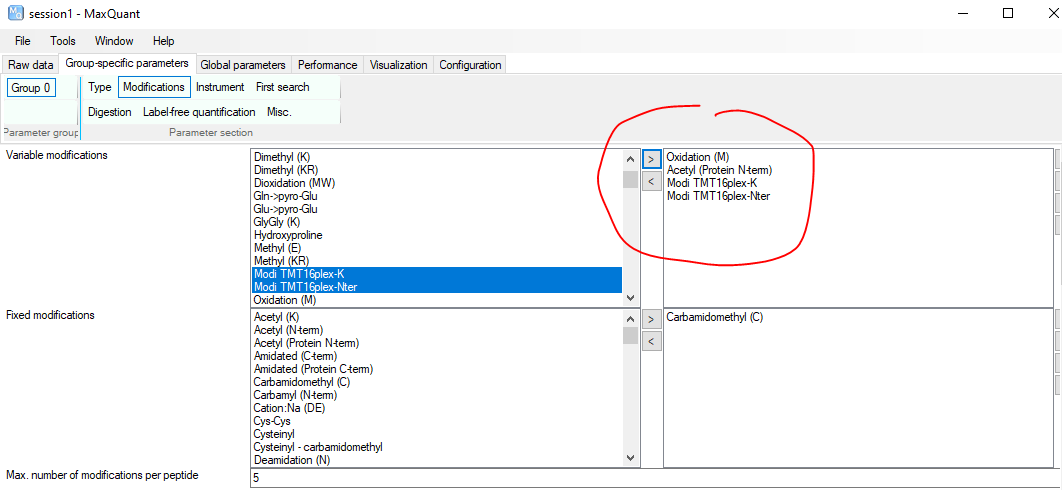
**Why Run this Script**

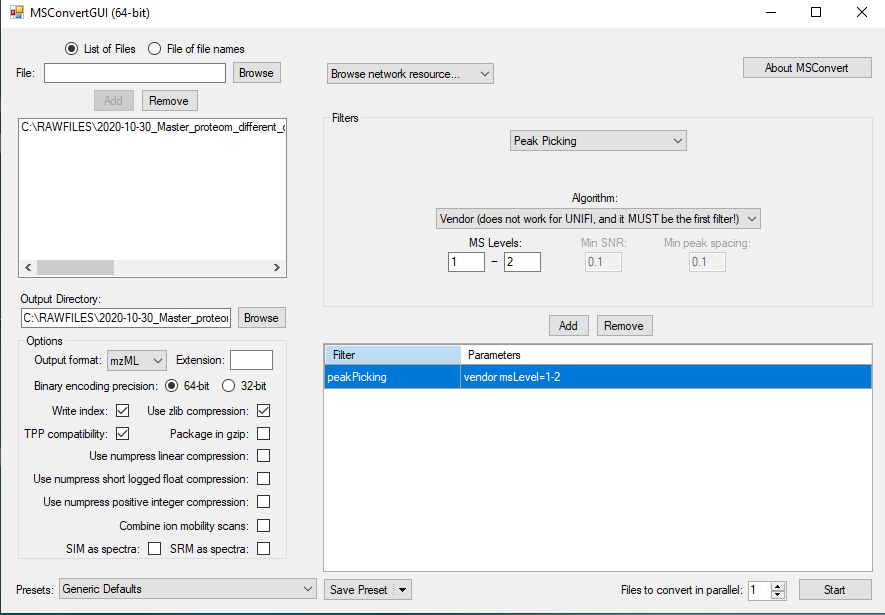
This script can be used to estimate the labelling efficiency of all individual channels present within a single TMT-sample. Consequently, by combining a small aliquote of each labelling reaction, a single measurement on the mass spectrometer is sufficient to check the labelling of all labelling reactions at once by using this script. Please read this User Guide carefully before running the Script.  
Note that unlike Cassiopeia, this script is not that easy to automate - this is because there is a tradeoff between quality and quantity of the remaining data (i.e. partially labeled PSMs) after each optional filtering step. Maybe in the future I will try to automate this script (e.g. code something like: “skip this optional block if number of partially labeled PSMs left is very low”).

