

# Supplement to “A flexible workflow for building spectral libraries from narrow window data independent acquisition mass spectrometry data”

Lilian R. Heil<sup>1</sup>, William E. Fondrie<sup>1</sup>, Christopher D. McGann<sup>1</sup>, Alexander J. Federation<sup>1</sup>, William S. Noble<sup>\*,1,2</sup>, Michael J. MacCoss<sup>\*,1</sup>, and Uri Keich<sup>\*,3</sup>

<sup>1</sup>Department of Genome Sciences, University of Washington, Seattle, WA, USA

<sup>2</sup>Paul G. Allen School for Computer Science and Engineering, University of Washington, Seattle, WA, USA

<sup>3</sup>School of Mathematics and Statistics, University of Sydney, NSW, Australia

November 22, 2021

## 1 Supporting Information

- **Supplemental File S1:** PDF containing pseudocode for the search procedure along with supplemental tables and figures.
- **Supplemental File S2:** Python script (`precursor_matrix.py`) used to convert Tide search results to matrix format.
- **Supplemental File S3:** R script (`process_precursors.R`) used to perform search.
- **Supplemental File S4:** Python script (`precursor_confidence.py`) used for FDR control.

<i>allS</i>	set of MS2 spectra structured as a list of arrays, one for each isolation window
<i>allD</i>	database of all possible peptides (where each peptide is a specific peptide/charge state combination)
<i>D</i>	structured as a list of arrays, each corresponding to one isolation window
<i>S</i>	database of all possible peptides within a single precursor isolation window
<i>P</i>	all MS2 spectra associated with a single precursor isolation window
<i>m</i>	all possible peptides associated with a single isolation window
<i>n</i>	number of peptides
<i>p<sub>i</sub></i>	number of MS2 spectra
<i>s<sub>j</sub></i>	a single peptide
<i>M<sub>i,j</sub></i>	a single MS2 spectrum
<i>A</i>	score assigned to the match between peptide <i>p<sub>i</sub></i> and spectrum <i>s<sub>j</sub></i>
<i>peak</i>	vector of changepoints
<i>TDpairs</i>	an m-dimensional structure array which stores the following information for each peptide: the maximally matching spectrum ( <i>peak.m</i> ), the corresponding score ( <i>peak.s</i> ), the width of the peak ( <i>peak.w</i> ), and its left and right boundaries ( <i>peak.l</i> , <i>peak.r</i> )
<i>groups</i>	a data structure containing target/decoy pairing information
<i>isTarget</i>	desired grouping designations for FDR control
<i>Ip</i>	a logical vector indicating the target/decoy status of each peptide in the isolation window
$\tau$	indices of considered peptides
<i>allGroups</i>	acceptance threshold
<i>allIsTarget</i>	desired grouping aggregated across all precursor isolation windows
<i>allScores</i>	target/decoy identification aggregated across all precursor isolation windows
<i>allPs</i>	assigned peptide scores aggregated across all precursor isolation windows
<i>qvalue</i>	IDs of all considered peptides aggregated across all precursor isolation windows
	q-values of all considered target peptides aggregated across all precursor isolation windows

