citFinder-0.1 User Guide

Aaron Maurais

16 December 2018

Contents

1 citFinder program outline

`citFinder` consists of 3 phases: Input, Analysis, and Output.

1.1 Input

1. Read each peptide from `DTAFilter-files` into a data structure in memory.
2. Get peptide modification parameters from existing `sequest.params` files.

1.2 Analysis

1. Calculate b, y and neutral loss ions for each peptide
   - By default, only $M + 1$ ions are considered.
   - The range of charge states to be considered can be specified with the `-minMZ` and `-maxMZ` options
2. Load the corresponding `.ms2` file into a buffer in memory
3. Find the location of the scan in the `.ms2` file buffer and parse it into a `ms2::Spectrum` object
4. Search the `ms2::Spectrum` object for fragment ions within a user defined tolerance.
   - The tolerance can be specified with the `--matchTolerance` option. The default tolerance is $M \pm 0.25Th$.
   - If multiple ions are found in the specified range, ties are broken by intensity, *i.e.* the most intense ion is chosen.
   - This behavior can be modified with the `--multipleMatchCompare` option.
   - In cases where multiple fragments have the same predicted mass, all possible fragments are considered to be found if the ion is found in the `ms2::Spectrum`.
5. Classify the fragment ions which were found.

- Fragment ions are classified into 5 possible groups according to the decision tree in Figure 1.
  - **frag**: Any fragment ion which was found.
  - **ambFrag**: B or Y ion which does not span the citrulline residue.
  - **detFrag**: B or Y ion spanning citrulline residue.
  - **detNlFrag**: Neutral loss fragment spanning citrulline residue and not containing N or Q.
  - **ambNlFrag**: loss fragment spanning citrulline residue containing N or Q.
  - **artNlFrag**: Neutral loss fragment not spanning citrulline residue.