**Step-by-Step Guide**

* **Generate a TMT-mix sample** by combining small volumes of each labelling reaction (Equal mixing is beneficial. While the script accounts for unequal mixing by normalization of total channel intensities, it can be assumed that the estimates for the individual labelling efficiencies will be less accurate when the mix is far from equal). Run the sample on the mass spectrometer with the required resolution for your TMT Experiment (i.e. TMT10 and TMT16/pro require at least 45k Resolution).
* **Search the resulting raw-file in MaxQuant** with TMT-modifications as variable (N-terminally as well as on Lysine), see:



* **When the search is done, use MSConvert to convert the thermo raw-file into an mzML – file**. Make sure that you perform peak-picking on all MSn-levels! This way, profile spectra will be converted into centroided spectra, i.e. all peaks will be reduced to a single intensity value on a certain m/z value.



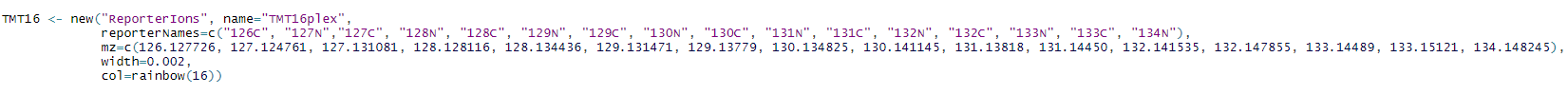
* **Copy the mzML-file as well as the MaxQuant msms.txt into the folder where you run this script**

* **Open the Script in R**, and set your working directory to where the mzML-file and the msms.txt are located.
* **Run section A) REQUIREMENTS of the script**. If the required packages are not installed yet, install them as instructed in this section.
* **Carefully go trough section B) PARAMETERS**. Enter the parameters that are specific to your experiment and MQ output, then run the code. The parameters will now be explained in more detail:

**-** ***wd***A character denoting the filepath of the directory where your files are stored. This parameter will set your working directory to where it is needed. Alternatively, you can set your working directory by using the R-studio interface functionalities; or use the function choose.dir() to retrieve the filepath.

**-** ***mzml\_file***A character denoting the filename of your mzmL-file

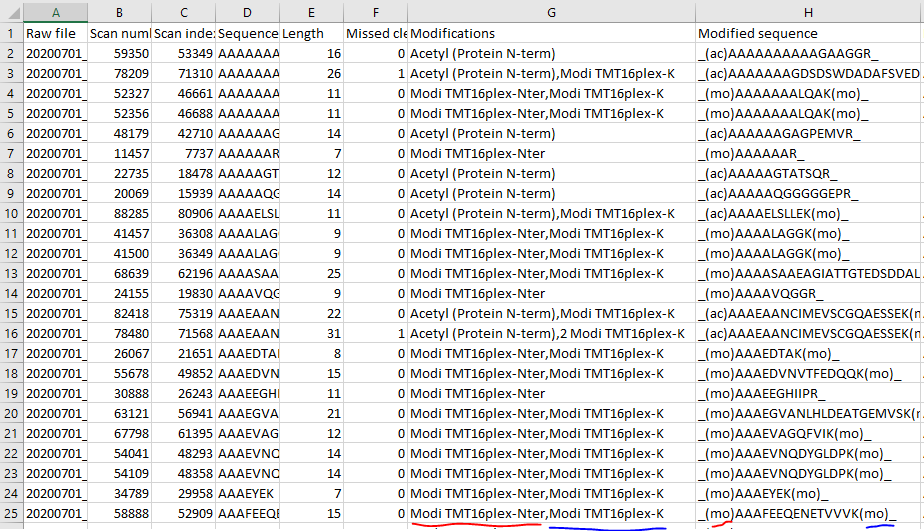
**- *msms.txt\_file***A character denoting the filename of your msms.txt file

