Supplementary Table 1

	Dataset Name	Num. Spectra	Num. Clusters	Database Size		Dataset Name	Num. Spectra	Num. Clusters	Database Size
1	*A. dehalogenans*	120.5 K	56.3 K	1.5 M	67	Monkeypox virus[3]	3.8 M	2.6 M	0.2 M
2	*A. fumigatus*	21.0 K	18.4 K	4.5 M	68	*N. Crassa*	185.0 K	134.0 K	5.5 M
3	*A. mirum*	752.6 K	608.3 K	2.2 M	69	*N. dassonvillei*	674.6 K	527.6 K	1.8 M
4	*A. nidulans*	583.6 K	175.7 K	4.9 M	70	*N. multipartita*	481.9 K	386.3 K	1.7 M
5	*A. niger*	6.8 M	4.3 M	6.2 M	71	*O. bacterium TAV2*	781.2 K	442.9 K	1.5 M
6	*A. phagocytophilium*	757.4 K	571.3 K	0.3 M	72	*O. cuniculus*	145.2 K	118.6 K	8.0 M
7	*A. robiniae*	793.6 K	679.1 K	-	73	*P. aeruginosa*	103.7 K	98.1 K	1.9 M
8	*A. thaliana*	943.4 K	597.9 K	19.7 M	74	*P. carbinolicus*	170.8 K	141.6 K	1.1 M
9	*A. variabilis*	1.4 M	1.0 M	1.9 M	75	*P. chrysosporium*	526.7 K	383.9 K	83.6 K
10	Arthrobacter	2.2 M	1.1 M	1.5 M	76	*P. falciparum*	2.0 M	1.2 M	7.1 M
11	BATS[1]	1.3 M	1.1 M	1.8 M	77	*P. fluorescens*	2.4 M	1.7 M	6.6 M
12	*B. Taurus*	3.4 M	2.3 M	15.9 M	78	*P. minatonensis*	545.6 K	450.8 K	1.0 M
13	*B. anthracis Sterne*	1.2 M	0.9 M	1.5 M	79	*P. placenta*	16.2 K	14.8 K	3.9 M
14	*B. burgdorferi*	828.2 K	541.6 K	0.4 M	80	*P. promelas*	163.3 K	105.8 K	29.2 K
15	*B. faecium*	671.1 K	553.9 K	1.1 M	81	*P. ubique*	1.7 M	1.2 M	0.9 M
16	*B. mallei*	1.4 M	0.9 M	15.1 M	82	Periphyton[4]	2.2 M	1.4 M	-
17	*C. aurantiacus*	406.8 K	344.6 K	1.4 M	83	*P. trichocarpa*	3.7 M	2.1 M	15.5 M
18	*C. crescentus*	8.2 M	4.3 M	1.2 M	84	*Prochlorococcus*	777.4 K	565.2 K	-
19	*C. curtum*	400.7 K	355.2 K	0.5 M	85	*R. capsulatus*	937.8 K	658.9 K	0.2 M
20	*C. elegans*	413.7 K	300.8 K	19.3 M	86	*R. castenholzii*	278.3 K	247.6 K	1.6 M
21	*C. flavigena*	1.1 M	0.8 M	1.2 M	87	*R. norvegicus*	1.5 M	1.0 M	17.1 M
22	*C. griseus*	292.0 K	242.5 K	0.2 M	88	*R. palustris*	3.0 M	2.2 M	1.6 M
23	*C. symbiosum*	459.0 K	410.1 K	0.6 M	89	*R. pickettii*	1.3 M	0.9 M	1.5 M
24	*C. synechocystis*	1.1 M	0.6 M	1.1 M	90	*R. sphaeroides*	11.0 M	5.9 M	5.5 M
25	*C. tepidum WT*	171.1 K	146.5 K	0.6 M	91	*S. Typhi*	3.1 M	1.5 M	1.4 M
26	*C. thermocellum*	24.3 K	22.3 K	1.0 M	92	*S. amazonensis*	1.0 M	0.6 M	1.3 M
27	*C. thermophilum*	243.3 K	205.3 K	0.8 M	93	*S. baltica OS155*	5.4 M	2.3 M	1.4 M
28	Cyanothece[2]	8.5 M	4.7 M	1.6 M	94	*S. baltica OS185*	1.5 M	1.1 M	1.5 M
29	Cyanothece PCC7424	396.8 K	335.2 K	1.6 M	95	*S. baltica OS195*	542.4 K	372.8 K	1.5 M
30	Cyanothece PCC7425	432.4 K	348.0 K	1.6 M	96	*S. baltica OS223*	478.8 K	394.4 K	1.5 M
31	Cyanothece PCC7822	61.5 K	55.4 K	1.6 M	97	*S. cerevisiae*	5.9 M	4.1 M	3.3 M
32	Cyanothece PCC8801	216.0 K	176.2 K	1.6 M	98	*S. denitrificans OS217*	1.0 M	0.6 M	1.3 M
33	*D. desulfuricans*	3.2 M	1.8 M	0.8 M	99	*S. frigidmarina* NCIMB 400	587.7 K	368.2 K	1.4 M

Table 1 (part I) Information about distribution of spectra among species represented in the PNNL datasets. For each species we note the number of spectra (after quality filtration), the number of clusters that contained these spectra, and the number of amino acids in the corresponding protein database (if available). About a quarter of the data consisted of spectra from quality control runs, or experiments for which we could not link spectra to a specific species; these were grouped as the Miscellaneous (bottom of the table). [1] BATS is a seawater microbial community from the Bermuda Atlantic; [2] mixture of 6 cyanothece strains (the database size refers to *Cyanothece sp. ATCC51142*); [3] the Human monkeypox virus sample contains both virus proteins and human proteins; [4] mixture of algae, cyanobacteria, heterotrophic microbes, and detritus;

