Supplemental Materials: MSAcquisitionSimulator: data-dependent acquisition simulator for LC-MS shotgun proteomics

I. DEFINITIONS

$\boldsymbol{e}$ = array of enzymes in the simulation
ion = an ion and all its attributes (sequence, charge, neutrons relative to the monoisotopic isotope, PTMs, etc)
ion_i = the residue at position i of the peptide represented by ion
N_{ion} = random variable for the index of the n-terminus of ion
C_{ion} = random variable for the index of the c-terminus of ion
Z_{ion} = random variable for the charge state of ion
IE_{ion} = random variable for the ionization efficiency of ion
M_{ion} = random variable for the number of neutrons greater than the monoisotopic form of ion
$\boldsymbol{PTM}_{ion}$ = array of the modification state for each residue on ion. This includes the state of no modification.
$\boldsymbol{P}_{ion}$ = array of proteins that can generate ion

II. DIGESTION

To model digestion, all enzymes specify a probability of cleavage given the residue present at the N-terminal side of the cleavage site, N, and the residue present at C-terminal side of cleavage site, C, and are assumed to be independent of each other. That is:

$$P(cleavage|N = n, C = c, e) \tag{1}$$

is given for all $e \in \boldsymbol{e}$. The probability of a peptide's cleavage from a single copy of its protein is then equal to the probability of cleavage at the peptide's N-terminus, C-terminus, and no cleavages in between, conditional on the enzymes in the simulation:

$$\begin{aligned} P(N_{ion} = n_{ion}, C_{ion} = c_{ion}|\boldsymbol{e}) = 1 - \prod_{e\in\mathbf{e}} 1 - P(cleavage|N = ion_{n_{ion}-1}, C = ion_{n_{ion}}, e)\times \\ 1 - \prod_{e\in\mathbf{e}} 1 - P(cleavage|N = ion_{c_{ion}}, C = ion_{c_{ion}+1}, e)\times \\ \prod_{i=n_{ion}}^{c_{ion}-1} \prod_{e\in\mathbf{e}} 1 - P(cleavage|N = ion_i, C = ion_{i+1}, e) \end{aligned} \tag{2}$$

III. MODIFICATIONS

Each modification has a user-defined probability of occupying a particular site (e.g. there are probably serines that are never phosphorylated), and a user-defined percentage of that residue that will be modified (e.g. if a particular serine *is* chosen to be phosphorylated, maybe only 1% of it ever exists in that form at any given time), given as $P(PTM)$. Candidate modification sites are first randomly assigned to each protein based on each PTM's probability of occupying a particular site. Afterwards, for each peptide created during the digestion process, modification combinations are created and their probability of existing is calculated as:

$$P(\boldsymbol{PTM}_{ion}) = \prod_{i=n_{ion}}^{c_{ion}} P(PTM_{ion_i}) \tag{3}$$

For the case where PTM_i is the state of no modification:

$$P(PTM_{ion_i} = \text{no mod}) = 1 - \sum_{\text{possible PTMs assigned to this residue}} P(PTM) \tag{4}$$

If the sum of probabilities for PTMs assigned to a particular residue is greater than 1, then their probabilites for that residue are normalized to sum to 1.

IV. IONIZATION EFFICIENCY

The ionization efficiency of an ion was randomly selected from a uniform random distribution. All ions of the same peptide have the same ionization efficiency.

$$\begin{aligned} P(ionization_{ion}) &= IE_{peptide} \\ IE_{peptide} &\sim uniform(0,1) \end{aligned} \tag{5}$$

V. CHARGE STATE DISTRIBUTION

The probability of an ion having a particular charge is modeled as a binomial distribution, with the binomial distribution's probability of success chosen randomly for each peptide in the simulation.

$$\begin{aligned} P(Z_{ion} = z_{ion}) &= binomial(n, z_{ion}, p) \\ n = 1 + \text{number of basic residues in pep} \\ p = 0.7 + r \\ r \sim uniform(0, 0.3) \end{aligned} \tag{6}$$

VI. ISOTOPIC DISTRIBUTION

The isotopic distribution for a molecule is computed using Mercury++ which uses a fast fourier transform to convolve the various element isotopes and their abundances (4). For the purposes of our model, we assume the value of $P(M_{ion} = m_{ion})$ is given.

VII. ION ABUNDANCE

We assume the probability of a particular ion's existence is based on all the previous equations, and that they are all independent of each other. Therefore the probability of an ion is given by:

$$P(ion|\boldsymbol{e}) = P(N_{ion} = n_{ion}, C_{ion} = c_{ion}|\boldsymbol{e}) \cdot P(\boldsymbol{PTM}_{ion}) \cdot P(ionization_{ion}) \cdot P(Z_{ion} = z_{ion}) \cdot P(M_{ion} = m_{ion}) \tag{7}$$

Furthermore, the number of copies of a particular ion A_{ion} is its probability of existence from a single protein, times the total protein abundance of proteins that can generate that ion $\boldsymbol{P}_{ion}$:

$$A_{ion} = P(ion|\boldsymbol{e}) \cdot \sum_{protein \in \boldsymbol{P}_{ion}} A_{protein} \tag{8}$$

VIII. RETENTION TIME

The retention time for each peptide is determined by the BioLCCC (Liquid Chromatography of Biomacromolecules at Critical Conditions) library (1). BioLCCC models the adsorption of polypeptide chains on porous media and can calculate the expected retention time for a particular molecule given the dimensions of column length and diameter, pore size, solvent concentrations, gradients, and flow rates. The effects of post-translational modifications can be modeled by specifying estimates of binding energy, which are user-specified parameters found in the ground truth configuration file.