**Table 1: Notation**

---

**Algorithm 1 DIA analysis**

---

```
1: procedure DIASEARCH(allS, allD, groups)
2:   allGroups[1 : length(groups)] = []
3:   allIsTarget = []
4:   allScores = []
5:   allPs = []
6:   for  $i_w = 1 : \text{LENGTH}(\textit{allS})$  do ▷ loop on isolation windows
7:      $D_T := \textit{allD}[i_w]$ ;  $S := \textit{allS}[i_w]$ 
8:     [ $M$ ,  $TDpairs$ ,  $isTarget$ ] := DATABASESEARCHING( $D_T$ ,  $S$ )
9:      $D_{ext} = [D_T, D_T]$  ▷  $D_{ext}$  includes precursor ID of targets and corresponding decoys
10:     $M := \text{CHANGEPOINTDETECTION}(M)$  ▷ Remove leading and trailing “junk” spectra from  $M$ 
11:     $peak := \text{CHROMATOGRAPHICPEAKDETECTION}(M, TDpairs, isTarget)$ 
12:     $M := \text{PEPTIDESCORENORMALIZATION}(M)$ 
13:     $Ip := \text{TDC}(M, TDpairs, peak, isTarget)$  ▷ keep the higher scoring peptide from each
    target-decoy pair
14:     $Ipt := \text{OPTIMALREPRESENTATIVESELECTION}(M, Ip, isTarget, peak)$ 
15:     $Ipd := \text{OPTIMALREPRESENTATIVESELECTION}(M, Ip, \text{NOT}(isTarget), peak)$ 
16:     $Ip := [Ipt, Ipd]$ 
17:     $mAll := 0$  ▷  $mAll$  is the windows-aggregated number of peptides
18:    for  $i_p$  in  $Ip$  do
19:      if  $i_p \neq \text{WHICH.MAX}(M[:, peak[i_p].m])$  then
20:         $Ip = Ip[-i_p]$  ▷ Retain only the maximum peptide per spectrum
21:      end if
22:       $mAll := mAll + 1$ 
23:       $i_g = \text{WHICHGROUP}(D_{ext}[i_p], isTarget[i_p], peak[i_p], groups)$  ▷ Determine the group of  $i_p$ 
    based on its features
24:       $allGroups[i_g] = [allGroups[i_g], mAll]$ 
25:      end for
26:       $allPs = [allPs, D_{ext}[Ip]]$ 
27:       $allScores = [allScores, peak[Ip].s]$ 
28:       $allIsTarget = [allIsTarget, isTarget[Ip]]$ 
29:    end for
30:    for  $i_g = 1 : \text{LENGTH}(groups)$  do
31:       $Ip = allGroups[i_g]$ 
32:       $qvalues[Ip] = \text{QVALUESVIATDC}(allScores[Ip], allIsTarget[Ip])$ 
33:    end for
34:    return ( $allPs[allIsTarget]$ ,  $qvalue[allIsTarget]$ ,  $allScores[allIsTarget]$ )
35: end procedure
36: procedure QVALUESVIATDC(scores, isTarget)
37:    $n := \text{LENGTH}(scores)$ 
38:    $sortPerm := \text{ORDER}(scores)$ 
39:    $scores := scores[sortPerm]$  ▷ sort scores in decreasing order
40:    $isTarget := isTarget[sortPerm]$ 
41:    $nTargetWins := \text{CUMSUM}(isTarget)$ 
42:    $nDecoyWins := [1 : n] - nTargetWins$ 
43:    $estFDR := \min(1, (nDecoyWins + 1) / \max(1, nTargetWins))$ 
44:    $qvalues[n] := estFDR[n]$ 
45:   for  $i = n - 1 : 1$  by  $-1$  do
46:      $qvalues[i] := \min(estFDR[i], qvalues[i + 1])$ 
47:   end for
48:   return ( $qvalues[\text{INVERSEPERMUTATION}(sortPerm)]$ )
49: end procedure
```

---

---

**Algorithm 2 Database searching**

---

```
1: procedure DATABASESEARCHING( $D_T = (p_i)_1^m, S = (s_j)_1^n$ )
2:    $D_D = \text{CREATEDECOYDB}(D_T)$ 
3:   for  $1 \leq j \leq n$  do
4:      $M_T[i, :] := \text{SCOREALLPEPTIDES}(s_j, D_T)$   $\triangleright$  SCOREALLPEPTIDES returns the scores of the
       matches between the spectrum  $s_j$  and every peptide  $p_i \in D$  (here we used Tailor-normalized XCorr)
5:      $M_D[i, :] := \text{SCOREALLPEPTIDES}(s_j, D_D)$ 
6:   end for
7:    $isTarget[1 : m] = \text{TRUE}$ 
8:    $isTarget[m + 1 : 2m] = \text{FALSE}$ 
9:    $TDpairs[1 : m] = [m + 1 : 2m]$ 
10:   $TDpairs[m + 1 : 2m] = [1 : m]$ 
11:  return  $[M := \text{CONCAT}(M_T, M_D), TDpairs, isTarget]$ 
12: end procedure
```

---

---

**Algorithm 3 Changepoint detection**

---

```
1: procedure CHANGEPOINTDETECTION( $M$ )
2:    $n = \text{NROWS}(M)$ 
3:   for  $j = 1 : n$  do
4:      $meds[j] := \text{MEDIAN}(M[:, j])$ 
5:   end for
6:   for  $ncp = 2 : 4$  do  $\triangleright$   $ncp$  is total number of changepoints
7:      $A := \text{CHANGEPOINT}(meds, ncp)$   $\triangleright$  Inputs for changepoint function (described by Killick et al.,
       2016) are vector of scores and number of changepoints,  $l$ 
8:     if  $A_{ncp} - A_1 \geq 0.5 \times n$  then
9:        $M := M[:, A_1 : A_{ncp}]$ 
10:    return  $M$ 
11:   end if
12: end for
13: return ERROR
14: end procedure
```

