Supporting Information

Khan et al. 10.1073/pnas.0904100106

SI Text

To further illustrate the simplicity of the algorithms presented in the text, we include pseudocode for each algorithm below. We assume an implementor has a well-designed kd-tree implementation and an implementation of a connected-components algorithm. Note that mass spectrometers typically operate in one of three modes: raw profile, profile, and centroid. In raw profile mode, the signal in a scan is finely sampled at discrete *m/z* values. In profile mode, only finely sampled *m/z* values above an estimated baseline are stored. In centroid mode, peaks have been located by a simple algorithm in baseline profile spectra. In this work, we assume that the data have been collected in centroid mode. Even though a centroiding algorithm has been applied, a single LC-MS scan will still consist of millions of peaks. In this work, we assume that the data have been collected in centroid mode or converted to centroid mode.

In the first step of processing a single scan, noise peaks are removed by using RemoveNoise(P) below.

RemoveNoise(P). Given a set of peaks, P returns a set of filtered peaks F

```
T = BuildIndex(P)
for each p in P
    Q = RangeQuery(T, p.rt − dt, p.rt + dt,
       p.mz − dmz, p.mz + dmz)
    if (size(Q) > R and p.intensity >minI) add p to F
return F
```

Each peak's retention time and *m/z* value are indexed by p.rt and p.mz, respectively. The parameters dt and dmz set the width and height of the planar orthogonal range query. R thresholds the number of peaks, and minI sets the minimum intensity.

Given a set of filtered peaks F, FindXICs() uses planar range queries to construct a graph structure. Connected components in that graph correspond to XICs. Note that the parameters dtF and dmzF set the width and height of the planar orthogonal range query.

FindXICs(F). Given a set of filtered peaks F returns XICs.

```
D = BuildIndex(F)
for each p in F
    Q = RangeQuery(D, p.rt − dtF, p.rt + dtF, p.mz −
    dmzF, p.mz + dmzF)
    For each returned point q in Q
        Add an edge between q and p in a graph G.
return FindConnectedComponents(G)
```

We made the following modifications in our implementation of the algorithm above:

- We removed XICs that were the result of incorrect centroiding at neighboring *m/z* values of high-intensity XICs. We use a depth-first approach to finding connected components that does not require explicit construction of the graph.
- We reestimated precursor intensity of an MS/MS peak by using surrounding peaks in an XIC.
- All intervals used were expressed in parts-per-million (PPM) where $m/z * \text{ppm} * 10^{-6} = \text{Da}$
- We allowed the user to specify *m/z* ranges of sample contaminants where peaks are filtered out in the LC-MS scan.

To further show the practical utility of using a spatial data structure, we include a function Draw() that illustrates how the spatial data structure T can be used to efficiently draw data points within a region of given width and height on a computer screen. This function plays a key role in allowing an experimenter to assess the quality of the data and set algorithm parameters.

Draw(T, width, height, rt1,rt2,mz1,mz2)

```
Q = RangeQuery(T, rt1, rt2, mz1, mz2)
a = width/(rt2 − rt1)
b = height/(mz2 − mz1)
for each q in Q, DrawPeak(a(q.rt − rt1), b(q.mz − mz1)).
```

Once each scan is processed to find XICs, the computer memory required for the peaks and their corresponding spatial data structures can be freed. All subsequent processing for multiple scans and labeled scans occurs on XICs indexed in a spatial data structure by the retention time and *m/z* dimensions of their most-intense peak.

Multiple scans are handled by AlignAndGroup() below. The algorithm translates all scans to the reference scan to adjust for differences in when data collection was started in both datasets. It then labels XICs with identifiers that indicate which dataset they belong to and combines all of the XICs into one merged dataset.

AlignAndGroup(S)

```
for each pair (R, C) in S where R is a reference scan
    drt = GetTranslation(C, R)
    Translate each XIC in C by drt in the
      retention time dimension
for each Si in S
    Mark all of the XICs in Si as originating from scan i
    Add these labeled XICs to the set Z.
return GroupXICs(Z)
```

Aligning and grouping requires that GetTranslation() uses reciprocal nearest-neighbor queries to determine corresponding XICs. The median difference in retention time between these XICs is used to align a scan to translation to a reference scan R.

