combine peptide rates to calculate protein rates. Default value for “Graph file type” in the settings menu.

**verbose\_output –** whether or not to include extra columns in Final\_Protein\_Rates.csv. The extra columns are detailed in the “Rate Calculation – Combine Sequence Rates” section of this read me. Default value for “How to combine sequences to protein” in the settings menu.

**sampling\_rate –** used in the extractor if chromatography division is used. How many scans should be looked at when performing a gaussian filter on the scan intensities. Must be an odd number. Best value will vary by instrument.

**smoothing\_width –** used in the extractor if chromatography division is used. How many scans should be looked at when using a simple smoothing width. Best value will vary by instrument.

**smoothing\_order –** used in the extractor if chromatography division is used. What order the simple smoothing algorithm should be used. How many scans should be looked at when using a simple smoothing width. Best value will vary by instrument.

**allowed\_peak\_variance\_min –** used in the extractor if chromatography division is used. The amount of time in minutes that peaks can vary by between charges/adducts to be considered the same. Default = 0.1 for intrafile .2 for interfile. Best value will vary by instrument.

allowed\_neutromer\_peak\_variance – used in the extractor if chromatography division is used. The amount of scans allowed between neutromer peaks to be considered a valid peak. Best value will vary by instrument.

**rel\_height –** used in the extractor if chromatography division is used. A value used to determine how close to baseline the gaussian fitting algorithm peak widths will be chosen from. Default is .9 (90%). Best value will vary by instrument.

**adduct\_weight –** used in the extractor if chromatography division is used. The weight that should be used for adduct reproducibility when calculating the score for chromatography peak picking. Default = 1.0. Best value will vary by instrument.

**variance\_weight –** used in the extractor if chromatography division is used. The weight that should be used for neutromer peak variance when calculating the score for chromatography peak picking. Default = 1.0. Best value will vary by instrument.

**ID\_weight –** used in the extractor if chromatography division is used. The weight that should be used for distance from the ID File's RT when calculating the score for chromatography peak picking. Best value will vary by instrument.

**intensity\_weight –** used in the extractor if chromatography division is used. The weight that should be used for the intensity when calculating the score for chromatography peak picking. Best value will vary by instrument.

**how\_divided –** used in the extractor if chromatography division is used. What created EIC should chromatography be divided on. default = "combined". Other options, "m0", "m1", "m2". Best value will vary by instrument.

**use\_chromatography\_division –** this setting attempts to find the best chromatography peak for extraction. It can be turned off (“No”) used to compare the chromatography of different charge states of the same peptide within each .mzML file (“Intrafile”) or to compare the same peptide and its charge states between files (“Interfile”) This will increase calculation time, generate extra intermediate files and may remove data. Default value for “Use Chromatography Division” in the settings menu.

# Adjusting Resource Files

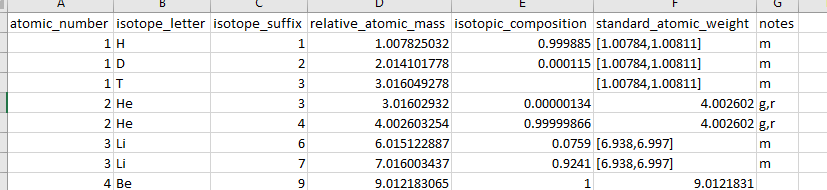
There are several locations in DeuteRater v6 where the elemental composition of a peptide, including fixed and variable post-translational modifications, are relevant. The resource files discussed in this section are used to tell DeuteRater v6 about what elements exist, what amino acids exist and what their element composition is, what post-translational modifications exist and what their elemental compositions are, and how many extra deuteriums can be biosynthetically incorporated into any amino acid or post-translational modification. All files discussed are in the “resources” subfolder of the folder containing your .exe or \_\_main\_\_.py version.



## Adding new elements

Adding new elements has two portions. If you wish to include new elements in the “Rate Calculation” step that is only doable in the python version. Currently there is an internal location in the DeuteRater v6 code which must know the element to add it. Currently, it knows, H, C, N, O, P, S, F, D (required deuterium for a standard or similar), Cl, Br, I, and Si. If you require a different element you must use the python version (the location to alter is the emass.py in the utils folder. Line 213 has the “master\_isotope” dictionary. Use the other elements as a model and add to the dictionary between the curly braces).

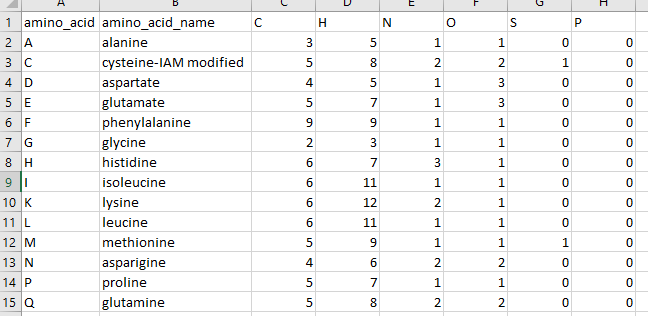
If you are using Peaks guide file generation, the elements are used to generate the theoretical mass of the peptides. The relevant file in the resources folder is “elements.tsv” which looks like:



This is based on data that came with the emass algorithm2. The only relevant columns are the isotope letter (the chemical symbol) and the “standard\_atomic\_weight” which is the atomic mass weighted by isotope abundance as you would find on the periodic table.