---

---

**Algorithm 4 Chromatographic peak detection**

---

```
1: procedure CHROMATOGRAPHICPEAKDETECTION( $M, TDpairs, isTarget$ )
2:    $m = \text{NROWS}(M)$ 
3:   for  $i = 1 : m$  do
4:      $\text{med} = \text{MEDIAN}(M[i, :])$ 
5:      $\text{MAD} = \text{MEDIAN}(|M[i, :] - \text{med}|)$ 
6:      $M^{RZ}[i, :] := (M[i, :] - \text{med})/\text{MAD}$ 
7:   end for
8:   for  $i_t = 1 : m$  do
9:     if  $isTarget[i_t]$  then
10:       $i_d = TDpairs[i_t]$ 
11:    else
12:      continue
13:    end if
14:     $[peak[i_t].m, l_t, r_t] := \text{FINDPEAKINROW}(M^{RZ}[i_t, :])$ 
15:     $[peak[i_d].m, l_d, r_d] := \text{FINDPEAKINROW}(M^{RZ}[i_d, :])$ 
16:    if  $l_t + r_t < l_d + r_d$  then
17:       $l = l_d; r = r_d$ 
18:    else
19:       $l = l_t; r = r_t$ 
20:    end if
21:     $peak[i_t].l = \max(peak[i_t].m - l, 1)$ 
22:     $peak[i_t].r = \min(peak[i_t].m + r, n)$ 
23:     $peak[i_t].w = l + r + 1$ 
24:     $peak[i_t].s = M[i_t, peak[i_t].m]$ 
25:     $peak[i_d].l = \max(peak[i_d].m - l, 1)$ 
26:     $peak[i_d].r = \min(peak[i_d].m + r, n)$ 
27:     $peak[i_d].w = l + r + 1$ 
28:     $peak[i_d].s = M[i_d, peak[i_d].m]$ 
29:  end for
30:  return  $peak$  ▷ an m-dimensional structure array storing peak data for each peptide
31: end procedure
32: procedure FINDPEAKINROW( $M_r$ )
33:    $m = \text{WHICH.MAX}(M_r)$ 
34:    $M_r = M_r/M_r[m]$ 
35:   for  $l = 0 : m - 1$  do
36:     if  $M_r[m - l] < 0.75$  then
37:        $l = l - 1$ 
38:     break
39:   end if
40: end for
41:   for  $r = 0 : n - m$  do
42:     if  $M_r[m + r] < 0.75$  then
43:        $r = r - 1$ 
44:     break
45:   end if
46: end for
47:   return  $[m, l, r]$ 
48: end procedure
```

---

---

**Algorithm 5 Peptide score normalization**

---

```
1: procedure PEPTIDESCORENORMALIZATION( $M$ )
2:    $m = \text{NROWS}(M)$ 
3:   for  $i = 1 : m$  do
4:      $q^{0.99} := \text{QUANTILE}(M[i, :], 0.99)$ 
5:      $M[i, :] = M[i, :] / q^{0.99}$ 
6:   end for
7:   return  $M$ 
8: end procedure
```

▷ return the peptide normalized scores

---

---

**Algorithm 6 Target/decoy competition**

---

```
1: procedure TDC( $M, TDpairs, peak, isTarget$ )
2:    $m = \text{NROWS}(M)$ 
3:    $I_p := \emptyset$ 
4:   for  $i_t = 1 : m$  do
5:     if  $isTarget[i_t]$  then
6:        $i_d = TDpairs[i_t]$ 
7:       if  $peak.s[i_t] > peak.s[i_d]$  then
8:          $I_p = [I_p, i_t]$ 
9:       else
10:         $I_p = [I_p, i_d]$ 
11:      end if
12:    end if
13:  end for
14:  return  $I_p$ 
15: end procedure
```

