>cat scripts.py

```python
#!/bin/python3
import re
import functools
import sys

aa_mono_masses = { # avg masses
        'A':71.03711 ,# 71.0788
        'R':156.10111,#     156.1875
        'N':114.04293,#     114.1038
        'D':115.02694,#     115.0886
        'C':103.00919,#     103.1388
        'E':129.04259,#     129.1155
        'Q':128.05858,#     128.1307
        'G':57.02146 ,# 57.0519
        'H':137.05891,#     137.1411
        'I':113.08406,#     113.1594
        'L':113.08406,#     113.1594
        'K':128.09496,#     128.1741
        'M':131.04049,#     131.1926
        'F':147.06841,#     147.1766
        'P':97.05276 ,# 97.1167
        'S':87.03203 ,# 87.0782
        'T':101.04768,#     101.1051
        'W':186.07931,#     186.2132
        'Y':163.06333,#     163.1760
        'V':99.06841 #  99.1326
        }

def trypticdigest(seq, minwt=500):
    # https://web.expasy.org/peptide_mass/
    # filters out fragments < 500 Da by default, so we do that as well
    # requires there to be no proline right before a cut, assumes full digestion
    return list(filter(lambda x: mass(x) > minwt, re.split(r'(?<=[K|R])(?!P)', seq)))

def mass(seq):
    if seq == '':
        return 0
    return sum(map(lambda x: aa_mono_masses[x], seq)) + 3*1.0078 + 15.9949 # add one H2O and one H

def fragmass(fragseq):
    # b fragments
    bfrags = [0] * len(fragseq)
    acc = 0
    for ind, aa in enumerate(fragseq):
```

```
45          acc += aa_mono_masses[aa] # avoid recalculating the fragment
               by caching the value
46          bfrags[ind] = acc + 3*1.0078 + 15.9949 # add one H2O and one
               H
47          # - 0.000548579909 # subtract one electron # i think thats
               not included in the proton mass?
48
49      # y fragments
50      yfrags = [0] * len(fragseq)
51      acc = 0
52      for ind, aa in enumerate(fragseq[::-1]):
53          acc += aa_mono_masses[aa] # avoid recalculating the fragment
               by caching the value
54          yfrags[ind] = acc + 3*1.0078 + 15.9949 # add one H2O and one
               H,
55          # - 0.000548579909 # subtract one electron
56
57      return (bfrags, yfrags)
58
59
60  with open(sys.argv[1], 'r') as f:
61      f.readline()
62      seq = functools.reduce(lambda x, y: x + y, [x.strip() for x in f.
           readlines()])
63      #print(seq)
64      print("digests:", len(trypticdigest(seq)))
65      print("no wt filter:", len(trypticdigest(seq, minwt=0)))
66      print("fragment masses:", fragmass('MAINHTGEK'))
```

```
>./scripts.py p00533-1.fasta
```

```
1  digests: 82
2  no wt filter: 122
3  fragment masses: ([150.05879000000002, 221.09590000000003,
       334.17995999999994, 448.22288999999995, 585.2818000000001,
       686.3294800000001, 743.3509400000002, 872.3935300000002,
       1000.4884900000002], [147.11326, 276.15584999999993,
       333.1773099999999, 434.22498999999993, 571.2839, 685.32683,
       798.41089, 869.448, 1000.48849])
```