**DeuteRater Instruction Manual and Tutorial**

**Table of Contents**

Introduction pg 2

Pre-Requisites pg 2

Limitations and Assumptions of DeuteRater pg 3

Installing and Opening DeuteRater pg 4

Extracting Data pg 6

Converting to mzML pg 6

Creation of Identification File pg 6

Calculating Labeling Shifts and Rates pg 12

Adjusting Folder Locations and Program Settings pg 15

Settings pg 17

Filters pg 20

Advanced Options pg 23

Altering amino acids or adding modifications pg 24

Adding or changing Labeling Site Patterns pg 25

Changing modifications pg 26

Output Files pg 27

Altering the Code pg 37

Adjusting the Defaults pg 37

Adding a New Element pg 37

Using DeuteRater for Other uses pg 43

Common Problems pg 44

Updates pg 46

Citations pg 47

**Introduction**

Welcome to DeuteRater, a program designed to use isotopic data to determine turnover rates of proteins. We have designed this tutorial to lead you step-by-step to beginning your analyses with DeuteRater.

**Pre-Requisites**:

DeuteRater\_py2: Python 2 (Python 2.7 or higher).

DeuteRater\_py3 or higher, Python 3 (Python 3.5 or higher)

py2 and py3 are functionally identical, py2 is for Python 2 and py3 is for Python 3.

Required Python Modules:

pyteomics

pandas

lxml

PyQt4

numpy

scipy

matplotlib

Required Python Modules that should be included in your default python installation:

multiprocessing

copy

collections

os

sys

csv

**Limitations and Assumptions of DeuteRater:**

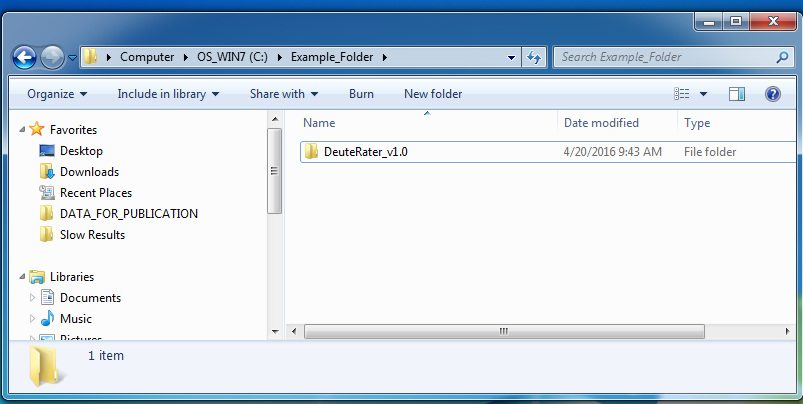
DeuteRater has been coded with a set of Assumptions and Limitations on its use. Before the tutorial the limitations will be laid out so you can be sure DeuteRater is correct for your application:

1. DeuteRater was designed first and foremost as a program for determining kinetic turnover rates of proteins. It will give outputs that can be used for analyzing the isotopic error of your mass spectrometer or other purposes, but those will require the use of intermediate files and are not directly covered in this tutorial.
2. It is assumed that whatever heavy isotopic label was introduced into your subjects was incorporated immediately (relative to the length of the experiment) and stayed constant the entire time. This is easily achieved with cell culture or injections in test animals. This is much more difficult to achieve with labeled diets or human test subjects.
3. It is assumed that your heavy isotopic label is ONE and ONLY ONE of the following 13C, 15N, or 2H.
4. DeuteRater must observe M0, the neutromer with no heavy isotopes both experimentally and theoretically. The theoretical calculation can still “see” a neutromer if its intensity is 1E-6 % of the total neutromer pattern but it does have limits. Because of this a traditional SILAC or SILAM approach of labeling all of one or two amino acids will probably fail. A mixture of heavy and light amino acids in cell media or an injection could probably be made to work, depending on the amount of labels in the heavy amino acid and the ratio of heavy to light.
5. DeuteRater is built for proteins. While the calculations are similar for other bio-molecules, DeuteRater is not set up to extract or analyze them by default.
6. DeuteRater was created by a lab which uses D2O as the labeling of 4-6% in mammals and mammalian cell culture. Filters and settings default to values appropriate to those types of experiments. If using a different label you must adjust settings and filters accordingly.

The code of DeuteRater is under a General Product License, so you are welcome to modify DeuteRater to undo these limitations. The code to do that is well beyond the scope of this tutorial.

**Installing and Opening DeuteRater**

First, download the folder from the internet and put it somewhere your computer. For this example I have placed it in the C drive, in a folder called Example\_Folder:



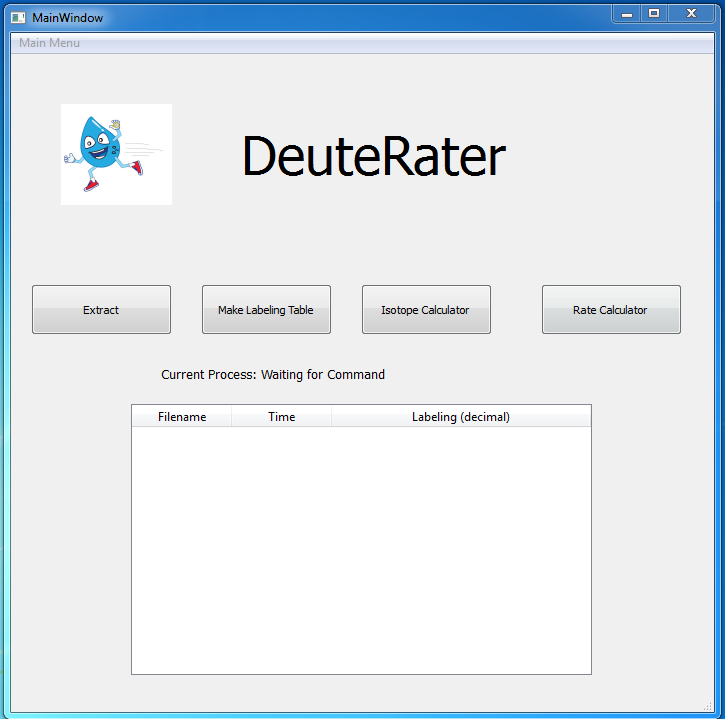
Next, you must open DeuteRater. Open the command line and type “python [codepath]” where [codepath] is the full file path for the code file. For our example, it looks like this:



Two Notes here:

1. The names of the folders cannot have certain characters in them or they will confuse the command line. White space (spaces, tabs, returns) are particularly problematic, so use an underscore instead of a space if such is necessary.
2. If you need to use spaces or don’t wish to type the full path in every time, you can change the working directory to the DeuteRater code folder and call the \_\_main\_\_.py file. (this is more advanced use of the command line so instructions are not included in this document.)

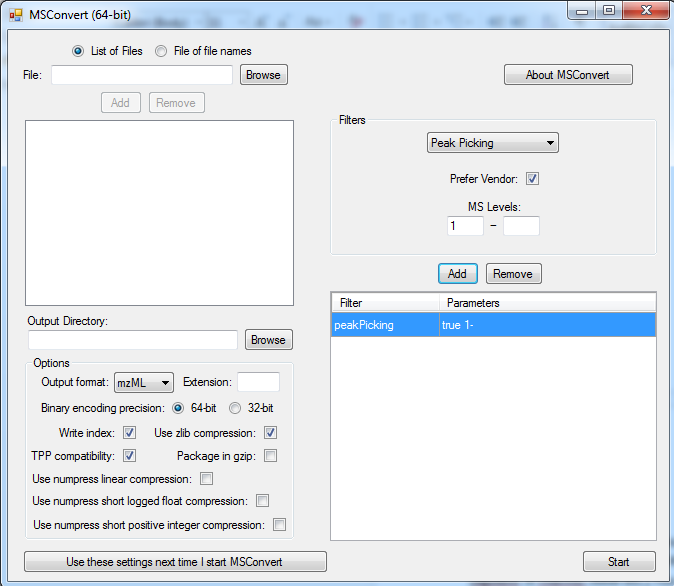
When you press enter, it should call up the DeuteRater interface, which should look like this:



**Extracting Data**

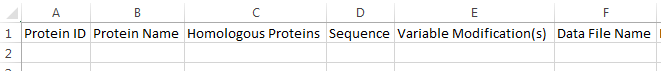
**Converting Data to mzML**

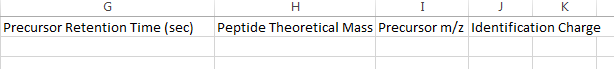
Now you should test the code with some data. You will need mzML files to search (the program is set to do this for MS-only runs). Note that if you have vendor specific output files (Agilient .d, Thermo .RAW etc.) you can change them to mzml using proteowizard1 (which can be downloaded here: http://proteowizard.sourceforge.net/downloads.shtml) and the following settings:



**Creation of Identification File**

DeuteRater has no protein identification software, so identify the peptides using whatever program you desire, such as Mascot, Protein Prospector, SearchGUI ,or any vendor specific programs that came with your mass spectrometer. This program should generate output with the protein IDs as well as other data. In the DeuteRater code folder is a csv file called peptide\_id\_template.csv. Open this file; it should look like this:





This is the template that you will need to follow; your identification file must have exactly these headers. Order doesn’t matter nor does the presence of extra columns, so feel free to just copy the headers and paste them onto the appropriate columns in your output document.

The columns are:

**Protein ID**: is the protein identification given by the id software. This is usually an accession code of some sort.

**Protein Name**: is the common name of the protein.

**Homologous Proteins**: gives some indication of homology for this peptide. This is just reported back by the program, and not used for anything, so “True”, “Yes”, a list of accession codes, or the number of homologous proteins, are all good indicators for this column.

**Sequence**: lists the peptide sequence using one letter codes for the amino acids.

**Variable modifications**: The format required is “modification name space (position)”, with parentheses required around position. For example, if a methionine was oxidized, and this methionine was the 3rd amino acid in the sequence, the code would be Oxidation (3). Note that any words between the first word and the parenthesis are ignored, so Oxidation of Methionine (3) will work as well.

WARNING: DeuteRater will not check that the replacement is reasonable. You can replace a lysine as easily as a methionine in the above example. Be cautious when putting together this column. Or avoid it as described in “Common Problems” on pg 43.

**Data File Name**: is the name of the MSMS file the peptide was identified in.

**Precursor Retention Time (sec)**: gives the time (in seconds) that the peptide was observed during the run on the mass spectrometer.