Table 1 describes the PNNL dataset used in our experiments. We collected $\approx$ 1.18 billion MS/MS spectra from over 100 organisms (referred to as the PNNL dataset). This data set was compiled by pooling the ion trap spectra that have been generated at the Richard Smith laboratory at PNNL in 2001-2009. The data was mostly generated on LCQ, LTQ, LTQ-FT and LTQ-Orbitrap instruments. A complete list of all datasets (originating from 134 species) is given in Table 1. The PNNL dataset contains 130.54 million human spectra (after spectral quality filtration) organized into 56.11 million clusters. The Table also provides the size of the corresponding proteome (if available). We remark that spectra from a single peptide may generate multiple clusters (see [6]) and that clusters containing spectra from several organisms contribute to multiple rows in the Table.

	Dataset Name	Num. Spectra	Num. Clusters	Database Size		Dataset Name	Num. Spectra	Num. Clusters	Database Size
34	*D. melanogaster*	688.0 K	409.3 K	12.4 M	100	*S. fumaroxidans*	540.5 K	443.1 K	1.3 M
35	*D. peptidovorans*	482.2 K	411.2 K	0.8 M	101	*S. heliotrinireducens*	654.2 K	550.1 K	0.9 M
36	*D. proteobacterium NaphS2*	1.1 M	0.6 M	-	102	*S. keddieii*	452.9 K	364.6 K	-
37	*D. radiodurans*	5.3 M	3.3 M	0.9 M	103	*S. nassauensis*	801.3 K	615.6 K	2.0 M
38	*D. vulgaris*	1.8 M	1.1 M	1.1 M	104	*S. oneidensis*	43.2 M	22.2 M	1.4 M
39	*E. caballus*	17.8 K	15.7 K	12.7 M	105	*S. putrefaciens 200*	562.5 K	362.2 K	1.3 M
40	*E. chaffeensis*	802.9 K	627.8 K	0.3 M	106	*S. putrefaciens CN-32*	664.2 K	413.8 K	1.3 M
41	*E. coli*	16.3 M	8.6 M	1.3 M	107	*S. thermophile*	851.1 K	677.3 K	-
42	*E. coli BL21*	1.0 M	0.7 M	1.3 M	108	*S. trabarsenatis ANA-3*	504.3 K	347.0 K	1.5 M
43	*F. graminearum*	95.9 K	66.5 K	0.3 M	109	*S. typhimurium*	13.8 M	7.0 M	1.4 M
44	*G. gallus*	18.9 K	15.0 K	10.8 M	110	*S. viridis*	264.8 K	234.8 K	1.2 M
45	*G. max*	3.6 M	2.6 M	- [1]	111	Sea Sediments[3]	806.0 K	583.6 K	-
46	*G. metallireducens*	1.4 M	1.0 M	1.2 M	112	*Shewanella MR-4*	539.5 K	356.5 K	1.3 M
47	*G. sulfurreducens*	6.1 M	3.0 M	1.1 M	113	*Shewanella MR-7*	525.0 K	357.5 K	1.4 M
48	*G. uraniumreducens*	1.6 M	1.2 M	1.4 M	114	*Shewanella PV-4*	666.3 K	410.1 K	1.3 M
49	*H. borinquense*	323.9 K	264.1 K	1.1 M	115	*Shewanella W3-18-1*	689.8 K	420.2 K	1.3 M
50	*H. modesticaldum*	312.6 K	255.5 K	0.9 M	116	Shewanella spp[4]	7.9 M	4.8 M	-
51	*H. sapiens*	122.8 M	64.1 M	34.2 M	117	*Synechococcus*[5]	1.1 M	0.8 M	1.0 M
52	*H. utahensis*	456.4 K	359.5 K	0.9 M	118	*T. bispora*	340.3 K	292.8 K	-
53	Human Cytomegalovirus	348.6 K	254.8 K	66.2 K	119	*T. elongatus*	8.5 K	8.1 K	0.8 M
54	*I. scapularis*	92.1 K	86.2 K	5.9 M	120	*T. pallidum*	31.0 K	29.1 K	0.3 M
55	*J. gansuensis*	670.3 K	609.7 K	4.0 M	121	*T. pseudonana*	49.1 K	44.8 K	5.9 M
56	*K. radiotolerans SRS30216*	2.1 M	1.1 M	1.5 M	122	*T. reesei*	6.0 M	3.0 M	77.0 K
57	Leaf Cutter Ant[2]	1.0 M	0.7 M	-	123	*T. terrestris*	640.4 K	497.3 K	-
58	*M. barkeri*	581.9 K	427.5 K	1.1 M	124	Termite Comm.[6]	677.6 K	504.3 K	-
59	*M. grisea*	41.4 K	30.1 K	6.1 M	125	Vaccinia virus	1.8 M	1.4 M	0.8 M
60	*M. hungatei*	382.0 K	297.8 K	1.0 M	126	*X. cellulosilytica*	1.8 M	1.3 M	1.2 M
61	*M. magneticum*	24.1 K	22.6 K	1.5 M	127	*Y. enterocolitica*	584.1 K	374.8 K	1.5 M
62	*M. musculus*	57.2 M	29.9 M	25.6 M	128	*Y. pestis*	1.3 M	1.0 M	1.2 M
63	*M. musculus B16*	115.4 K	105.2 K	25.6 M	129	*Y. pestis CO92*	260.7 K	106.3 K	1.2 M
64	*M. musculus cortical neuron*	14.8 K	14.1 K	25.6 M	130	*Y. pseudotuberculosis*	1.3 M	0.8 M	1.3 M
65	Macaque	2.9 M	2.1 M	17.5 M	131	Miscellaneous	155.4 M	80.6 M	-
66	Microbial Communities	218.9 K	181.8 K	-					