---

---

**Algorithm 7 Selection of optimal representatives**

---

```
1: procedure OPTIMALREPRESENTATIVESELECTION( $M, I_p, isTarget, peak$ )
2:    $Ipt := Ip[isTarget]$ 
3:    $Ipd := Ip[NOT(isTarget)]$ 
4:    $Ipt := IIpt[ORDER(peak.s[Ipt])]$  ▷ Sort scores in decreasing order
5:   for  $i_p$  in  $Ipt$  do
6:      $pRow := M[i_p, :]$ 
7:      $overlaps = \{i \in Ipt : peak[i_p] \cap peak[i] \neq \emptyset \text{ AND } peak.s[i] < peak.s[i_p]\}$ 
8:      $acosNull = 0$ 
9:      $draws := PERMUTE(Ipd)$ 
10:    for  $l = 1 : \min(1000, LENGTH(Ipd))$  do
11:       $pRow2 = M[draws[l], :]$ 
12:       $acosNull := \max(acosNull, (|pRow \cdot pRow2|) / (\|pRow\| \|pRow2\|))$ 
13:    end for
14:    for  $i$  in  $overlaps$  do
15:       $pRow2 = M[i, :]$ 
16:       $acosTest = (|pRow \cdot pRow2|) / (\|pRow\| \|pRow2\|)$ 
17:      if  $acosTest > acosNull$  AND  $[(pRow2 > 0) \cdot (pRow > 0)] / n > 0.25$  then ▷ Check if acos
18:         $Ipt = Ipt[-which(Ipt == i)]$  ▷ Remove the lower scoring peptide from the final matrix
19:      end if
20:    end for
21:  end for
22:  return  $Ipt$ 
23: end procedure
```

---

<b>Parameter</b>	<b>Value</b>
min-length	6
max-length	50
min-mass	200
max-mass	7200
enzyme	trypsin
deisotope	0
digestion	full-digest
missed-cleavages	0
keep-terminal-aminos	NC
num-decoys-per-target	1
min-mods	1
max-mods	1
mods-spec	1STY+79.966331

**Table 2: Parameters for Tide index of phosphopeptide enriched samples.**

<b>Parameter</b>	<b>Value</b>
min-peaks	20
deisotope	0
precursor-window	1.007
precursor-window-type	mz
mz-bin-width	0.02
mz-bin-offset	0.4
spectrum-charge	2
top-match	100000
use-tailor-calibration	true
concat	true

**Table 3: Parameters for direct Tide search of phosphopeptide enriched samples.**



<b>Parameter</b>	<b>Value</b>
RPmax	25
RFmax	300
CorrThreshold	0.2
DeltaApex	0.6
RTOverlap	0.3
AdjustFragIntensity	true
BoostComplementaryIon	true
ExportPrecursorPeak	false
ExportFragmentPeak	false
SE.MS1PPM	20
SE.MS2PPM	40
SE.SN	2
SE.MS2SN	2
SE.MinMSIntensity	5
SE.MinMSMSIntensity	1
SE.MaxCurveRTRange	1
SE.Resolution	15000
SE.StartCharge	2
SE.EndCharge	3
SE.MS2StartCharge	2
SE.MS2EndCharge	3
SE.NoMissedScan	1
SE.MinFrag	10
SE.EstimateBG	true
SE.MinNoPeakCluster	1
SE.MaxNoPeakCluster	3
SE.StartRT	0
SE.EndRT	9999
SE.MinMZ	200
SE.IsoPattern	0.8
SE.MassDefectFilter	true
WindowType	MSX

**Table 4: DIA-Umpire parameters for generating pseudospectra from phosphopeptide-enriched data.**

<b>Parameter</b>	<b>Value</b>
min-peaks	20
deisotope	0
precursor-window	20
precursor-window-type	ppm
mz-bin-width	0.02
mz-bin-offset	0.4
spectrum-charge	2
top-match	3
use-tailor-calibration	true
concat	true