**Peptide Theoretical Mass**: is the neutral theoretical mass of the identified peptide.

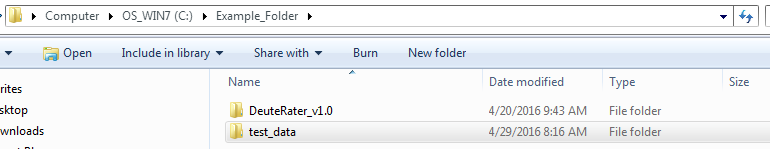
**Precursor m/z**: lists the observed m/z value of the peptide.

**Identification Charge**: is the charge with which the peptide was observed. DO NOT include a + sign for positive charges.

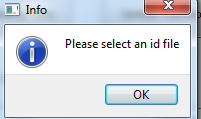
When you have finished creating this file, save it as .csv or .tsv. (Note that the file code **must be** .tsv or .csv. If the file is properly formatted but has a different ending, it will be rejected by DeuteRater.) Most spreadsheet editors have this option. For example, when saving in Microsoft Excel there is a pulldown menu under the save filename:



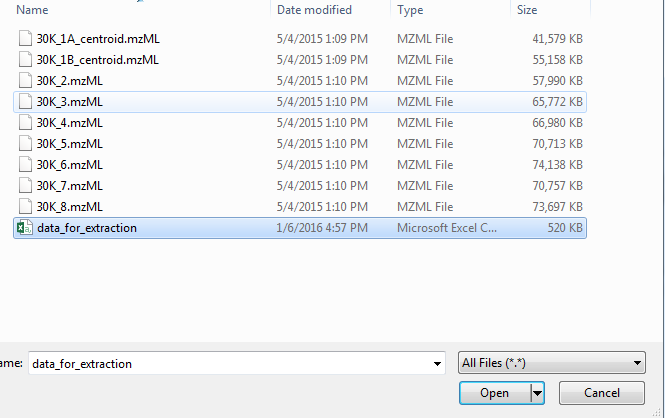
For our tutorial example, I have created a test data folder next to the code folder:



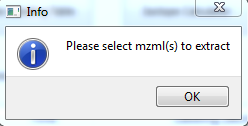
Now, go back to the DeuteRater window and hit the “Extract” button. The following window will pop up:



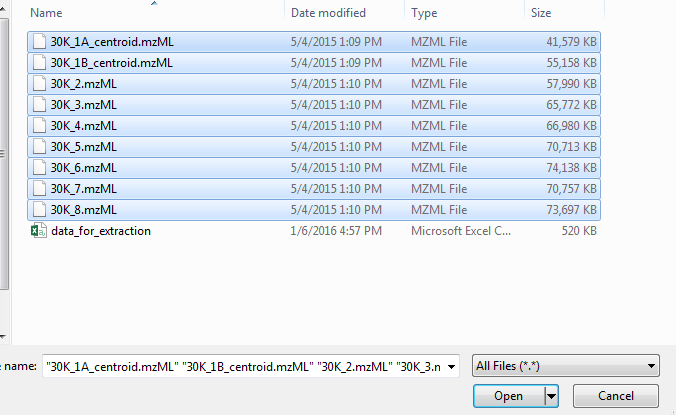
When you hit “OK” you will be in a new Extracted\_Files folder; navigate to your .csv id file, highlight it, and then press the “Open” button (outlined in blue below).



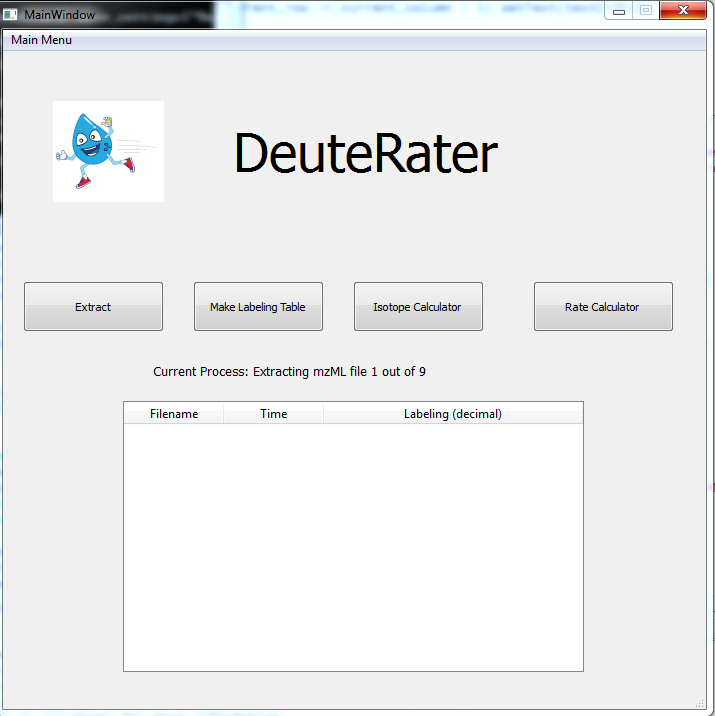
Pressing open will cause this window to close; the program will next prompt you for mzml files:



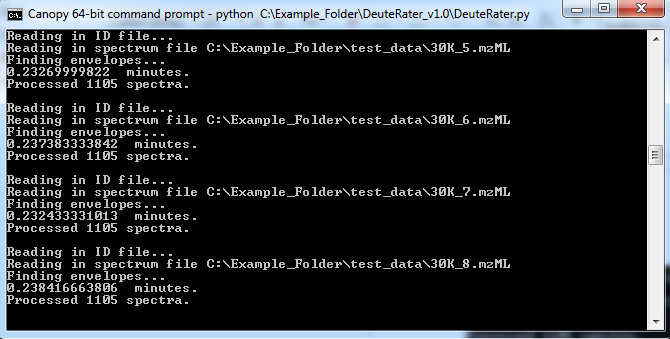
Hit “OK” and then select your mzmls as shown below. You may select all of the mzml files if you desire:



After clicking “Open” the program will be unresponsive, but will give you periodic updates:



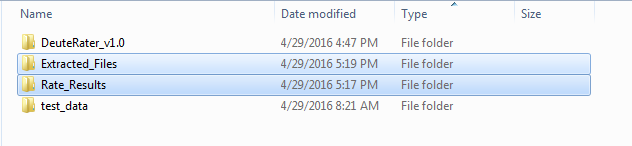
The command line gives more detailed output if you desire more information:



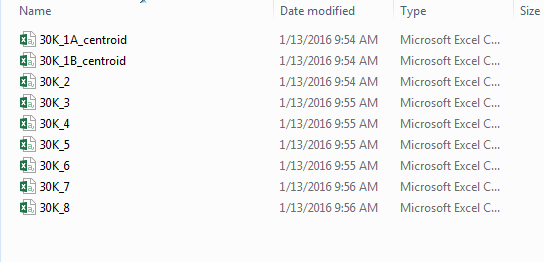
When the extraction is finished, the Current Process on DeuteRater will return to “Waiting for Command”. This can take quite some time depending on the size of the input files. If there is a serious error it will be reported in the command line.

If you cannot or do not wish to select all of the mzml files at once, or you need to use multiple id files, you can hit the extract button multiple times. As long as none of the mzml files have the same name it will put all of the extracted files in the same place.

Note that two new folders were created when the extract button was pushed:

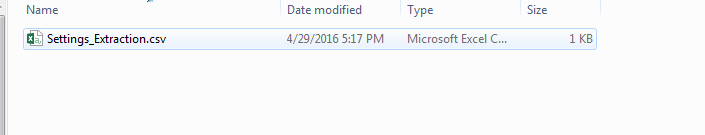


This only occurred because these folders did not already exist. The folder Extracted\_Files holds the extracted mzmls in the form of csvs:



This folder can hold many .csvs, and will be added to every time you extract new csvs without changing the folder. DO NOT put any .csvs in here not made by the extractor! Doing so will disrupt later parts of the analysis (you may put folders or files that are not .csvs in this folder without disrupting anything).

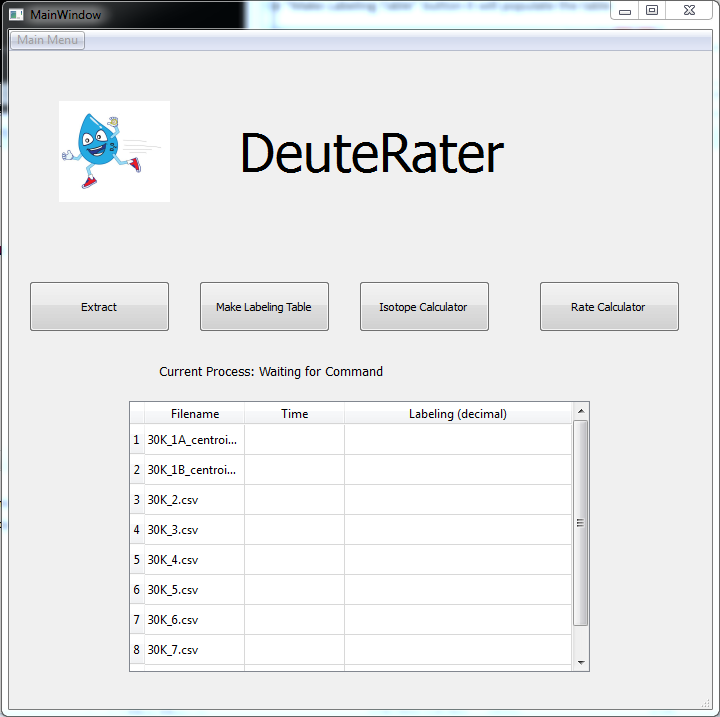
The other folder is called Rate\_Results and contains just one file at this point in the tutorial:



This saves the settings used in the extractor. Since the file contents are identical to other settings files produced later, we will discuss them on pg. 29. This file is overwritten each time you run the extractor. When running the extractor many times, make sure to use the same settings for each, or rename the newly made Settings\_Extraction file after each time the Extractor finishes so you have a record of your settings**.**