Table 1: (part II) Information about distribution of spectra among species represented in the PNNL datasets. For each species we note the number of spectra (after quality filtration), the number of clusters that contained these spectra, and the number of amino acids in the corresponding protein database (if available). About a quarter of the data consisted of spectra from quality control runs, or experiments for which we could not link spectra to a specific species; these were grouped as the Miscellaneous (bottom of the table). [1] The proteome of *G. max* (soybean) is not yet available; [2] microbial community from leaf cutter ants; [3] seawater microbial community; [4] microbial community of 7 unsequenced *Shewanella* species; [5] mixture of two cyanobacteria (*Synechococcus_sp_PCC7002* and *Synechococcus_sp_CC9605*); [6] termite hindgut microbial community.

Supplementary Table 2

Cluster size	Num. spectra	(%)	Total clusters	Number of k-clusters				
				$k=1$	$k=2$	$k=3$	$k=4$	$k=5^+$
1	277,602,847	47.75%	277,602,847	277,602,847	0	0	0	0
2	17,622,936	3.03%	8,811,468	7,737,404	1,074,064	0	0	0
3	9,071,289	1.56%	3,023,763	2,454,196	466,225	103,342	0	0
4	6,742,092	1.16%	1,685,523	1,304,746	272,307	89,122	19,348	0
5	5,471,735	0.94%	1,094,347	812,204	183,272	70,444	23,882	4,545
6	4,830,618	0.83%	805,103	574,355	140,380	57,329	24,034	9,005
7	4,323,172	0.74%	617,596	424,876	111,022	47,040	22,389	12,269
8	3,940,240	0.68%	492,530	328,909	90,048	39,291	19,823	14,459
9	3,127,068	0.54%	347,452	227,651	64,649	28,319	14,437	12,396
10	3,071,200	0.53%	307,120	195,040	58,264	26,032	13,908	13,876
11-15	13,331,219	2.29%	1,045,276	620,121	203,528	96,009	53,349	72,269
16-20	10,004,260	1.72%	560,189	298,993	113,622	56,581	32,509	58,484
21-30	19,567,013	3.37%	780,185	356,144	161,911	87,196	51,521	123,413
31-40	16,953,805	2.92%	482,297	187,312	99,543	57,215	35,564	102,663
41-50	14,445,101	2.48%	319,597	108,296	65,754	39,408	24,581	81,558
51-100	49,289,878	8.48%	701,979	181,478	137,699	90,644	59,656	232,502
101-200	47,775,465	8.22%	346,890	51,614	56,733	45,015	32,474	161,054
201-500	41,965,665	7.22%	144,107	10,264	14,971	15,727	13,052	90,093
500+	32,184,809	5.54%	32,192	934	1,105	1,672	1,823	26,658
Total	581,320,412	100.00	299,200,461	293,477,384	3,315,097	950,386	442,350	1,015,244

Table 2: Clustering of the PNNL dataset. The set of ≈ 1.18 billion spectra was first reduced to 581 million spectra that passed quality filtering and grouped into ≈ 299 million clusters. The table holds the number of spectra in clusters of different sizes and also lists how many of those clusters included spectra from multiple organisms. A k-cluster is defined as a cluster with spectra from exactly k organisms (similarly, a k^+-cluster includes spectra from at least k organisms).

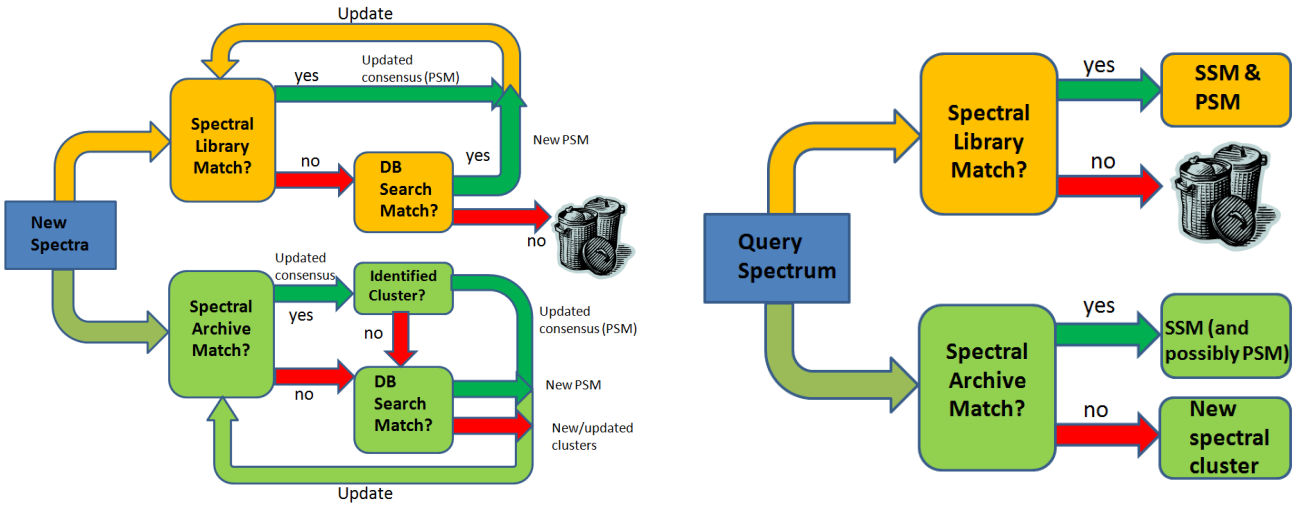

Figure 1: Comparison between the construction (left) and use (right) of spectral libraries and spectral archives. Spectral archives utilize all spectra either as identified or unidentified clusters while spectral libraries discard unidentified spectra. New spectra that are searched against an archive can result in either a PSM (when matched against an identified cluster) or an SSM (when matched with an unidentified cluster). Unmatched spectra in an archive search are used to created new spectral clusters. Since spectral libraries only utilize PSMs, the majority of the spectra remain unassigned.