**- *tmt\_reporter***An object of class “ReporterIon”, as defined by the R-package MSnBase. This parameter specifies which reporter ions you expect in your sample. If you are using TMT6 plex or TMT10 plex, just specify this parameters as “TMT6” or “TMT10” (without the quotation marks); these are predefined objects by MSnBase. If you want to define your own set reporter ions, you can create your own ReporterIon object by the function new( ), as can be seen at the end of section A) REQUIREMENTS, where “TMT16” is defined:

For more information on how to define your own ReporterIon object, see the following link:  
<https://lgatto.github.io/MSnbase/reference/ReporterIons-class.html>

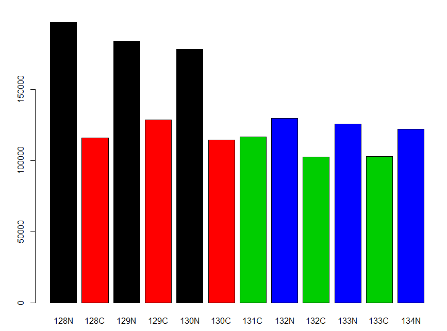
**- *TMT\_mod\_abbreviation***A character denoting how TMT-modifications are specified in the MaxQuant msms.txt column “ModifiedSequence”. Please open the msms.txt in Excel and inspect this column yourself. Based on how TMT variable modifications were saved in MaxQuant (as well as MQ version itself) this can differ.

In the following example (see picture), this parameter would be specified as:  
TMT\_mod\_abbreviation <- “mo”



**- *N\_term\_acetyl\_abbreviation***A character denoting how N-terminal acetylation is specified in the MaxQuant msms.txt column “ModifiedSequence”. Please open the msms.txt in Excel and inspect this column yourself, as this can differ between MQ versions,

In the example above (see picture), this parameter would be specified as:  
N\_term\_acetyl\_abbreviation<- “ac”

**- *groups***  
A numerical vector with length equal to the number of samples in your experiment, specifying the group of each sample. For example, if you have a 6plex with biological replicates A,A,A,B,B,B (group A and group B are different groups), this parameter would be set as:  
groups <- c(1,1,1,2,2,2)  
This parameter is required for filtering out PSMs of partially labeled peptides that have high variance between groups but low variance within groups (i.e. their expression pattern shows strong difference between groups). If these PSMs were kept, these group-biased PSMs would bias the estimation of label efficiency. Therefore they are filtered out by a ANOVA-cutoff. Example of observed spectrum that gets removed by this filter (colors correspond to groups):

Note: for this experiment, the parameter groups would be:  
groups <- c(1,2,1,2,1,2,3,4,3,4,3,4)

If you have no groups or you want to skip this filtering step, just set this parameter to:

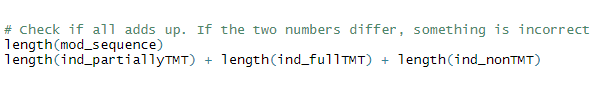
groups <- NULL

- ***ANOVA\_pval\_cutoff; pif\_cutoff; precursor\_intensity\_quantile\_cutoff\_partiallyTMT; score\_cutoff***

These parameters you can keep at their default values. They will be used to filter out PSMs of partially labeled PSMs that are deemed less trustworthy/accurate, which ultimately ensures better estimation of labeling efficiencies. If you want a more stringent filtering because you have many partially labeled PSMs, you can change them and then rerun part of the script

* **Run section C) READ IN ID DATA (MSMS.TXT).**

* **Run section D) FIND FULLY LABELED/UNLABELED/PARTIALLY LABELED PSMS.**  
   In this section, you can (and should!) check 3 things:

**-1)** check the number of fully labeled PSMs , unlabeled PSMs and partially labeled PSMs via length(ind\_fullTMT), length(ind\_nonTMT), length(ind\_partially\_TMT). It is unlikely that any of those should be 0!  
  