**Calculating Labeling Shifts and Rates**

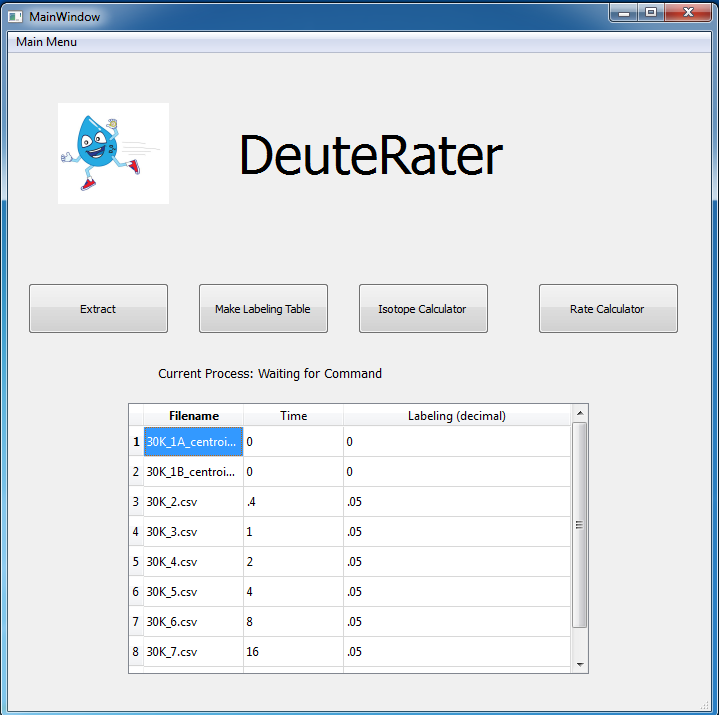
If you now push the “Make Labeling Table” button in the main DeuteRater window, it will populate the table as shown below:



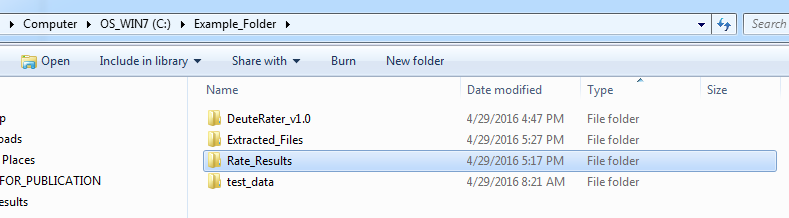
This table lists all csv files in the Extracted\_Files folder. If you do not wish to use one or more of the files in the list, go into the Extracted \_Files folder, find the offending file(s), and either change the file extension(s) (the three letters after the .) or move the file(s) to another folder. Then push the “Make Labeling Table” button again.

Fill in the columns for “Time” and “Labeling (decimal)”. The “Time” must be the duration of labeling. You do not enter units, but you should keep track of them and keep them consistent. For example, if you have a labeling length of 6 hrs and 1 day you should enter .25 and 1 or 6 and 24. If you use 6 and 1 the computer does not understand units and so will treat the 6 hr time as 6 times later than the 1 day time. Labeling (decimal)” must be a decimal value between 0 and 1, inclusive. This number represents the amount of labeling that the organism has received to make labeled peptides (the tutorial example uses deuterium at 5% (0.05); you can use much higher amounts if needed).

After the table is filled out, press the “Isotope Calculator” button:

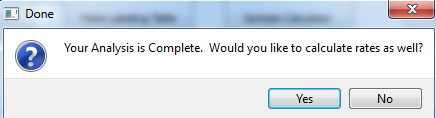


While running, updates will be listed fairly often on the Current Process option and the command line. DeuteRater will begin pulling in and filtering data, as well as generating theoretical curves for how the various isotopic peaks will respond as labeling increases. It will then calculate percent changes based on the labeling values you previously entered in the table. These will be written to various .csv documents in the Rate\_Results folder created earlier:



NOTE: if you deleted the Rate\_Results folder, or it does not exist for some other reason, it will be created here, next to the extractor and code folders.

Once the results are complete, DeuteRater will ask a question:



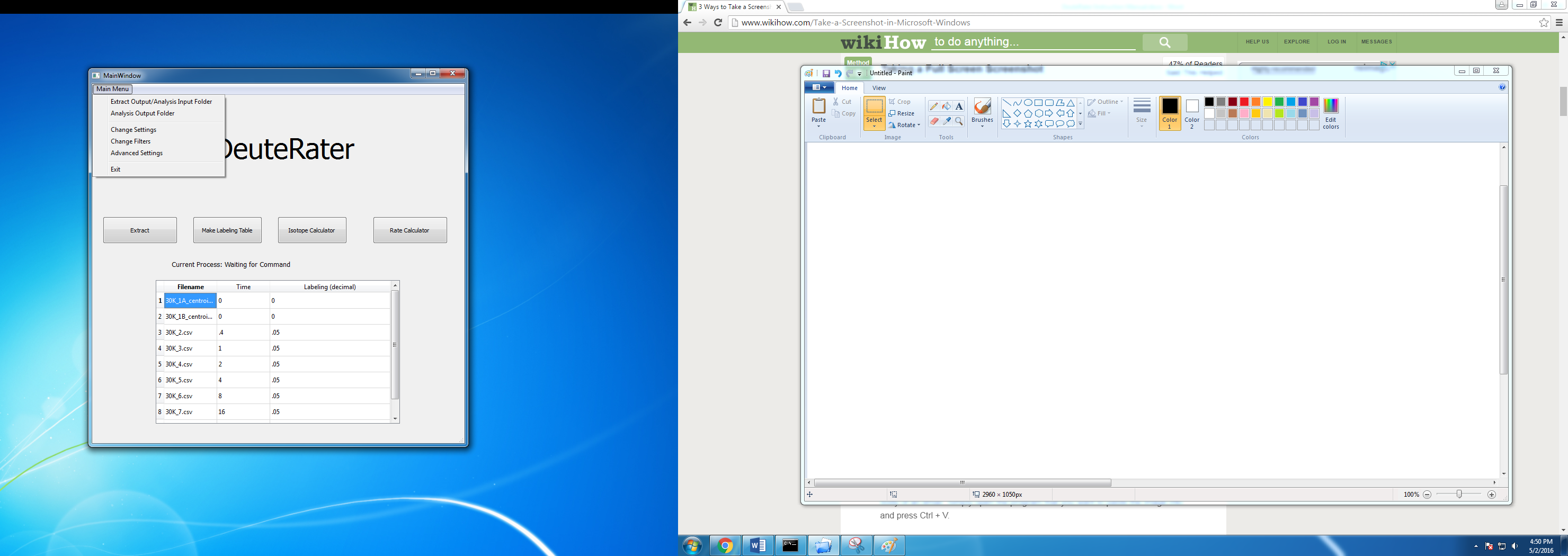
This is asking if you wish to proceed to the last button, “Rate Calculator”, which will plot rate equations and then create graphs of them. If you select “Yes”, the program will inform you when your analysis is done.

A final note about the buttons before moving on: There is nothing stopping you from opening up DeuteRater and starting with Rate Calculator, as long as the other buttons have been used previously and the folder it needs (Rate\_Results in our example) exists. This is also true with any of the other buttons: as long as their pre-requisite files are filled and the program knows where the data is, the program will work.

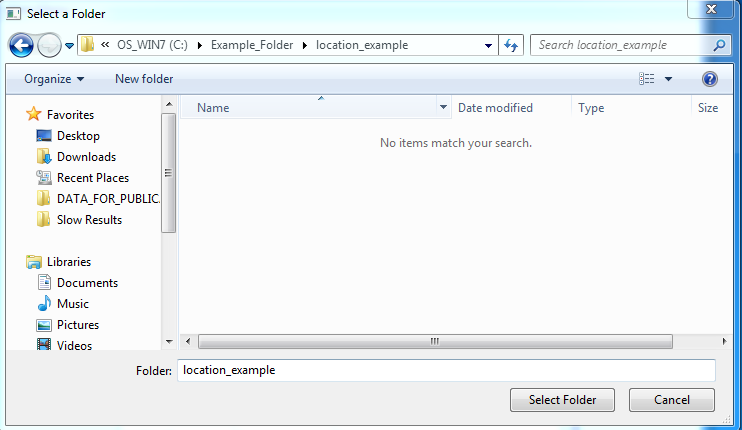
The Isotope Calculator button always requires a filled Labeling Table, but Ctrl+v works as the paste command for the table; just open the Time\_and\_Labeling.csv file, and copy and paste the values into place if doing it again. Just make sure to close the file when you are done.

**Adjusting Folder Locations and Program Settings**

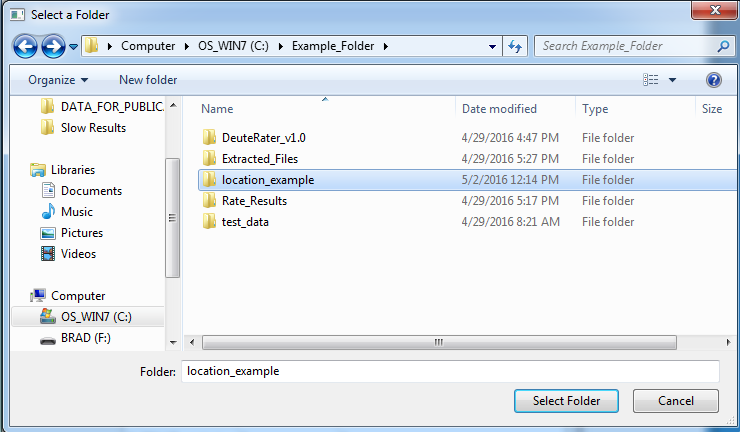
At this time, it is worthwhile to take a look at the Main Menu options in the top left hand corner.



The first option reads “Extract Output/Analysis Input Folder”. This sets the folder that the “Extract” button will put the extracted files into, and “Make Labeling Table” and “Isotope Calculator” obtain data from. When you select this option, you will be taken inside the current folder (by default the “Extracted\_Files” folder next to the code folder). To change, navigate to the folder of your choice, then hit “Select Folder” when you’ve found the folder you want. Note: you must either be inside the folder you wish to select when you hit the button…



Or have it selected:

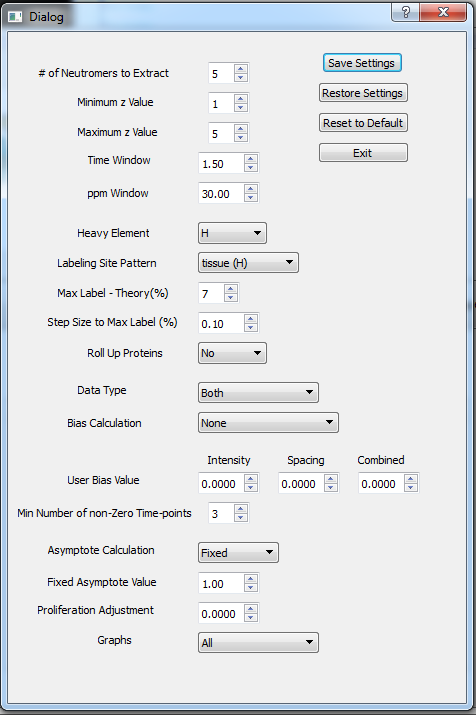


Make sure to hit “select folder.” Both of the above examples set the new folder as C:\Example\_Folder\location\_example.