IX. CHROMATOGRAPHIC PROFILES

The shape of an ion's chromatographic profile is modeled by an Exponential Gaussian Hybrid (EGH) function (2). The EGH takes into account the tailing typically observed at the end of a chromatographic profile. The two user-specified parameters of the EGH are elution width (σ) and the amount of tailing (τ) (S1). These chromatographic profiles are used to determine the number of ions reaching the ion detector by numerically integrating the profiles using Simpson's method at one-millisecond intervals for every ion present at the current time and m/z constraints. This integration continues until either a user-specified target total ion count is reached, or the maximum injection time is reached.

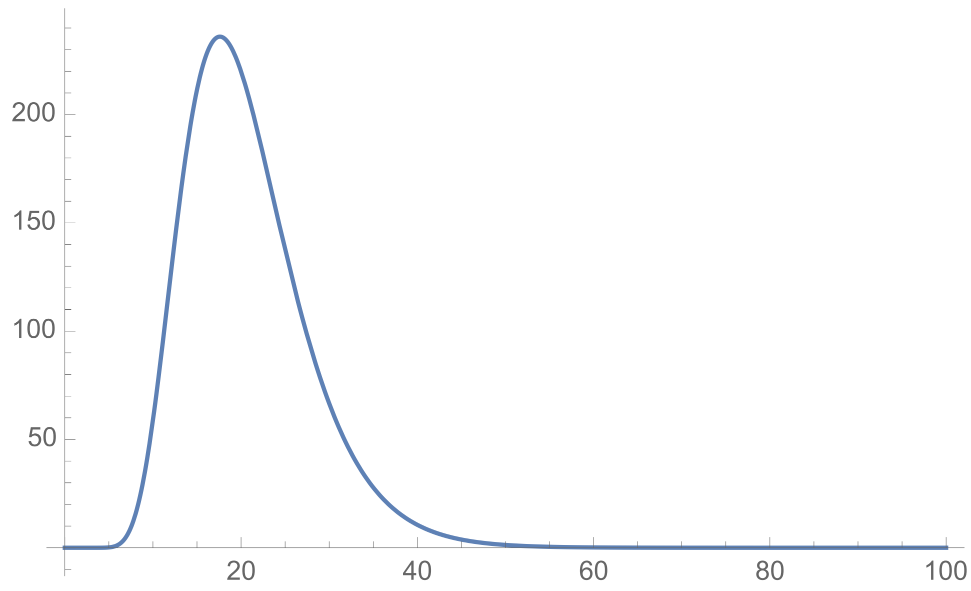


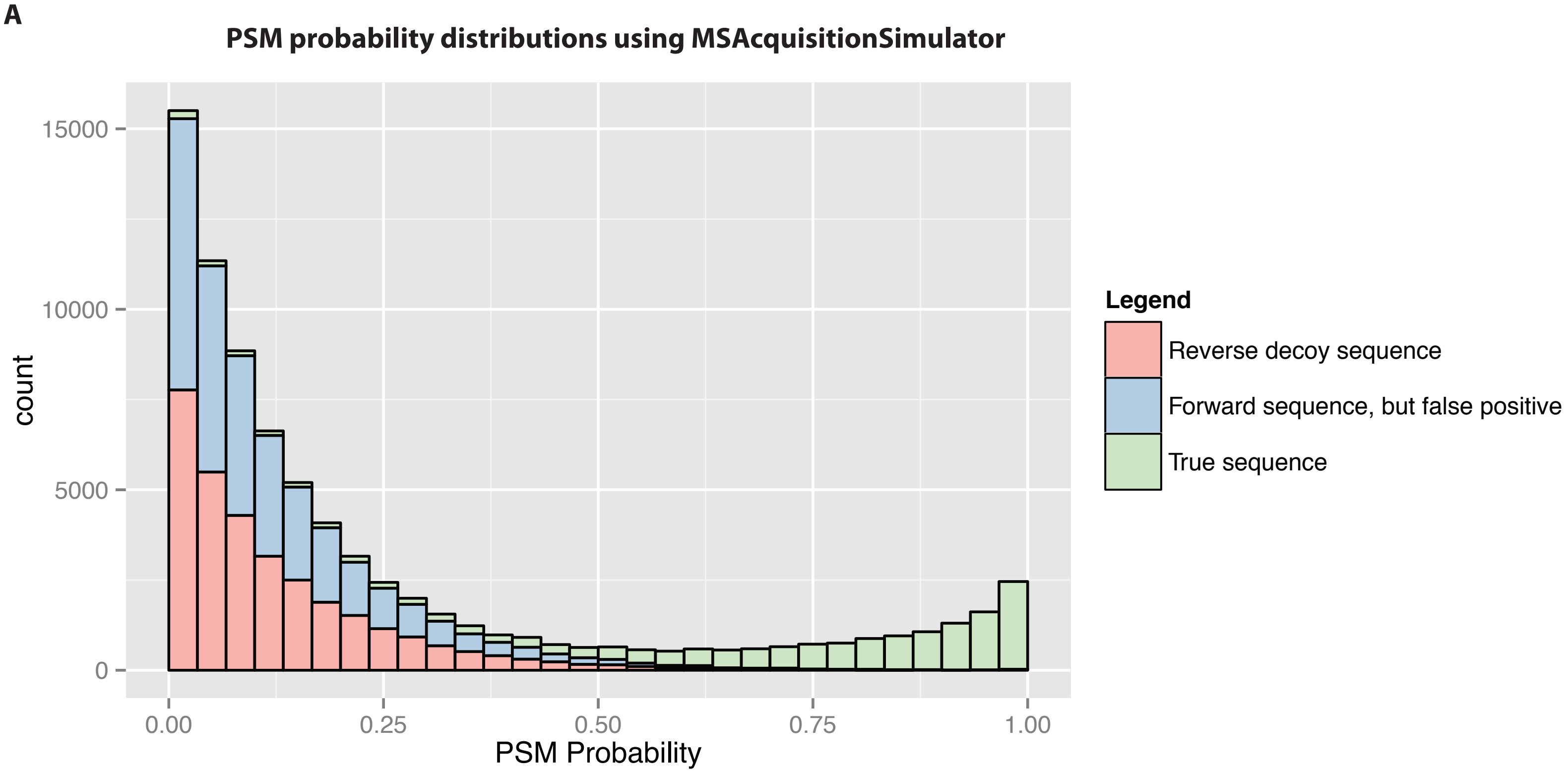
FIG. S1: Chromatographic profile with default parameters of $\sigma = 6$ and $\tau = 4$