Supplementary Note 1 - Spectral Archives Complement Spectral Libraries

Most mass spectrometry studies attempt to identify Peptide-Spectrum Matches (PSMs) and often ignore Spectrum-Spectrum Matches (SSMs) if PSMs for these SSMs are not established. We argue that SSMs are also useful (even if the corresponding peptide is not identified) since they allow to cross-reference spectra generated by different researchers and to query all spectra ever generated against a single repository.

Spectral libraries are essentially databases of PSMs while spectral archives are databases of both PSMs and SSMs. While construction of PSMs (via MS/MS database search) is a well-studied topic, construction of *all* SSMs represents a formidable clustering problem. Fig. 1 reveals similarities and highlights the differences between construction (Fig. 1 left) and use (Fig. 1 right) of spectral libraries and spectral archives.

With an archive we first cluster, then search the clusters against a protein database to generate Peptide-Cluster Matches (PCMs). These PCMs in turn get propagated to all spectra in the identified clusters to generate PSMs. With the library, we first search the spectra against a protein database to generate PSMs, group PSMs corresponding to the same peptide, and finally deposit the *curated* consensus PSM in the spectral library. The spectral library can then be used to identify spectra from new spectral datasets (but cannot identify new peptides).

To illustrate the similarities between the two approaches, we analyzed a dataset from the human HEK293 kidney cell line [13] with ≈ 0.75 million spectra. A traditional database search of this dataset with InsPecT [14] against the IPI human database at 1% FDR identified 96536 spectra from 20828 peptides and 5332 proteins. The spectral archive of this dataset includes ≈ 0.34 million clusters. We searched the consensus spectra of these clusters against the same protein database and identified 115330 spectra from 20801 peptides and 5343 proteins with the same 1% FDR ($\approx 8\%$ of these peptides were missed by the searches of individual spectra). We further restricted the database to 5343 identified proteins (akin to the two-stage search performed by X!Tandem [2]) and identified 122703 spectra from 22204 peptides in the resulting (smaller) database with 1% FDR.

To illustrate how spectral libraries contribute to peptide identifications, we constructed a *non-curated*

spectral library consisting of the consensus spectra of PSMs identified in the traditional database search and searched all spectra against this spectral library. This procedure increased the number of identified spectra from 96536 to 115743 with a maximal p-value of 0.01 (see below for more details on p-value assignment). We note that while the number of identifications is quite similar in the spectral archive and the spectral library approaches, the *amortized* time required to create the archive is small compared to the total time required for a database search. Building the archive took ≈ 0.03 seconds per spectrum, faster than the typical runtime for an MS/MS search against a large database.

We emphasize that 1% FDR computed via Target-Decoy Approach (TDA) and 0.01 p-value represent different (and not equivalent) ways to select statistically significant matches. Since it remains unclear what represents an analog of a decoy database for spectral libraries, we used a recently proposed Target-Decoy Library Approach (TDLA) approach [8] to better compare the results of the database search with the spectral library search. We used a large library from an evolutionary distant species than the one being searched (e.g., insects versus mammals) to serve as a decoy library and removed from this library any PSMs representing peptides that were common to both species. Similarly to TDA, after searching the spectral dataset against target and decoy libraries, the results are sorted by scores and FDR is computed by taking note of ratio between the number of decoy and target hits at each score level.

Searching the human kidney dataset using the NIST human library [11] (with $\approx$ 300000 spectra) led to the identification of 64021 spectra from 16290 peptides and 4270 proteins (results were filtered to 1% FDR using the TDLA approach with spectral library of *D. melanogaster* as a decoy and adjusting for the difference in sizes). The peptides identified by the search of the NIST spectral library included 2045 peptides that were not identified by the regular database search ($\approx$ 10% of the peptides identified by regular search). As expected, a significant portion of peptides from the kidney dataset is missed by spectral libraries (since they remain incomplete), a reason why library searches are typically followed by regular database searches. On the other hand, library search represents a valuable addition to the regular database searches since they generate additional peptide identifications.

Supplementary Note 2 - Reducing Running Time and Memory Requirements

Our clustering algorithm follows the design of the algorithm in [6] but uses various heuristics to reduce its running time (currently, 10^6 similarity computations per second on a 3.2Ghz desktop PC) and memory requirements (enabling clustering of two orders of magnitude more spectra than the the previous algorithm). Note that the pre-processing rate of the data is $\approx$ 2000 spectra/second (drops to 700 spectra/second if quality filtering is also performed). We employ various heuristics aimed at reducing the number of similarity computations performed by our clustering algorithm.

The first heuristic evaluates how likely it is that two spectra belong to the same peptide, without explicitly computing the similarity between them. For example, spectra from the same peptides have similar sets of strong peaks: in our data, 96% of the pairs of spectra from the same peptide had at least one peak in common in their respective sets of the four strongest peaks. However, only 3.5% of the pairs of spectra from different peptides match one or more of their top four peaks. Thus we are able to forgo most of the unnecessary similarity computations. To further reduce the running time we organized the spectra in lists according to the masses of their top four peaks (each spectrum appears in four lists that correspond to its four top peaks' masses). Thus in practice we do not have to evaluate all pairs of spectra to check if the sets of their top four peaks intersect, rather we confine the similarity computations to pairs of spectra appearing in the same list.