The next item on the pull down menu is “Analysis Output Folder.” This sets the output location for the Isotope Calculator and Rate Calculator, as well as the Input Folder for Rate Calculation (like the Rate\_Results folder in our example). It works just like the “Extract Output/Analysis Input Folder” option.

Note: The program will not check the contents of these folders so it may overwrite data if you are not careful.

The next option is “Change Settings,” which opens the following settings menu:



Starting with the buttons in the top right:

**Save Settings** saves the current settings. **Restore Settings** makes the settings whatever they were when you last saved settings. **Restore Defaults** sets the settings to whatever they were when the program started. **Exit** leaves the settings menu.

Whenever you open the settings menu it will open whatever settings are currently in use by the program.

The first five settings are for the extractor:

**# of neutromers to extract** - how many isotope peaks to look for (remember that no heavy isotopes is also a neutromer).

**Minimum z value** and **Maximum z value** - what charges to look for; the program will look for every charge state between the minimum and maximum, inclusive, for every unique sequence.

**Time Window** and **ppm Window** - how much variance is acceptable in an id. When making an identification, the retention time and ppm can be off by whatever the settings are, and it will still be accepted. Time is in minutes.

The next four settings (Heavy Element to Step Size) are for the theoretical calculation.

**Heavy Element** - which element bears the heavy label: C,H or N.

**Labeling Site Pattern** – This is a bit more complex. It refers to a key for the number of possible labeling sites in each amino acid. These can be added to the program as needed (instructions detailed later). The program comes with patterns for mammalian tissue and mammalian cell culture with deuterium. Additionally, there is the option “global”. Global indicates that every atom of the heavy element can be heavy.

The following two settings are only relevant if **Slow DeuteRater** is selected from the Advanced Options (described later in this section):

**Max label** – If the Slow DeuteRater option is selected, this is the maximum percent labeling the theoretical curve for each peptide will go out to. The further you are from your actual values, the worse the fit, but if your actual values are greater than the maximum label, your calculated values are highly likely to be wrong. Setting a percentage point or two above your highest value (6% or 7% if your highest labeling value is 5.3% for example) is recommended.

**Step size to max label** – in Slow DeuteRater the computer will calculate steps from 0% to max label, and then fit theoretical curves to those values. This sets the size of the steps - the smaller the steps, the greater the calculation time and fit accuracy.

The final settings have to do with rate calculations:

**Roll-Up** – If this is set to “Yes,” then all peptide turnover percentages from one protein for one time will be combined into one number, after checking for and removing outliers using a Median Absolute Deviation test. That number and its error will be used for the rate fitting graph. If this is set to “No,” then all points will be used without error to fit the graph.

**Data Type** – Do you wish to use change in abundance or change in neutromer spacing for the rate calculations? If “both” is selected, another calculation called “Combined” is performed using the median of the points with the best agreement from both of the other calculation types.

**Bias Calculation –** Do you want to assume a global bias from your instrument? “None”, the default, assumes that the instrument has no bias (or it varies and so a global correction is not appropriate). “User Defined” uses the values set in the User Bias Value below this option. “DeuteRater Calculated” has DeuteRater calculate the median turnover value for each calculation type used where the labeling is 0. This is then assumed to be the instrument bias and subtracted from all measurements. Note that DeuteRater Calculated requires a sample with no labeling to function.

**User Bias Value** – the values to assume for instrument bias for each measurement type. This is the value of the Bias NOT the correction value. If “User Defined” is selected for Bias Calculation, these numbers are subtracted from the turnover values of the appropriate calculation type.

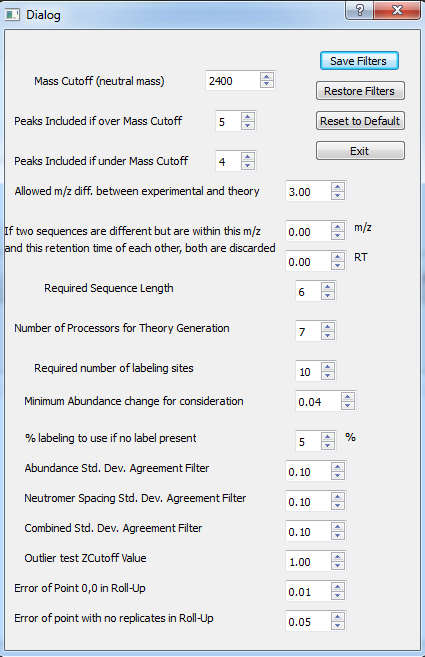
**Min number of non-zero time-points** – the minimum number of time-points a protein must be seen at in order to have a rate fit. Note that zero time or zero labeling points do not count (zero labeling gives non-real answers and the line is forced through zero by the nature of the equation used to fit it). Keep in mind that to the computer .1 and .1000001 are different times.

**Asymptote Calculation** – The equation used to fix the line is a – a\*e^(-kt). If this value is “Fixed,” then the computer will set “a” in the equation to the value in Fixed Asymptote Value. If set to “Variable,” the computer will attempt to calculate “a” and “k” at the same time for each protein.

**Fixed Asymptote Value** – If Asymptote Calculation is set to Fixed then this determines what the asymptote for the rate equation is. If Asymptote Calculation is Variable this number is not used.

**Proliferation Adjustment –** This filter changes the fit equation to a – a\*e^(-(k-proliferation adjustment) \* t). The goal is to remove proliferation as a factor in k. It can also be used to manually adjust k globally if such is needed. Based on word done by others2.

**Graphs** – Which method of calculating rates do you wish to graph? The program will not produce graphs of calculation methods that have not been used. For example, if you change Data Type to “Abundance” and leave Graphs at “All” the program will only produce Abundance graphs.

The next option in the pull-down menu is “Change Filters,” which looks like this:

Starting with the buttons in the top right:

**Save Filters** – saves the current values.

**Restore Filters**- restores the values to what they were when you last hit Save Filters.

**Reset to Default**- resets the filters to whatever they were when the program started.

**Exit** – exits this menu.

The first three menu values are connected:

**Mass Cutoff**- the neutral mass that governs the switch between the next two values.

**Peaks Included if over mass cutoff** – the number of neutromers to include in calculations if the mass is over the Mass Cutoff value.

**Peaks Included if under mass cutoff** – the number of neutromers to include in calculations if the mass is under the Mass Cutoff value.

The above filters are designed to catch all relevant neutromers, and heavy sequences have more relevant neutromers. You may set the under and over values to the same number, or you may make them very different. Keep in mind that if the program cannot find sufficient neutromers for a peptide (due to the extractor not looking for or not finding enough), that peptide will be discarded.

**Allowed m/z difference between experimental and theory** - If the experimental identification is more than this many m/z units away from the theoretical m/z for the peptide, it is discarded. This is mainly for removing unidentified modifications, so it can be set at 3 or 4 and still function perfectly.

**If two sequences are different but are within this m/z and this retention time of each other both are discarded** – This filter is primarily for addressing trust issues. If sequences are too close together in retention time and m/z, it could result in a mistaken id. Note that the sequences must be different and within both filters to be discarded. This is 0 for both by default (and thus essentially off).

**Required Sequence Length** – A peptide must be at least this long or it will be discarded. Again, this addresses trust in the id, as well as provides a preliminary filter for Required Number of Labeling Sites.

**Number of Processors for Theory Generation** - The theoretical calculations can be time consuming, especially if Slow DeuteRater is selected from the Advanced Options menu. As a result, multiple processors should be used whenever possible. The program will not let you set this to more processors than you have. By default you will use all but 1 (if you have more than 1). Note that the more processors you use the faster the process will go, but the less your computer will be able to do while the analysis is running. Also, using all processors will disable the update process while the calculation is going.

**Required Number of Labeling Sites**: If a sequence does not have potential sites for this many labels it will be discarded. This is primarily a noise filter. If there are not enough potential labels, then the maximum possible change in the neutromer pattern of that sequence is small. This allows noise to dominate how much the peptide changes at a time.

**Minimum Abundance Change for Consideration** – If a predicted change is too low, noise might be influencing it, so it is discarded. The abundance is a proportion of the total abundance of all peaks being considered, so the .04 default translates essentially as “if the maximum possible abundance changes by less than 4% of the total abundance for one of the neutromers, don’t consider it”. Neutromer Spacing does not have such a filter because it uses absolute spacing distance. This biases a potential filter against M1-M0 spacing and small peptides which can both be useful. Therefore no such filter exists.

**% labeling to use if no label present** – This filter is purely for the Calculation\_of\_Fraction\_New\_Protein.csv. When calculating the amount of turnover, we can’t have a zero due to a divide by zero error. Instead of deleting these values, we assign them an arbitrary maximum amount of label for troubleshooting and analysis purposes. This sets that maximum.

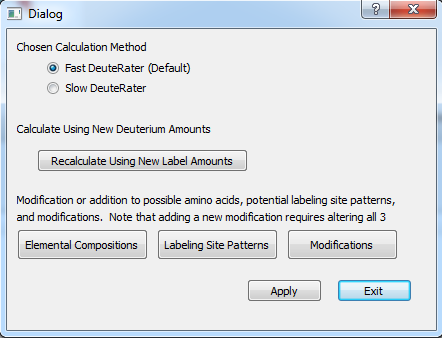
**Std. Dev. Agreement Filters** – All of the neutromer peaks for a peptide identification are averaged together. Even though the average is not used for the rate calculations, the standard deviation still tells how well the peaks agree. If the standard deviation is above this set value, there is too much noise to trust the measurement, so that particular peptide in that charge state in that file is discarded. The lower this value is, the better the data, but there will be less of it. Note that all calculation types can have separate filter values.

**Outlier test ZCutoff Value** – A Median Absolute Deviation test is used to filter points for Neutromer Spacing and Combined peptide measurements, as well as the combinations used in the Roll-Up function. This option sets the Z value used as a cutoff for this test.

**Error of Point 0,0 in Roll-Up** - When fitting rates, a zero value is provided. In Roll-Up everything needs an error value; this provides the error value for the zero point. The effect of this on aspects other than error will be minimal due to the equation forcing the line through 0. (The line being forced through zero by theory and the equation are also why we provide a zero instead of using zero from the provided mass spectrometry data).

**Error of point with no replicates in Roll-Up** - If you roll-up proteins, every point needs an error value; standard deviation is used. If there is a protein for which a time has only one peptide, the standard deviation is 0, which will skew the graph in favor of this untrustworthy point. This option sets the error value of those points to correct this issue.

