Figure S1: Pseudo-code for the *degradation graph* construction

Pseudo code for the *degradation graph* construction. The notation was already defined in the text but for completeness and an easier reading we repeat it here.

Given a node v in the *degradation graph*, $s(v)$ denotes the amino acid sequence of the peptide associated with the node v. The length of the amino acid sequence is given by $|s(v)|$. $s(v)[a,b]$ with $1 \leq a \leq b \leq |s(v)|$ is the subsequence of the amino acid sequence from position a to position b. $m(v)$ denotes the mass of the peptide associated with the node v. If we could identify a signal that corresponds to the peptide associated with v, we will denote it's intensity with $I_{m(v)}(t_i)$. The association between mass and intensity takes into account, that mass spectrometers measure only mass to charge ratios and therefore cannot distinguish peptides with equal mass. Therefore different peptides with equal mass can be associated to the same intensity value, without counting the signal twice in the later analysis. The set of all peptide masses in the graph is denoted by M. We further introduce a queue of nodes L, which is empty at the beginning of the construction.

```
1: function VERIFICATION(Degradation Graph g, Spectrum P, Time t)
2:     L ← {}
3:     for each node v in g do
4:         if P contains a signal p for peptide s(v) then
5:             L ← {L, v}
6:             I_{m(v)}(t) ← intensity of p
7:         end if
8:     end for
9:     return L
10: end function
```

```
1: function EXTENSION(Degradation Graph g, Spectrum P, Time t, Node List L)
2:     for each node u in L do
3:         if P contains a signal p for peptide s(u)[2, |s(u)|] then
4:             create node v, with s(v) ← s(u)[2, |s(u)|] and I_{m(v)}(t) ← intensity of p
5:             add edge u → v
6:             L ← {L, v}
7:         end if
8:         if P contains a signal p for peptide s(u)[1, |s(u)| − 1] then
9:             create node v, with s(v) ← s(u)[1, |s(u)| − 1] and I_{m(v)}(t) ← intensity of p
10:            add edge u → v
11:            L ← {L, v}
12:        end if
13:        for each c, 2 < c < |s(v)| − 1 do
14:            if P contains signals p_v, p_w for peptides s(u)[1, c] and s(u)[c + 1, |s(u)| − 1] then
15:                create nodes u_c
16:                add edge u → u_c
17:                create node v, with s(v) ← s(u)[1, c] and I_{m(v)}(t) ← intensity of p_v
18:                create node w, with s(w) ← s(u)[c + 1, |s(u)|] and I_{m(w)}(t) ← intensity of p_w
19:                add edge u_c → v
20:                add edge u_c → w
21:                L ← {L, v, w}
22:            end if
23:        end for
24:    end for
25: end function
```