GetTranslation(C, R). Computes the median translational difference between scans.

```
for each XIC x in C
    Q = RangeQuery(R, x.mz − dmzA, x.mz + dmzA, x.rt − drtA,
    x.rt + drtA) find nearest XIC b in Q
    Q = RangeQuery(C, b.mz − dmzA, b.mz + dmzA, b.rt − drtA,
    b.rt + drtA) find nearest r in Q
    if x = r save drt = x.rt − b.rt
return median of drt values
```

Once scans are aligned to translation GroupXICs() uses planar orthogonal range queries to construct a graph connecting XICs in a merged dataset Z. The range queries use the start and end time of the XIC and a user-specified parameter width in the *m/z* dimension of an XIC. Connected components in this graph correspond to XICs that have been grouped across scans. Instead of using nonlinear parametric alignment, the range queries automatically account for variance in the position of XICs.

GroupXICs(Z). Groups XICs in Z.

```
T = BuildIndex(Z)
for each x in Z
    Q = RangeQuery(T, x.start, x.end,
       x.mz − dwidth, x.mz − dwidth)
    For each returned point q in Q
        Add an edge between q and x in a graph G.
```

return FindConnectedComponents(G)

We made the following modifications in our implementation of the multiscan processing algorithms described above:

- For datasets with replicate runs, we apply grouping hierarchically. We group replicates first, eliminating XICs that do not occur in a sufficient number of replicates.
- We identified ambiguous grouping of XICs when two or more XICs from the same scan occurred in a group. All of the XICs in this group for this scan were removed.

In IsotopePairs(), isotope-labeled data are handled by grouping XICs within a single scan to find light and heavy isotope pairs. The algorithm iterates through XICs by increasing value in the *m*/*z* dimension, pairing light XICs with XICs caused by the heavier species first. By building a spatial index on the XICs, the algorithm assures the same XICs are returned by reciprocal planar orthogonal range queries by using the given label shift in the *m*/*z* dimension.

IsotopePairs(X). Finds isotope pairs in a given XIC set X

```
T = BuildIndex(X)
Sort-by-increasing-m/z(X)
for each x in X
    if x has been paired then skip
    Q = RangeQuery(T, x.start, x.end,
        x.mz + labelshift − tol,
        x.mz + labelshift + tol)
    find largest XIC x2 in Q
    Q2 = RangeQuery(T, x2.start, x2.end,
        x2.mz − labelshift − tol,
        x2.mz − labelshift + tol)
    find largest XIC r in Q2
    if r = x then save isotope pair (x, x2)
```

We made the following modifications to the algorithm presented above:

- We repeat this process in reverse starting from XICs with large *m*/*z* values to handle the case where the heavy XIC was the only XIC with a fragmentation spectrum.
- We rely on the instrument determined precursor charge to compute isotopic spacing between two XIC pairs.