6. Determine whether the peptide contains citrulline based off the decision tree in Figure 2.

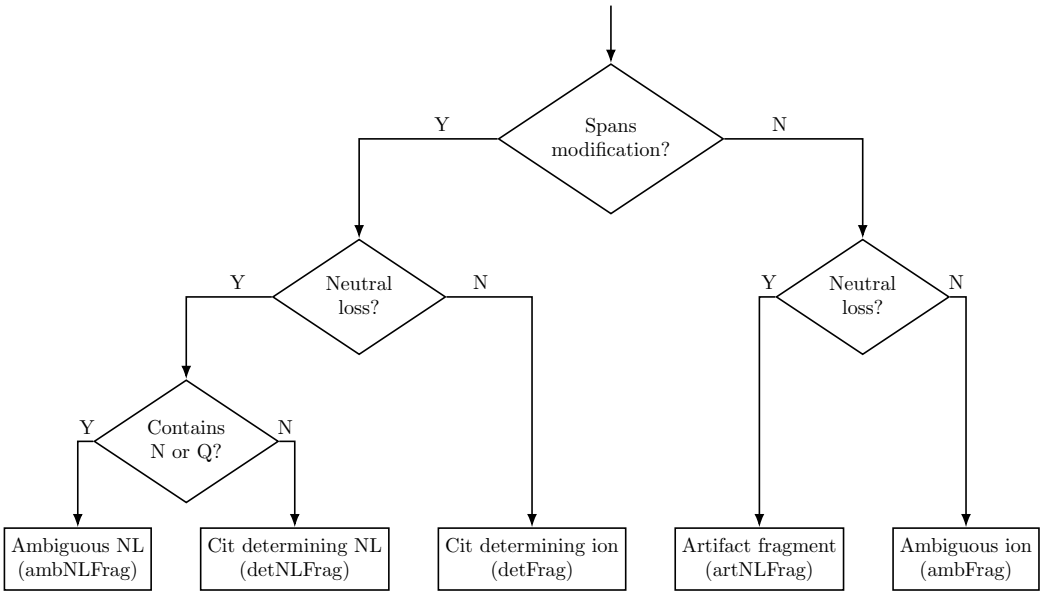


Figure 1: Decision tree for classifying fragment ion type

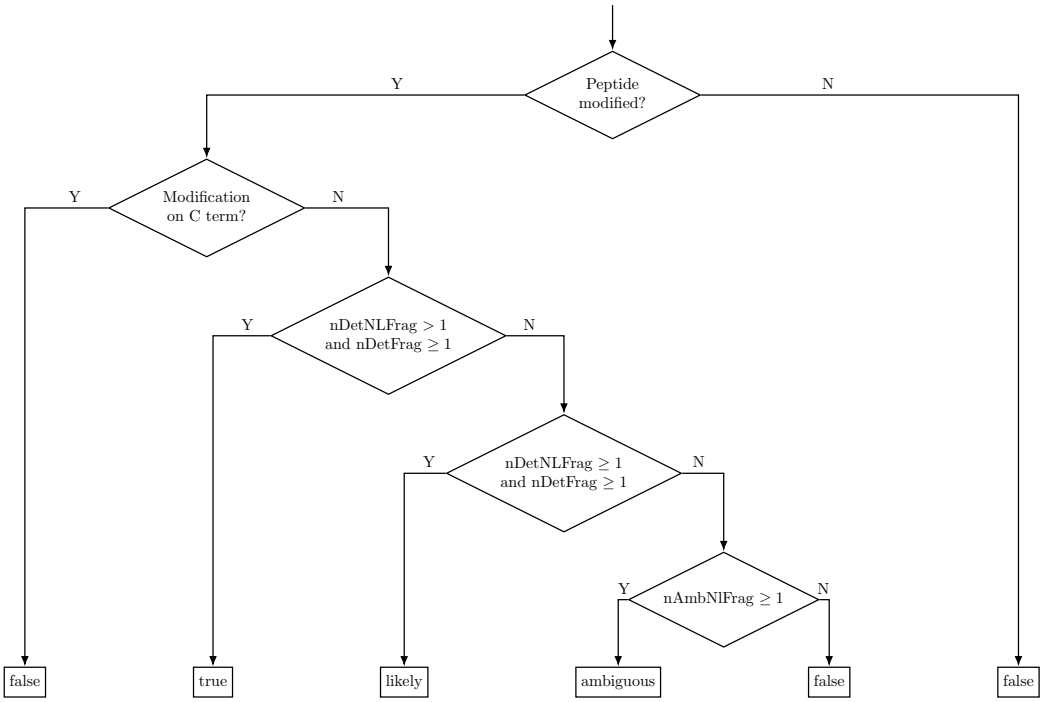
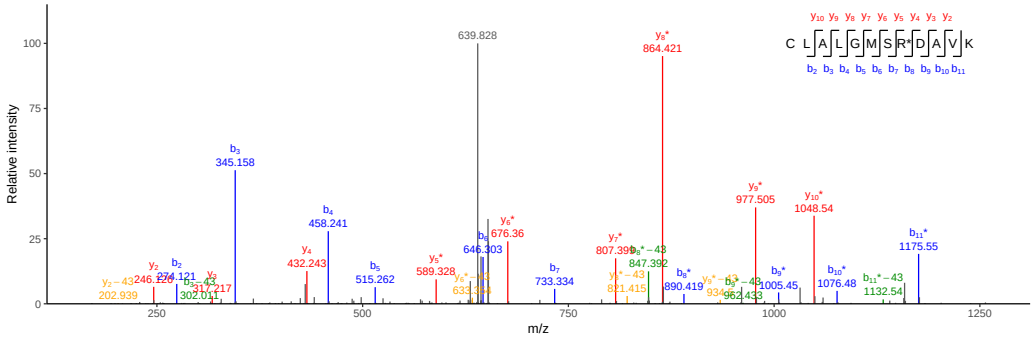


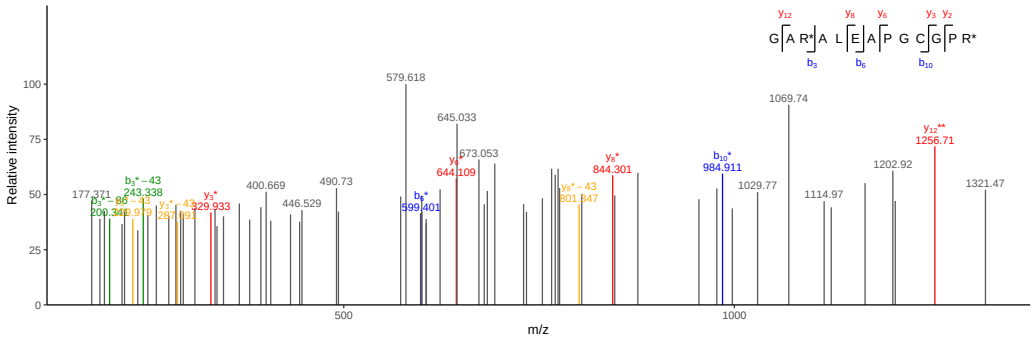
Figure 2: Decision tree for determining whether a peptide contains citrulline.

1.3 Output

- `peptide_cit_stats.tsv`
  - Excel file with a line for each peptide, and columns for sequence, scan number, parent file, whether the peptide contains citrulline (output of decision tree in figure 2) and numbers and types of ion classifications from decision tree in 1.
- If the `--printSpectra` option is specified, a `.spectra` file is generated which can be used to automatically generate a labeled ms2 from a peptide sequence.
  - Example ms2s are shown in figure 3.



(a) A good ms2



(b) A bad ms2

Figure 3: Example labeled ms2s

2 TODO

1. Figure out how to compile the program on `pleiades`.
    - This should be trivial, I just have to take the time to do it.
2. Write comprehensive documentation.
    - I'd like to get the documentation to at least the point that the program is easy for you to use.
3. Figure out how to classify fragment ions in peptides with multiple modified residues.
    - Right now the fragments are classified according to the decision tree in Figure 2 for each citrulline modification, *i.e.* a fragment will end up with two distinct classifications for each each citrulline modification.
    - There is currently no way to determine which modification the classification is referring to in the output file.
    - I'd like your input on this. Is the way things are classified good enough for your use case, or would you rather be able to tell which individual citrulline modification was confirmed without manually looking at the ms2 your self?
    - Again this is only a problem in cases where there are multiple citrulline residues on a peptide.