X. PEPTIDE-SPECTRUM-MATCHES

To generate realistic PSMs, the precursor ion fraction (PIF) is first calculated for each peptide in an MS/MS scan. The PIF is defined as the sum of ion intensities for a given peptide (i.e., the sum of all its isotope intensities) divided by the total ion intensity of the scan. Next, one peptide is randomly selected from these peptides - weighted by their PIF. If the peptide is contained in the peptide database (including reverse decoy sequences) that is being used to mimic a database search, and within the user-defined precursor mass tolerance, then the PSM sequence is set to this peptide and the PSM probability is set to the corresponding PIF. If peptide is not in the peptide database, then a peptide is chosen in a uniform random fashion from the peptides in the database search within the mass tolerance of the targeted precursor m/z, and given a PSM probability sampled from a truncated exponential distribution. Psuedocode is presented below:

```
Data: $ions$ eluting at time $t$ within precursor $m/z \pm ms2IsolationWidth$.
      $candidateCharges$ for the targeted precursor peak.
Result: peptide-spectrum-match $PSM$ consisting of $PSM.peptide$ and $PSM.probability$
$peptide2PIF$ = map();
$totalIntensity = 0$;
foreach $ion \in ions$ do
    $peptide2PIF[ion.peptide]$ += $ion.intensity$;
    $totalIntensity$ += $ion.intensity$;
end
foreach $peptide \in peptide2PIF.keys()$ do
    $peptide2PIF[peptide]$ /= $totalIntensity$;
end
$randomPeptide$ = sampleWeightedRandom($peptide2PIF$);
$DB_{m/z}$ = getCandidatePeptidesFromSequenceDB($m/z$, $candidateCharges$, $precursorMassTol$);
// The peptide sequence database used to mimic a database search
// is typically a subset of the sequences used to generate the ground truth.
// The database search sequences sometimes even include sequences not in the ground truth.
if $randomPeptide \in DB_{m/z}$ then
    $PSM.peptide = randomPeptide$;
    $PSM.probability = peptide2PIF[randomPeptide]$;
else
    $PSM.peptide$ = sampleUniformRandom($DB_{m/z}$);
    $PSM.probability$ = sampleNullDistribution(); // exponential($\lambda = db\_search\_null\_lambda$)
end
return $PSM$;
```

A



B

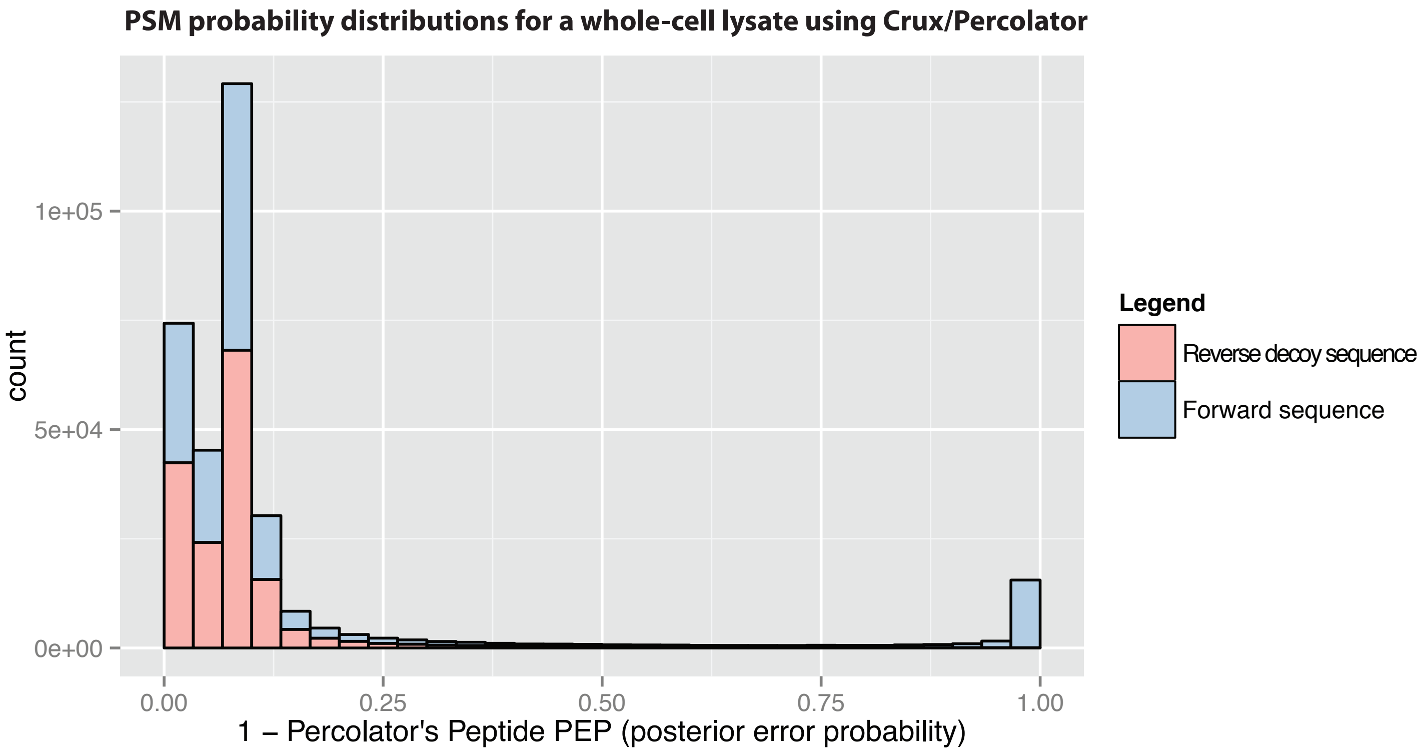


FIG. S2: **A**. Distribution of PSM probabilities from AcquisitionSimulator using TopN. Both false positive forward PSMs and reverse decoy PSMs are generated. False positive and decoy PSM probabilities are sampled from an exponential distribution with a user-specified lambda parameter. Lamba=8 was used for this simulation. Both peptide and protein-level false discovery rates can be estimated using the decoys. **B**. Peptide probability distributions from a real whole-cell lysate experiment using Crux and Percolator (3).