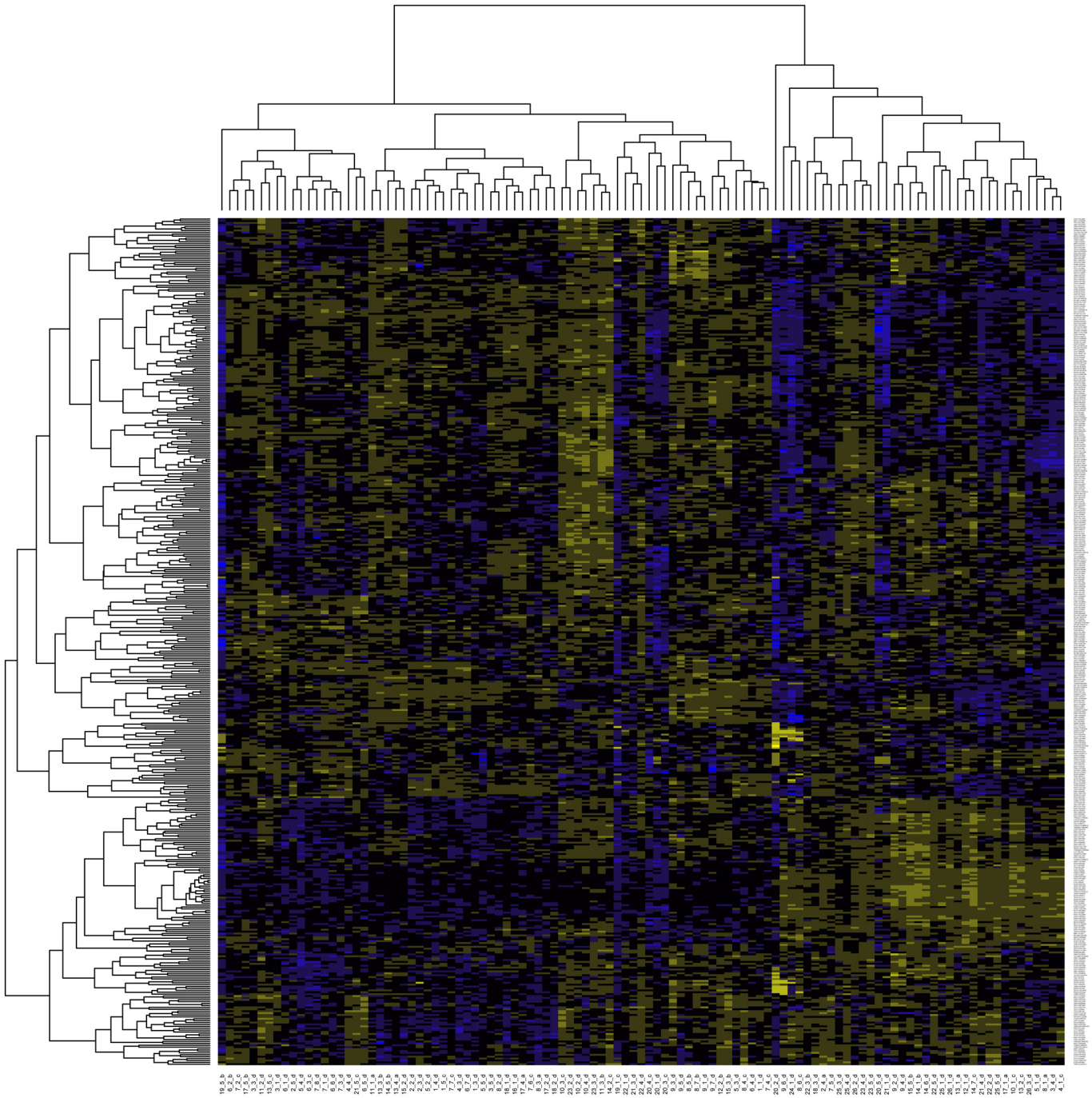

Fig. S1. Hierarchically clustered protein abundance data. 447 unique proteins were quantified in at least 80 progeny from a cross between *Saccharomyces cerevisiae* strains BY4716 and RM11-1a. Rows correspond to the 447 proteins and columns correspond to the 107 progeny. Nearest-neighbor averaging was used to impute missing data. Yellow designates high and blue designates low abundance relative to the average abundance for the protein.