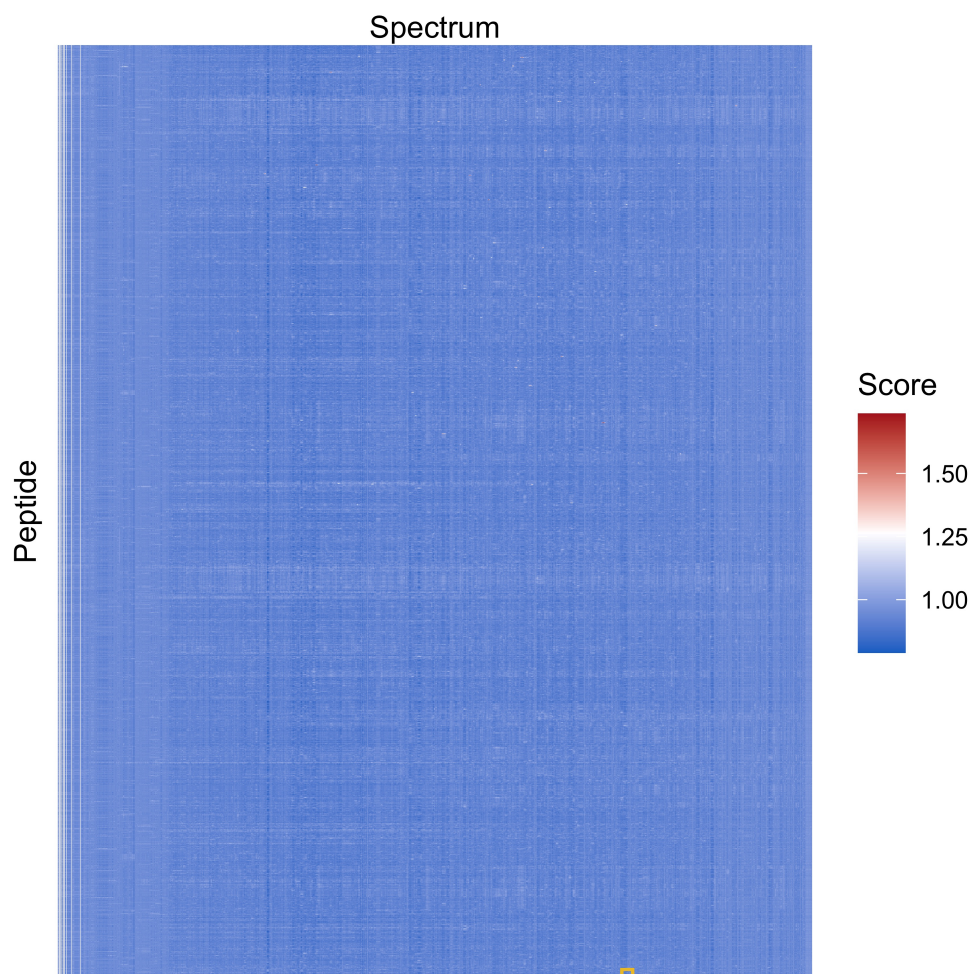
**Table 5: Parameters for Tide search of pseudospectra from phosphopeptide enriched samples.**

<b>Parameter</b>	<b>Value</b>
min-length	6
max-length	50
min-mass	200
max-mass	7200
enzyme	trypsin
deisotope	0
digestion	full-digest
missed-cleavages	0
keep-terminal-aminos	NC
num-decoys-per-target	1
min-mods	0
max-mods	255

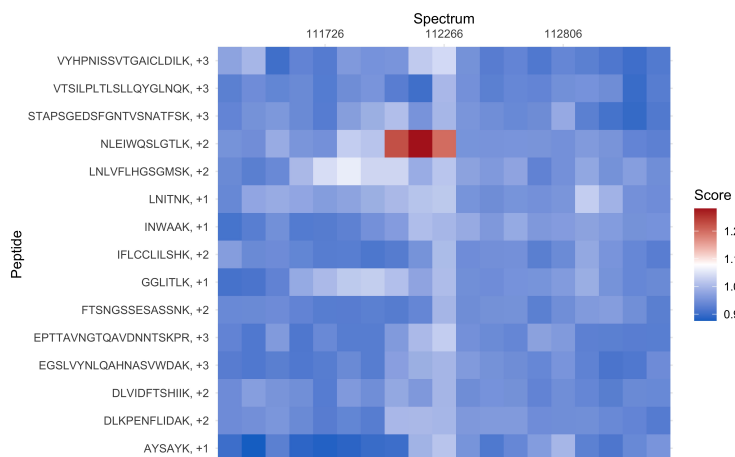
**Table 6: Parameters for Tide index for yeast search.**

<b>Parameter</b>	<b>Value</b>
min-peaks	20
deisotope	0
precursor-window	1.007
precursor-window-type	mz
mz-bin-width	0.02
mz-bin-offset	0.4
spectrum-charge	all
top-match	10000
use-tailor-calibration	true
concat	true