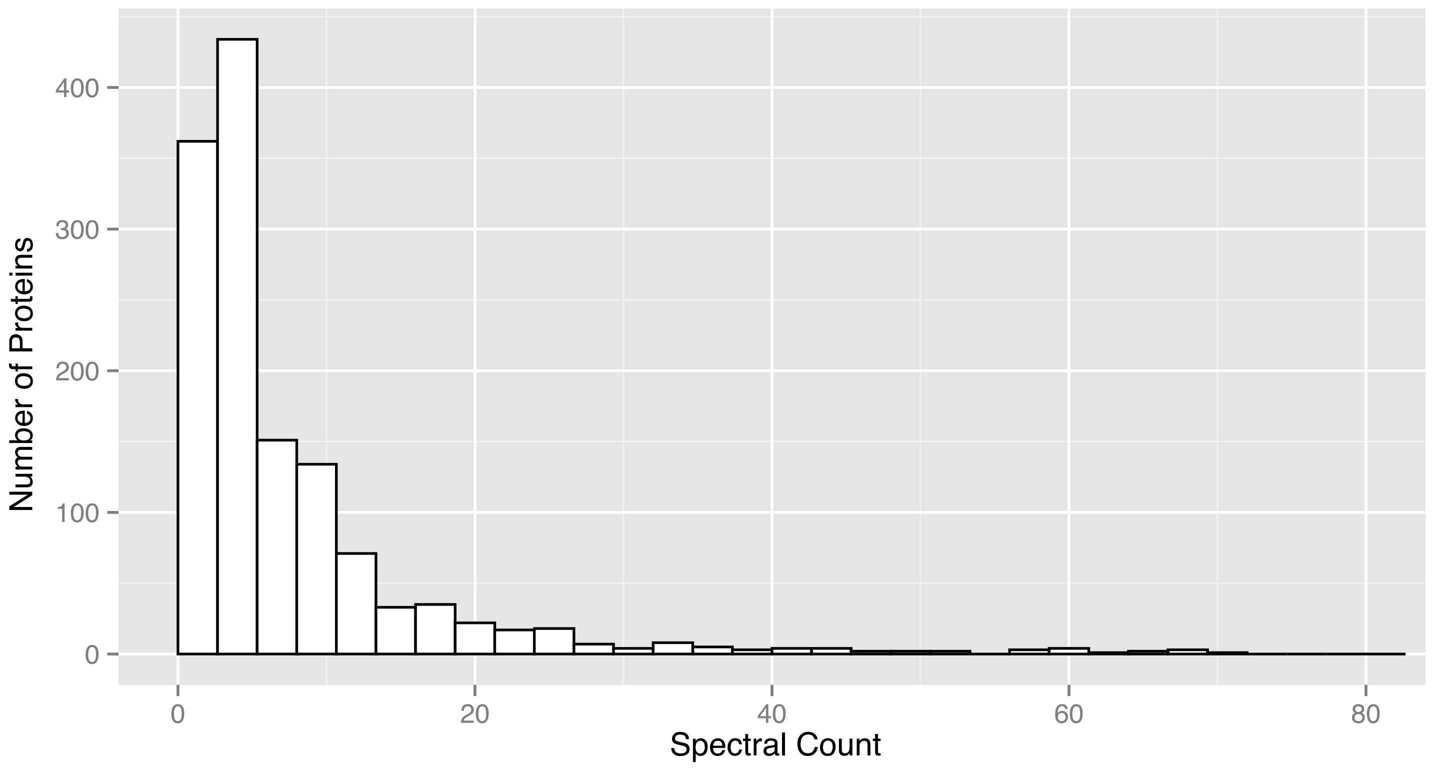


FIG. S3: Histogram of spectral counts per protein. Titin, the largest human protein with 34,350 amino acids was omitted for visual clarity. It was observed with 307 spectra. Only true positive proteins and PSMs identified at their respective 1% FDRs were included.