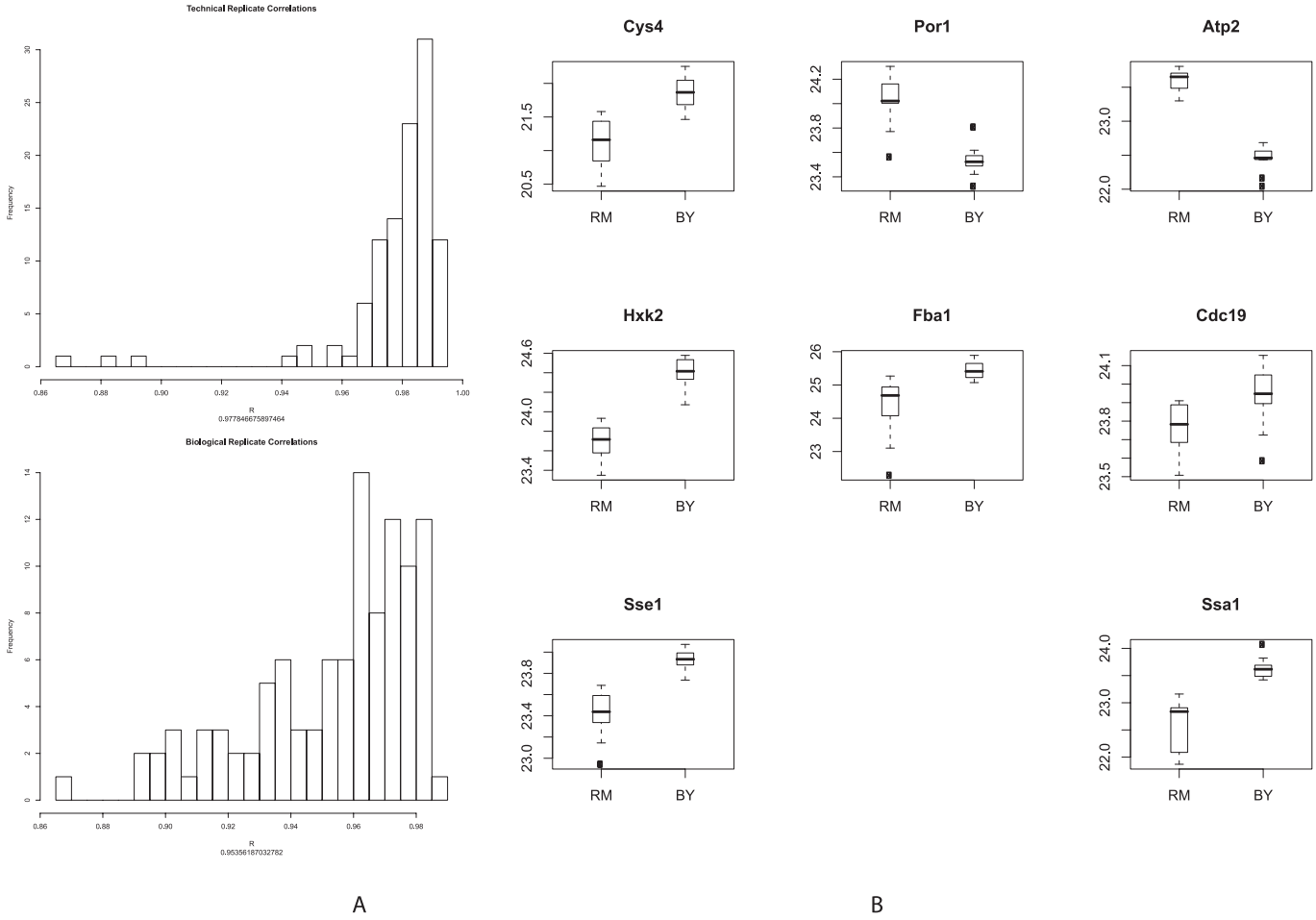

Fig. S2. (*A*) Distribution of Pearson correlations between protein abundance values measured between technical and biological replicates in the large Foss et al. dataset. (*B*) Box plots show the distribution of proteins assayed by Western blot analysis in the original Foss et al. study. GND1, which was present in the original study, was eliminated because we found that the measured tryptic fragment contained a polymorphism.