The second heuristic we use relies on the fact that our algorithm uses multiple rounds of cluster joining with decreasing similarity thresholds. Using this approach we approximate a hierarchical-clustering algorithm. Instead of recomputing the similarity between pairs of consensus spectra at each round, we carry over similarity results from one round to the next by taking note of the top 25 most similar clusters that matched each spectrum (it is unlikely that the most similar consensus for a given spectrum was not one of the top 25 in a previous round). Thus in subsequent rounds we do not compute the similarities between all pairs of spectra, but rather only a much smaller fixed number of computations is involved.

In our clustering algorithm we also made several design decisions aimed at reducing the memory requirements. First we removed all quadratic-space elements that were part of the previous version (e.g., a bit vector that indicated if pairs of spectra should be compared or not). In addition, to reduce memory fragmentation and the overhead cost of memory reallocation, all data structures are centrally allocated in beginning of the run (space for peaks, consensus spectra, etc.) and reused by the program as different spectra batches get processed without explicitly freeing the memory until termination. In addition, due to the way the binary data is stored, the program can easily load to memory only the spectra that need to be evaluated (these spectra all fall in within a specific range of precursor m/z units). Once a spectrum's precursor m/z is below the m/z units being processed, the memory it used (spectrum structure and peaks) is immediately made available for the next batch of spectra.

Figure 2 displays the running time of the algorithm (without data preprocessing) as a function of the number of spectra (left) and the number of similarity computations (right). There are two types of operations that need to be performed: Input/output tasks such as reading spectra, creating consensus spectra for clusters, and writing the results to the output. The time required to perform these actions tends to be linear in the input size (the number of spectra). The other type of operations are the ones involved in the clustering of the data which are mainly computing the similarity between pairs of spectra. The number of pairs that need to be compared typically grows quadratically with the input size. The plot on the left shows the quadratic nature of the running time. From it we see that up to an input size of $\sim$ 1.5 million spectra the linear-time operations take longer than the quadratic time, but for larger datasets the quadratic-time operations begin to dominate (as expected). This quadratic growth of the clustering time may become a limiting factor in the future when even larger datasets (e.g., a trillion of spectra) are clustered.

The main design consideration when approaching this problem was to minimize the running time. Hence we developed a fast greedy approach which at times might yield "suboptimal" clustering (e.g., a partition that does not minimize the squared distance from clusters to centroids). Unlike most clustering algorithms, the order in which the spectra are evaluated can slightly change the composition of the clusters

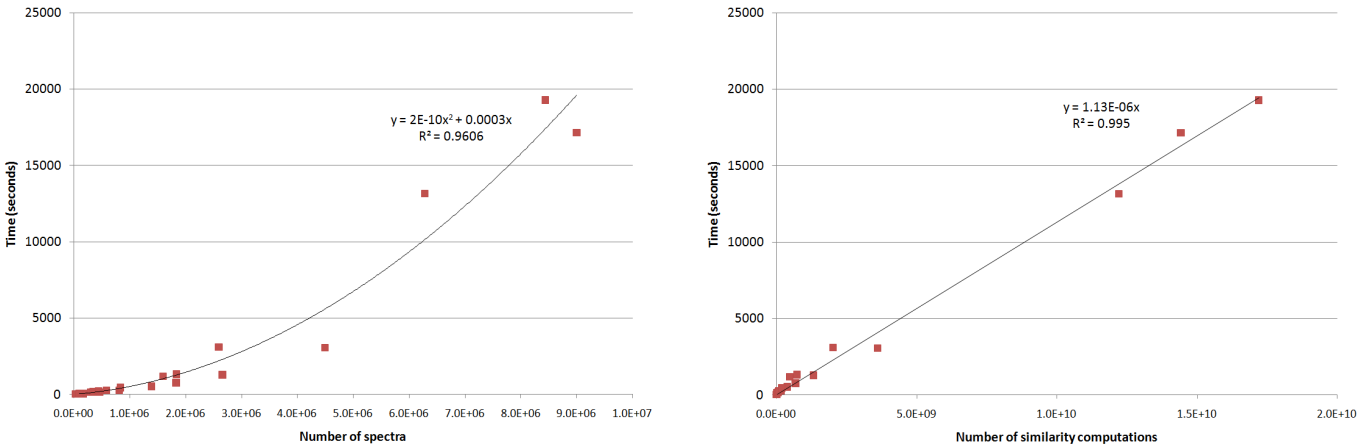

Figure 2: The running time of the MSCluster for datasets of various sizes (excluding time required to preprocess the data). The graphs show the running time in seconds vs. (i) the number of spectra in the dataset (left), and (ii) the number of similarity computations (right). The running time is dominated by the number of similarity computations and is approximated as quadratic in the number of spectra.

(the algorithm greedily joins pairs of consensus spectra whose similarity exceeds a threshold, these are not always the most similar pairs of spectra). In addition, MS-Cluster can only add spectra to a cluster but it never splits a cluster into sub-clusters. Splitting clusters is not practical when constructing large-scale archives since the peak information of individual spectra is not stored (only peak information for consensus spectra is kept). We believe the resulting gain in speed outweighs the slight decrease in clustering quality, and is a worthwhile sacrifice for gaining the capability to cluster billions of spectra. Furthermore, as we argued previously [6], since in the domain of MS/MS, the spectra display significant variance which leads to a natural fragmentation of clusters of spectra of the same peptide, there is little practical value in trying to obtain an "optimal" clustering of the data.