**Table 7: Parameters for direct Tide search of yeast samples.**

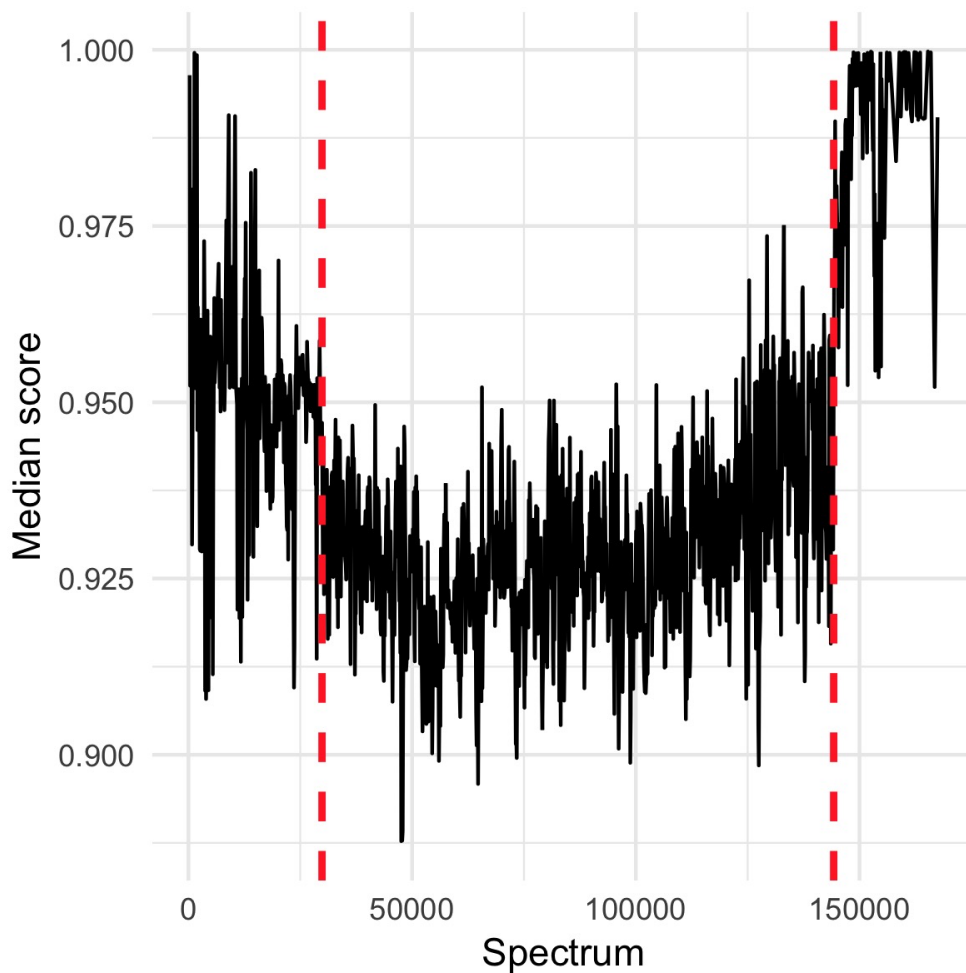


(a) Full matrix

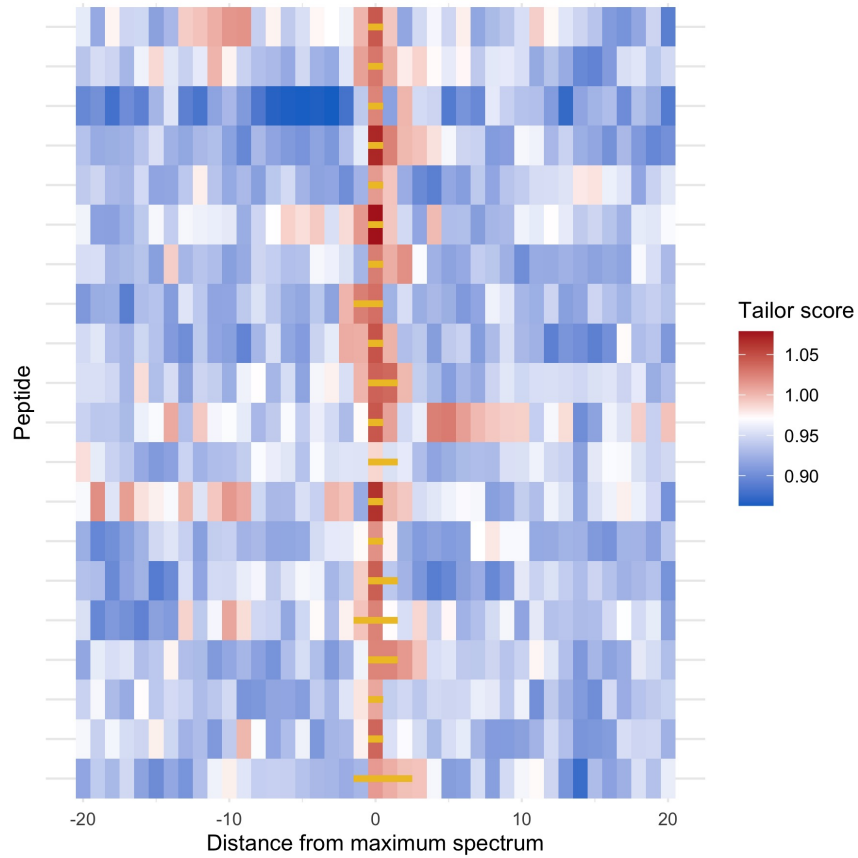


(b) Zoom in of (a)