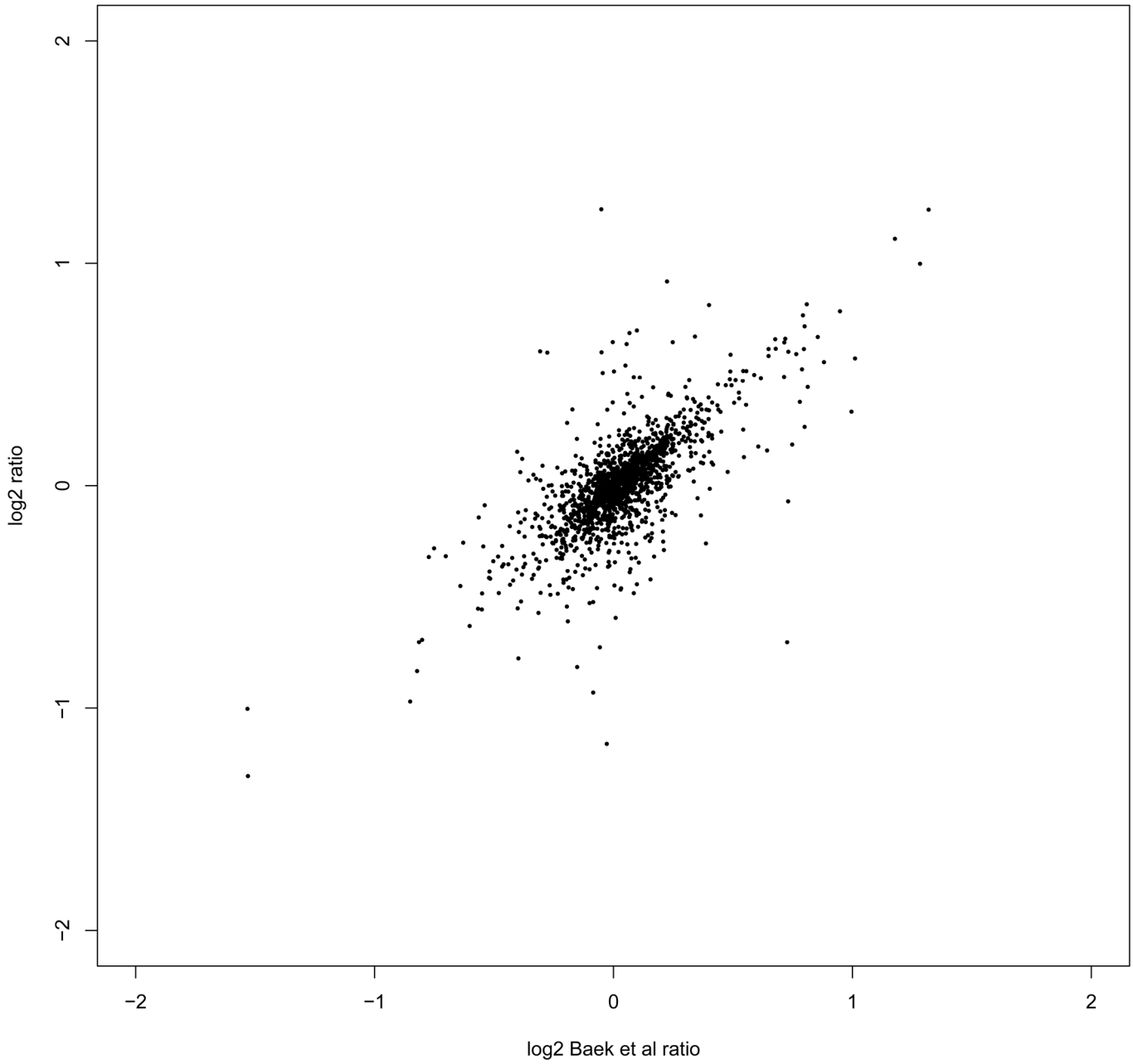

Fig. S3. Log_2-fold change measured by our algorithm and the algorithm used in the Baek et al. study. $r^2 = 0.69$ Spearman's correlation across 1,602 unique protein-coding genes.