## Adjusting Amino Acids

The relevant file here is “aa\_elem\_compositions.tsv”. Inside the file looks like this:

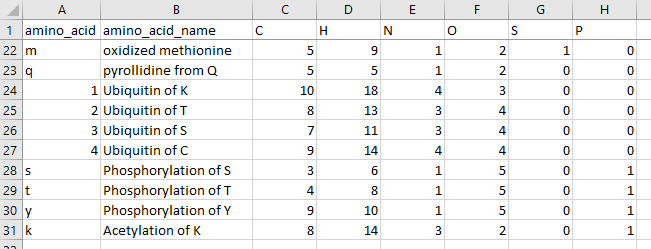


The first column lists the one letter code for the amino acid or modified amino acid. This does not have to be the traditional one letter code, or even a letter so long as it is unique and only one character long. The second column is the name, mostly for your benefit. For example, the cysteine above has the chemical formula for having an iodoacetamide modification and the name reflects this so we are not confused. All the other columns are elements that compose the amino acid. A few notes:

* the elemental composition is based on the elemental composition in the middle of a peptide chain. So the peptide bonds are formed. This is why alanine only has one Oxygen for example.
* Amino acids may be freely adjusted, renamed or have their composition changed. But remember that the change will affect all future DeuteRater v6 runs until it is changed again.
* if you wish you may add a new element to the right of the table. The column must be filled but it may be filled with zeroes. It must be a known element in the “elements.tsv” file discussed in section “Adding new elements”. “X” is not valid.
* There is an internal location in the DeuteRater v6 code which must know the element to add it. Currently, it knows, H, C, N, O, P, S, F, D (required deuterium for a standard or similar), Cl, Br, I, and Si. If you require a different element you must use the python version (the location to alter is the emass.py in the utils folder. Line 213 has the “master\_isotope” dictionary. Use the other elements as a model and add to the dictionary between the curly braces).
* If you add a new amino acid, remember to add the deuterium amount of any new amino acids you add as described in “Adding new Labeling Schemes”

## Adjusting Post Translational Modifications

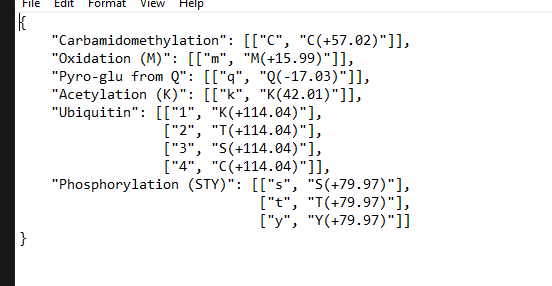
Post-translational modifications can be used in two different ways. The first, required no matter what, is to adjust the “aa\_elem\_compositions.tsv” as described in “Adjusting Amino Acids”. As an example:



These are variable modifications. Replace the relevant amino acid in your Guide file. So, if a methionine is oxidized, replace an “M” with an “m” in the relevant peptide sequence. Fixed modifications can be represented as a separate amino acid or by altering the primary amino acid. So “C” can be altered to always have Iodoacetamide as part of its chemical formula.

If you add a new amino acid modification, remember to add the deuterium amount of any new amino acids you add as described in “Adding new Labeling Schemes”

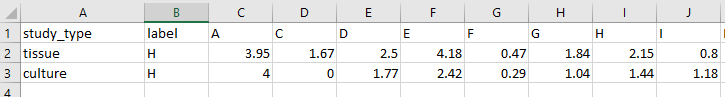
The second file having to do with post-translational modifications is “ptms.json”. This is only relevant for creating a Guide file for peaks. This file can be opened in notepad or any coding environment. It looks like this:



The text is the name of the modification, which will be added to the appropriate column in your guide file. Inside the brackets, the first element is the one letter code for the modification. The second element is the name for the modification produced by peaks. This allows DeuteRater v6 to replace the modification name from Peaks with the one letter code DeuteRater v6 recognizes. Make sure if you add to this, you also add to the “aa\_elem\_compositions.tsv” file as described above.

## Adding new Labeling Schemes

Every amino acid has different amounts of deuterium that can be stably incorporated based on biosynthetic pathways. These should be stable covalent bonds, not ions, or hydrogen bonds capable of hydrogen exchange. The file that governs this is called “aa\_labeling\_sites.tsv”. It looks like this:



Study\_type is the value that will go into the “Label Key” option in the settings menu or “label\_key” in settings.yaml. the “label” column is the element being used for heavy labeling. Currently only “H” is supported. All other columns are amino acids or post-translationally modified amino acid from the file “aa\_elem\_compositions.tsv”. The value under an amino acid indicates the number of deuteriums that can be added. Since this is a population average, decimal values are allowed.

The two default values are “tissue” which is values from mammalian tissue and “culture” which is values from mammalian cell culture3. If new amino acids or modifications are added, be sure to add new columns to this file.

# Citations

1. Kessner, D.; Chambers, M.; Burke, R.; Agus, D.; Mallick, P., ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics* **2008,** *24* (21), 2534-6.

2. Rockwood, A. L.; Haimi, P., Efficient calculation of accurate masses of isotopic peaks. *J Am Soc Mass Spectrom* **2006,** *17* (3), 415-9.

3. Naylor, B. C.; Porter, M. T.; Wilson, E.; Herring, A.; Lofthouse, S.; Hannemann, A.; Piccolo, S. R.; Rockwood, A. L.; Price, J. C., DeuteRater: a tool for quantifying peptide isotope precision and kinetic proteomics. *Bioinformatics* **2017,** *33* (10), 1514-1520.