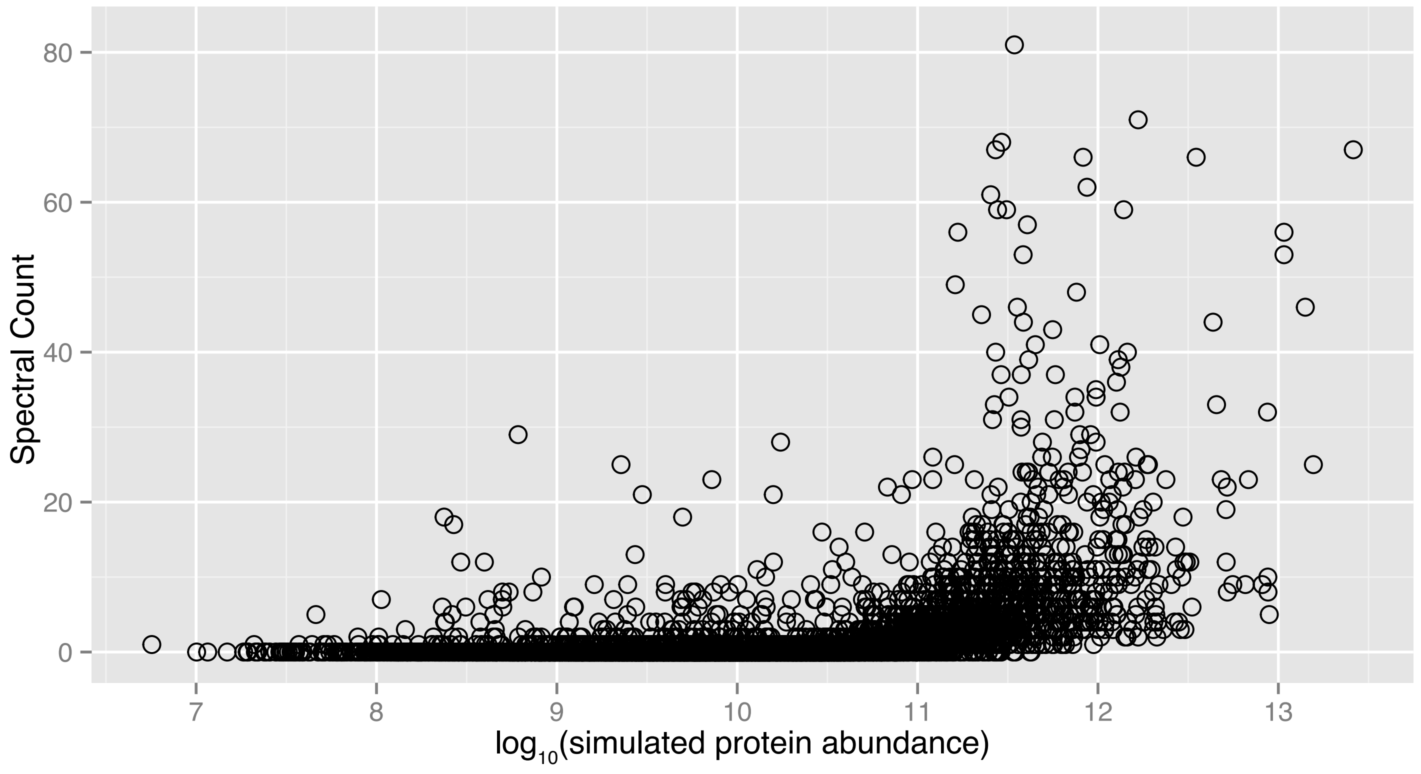
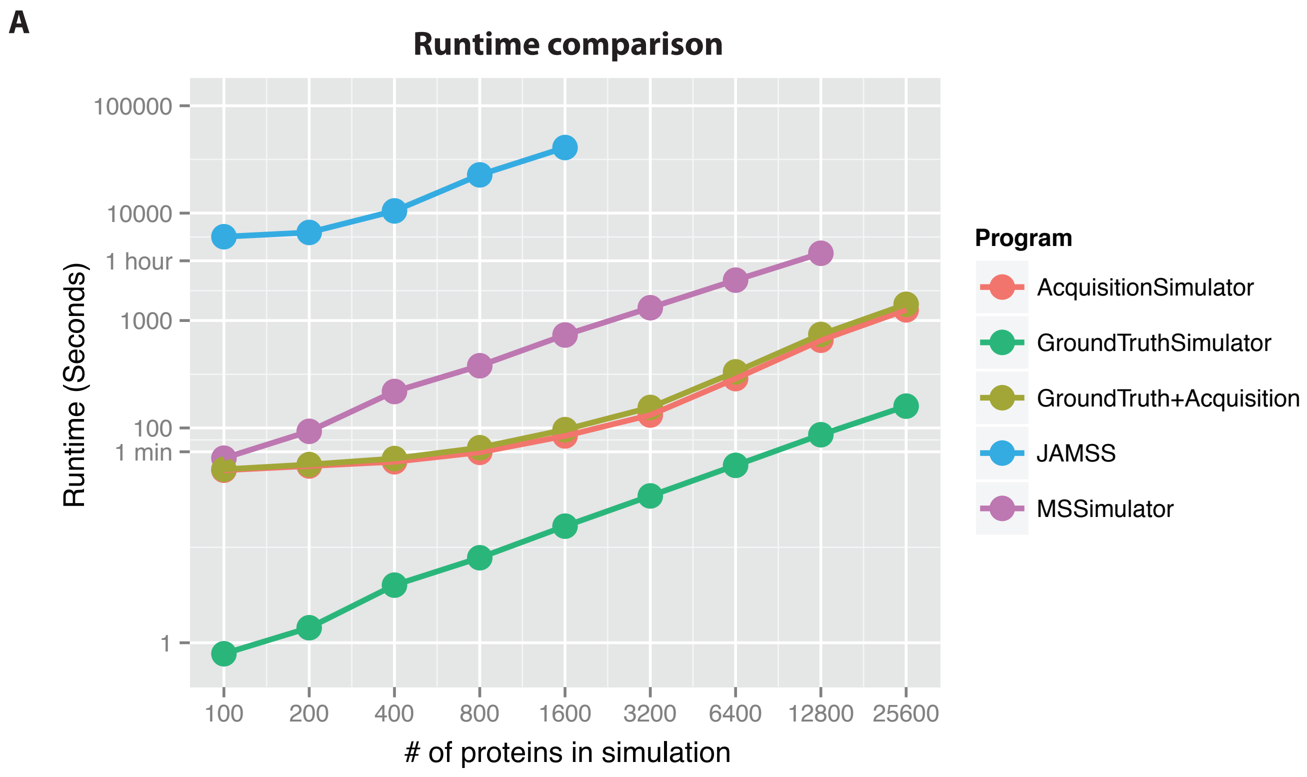


FIG. S4: Spectral counts vs simulated abundance plotted for simulated proteins. Titin, the largest human protein with 34,350 amino acids was omitted for visual clarity. It was observed with 307 spectra. Only true positive proteins and PSMs were included. Only the PSMs were required to be identified at a 1% FDR.