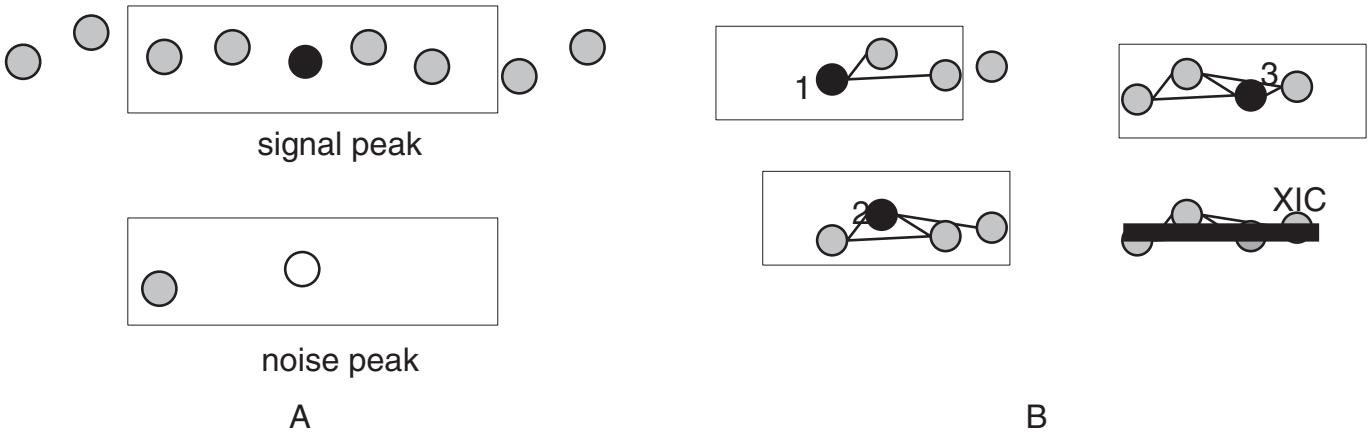

Fig. S4. (*A*) A planar orthogonal range query determines whether or not a peak is labeled as signal or as noise. A peak is labeled signal if the query returns a threshold number of peaks. (*B*) After filtering, another set of planar orthogonal range queries are used to connect signal peaks in an undirected graph. XICs correspond to connected components in this undirected graph.

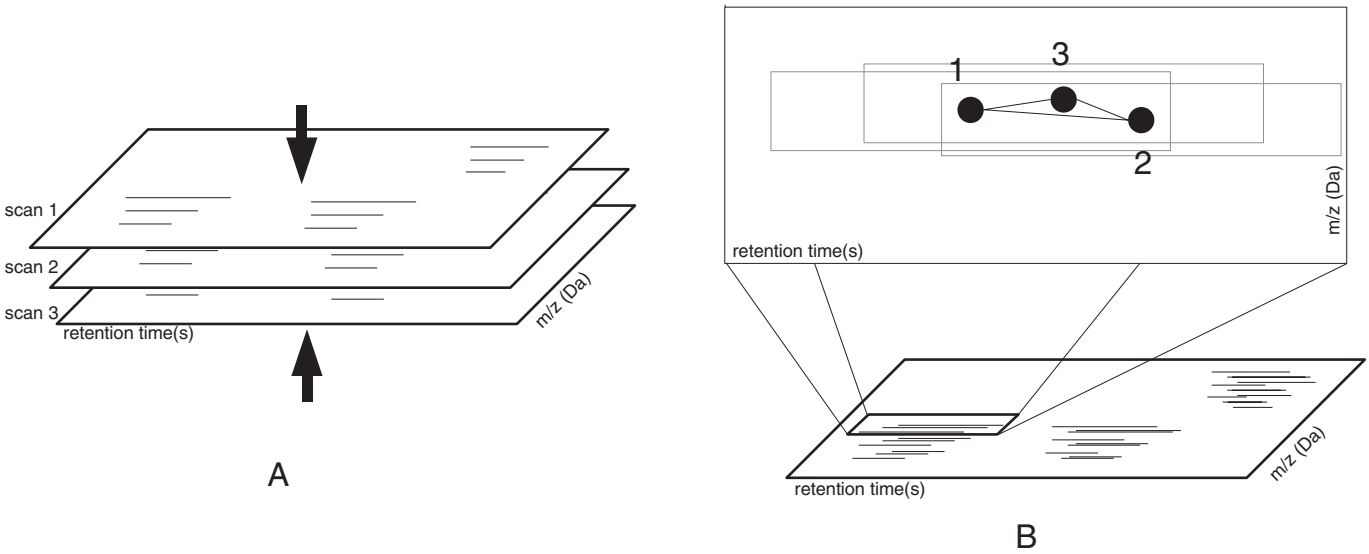

Fig. S5. (*A*) XICs are labeled according to the scan from which they originate and combined into a single scan. (*B*) Planar orthogonal range queries are used to connect labeled XIC centers in an undirected graph. The range queries compensate for any nonlinear differences in retention time between LC-MS scans. Connected components in the graph correspond to grouped XICs of the same tryptic peptide.

Table S1. Known femtomole amount of the spiked-in protein injected into each of six datasets collected in triplicate

	Protein injected (fmol) per sample					
Protein name	1	2	3	4	5	6
Myoglobin	800	25	50	100	200	400
Carbonic anhydrase	400	800	25	50	100	200
Cytochrome *c*	200	400	800	25	50	100
Lysozyme	100	200	400	800	25	50
Alchohol dehydrogenase	50	100	200	400	800	25
Aldolase A	25	50	100	200	400	800

This table is from the Mueller et al. study.