Supplementary Note 3 - Selecting Spectral Similarity Thresholds and Computing P-Values

We employ the widely-used dot-product [9, 12] for fast computation of the similarity between spectra. To speed up the computations we consider only the top 15-40 peaks in each spectrum (which does not compromise the quality of clustering), depending on the maximal m/z of peaks in the spectrum. The mass lists are selected in such a way that every pair of masses are at least ε apart (ε represents the accuracy of the mass measurements). See supplemental information for more details on the similarity computation.

Figure 3 (top left) shows the probability distributions of similarity values between spectra in a cluster and their consensus, randomly selected pairs of spectra of the same peptides (identified by InsPecT at 2% FDR) and of pairs of spectra from different peptides. Pairs of spectra of different peptides were selected to have a precursor m/z of at least 8 m/z units apart to avoid comparing spectra of the same peptide that got erroneously assigned. As expected, pairs of spectra of the same peptide display a much greater similarity to each other compared to pairs of spectra from different peptides. In addition, the similarity between spectra in a cluster and their consensus is, on average, much higher, than the similarity between an arbitrary pair of spectra of the same peptide (that are not necessarily from the same cluster). This difference has two main causes. First, for spectra from the same cluster, the similarity to the consensus is typically greater than the similarity between cluster members, since the consensus represents the "center" of the cluster. Second, spectra from the same peptide may end up in different clusters (Table 1 in the paper illustrates the difference between number of identified peptides and number of identified clusters). These multiple clusters typically have smaller "radius" compared to the case when all spectra from the same peptide end up in the same cluster.

The similarity threshold t that is used to determine if a spectrum S should be joined into a cluster depends on several parameters: N - the total number of spectra that S was compared with while clustering; p - the mixture probability we are willing to tolerate for joining spectra that are generated from different peptides; and the empirical cumulative probability function $CDF(t)$ which measures the proportion of random pairs of spectra from different peptides with a similarity value that is less than or equal to t ($CDF(t)$ is depicted as the dash line in Figure 3, top right).

Assuming that the spectra being compared are independent of each other, we can select for a spectrum a minimal similarity threshold t that satisfies $[CDF(t)]^N > 1 - p$, in order to guarantee that the overall proportion of cases where spectra are joined erroneously will be less than p. In a similar fashion, $CDF(t)$ can be used as the null hypothesis distribution to assign p-values to the similarities observed between pairs of spectra during clustering, which in turn can be used to decide if spectra should be joined or not. For example, the event where the best match to spectrum S, after being compared to N spectra, showed a similarity value t can be assigned $p-value = 1-[CDF(t)]^N$. As N increases, for instance, when clustering larger datasets, the similarity threshold t also needs to increase in order to maintain the same clustering quality. Figure 3 (bottom left) depicts this phenomenon. The precision/recall curves in the figure are generated from experiments in which the similarity between a spectrum S and another random spectrum of the same peptide was compared to the maximum similarity between S and N randomly selected spectra of other peptides. S is considered correctly matched only if its similarity to the other spectrum of the same peptide is larger than the maximal similarity between it and the N spectra from other peptides. By considering different minimal similarity thresholds that are needed in order to join pairs of spectra we can create precision/recall curves. The recall is the proportion of instances in which the spectrum S and the other spectrum from the same peptide had a similarity exceeding the threshold, and the precision is the proportion of those cases in which the pair of spectra of the same peptide had a similarity that was higher than the maximal similarity observed between S and the N spectra from other peptides. The curves in the figure show that for low values of N it is not likely to mismatch the spectrum S, however as N increases, we are more likely to observe cases where spectra of other peptides show significant similarity to our spectrum, so fewer pairs of spectra can be confidently joined. For example, at a fixed precision rate of 0.95, approximately 95% of the pairs of spectra can be joined if S is compared to $N = 1$ additional spectrum. This proportion drops to 60% and 21% when N is increased to 1,000 and 1,000,000, respectively.