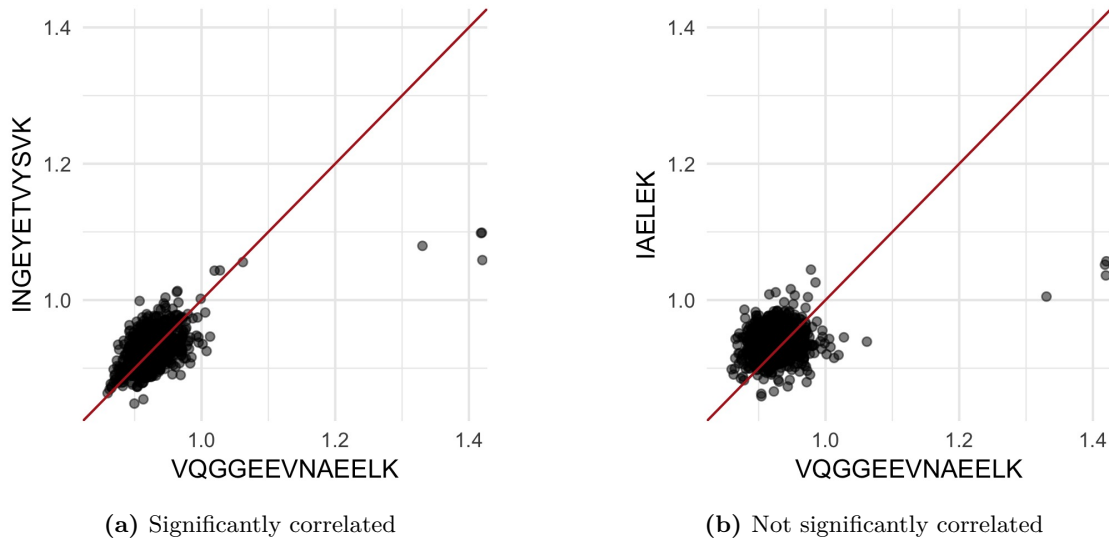
**Figure 1: Graphical representation of a single 2- $m/z$  matrix produced after Tide search (A)** The full matrix has 1733 peptides (rows) and 1439 spectra (columns). (B) A zoom in on the area of (A) marked by the yellow box. The peptide NLEIQQLSLGTLK is accepted at 1% FDR based on this example.



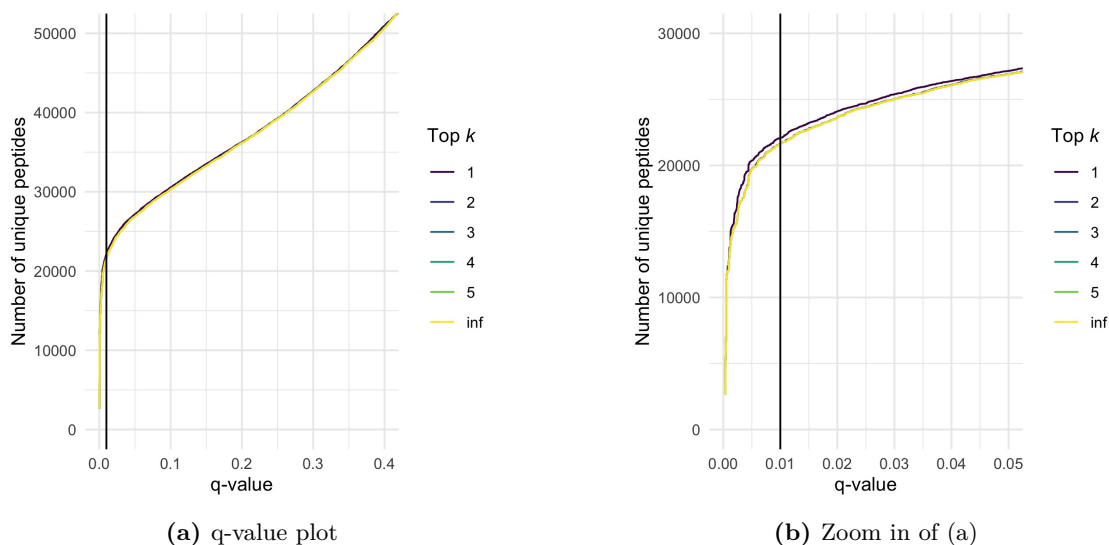
**Figure 2: Plot of median scores for each spectrum across a DIA window.** Changepoint detection automatically removes spectra with median scores distinct from the rest of the run. To account for shifts in score distributions early in the chromatographic gradient, only spectra between red dashed lines are retained for analysis. We first identify two changepoints and retain only the  $n'$  mass spectra between these two changepoints provided  $n' \geq 0.5n$ , where  $n$  is the original number of spectra. If  $n' < 0.5n$  then we repeat the changepoint detection while increasing the number of changepoints, this time considering the  $n'$  spectra between the first and the last changepoints.