The final option in the pull-down menu, aside from “Exit,” is “Advanced Settings,” which looks like this:

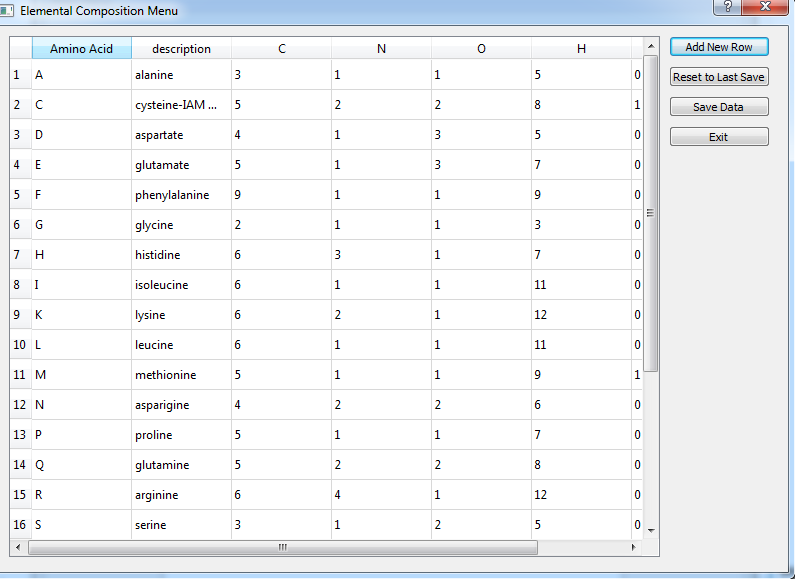


**Fast DeuteRater**, the default, is faster. When it calculates the theoretical maximum change in the neutromer abundances and spacing, it assumes that the labeling amounts entered in the labeling table are entirely accurate. So the program only bothers to calculate the baseline at no labeling, and then the change from that baseline at each labeling amount you entered.

**Slow DeuteRater** is for those who are less sure of this. The program will calculate a theoretical curve for each peptide from no labeling up to the **Max label** value in the settings menu. It will calculate the points for this curve in increments of **Step size to max label**, also in the settings menu. This allows you to have these curves if you desire, and also allows the recalculation of turnover with new amounts of label without needing to run through the Isotope Analysis button again.

**Recalculate using New Label Amounts** – This button allows the recalculation of turnover without needing to fill out the Labeling Table and press the Isotope Analysis button again. The main restriction with this button is that the Isotope Analysis button must have previously been used with **Slow DeuteRater** selected. Also, the new labeling must be added in the Time\_and\_Labeling\_Data.csv file, not the labeling table. After this button is pressed, the program will go into the Analysis Output folder and use the Time\_and\_Labeling\_Data.csv and the Combination\_of\_Experimental\_Data\_and\_Theoretical\_Curves.csv to create the Calculation\_of\_Fraction\_New\_Protein.csv file again. Of course, if you want to save the original analysis, you may simply copy these .csv files to a new folder and then set that to the Analysis Output Folder.

**Elemental Compositions:** This will open the following window:

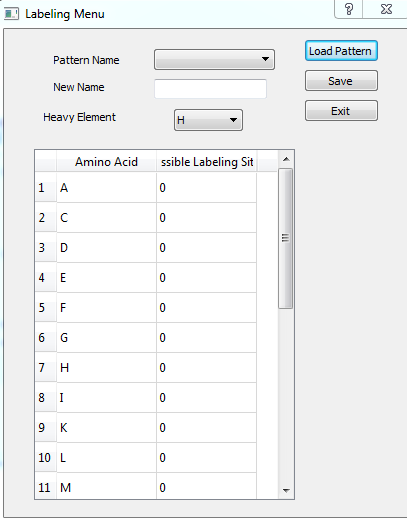


This window shows all amino acids that the program recognizes. This includes standard amino acids, Fixed modifications (like row 2 showing IAM modified cysteine), and variable modifications (m for oxidized methionine on line 21, not shown in the above picture). These should be elemental compositions at pH 4 INSIDE a peptide (2Hs and an O are automatically added to account for this in DeuteRater). If your peptides need different amino acids or new modifications you can hit Add New Row to make a new blank row and then add any one letter code you wish. You can also alter current amino acids to represent fixed modifications or different conditions.

Be sure to Save Data before leaving if you have made changes you wish to keep. The program will stop problematic entries (blank spaces, Amino Acid codes longer than one character, duplicate amino acid codes, etc.). There is no penalty for having unused Amino Acids, but if you wish to delete a row for any reason, delete all entries in that row and then Save Data. The row will be gone.

WARNING: If you add an amino acid it will not automatically be added to the Labeling Site Patterns, and Isotope Calculator will not run. This is to ensure that you check the amount of potential labeling sites in your new amino acid or modification (especially important in metabolic labeling). If you add a new amino acid, use the Labeling Site Patterns button. That window will update itself with the new amino acid and update the program as soon as you save a Labeling Site Pattern. If you want the new amino acid to have 0 potential labeling sites in all conditions, load a pattern and immediately save it.

**Labeling Site Patterns:** this is for examining the Labeling Site Pattern options that can be selected in the settings menu. The window looks like this:



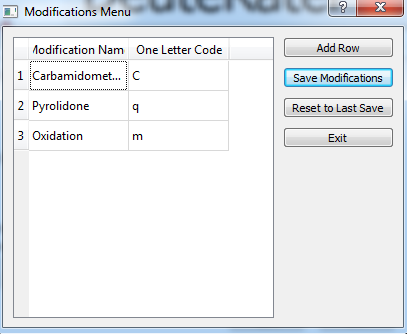
The window opens with a blank pattern and 0s for every amino acid and modification. To check a pre-existing pattern use the Pattern Name pull down menu. Select the appropriate name and press Load Pattern. This will populate New Name, Heavy Element, and the table with the values for that name.

To save modifications to a current pattern: load the pattern, make whatever changes you require and then press save.

To create a new pattern: start with the blank pattern or a pre-existing pattern, give the pattern a unique name, change the heavy element to whatever you wish, update the table appropriately, and then press Save.

Note: unlike the other menus, you cannot add rows to the table or delete them. The table is in some ways an extension of the Elemental Composition table. DeuteRater will check to make sure that you do not have more Potential Labeling Sites than atoms of that type in the elemental composition (e.g. Alanine has 5 hydrogens. It cannot have 10 Potential Labeling Sites for Deuterium. There is no space for them). If you are changing Elemental Compositions to add a new modification or alter an existing amino acid, do that first. Then change the Labeling Site Patterns.

**Modifications:** This page is for the modifications that DeuteRater will recognize, specifically for use in the Modifications column of peptide\_id\_template.csv (this is used to tell the Extract button what to find in the mzML files). The window looks like this:



Remember that the Modification format is “Name (location)” for the Modification column. “Modification Name” in this table is the Name and the “One Letter Code” will be the letter that the amino acid is replaced with. For example, if you had the sequence “LVMEGR” and the modification was “Oxidation (3)” DeuteRater would make the new sequence “LVmEGR”.

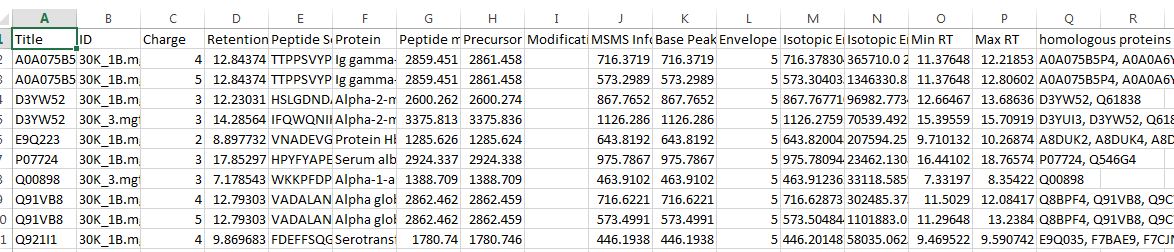
Keep in mind that this will only replace the letter in the Amino Acid sequence and keep track of allowed modifications. It will NOT make an elemental composition suddenly appear. If DeuteRater sees an amino acid with only a letter it will discard the peptide.

Note: The computer will not check what it is discarding so be careful of location. Also, Name is case sensitive.

**Apply –** saves changes. This only affects the Fast and Slow DeuteRater option currently.

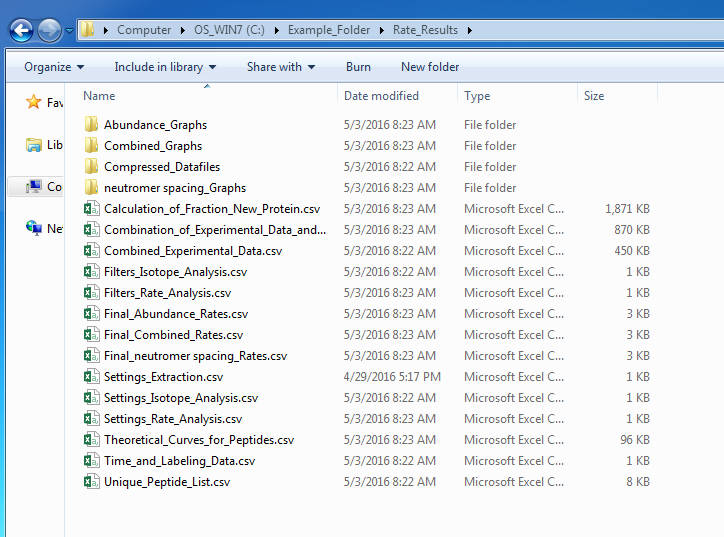
**Output Files:**

In the Extracted\_Files folder, the files look like this:

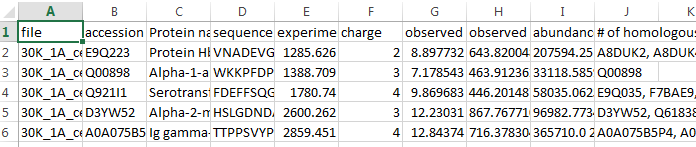


These list the data from your mzmls in a different format. The Isotopic Envelope m/z values and Isotopic Envelope abundance values contain all relevant values in a space-separated format.