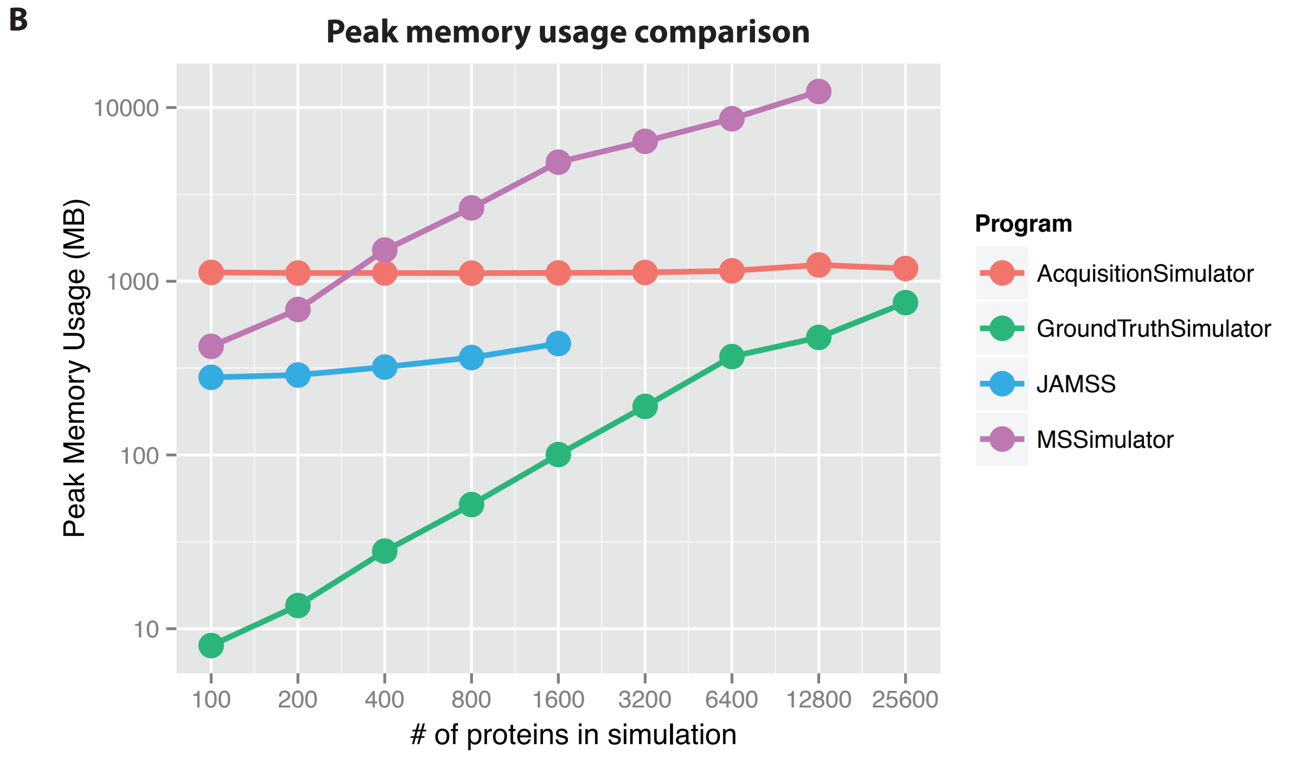


FIG. S5: Performance comparisons were performed on an Apple MacBook Pro with 16 GB 1600 MHz DDR3 RAM and two quad-core 2.8 GHz Intel Core i7s using the GNU time command. FASTASampler was used to generate increasingly larger datasets for simulation. In order to achieve reliable comparisons, similar parameters were used for each program, namely strict trypsin cleavage, no missed cleavages, no post-translational modifications, a 1 hour gradient, and 1 MS scan per second. JAMSS and MSSimulator generate MS/MS spectra in a very different fashion from MSAcquisitionSimulator, so this feature was turned off for JAMSS and MSSimulator, but still enabled for AcquisitionSimulator. JAMSS simulations were stopped after 1,600 proteins due to the long runtime, and MSSimulator was stopped after 12,800 proteins due to memory constraints.

[1] Gorshkov, A.V. *et al.* (2006) Liquid Chromatography at Critical Conditions: Comprehensive Approach to Sequence-Dependent Retention Time Prediction *Anal. Chem.*, **78**, 7770-7777.

[2] Lan, K. *et al.* (2001) A Hybrid of Exponential and Gaussian Functions as a Model of Asymmetric Chromatographic Peaks *Journal of Chromatography, Part A*, **915**, 1-13.

[3] Park, C. *et al.* (2008) Rapid and accurate peptide identification from tandem mass spectra *Journal of Proteome Research*, **7**, 3022-3027.

[4] Rockwood, A.L. *et al.* (2006) Efficent calculation of Accurate Masses of Isotopic Peaks *Journal of The American Society for Mass Spectrometry*, JASMS 03-2263, 2006.