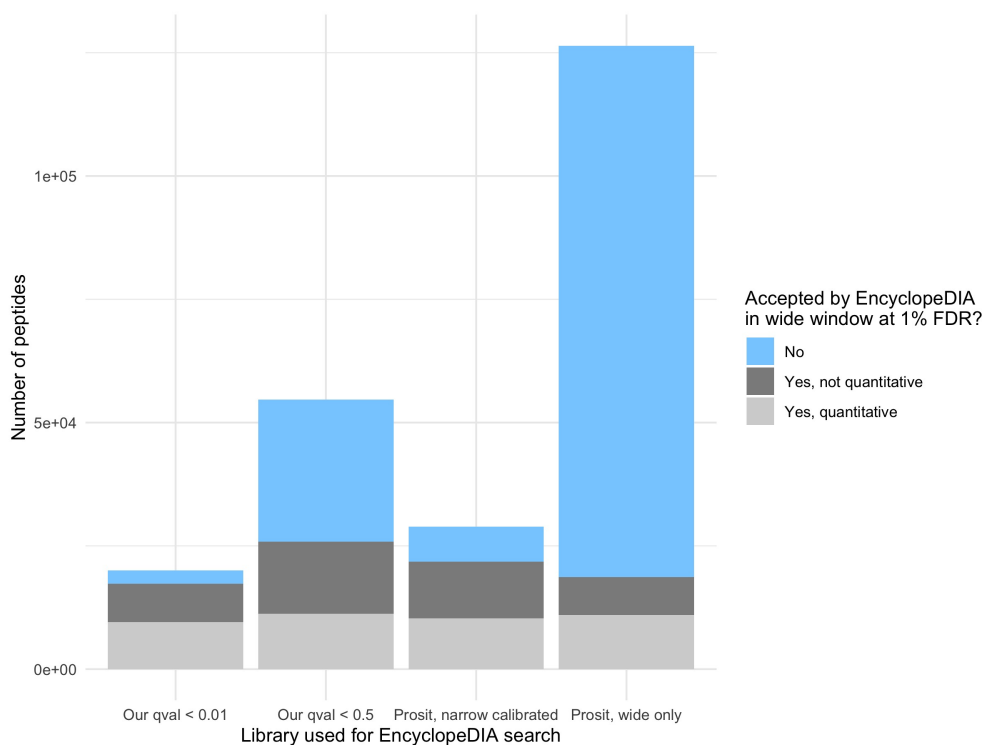
**Figure 3: Peak boundaries assigned with our method.** The figure graphically depicts the  $\pm 20$  scores surrounding the top scoring spectrum for a number of target peptides. Peak boundaries (assigned by Supplemental Algorithm 4) are shown with yellow lines.



**Figure 4: Pairwise scatterplot with each point representing a single spectrum.** In (a), the peptides are identified as significantly correlated. In (b), the peptides are not significantly correlated. If the peptides in (a) share more than 25% of their ions, the lower scoring peptide (y-axis) will be removed.



**Figure 5: Plot of q-values when the top  $k$  peptides are retained.** Here, the top  $k$  scoring peptides for each spectrum are retained for FDR control. Although many DIA spectra are chimeric, it is rare that multiple "true" peptides would share the same optimally matching spectrum. Therefore, using a  $k$  value of 1 does not seem to eliminate a significant number of peptide matches, rather it increases power presumably by culling some high scoring fake matches. The vertical line represents 1% FDR threshold.



**Figure 6: Summary of the number of peptides in EncyclopeDIA runs using alternative libraries.** The figure summarizes the results of running EncyclopeDIA on the wide-window yeast DIA data using four different libraries: the first two runs used narrow-window yeast DIA data processed by our novel tool using a q-value cutoff of 0.01 and 0.5 to generate the library, and the third run used EncyclopeDIA's ability to take advantage of the same narrow-window data to significantly reduce the initial Prosit-generated library. The last run used the full-size Prosit library of all possible tryptic peptides.