The contents of the Rate\_Results Folder should look like this:



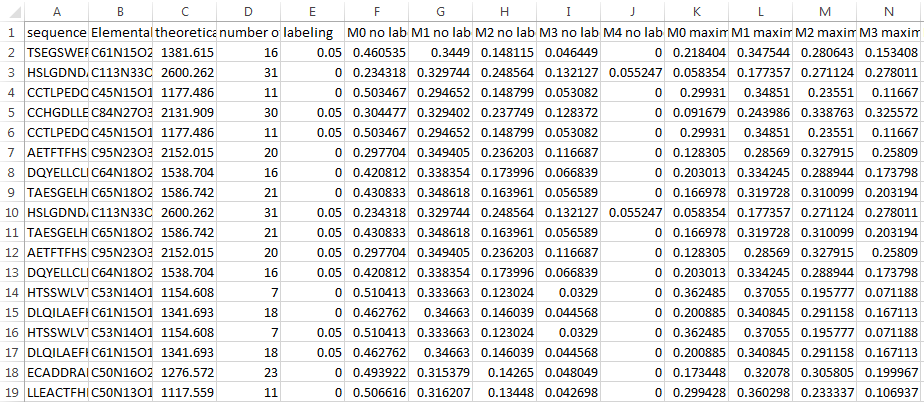
After writing the Extracted\_Files, DeuteRater will next write the Compressed\_Datafiles folder. This folder contains all the files from Extracted\_Files but with the columns re-ordered and some basic filtering applied (no missing peaks in the isotope pattern, sufficient sequence length, no unrecognized modifications, and so on). These are all combined into the Combined\_Experimental\_Data file, which looks like this:



All information is as before, plus filenames added for later use.

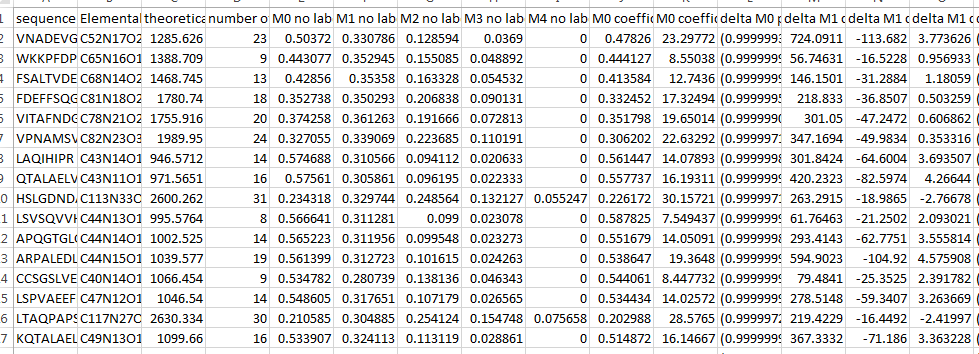
Next comes the Unique\_Peptide\_List, which contains a list of all unique peptide sequences. This differs slightly based on whether you have selected fast or slow in the advanced options menu. In the fast default there are sequences and labeling values; each pair is unique. If slow is selected only the sequences are in this file and they are all unique.

The sequences (and labeling in fast mode) are fed into the theory generator, which then produces the file Theoretical\_Curves\_for\_Peptides. This file differs depending on whether you are going fast or slow. The fast output looks like this:



There are more columns but this gives the general idea. The file contains some information about the peptide, sequence, elemental composition, mass and so on. It also includes the baseline for no labeling and the maximum values at the amount of labeling given. You may note that the no labeling and maximum labeling values are different even if the labeling amount is 0. This is because the 0’s are calculated with the **% labeling to use if no label present** setting in the filters menu.

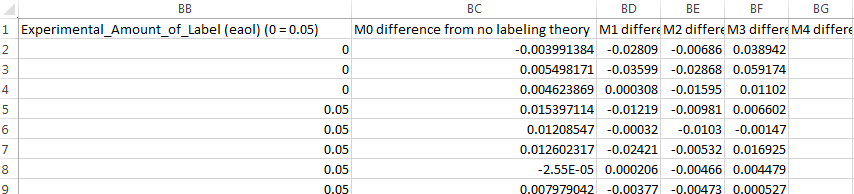
If you do the slow version you get a longer document, which looks like this:

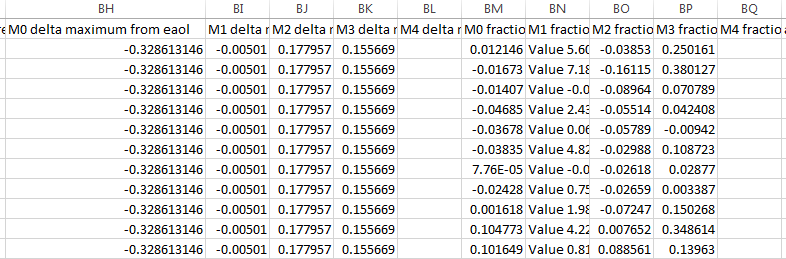


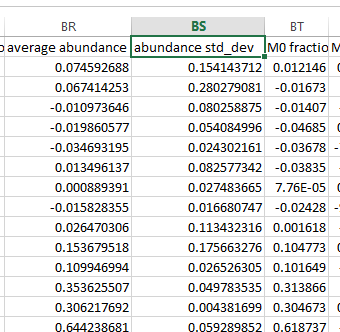
The file is quite a bit longer, but I have cut the rest to save space. This has each unique sequence with baseline values with no labeling. The next columns list a bunch of coefficients; the coefficients are for the change in each peak as labeling increases. Abundance M0 is delta = -(a-ae^(-kt)). The rest of the abundances are third order polynomials (neutromer spacing uses fourth order polynomials). None of the polynomials include a constant. They go in order from highest power of x to lowest so for the values in the box the equation is y = -724.0911 \* x ^3 + 113.682 \* x^2 -3.773626\*x for the M1 (one heavy isotopes peak), where y is change in signal and x is amount of label. The neutromer spacing values are listed after the abundance values.

The Theoretical curves are referenced to the experimental data by sequence and saved as either Combination\_of\_Experimental\_Data\_and\_Theory for Fast DeuteRater or Combination\_of\_Experimental\_Data\_and\_Theoretical\_Curves for slow DeuteRater. Note: if you look at the neutromer spacing values, they have changed from the theoretical curves. This is because the experimental data has the charge values, allowing correction of spacing equations by charge.

The final output produced by the Isotopomer Calculator is the Calculation\_of\_Fraction\_New\_Protein document. Most of the contents are the same as previous documents, so you can get all the same data without shifting through multiple documents. The differences are as follows (Note: this discussion is for the slow setting of DeuteRater. The only difference between slow and fast is that the fast document is smaller, and the maximum values are further to the left.):







These figures show the columns for abundance. The first column is the Experimental amount of label (eaol). You’ll notice that in the header there is a line that reads 0 = .05. This is because the calculations will error out if 0 is used, so it gives you a calculation using .05 for troubleshooting purposes. If you want a different number here, enter an alternate value in the filter menu with option **% labeling to use if no label present**. For example, if you desire to be close to zero, enter .01 or something similar. You may also give a number close to 0 in the Labeling (Decimal) spot in the Labeling table and the 0= value will never activate.

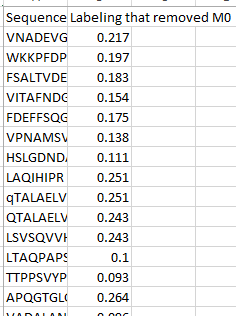
Columns “M0 difference from no labeling theory” to “M4 difference from no labeling theory” list the differences from no labeling theory. This is the experimental value subtracted from the baseline for each peak.

Columns “M0 delta maximum from eaol” to “M4 delta maximum from eaol” the maximum label from eaol. This is the result of putting the labeling value into the coefficients calculated earlier.

The Fraction new for each peak results from dividing the difference by the maximum, giving the amount of label incorporated. This is summarized in”M0 fraction new” to “M4 fraction new”. The average of these values, their standard deviation and a statistic used in further analysis are listed after. Averaging is too subject to noise; abundance uses M0 because it is consistent (always goes down often by a very noticeable amount), and the other metrics use medians to control for noise.

Of final note are some error metrics. In the lines above you will notice some fraction new values are words, not numbers. This is due to a filter cut off on the possible amount of change to prevent noise from interfering. The neutromer spacing section of the file has some columns called “Final” which are the same as their fraction new, but report if the point was an outlier based on a Median Absolute Deviation test. Finally, if using both abundance and neutromer spacing, the combined file will have a column containing all points it combined. It will contain the fraction new values in order from the first abundance peak to the last neutromer spacing. If a number is missing, it was not included again due to a Medain Absolute Deviation test for outliers.

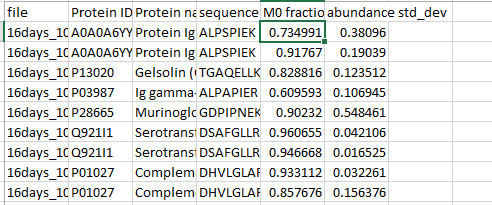
There is a potential for one more type of file. This file is named “Peptides Removed for too much Label.csv”, which contains columns that look like this:



This is for peptides whose amount of labeling grew to the point of removing the M0 peak. Since there are several assumptions that require M0, if the labeling is high enough to remove that peak, the peptide is discarded. Fast DeuteRater indicates which label you entered was too high, while slow DeuteRater indicates which step on its way to maximum labeling caused the problem.

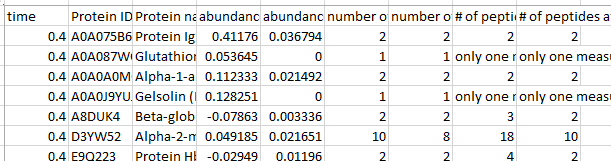
The output files generated by the Rate Analysis button are described below.

The first output is only present when a Bias Calculation other than “None” was selected in the Settings menu. The file name is [Analysis\_Type]\_ bias\_corrected.csv where [Analysis Type] is Abundance, neutromer spacing or Combined. The file will look like this:



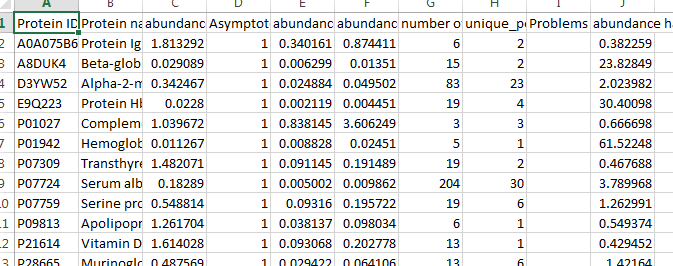
It contains filename, protein id and name, sequence, turnover number and the standard deviation. This is all the data taken on into the next step of the calculation.