**-2)** check if the number of fully labeled + unlabeled + partially labeled PSMs add up to the total number of PSMs observed. Basically, these two lines of code should result in the same number:

If either 1) or 2) seem off, check if your parameters are correct! (i.e. TMT\_mod\_abbreviation)  
  
  
**-3)** check the global labeling efficiency label\_eff\_global, calculated as the overall percent non-acetylated amino-groups (i.e. amino groups that can react with TMT to get labeled) that were reported as labeled in the msms.txt file.   
Example: First all acetylated amino groups are excluded for this calculation. Then, if say 90% of the remaining amino groups were found labeled in the msms.txt, and 10% were reported unlabeled (i.e. free, consequently sitting on unlabeled or partially labeled peptides), the global labeling efficiency would be calculated as 90%.  
If this global labeling efficiency very high (>99%), I recommend you skip all code-blocks in section E) that are described as OPTIONAL. These code blocks are for the most part additional filter steps and are more relevant if labeling efficiency did not go too well, and you want accurately estimate which channels have had bad labeling. If the labeling efficiency is <99%, I would run the whole script including the OPTIONAL code blocks to make sure you don’t have a single channel or two that are worse than all the rest (as this would suggest a potential need for relabelling).

At the End of section D), If not chosen otherwise, the script will automatically determine if the optional code-blocks in section E) should be run. This is determined based on the calculated global labeling efficiency label\_global; and the cutoff is 98% (If below this cutoff, OPTIONAL is set to TRUE)

* **Run section E) READ IN MZML-FILE; FILTER MORE STRINGENTLY (OPTIONAL) AND EXTRACT REPORTER INTENSITIES**,  
   but execute all code blocks (there are 10 in total in this section, and they are numbered in the script) one after another! In case you want more stringent filtering of partially labeled PSMs, also run the OPTIONAL code blocks, and after each filtering step, check how many PSMs of partially labeled peptides are left via >table(msLevel(df\_spec\_partiallyTMT)). Note that this code line is also in the script, you don’t have to type it yourself.   
  If the number of partially labeled PSMs becomes too low (say <10, but use your own discretion - the default filter parameters are not really that strict), skip all other following OPTIONAL code blocks (otherwise you might filter out every single PSM of partially labeled peptides, and then there is no partially labeled PSM left to calculate the labeling efficiency for the individual channels!).  
  If you encounter an error, you might have filtered out the every single PSM (in this case, rerun the code but skip all or some of the OPTIONAL code blocks). You might have also incorrectly specified a parameter, e.g. groups.
* **Run section F) CALCULATE INDIVIDUAL LABELING EFFICIENCIES.**  
   This section calculates the labeling efficiencies of each channel individually in a simple linear equation.  
  This equation states that the global labeling efficiency (as calculated in section D) equals a weighted mean of all individual channel labeling efficiencies; and the weights correspond to the relative channel intensities in the mix. The summed intensity of partially labeled peptides, normalized by the relative channel intensities, is used to calculate “relative labeling efficiencies” (e.g. channel1’s labeling efficiency is 1.3 times worse than channel2’s labeling efficiency). This reduces the equation with x unknown parameters (where x is the number of channels in your experiment) to just 1 unknown parameter, thereby allowing to solve the equation.
* **Run section G) RESULTS.**  
  This section lists the results of this script, i.e. estimated labeling efficiency of all individual channels, global labeling efficiency, relative channel intensities, factors for equal mixing, etc.

**Additional remarks**

* Partially labelled peptides are peptides that have 2 more more free amino-groups ready to be labelled, but at least one of those amino-groups is unlabelled. The main assumption of this script is that labelling reactions with bad labelling efficiencies produce higher numbers of these specific peptides (as well as higher numbers of unlabelled peptides - but unlabeled peptides have no channel/sample identity in a mixture of labelling reactions! That’s why we can’t use unlabeled peptides in a mixture). We therefore expect high reporter intensities for partially labeled peptides in channels that had suboptimal labeling efficiency. I think that this is a fair assumption; and retrospectively testing it on a dataset where the labelling efficiency was first calculated by measurement of individual labelling reactions, where 1 channel was labelled worse than the rest (this 1 channel had 96% labelling efficiency, the others above 99%), the subsequent mix being analysed with this script again indicated quite accurately that this 1 channel had a labelling efficiency worse than the other channels.
* This script does not correct reporter ion intensities for isotopic impurities. It would be possible to do it, but that would require a lot more work for the user of this script. Still I think this effect is negligible.