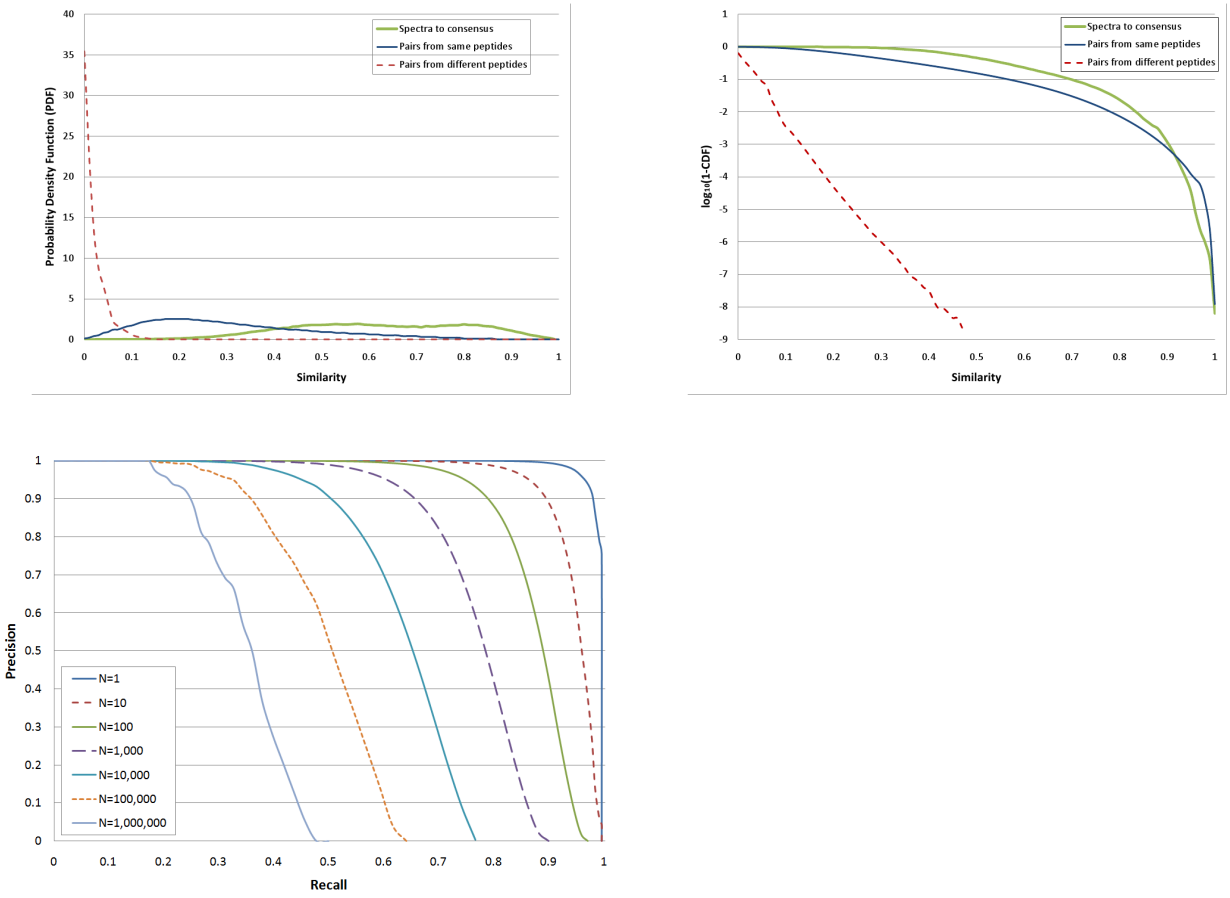

Figure 3: Probability density functions (top left) and cumulative distribution functions (top right) for similarity between spectra in clusters and their consensus, pairs of spectra of the same peptide (but not necessarily from the same cluster) and pairs of spectra from different peptides. The cumulative distribution functions use a logarithmic scale and plot $\log_{10}(1 - CDF)$ to display the difference in the values of the functions more clearly. The dash line which corresponds to the CDF of pairs from different peptides is used as the null hypothesis for computing p-values for similarity scores. Each plot was generated using over 10^8 pairs of spectra.The plot on the bottom left depicts precision/recall curves generated in experiments where the similarity between a spectrum S and another random spectrum of the same peptide was compared to the maximum similarity between S and N randomly selected spectra of other peptides (with a similar precursor mass). S is considered correctly matched only if its similarity to the other spectrum of the same peptide is larger than the maximal similarity between it and the N spectra. By considering different minimal similarity thresholds that are needed in order to join pairs of spectra we can create precision/recall curves. The recall is the proportion of instances in which the spectrum S and the other spectrum from the same peptide had a similarity exceeding the threshold, and the precision is the proportion of those cases in which the pair of spectra of the same peptide had a similarity that was higher than the maximal similarity observed between S and the N spectra from other peptides.

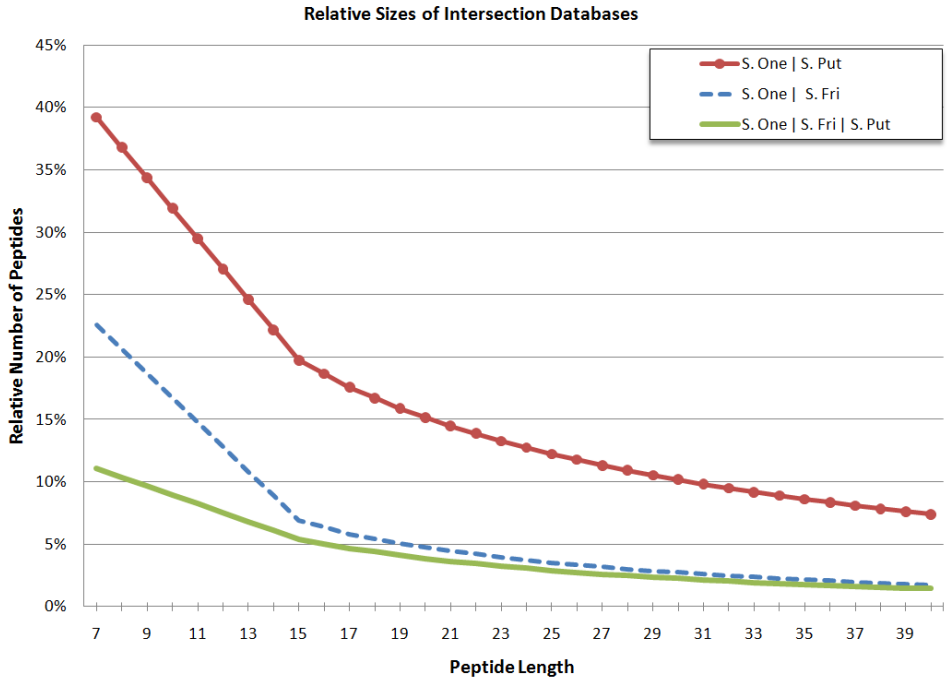

Figure 4: Intersection of protein sequence databases. The figure shows the level of conservation of peptides of different lengths when intersecting the six-frame translations of the genomes of *Shewanella oneidensis* (*Sone*), *Shewanella frigidimarina* (*Sfri*) and *Shewanella putrefaciens* (*Sput*) (for the sake of simplicity we substitutes intersection proteome by the intersection of six-frame translations). The x-axis denotes peptide length, and the y-axis shows the proportion of peptides that are present in the intersection database compared to the number of peptides in the database of *Shewanella oneidensis*.