XI. SIMULATION PARAMETERS

```
$ FASTASampler -dlognormal -m10 -s0.9 -p0.50 -o sampled_human_swissprot.fasta uniprot_homo_sapiens_proteome.fasta.gz
$ GroundTruthSimulator -c ../ground_truth.conf -o ground_truth_human.tab sampled_human_swissprot.fasta
$ AcquisitionSimulator -c ../acquisition_TopN.conf -f topN.fido ground_truth_human.tab
$ AcquisitionSimulator -c ../acquisition_WeightedN.conf -f weightedN.fido ground_truth_human.tab
$ AcquisitionSimulator -c ../acquisition_RandomN.conf -f randomN.fido ground_truth_human.tab
```

```
############################################################################
## GroundTruthSimulator parameter file
## Version 1.0.0, Release Date: Sept 20, 2015
## See README.txt for details
############################################################################
## All parameters are separated from their values by an equals sign ('=')
## '#' are comments and everything after them is ignored until the next line
############################################################################


############################################################################
## Modifications
## Each line represents a different modification
## -n = Full name of modifications
## -b = Abbreviation for modification. This will be used when writing the
##      sequence of a peptide, e.g. PEPpTIDE, AcarCDEFG
## -r = List of residues the modification can be on. "n" is for N-terminal
##      modifications and "c" is for C-terminal modifications
## -p = Probability that a residue will ever have this modification.
##      For example, -p1 would mean every residue defined by "-r" will have
##      at least some copies with this modification. -p0.5 will result in a
##      random 50% of residues never having this modification.
## -a = Relative abundance of the modification on a particular residue. For
##      those residues randomly chosen to have this modification, how many
##      copies will exist with it? For example, -a1 would mean that if the
##      residue was selected to have this modification, then all copies of
##      this residue will have the modification. -a0.9999 would result in
##      0.0001 percent of the copies to not have the modification. -p1 -a1
##      would create a classic "static" modification. Low aboundance PTMs
##      such as ubiquitination should have a low value for this parameter
## -m = Molecular formula for this modification. Format is the standard
##      abbreviation used for the element followed by the number of the
##      element. Each element and count combination is separated by a comma.
##      Negative numbers are allowed.
## -e = Bind energy of the modification. This affects the retention time of
##      peptides having this modification. Retention time is predicting with
##      the BioLCCC library. Please refer to their documentation for how to
##      set this: http://pythonhosted.org/pyteomics.biolccc/
##      Phosphorylation bind energy was taken from their document. Depending
##      on the residue, the bind energy is different, which is why the
##      modification is split into 3 separate lines. If you don't know what
##      the value of this parameter should be, try a guess! Just don't make
##      it too big. Positive numbers increase retention time. 0 Has no effect.
## --blocks_cleavage = This PTM will inhibit digestion.
##                     Excluding this parameter means the PTM does not block
##                     cleavage
##                     ** NOT IMPLEMENTED YET **
## --stackable       = This PTM can stack with other PTMs.
##                     Excluding this parameter means the PTM does not stack
##                     ** NOT IMPLEMENTED YET **
## --post_digestion  = This PTM should be added after in silico digestion
##                     Excluding this parameter means the PTM is present pre
##                     digestion
##                     ** NOT IMPLEMENTED YET **
############################################################################
PTM= -nCarbamidomethyl  -bcar -rC  -p1 -a0.9999 -mH3,C2,N1,O1 -e0.77
#PTM= -nPhosphorylationS -bp -rS -p0.1 -a0.01 -mH1,O3,P1 -e-0.45
#PTM= -nPhosphorylationT -bp -rT -p0.1 -a0.01 -mH1,O3,P1 -e-0.74
```

```
#PTM= -nPhosphorylationY -bp -rY -p0.1 -a0.01 -mH1,O3,P1 -e-1.32
#PTM= -nProteinAcetylation -bAc--rn -p0.5 -a0.1 -mC2,H2,O1 -e0.0
#PTM= -nAcetylation  -bac -rK -p0.01 -a0.002 -mC2,H2,O1 -e0.1 --blocks_cleavage
#PTM= -nGlyGly   -bubi -rK -p0.05 -a0.001 -mH6,C4,N2,O2 -e0.05 --blocks_cleavage
#PTM= -nDeamidation  -bd -rNQ -p0.2 -a0.01 -mH-1,N-1,O1 -e-0.1
#PTM= -nOxidation -box -rM -p0.5 -a0.05 -mO1 -e1.8215
#PTM= -nLys6 -bl6 -rK -p1 -a0.99 -mn6 -e0 --stackable
#PTM= -nPropionyl -bprop -rK -p1 -a0.99 -mH4,C3,O1 -e4.0 --post_digestion

###############################################################################
## Enzymes
## Each line represents a different enzyme
## -n = Name of the enzyme
## -r = List of residues this enzyme should cleaved at. '*' is a wild card
##      and means all residues can be cut. Useful to model non-specific
##      cleavage
## -p = Relative frequency of cleavage at this residue. -p1 would result in
##      no missed cleavages.
## -t = Terminus of cleavage. 'C' = C-terminus, 'N' = N-terminus
## -b = Residues that block cleavage. ** NOT IMPLEMENTED YET **
###############################################################################
Enzyme= -nTrypsinR -rR -p0.99 -tC -bP
Enzyme= -nTrypsinK -rK -p0.95 -tC -bP
Enzyme= -nRandom -r* -p0.001 -tC

###############################################################################
## BioLCCC parameters for liquid chromatography retention time prediction
## Please refer to their documentation for details:
## http://pythonhosted.org/pyteomics.biolccc/
###############################################################################
column_length= 150  # length in mm
column_diameter= .075 # diameter in mm
column_pore_size= 100  # pore size in angstroms

second_solvent_concentration_a= 2.0 # Percentage * 100
second_solvent_concentration_b= 80.0 # Percentage * 100

gradient_percent_b_start= 0.0 # Percentage * 100
gradient_percent_b_end= 50.0 # Percentage * 100
gradient_duration= 185.0  # Time in minutes
gradient_flow_rate= 0.00025  # ml/min

###############################################################################
## Elution profile shape parameters
## A exponential-Gaussian hybrid function (EGH) is used to model the shape
## Please refer to "A hybrid of exponential and gaussian functions as a
## simple model of asymmetric chromatographic peaks" by Kevin Lan and James
## W. Jorgenson, Journal of Chromatography, Part A, 915, 1-13 (2001).
###############################################################################
elution_sigma=6  # width of elution profile = ~6 x elution_sigma seconds
elution_tau=4 # influences amount of tailing, 0 = gaussian shape

###############################################################################
## Miscellaneous
###############################################################################
max_mass= 10000 # amu
max_mz= 3000 #
min_mz= 300 #
```

```
prune_threshold= 1e4 # The minimum abundance for any ion to be