The second output is conditional on doing the roll-up. If you selected this setting there will be files labeled [Analysis\_Type]\_Protein\_Roll\_Up, where [Analysis Type] is Abundance, neutromer spacing or Combined. When you open your file it will look similar to this:



Time is the time the protein came from, protein ID gives the accession number, protein name is the common name, fraction new is the median fraction new for that protein in that file, and std. dev. is the standard deviation for that protein in that file. Number of peptides is the total number of peptides after the outlier check (Median Absolute Deviation), while unique lists only those with unique sequences that remain after the outlier check. The far right columns show the number of sequences before and after the outlier check. Keep in mind that charge states have not been combined yet, so some of the measurements represent the same protein seen in different charge states. All analysis methods produce files of this form.

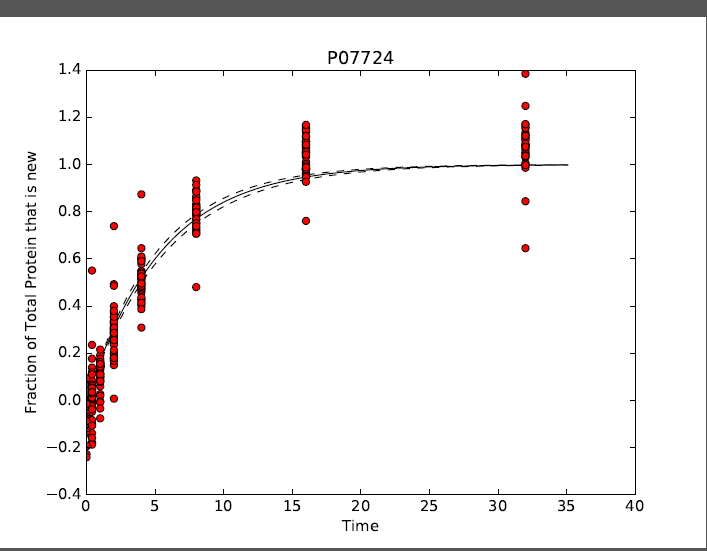
The final rate output file, called Final\_{}\_Rates, (or Final\_{}\_Rates\_Roll\_Up if roll up was performed) where {} is Abundance, neutromer spacing or Combined, looks similar to this:



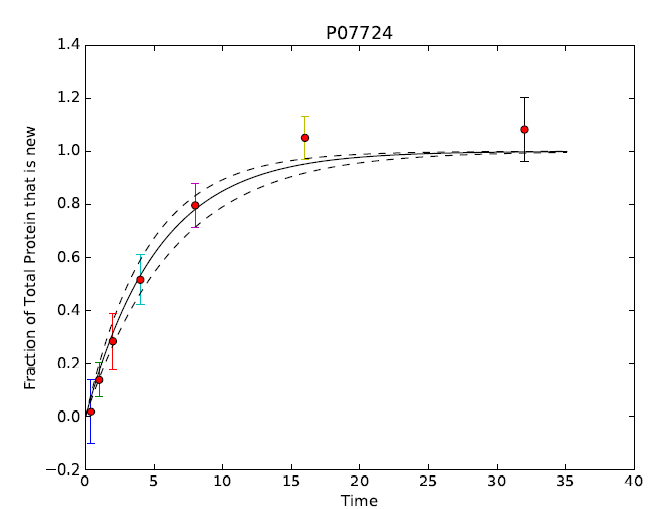
The protein ID is the accession number or other identifier provide for the protein, and Protein name is the common name. The rate lists the k values from the formula labeling = a- e^(-k\*time). Effectively it is change per unit of time. Asymptote is “a” from the rate equation. Std. dev and 95% Confidence list the standard deviations and 95% confidence intervals of the predictions of k. The number of measurements gives the total number of measurements for this protein through all files. Unique peptides shows how many unique sequences were used (this discounts things like different charge states and the same sequence in multiple files, which are all included in number of measurements). Problems indicate problems the fitter had with the data. The most common is an optimize warning which is written as “OptimizeWarning: fit is not optimal” in the problems column, meaning an optimal fit could not be found within the constraints; check the graph to see how non-optimal it is before trusting the k value. Another warning indicates that the program could not find a and k. This shows up as “a fit could not be found” in the problems column and “(value) could not be determined” in the rate, asymptote, std. dev. and 95% C.I. columns, with value being the appropriate column name. This generally only happens in variable asymptote fittings. Half-life is the half-life of the protein in whatever time units you provided in the labeling table.

The graph output files are in the graph folders created in the output folder. These are pdfs named by protein ID.

If roll up is left off the graph will look something like this:



If roll up is selected the graph will instead look something like this:



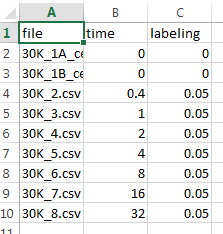
Both graphs have the same data, but differ in displaying one point with an error bar or simply many points to fit the graph. The solid black line is the rate, and the red dots are the points. In a roll-up graph the points have error bars. The black dotted lines indicate the 95% confidence interval for the rate.

WARNING: The errors are for the k value, which does not affect the intercept or asymptote of the graph. Thus the error lines will be closer around the intercept and asymptote, and may seem far smaller than they actually are. Interpret the graphs with caution.

WARNING: Even with the filters, bad points can make it through. The rates can be heavily skewed by errors that agree well. Often this takes the form of a time point with few values and two of the bad points are the same id in different charge states (which agree very well in most cases). Check for these sorts of problems before fully trusting the data. The bad points should be very obvious to the eye.

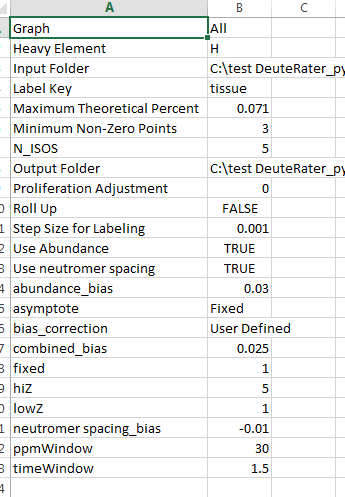
Other files:

Time\_and\_Labeling\_Data is simply the labeling table:



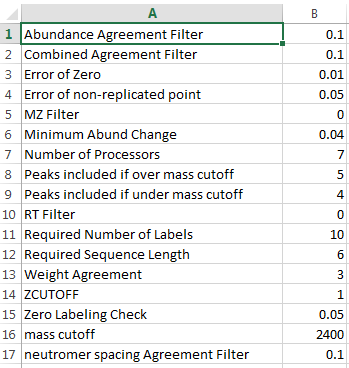
This serves as a reference for you and is also used by the Rate Calculator button.

Settings\_{} (where {} is either Extraction, Isotope\_Analysis, or Rate\_Analysis): These result from the settings menu. Settings are created after the most recent time you pushed the corresponding button on this data. The titles indicate which button they originate from, but content is the same:



The names of these values are not the same as those in the settings menu, but should be fairly obvious. hiZ is maximum z for example, N\_ISOS is short for number of isotopes, which is the number of neutromers to extract, and so on.

The final files are the Filters\_{} files, where {} is Isotope\_Analysis and Rate\_Analysis (the extractor does not use anything from the filters menu) which is like the settings output, but for the filters. Again, there are currently filters for both the Isotope Calculator and Rate Calculator buttons; the content is the same:



If you look at the filters, the names are similar to those in the filter menu, but in different order. The MZ and RT filters translate to “if two sequences are different but are within this m/z and this retention time of each other, both are discarded.” Weight agreement translates as “allowed m/z diff between experimental and theory.”

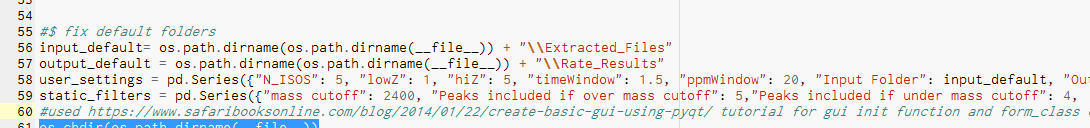
**Altering the Code**

Before beginning, note that altering the code has far fewer checks to prevent issues than the standard user interface. Be cautious making changes beyond the following instructions until you are familiar with the code. Also, always make an archive copy of the module you are modifying and store it somewhere before making large changes.

Finally, when you open the code folder you will notice two types of files: “.py” and “.pyc”. Only modify the “.py” files. The “.pyc” files will be automatically adjusted the next time you run the program.

**Adjusting the defaults**:

The default settings and filters are found near the top of the DeuteRater.py file. These are the values displayed when you first open either the Change Settings or Change Filters option. The filters are shown below surrounded by a red box, and the settings are in the line right above it:

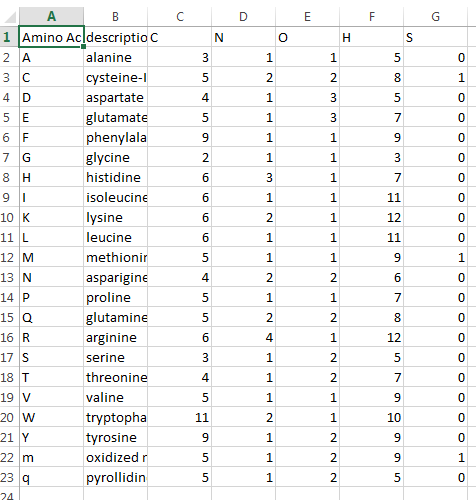


To adjust, find the value you wish to alter and change the value after its colon. The keys are the same as those in the filters and settings output documents discussed above. Save the file when you are done. Changes will take effect when you next start the program.

**Adding a New Element:**

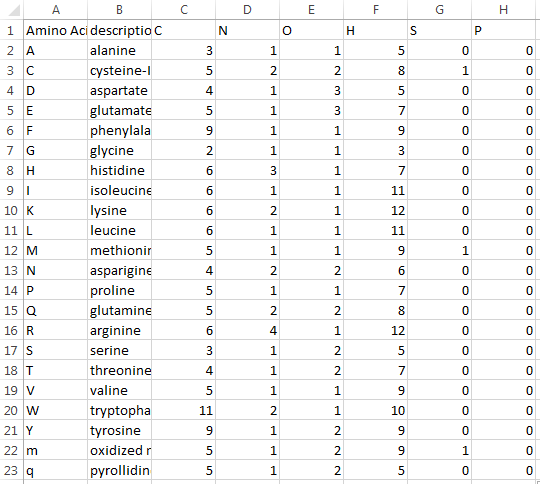
This is a more complex code alteration than adjusting defaults detailed in this tutorial. (Note that this does not cover an element that you are artificially altering the isotopic ratios of; that change can be done but it’s a bit more complex and will not be covered here). For this example, phosphorous will be added in order to look at protein phosphorylation.

