The .tsv file are the output of the 'search' for peptides, which is a table of data as 'tab-seperated values'.
You can view tab-separated text files in Excel, and many text editors (e.g. Visual Studio Code) have a plugin that allows them to be viewed as a table.
You can easily load them with Python using the csv package - for example, read_tsv = csv.reader("input.tsv", delimiter="\t")
(The csv package normally reads 'comma separated values', so the delimiter paramters tells the package to use tabs instead of commas)

Each line in the file is separated with a newline character \n. Every column is separated with a tab character \t
The first line contains the 'header' - a description of what each column means. (These names should not change).
Each line then contains a peptide that was found by the search, including various details.
The peptide name can be found in the 'peptide' column, and the count of the number of times it was detected is in the 'tot_num_ions' columns.

The same peptide may be found more than once, as it may have some modificaiton added on to it (this is a biological process that frequently occurs).
Don't worry about this, you can just take the one with the largest count ('tot_num_ions')

The .fasta file describes all of the proteins believed to be in the species.
There is a line beginning with > which describes the protein (you don't really need to worry about that)
Then there is a long string of Amino Acid codes, until either you get to the next protein, or the file ends.
The newline characters in this file don't mean anything, they are just there to make the file easier to view as a text file.

The output of this will go into the Random Forest Code, so another .tsv file is fine for output.
There is an example as toy_output.tsv

I have added some toy examples of only 1 or 2 values, so hopefully you can more easily see how this stage would work.
Here is the data they contain:

?? label for random forest regression (after standardized)

Proteins:
AAKAARAAM
AAKAAD

see directly from one .fasta file

Peptides in sample 1:
AAR 100
AAK 50
Peptides in sample 2:
AAK 100

?? how to get 100,50

see directly from the .tsv files

Trypsin breaks up proteins, but in very predictable ways
- After Argenine or Lysine, as long as they aren't followed by Proline

OUTPUT:
AAM 0
AAR 100
AAK 100

?? name from .fasta (split after K/R, ?? proline)
?? number from .tsv (largest count)

?? why no AAD

(We know that AAM was present, but we didn't see it. As AAR is unique in Protein 1 - AAKAARAAM, so we know that Protein AAKAARAAM was present.)
For now, let's use the largest count 'tot_num_ions' that we see across all of the sample files in the output, and 0 if we know we didn't see it.