# included in the simulation. it is used for
# filtering purposes. Decreasing this will
# increase run-time and file sizes. If you used
# the recommended abundances for proteins then
# this is probably a good number. Ions below
# this threshold are very unlikely to be
# observed in a typical MS1 scan.
```

```
############################################################################
## AcquisitionSimulator parameter file
## Version 1.0.0, Release Date: Sept 20, 2015
## See README.txt for details
############################################################################
## All parameters are separated from their values by an equals sign ('=')
## '#' are comments and everything after them is ignored until the next line
############################################################################


############################################################################
## Acquisition Algorithm
## acquisition_algorithm=  The name of the acquisition algorithm to use.
##  Built-in options are TopN, RandomN, StochasticN.
## TopN targets the most abundant ions. RandomN
## targets them completely at random. StochasticN
## randomly choose peaks weighted by their
## abundance.
## Please consult the documentation for instructions
## to create custom algorithms.
## acquisition_algorithm_params=  A string of parameters for the chosen
## algorithm. The built-in methods share
## the same parameters.
## --num_ms2= # of MS2 scans per MS1
## --ms1_min_mz= lower limit of MS1 range
## --ms1_max_mz= upper limit of MS1 range
## --ms2_isolation_width= +- m/z on each
## side for MS2
## --dynamic_exclusion_enabled=1 for true
## 0 for false
## --dynamic_exclusion_tolerance= +- mz
## on each side
## --dynamic_exclusion_time= seconds on
##   exclusion list
############################################################################
acquisition_algorithm= TopN
acquisition_algorithm_params= --num_ms2 10 --ms1_min_mz 300 --ms1_max_mz 2000 --ms2_isolation_width 1 --dyn

############################################################################
## Elution profile shape parameters
## These should be identical to those used with the GroundTruthSimulator
## A exponential-Gaussian hybrid function (EGH) is used to model the shape
## Please refer to "A hybrid of exponential and gaussian functions as a
## simple model of asymmetric chromatographic peaks" by Kevin Lan and James
## W. Jorgenson, Journal of Chromatography, Part A, 915, 1-13 (2001).
############################################################################
elution_sigma=6  # width of elution profile = ~6 x elution_sigma seconds
elution_tau=4 # influences amount of tailing, 0 = gaussian shape

############################################################################
## Database search parameters
############################################################################
fasta= /Users/dennisg/Research/fasta/human_sp_022015_raw.fasta # path
db_search_min_mass=300
db_search_max_mass=9000
db_search_max_missed_cleavages=0
db_search_min_enzymtyic_termini=2
db_search_max_dynamic_mods=0
```

```
db_search_mass_tolerance=.02 # amu on each side
############################################################################
## Modifications
## Each line represents a different modification
## -n = Full name of modifications
## -b = Abbreviation for modification. This will be used when writing the
##      sequence of a peptide, e.g. PEPpTIDE, AcarCDEFG.
## This MUST match the abbreviation used for the GroundTruthSimulator
## -r = List of residues the modification can be on. "n" is for N-terminal
##      modifications and "c" is for C-terminal modifications
## -m = Molecular formula for this modification. Format is the standard
##      abbreviation used for the element followed by the number of the
##      element. Each element and count combination is separated by a comma.
##      Negative numbers are allowed.
## -s = 1=static 0=dynamic
############################################################################
db_search_PTM= -nCarbamidomethyl  -bcar -rC -mH3,C2,N1,O1 -s1
############################################################################
## Enzymes
## Each line represents a different enzyme
## -n = Name of the enzyme
## -r = List of residues this enzyme should cleaved at. '*' is a wild card
##      and means all residues can be cut. Useful to model non-specific
##      cleavage
## -t = Terminus of cleavage. 'C' = C-terminus, 'N' = N-terminus
## -b = Residues that block cleavage. ** NOT IMPLEMENTED YET **
############################################################################
db_search_enzyme= -nTrypsin -rRK -tC -bP
db_search_null_lambda=8 # When an MS2 scan is assigned a
# peptide-spectrum-match (PSM) that does not
# match a peptide that was present in the scan,
# and is therefore an incorrect match, the PSM
# probability is sampled from an exponential
# distribution with this specified lambda value.
# This models the "null" distribution of PSMs

############################################################################
## Miscellaneous
############################################################################
acquisition_length= 10800  # seconds

############################################################################
## Instrument parameters
## Note: The timing model for the simulator reflects an QExactive instrument
## The parameters correspond to transient times and fill times.
############################################################################
max_ms1_injection_time= 0.2  # Seconds. The maximum amount of time ions
# will be accumulated for an MS1 in order
# to hit the target total ion count.
max_ms2_injection_time= 0.5  # Seconds. The maximum amount of time ions
# will be accumulated for an MS1 in order
# to hit the target total ion count.
ms1_target_total_ion_count= 1e10
# The maximum number of ions accumulated for
# an MS1. This number should not match the
# parameter used on an actual instrument.
# The reason it is set high by default is
# because the simulator does not model the
# transmission rate of ions. It assumes 100%
# transmission. This number should be adjusted
# if the abundance distribution for the
# FastaSampler was changed.
ms2_target_total_ion_count= 5e8
# The maximum number of ions accumulated for
# an MS2. This number should not match the
# parameter used on an actual instrument.
# The reason it is set high by default is
# because the simulator does not model the
# transmission rate of ions. It assumes 100%
# transmission. This number should be adjusted
# if the abundance distribution for the
# FastaSampler was changed.
resolution= 60000  # Currently only a constant resolution is
# supported
dynamic_range= 5000 # The minimum peak intensity for a scan is
# equal to the most intense peak (base peak)
# divided by the dynamic_range. All other
# peaks are filtered out.
ms1_scan_time= 0.256  # Seconds. The amount of time for an MS1
# scan to complete. This is the transient
# time for an Orbitrap instrument.
ms2_scan_time= 0.064  # Seconds. The amount of time for an MS2
# scan to complete. This is the transient
# time for an Orbitrap instrument.
scan_overhead_time= 0.015  # Seconds. The total time for each scan is
# max(scan_time, injection_time) +
# scan_overhead_time
```