To begin, go into the code folder and open a document called “Amino Acid Elemental Composition.csv”; it will look like this:



There needs to be a one letter code for the element we will add. It cannot currently be used by other elements, which are in row 1, columns “C” and after in the document above. “X” is also off limits. For the phosphorylation example, any one letter code that is not C, N, O, H, S or X will work. Keep in mind that this is case sensitive, so if you want to add selenium for example, which has an elemental symbol “Se” and S is already used, you can use “s”.

In our example we will add “P” for phosphorous:



In the column below, you must also add the appropriate number of the new element each current Amino acid has, which is 0 phosphorus for all amino acids in our example. Don’t forget to save the file when you are done. (Usually you will also need to add an extra amino acid or modification, but this has been discussed previously in the Advanced Options Menu.)

Next open a file called Isotopomer\_Class\_vrs2.py (.py files can be opened in a text editor if you do not have a python editor). Go to line 37 of the file, which will look like this:



Add your new one letter code (‘P’ in our example) to the list of letters. In this tutorial, ‘P’ will be added to the end of the list. The values need to be comma separated, remain in the brackets, and the letter must be in apostrophes. It should now look like this:



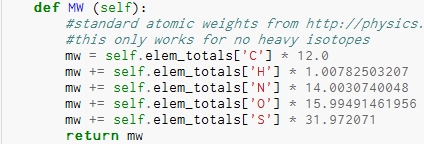
Next go to line 43 in Isotopomer\_Class\_vrs2.py, which looks like this:



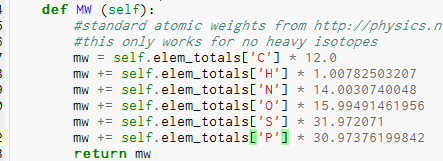
This lists the default values of 1 oxygen and 2 hydrogens for the ends of the peptide backbone. The new element must be added to this list. You will likely simply need to add a 0 after the colon for the new element. In this example, adding the new element so it is similar to the other elements looks like this:



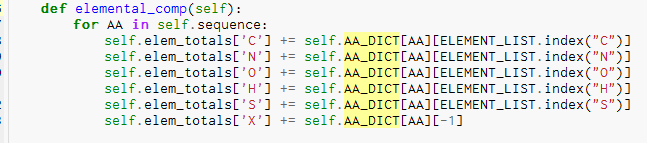
Next go to a block of code starting on line 54, which looks like this:



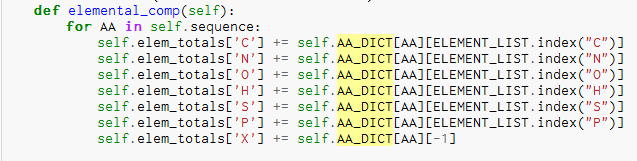
Add the new element to the bottom, above the line that says “return mw”. The number after the asterisk is the mass of the light isotope (e.g. 14N, not 15N, nor the average listed on the periodic table for nitrogen). This value can be found at <http://physics.nist.gov/cgi-bin/Compositions/stand_alone.pl>. For phosphorous this is 30.97376199842, so the addition looks like this:



Finally, go to the code block that starts at line 66, which looks like this:



Add the new element above element X using the same syntax as the other lines, as shown below.



Save Isotopomer\_Class\_v2.py and close it.

Open \_\_main\_\_.py. Go to line 218, which looks like this:

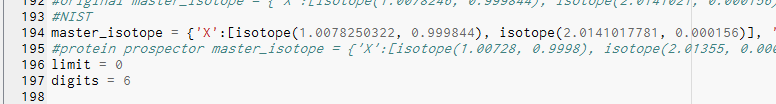


Alter this line to look exactly like the first modification you made to Isotopomer\_Class\_vrs2.py. The order must be exactly the same:

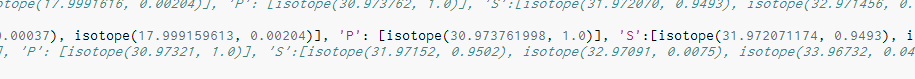


Save and close \_\_main\_\_.py

The last two code modifications will be very similar: either open Emass\_Port\_vrs4.py or Emass\_Port\_vrsFast2.py (the change is the same for both files, so when you are done with one, modify the other file). For either file, go to line 194:



Add phosphorous like this:



A couple of notes on this one before proceeding: The format for entries is “element” : [ isotope(a,b), isotope(a,b)]. The letter “a” is the exact mass for that isotope and “b” is the abundance of that isotope in the total population. Only include biologically relevant isotopes (e.g. note that tritium and 14C are not listed in the code because they are too rare). Isotopes need to be entered in order: +0 neutrons, +1 neutrons, etc. until all are done. Finally, if there is a gap, such as in Sulfur (which has +0,+1,+2,+4 but not +3), you need to enter -1000000 for mass and 0 for abundance for the missing +3 isotope (see below):





Remember to save and close the files after you are done altering the code.

**Using DeuteRater for Analysis of Mass Spectrometry Accuracy and Precision:**

In addition to kinetics, DeuteRater can also be used to analyze how well the mass spectrometer is identifying Intensity or m/z of neutromers. This can be degraded by instrument noise, co-elution with other species of similar m/z, and other factors.

Accuracy: Accuracy checks require non-labeled data. Extraction proceeds as described above. When you create the labeling table, use 0 as the Labeling (Decimal) value for your unlabeled samples. Then press the Isotope Calculator button. Afterwards, open the Calculation\_of\_Fraction\_New\_Protein.csv. Select only those rows which represent the unlabeled sample (sorting by filename or a labeling column is the easiest way to do this). DeuteRater will have attempted to calculate the Turnover at 0 label according to the “% label to use if no label present” in the filters menu. Since there is no heavy label, DeuteRater should have calculated 0 as the amount of turnover. Therefore “M0 fraction new for calculation”, “median neutromer spacing fraction new”, and ”median combined fraction new” should all be close to 0 for every point (how many of these three columns are present depends on the Data Type value in the Settings menu). After collecting these points, you can use statistical tools (e.g. a scatterplot, mean, or median) to evaluate if the points are close enough to 0 consistently enough for your machine.

Precision: Precision follows the same steps as accuracy except that unlabeled samples are not required. Instead of the “fraction\_new” columns you will need to examine the “abundance std\_dev”, “neutromer spacing std\_dev”, and “combined std\_dev” columns. These list how well the neutromer peaks in the pattern agree in their determination of fraction new peptide at that time. The lower the values the better they are. Again you will need to use third party software or manually analyze how close the values are to 0; DeuteRater will not do so.

Note: For Precision, if labeled samples are used, the measurements will be affected if the amount of label is different from the amount entered in Labeling (Decimal) in the Labeling Table. This does not apply to unlabeled samples because no matter how high or low the labeling you tell DeuteRater, the answer should always be 0 for every neutromer.

**Common Problems:**

Q: DeuteRater will not open even though all of the python parts are there.

A: Make sure that your default python meets the criteria. If you already had python on the machine, the command line may default to an older version instead of the one you just updated. If it does, make sure you installed all necessary modules in the PythonPath so the program knows where to find them.

Q: The Current Process line on the interface has not changed in a long time.

A: This could be one of two things. First, some processes take much longer than expected, especially on big data. Second, there may have been an error. Check the command line, which will display an error message if an error has occurred.

Q: There is no error message, but all the extracted files have only a header.

A: Check that your headers are on the right columns. If they are, check that the time is listed in seconds. This problem usually results from everything being erroneously filtered out.

Q: The Variable Modification column takes a long time to change when making the id file. Do I have to use that format?

A: No, but you must do a different check. All the variable modifications do is tell the program which letters to replace with which other letters. If you do that manually (the sequence gets read into the program with all pyrolidines already indicated by q for example), you can just put Variable Modification(s) over an empty column and the program won’t notice. Some search programs like Agilent MassHunter do this automatically. However, if you don’t change the sequences and ignore the variable modifications, you will lose the sequences due to disagreement with the theoretical weights. Also, any unrecognized modifications present will be discarded.

Q: My graph has no bottom error line.

A: There are some cases in which the 95% confidence interval is greater than the rate, which causes the error bar to be very negative. In these cases, DeuteRater forces the error bar to be zero. If this is the case and you have no points below zero, the line will not be visible against the x-axis at the bottom of the graph.

Q: I don’t like the Advanced Options sub-menus. Is there a way to avoid them?

A: Yes. The sub-menus (Elemental Composition, Labeling Site Patterns, and Modification) of the Advanced Options menu can be circumvented. They are altering 3 .csvs in the code folder: Amino Acid Elemental Compositions.csv, Number of Labeling Sites.csv, and Modifications.csv respectively. If you desire you can open these files and alter them directly. The same caveat that applies to altering the code applies here: the sub-menus protect the program from improper input, so the program doesn’t check for bad input. If you break the rules when altering the .csvs you can get bad data or cause the program to error out. Use caution and make back-up copies.

Q: I am getting an error you didn’t address.

A: Contact the corresponding author with detailed information. What were you doing? Did this work before? Etc. A screen shot of the command line report of the error is helpful. We will attempt to contact you to resolve the error.

**Updates:**

**2.1 and 3.1:**

There are two major updates in this patch. The first is the addition of bias corrections as described on pg. (). The second is an alteration to the extractor. In the extractor is designed to extract information for rates so the Intensities reported are the best ratios for a particular sequence in a particular file normalized to m0. Originally this was multiplied by the amount of the highest m0 value observed for that sequence in that file to give an idea of the maximum intensity during testing. This confused some users to think that was area under the curve and so could be used for quantitation. We have removed the multiplication step and apologize for any misunderstanding.

**Citations:**

1. A cross-platform toolkit for mass spectrometry and proteomics. Chambers, M.C., et. al. *Nature Biotechnology* **30**, 918-920 (2012).

2. Measuring the dynamics of *E. coli* ribosome biogenesis using pulse-labeling and quantitative mass spectrometry. Chen, S.C. et al. *Mol. BioSyst*., 2012, **8**, 3325–3334

Though not mentioned in the tutorial, the calculations of intensities and m/z values for theoretical amounts is a python port of emass, which is code from:

Rockwood, A.L. and Haimi, P.: "Efficent calculation of Accurate Masses of Isotopic Peaks",

*Journal of The American Society for Mass Spectrometry*, JASMS 03-2263