Supplementary Note 4 - Searching intersection proteomes

Since proteomes of related species may share many peptides, clusters resulting from multiple species can be searched against smaller *intersection proteomes*, thus increasing the number of peptide identifications for a given FDR. When spectra are entered into a spectral archive, we keep track of the organism from which they were produced, which allows us to detect clusters of spectra that originate from multiple organisms. If a spectrum belongs to a peptide that appears in multiple proteomes, we can search it against the significantly smaller *intersection proteome*, which is defined as the set of all peptides that belong to all these proteomes.

Figure 4 shows the relative sizes of the intersection proteomes compared to the size of the database of the *Sone* proteome. Shorter peptides have a much higher chance of being in the intersection proteome than longer peptides (the longer a peptide, the higher the probability that it incurs a mutation in a diverged species). For 7 amino acids long peptides we see that 39% of the 7-mers are common to the database of *Sone* and *Sput*, 23% of 7-mers are common to *Sone* and *Sfri* and 11% are common to the intersection of all three. These rates further reduce as the peptide length in the intersection database increases.

Cluster size	Num. spectra	(%)	Total clusters	Num. 1-clusters	Num. 2-clusters	Num. 3-clusters	Num. 4-clusters	Num. 5^+-clusters
1	3,894,311	19.89%	3894311	3894311	-	-	-	-
2	1,066,320	5.45%	533160	366679	166481	-	-	-
3	701,622	3.58%	233874	119097	80241	34536	-	-
4	491,296	2.51%	122824	53189	36297	25209	8129	-
5	367,830	1.88%	73566	28251	18682	15420	8835	2378
6	299,718	1.53%	49953	18117	10874	9881	7094	3987
7	252,700	1.29%	36100	12466	6992	6500	5355	4787
8	218,608	1.12%	27326	9248	4901	4311	3895	4971
9	180,945	0.92%	20105	6684	3266	2842	2742	4571
10	163,330	0.83%	16333	5316	2527	2131	2093	4266
11-15	654,280	3.34%	51483	16283	7313	5603	5583	16701
16-20	487,344	2.49%	27326	8102	3557	2377	2127	11163
21-30	1,020,942	5.21%	40588	10300	4951	2934	2583	19820
31-40	939,100	4.80%	26753	5431	2818	1481	1348	15675
41-50	810,589	4.14%	17938	3387	1746	870	761	11174
51-100	2,470,047	12.61%	35678	5856	3033	1604	1269	23916
101-200	2,109,678	10.77%	15439	2173	1182	620	425	11039
201-500	1,861,874	9.51%	6367	690	307	206	130	5034
500+	1,590,460	8.12%	1497	149	37	26	16	1269
Total	19,580,994	100.00%	5,230,621	4,565,729	355,205	116,551	52,385	140,751

Table 3: Clustering of spectra of short peptides in PNNL dataset. We selected ≈ 5 million clusters (with 19 million spectra) from the PNNL dataset that had a predicted precursor mass of less than 850 m/z units. The table holds the number of spectra in clusters of different sizes and also lists how many of those clusters included spectra from multiple organisms. A k-cluster is defined as a cluster with spectra from exactly k organisms (similarly, a k^+-cluster includes spectra from at least k organisms).

Supplementary Note 5 - Using spectral archives to identify short peptides

MS/MS database search algorithms analyze all peptides in a proteome, usually a much smaller computational space than the set of all possible peptides. This enables the Target-Decoy Approach (TDA) to evaluate the statistical significance of the results [3, 7]. However, the TDA paradigm does not apply to short peptides in peptidomics studies (e.g., many neuropeptides are shorter than 7 aa) short peptides are present in *both* target and decoy protein databases. Thus, a database search in this case is not unlike a de novo interpretation (since it needs to consider all possible peptides), which typically achieves lower accuracy and does not provide a possibility to evaluate FDR (because score distributions in target and decoy databases are nearly identical).

To interpret spectra of short peptides we need to design new algorithms and train new scoring models, tailored for the unique characteristics of the fragmentation of short peptides. However, to do this we need large training data, which is usually obtained via a database search, a catch-22. Though generating spectra for a large number of synthetic peptides is a way out of this deadlock, this approach is costly.

In this section we demonstrate how spectral archives can be utilized to bootstrap generation of a training set of spectra of short peptides and provide false discovery rates for the identifications. We started off by selecting a subset of the PNNL data set which held all the clusters with precursor mass below 850 m/z units (typically corresponding to peptides of length 7 amino acids and shorter). This subset contained over 5 million clusters (See Table 3 above). For each cluster we also took note of the organisms that contributed spectra to it. Following that we ran the PepNovo [4, 5] de novo sequencing algorithm on each cluster and retained the highest scoring peptide (we use de novo sequencing since its scoring is less sensitive to the fact that the peptides are short, compared to a database search). We retained clusters whose de novo sequenced peptide was between 4 and 6 amino acids in length, leaving us with 718,437 clusters. At this

stage we run into a problem since we cannot tell which of the generated de novo peptide reconstructions are correct. To provide false discovery rates, we needed to come up with a new variant of the target-decoy approach that generates individual target/ decoy databases for each spectral cluster (in contrast with the standard TDA approach that generates one target and one decoy database).

The first step in our TDA analysis, is to partition the de novo results according to the length and charge. De novo solutions with the same length and charge are then sorted according to decreasing PepNovo scores and evaluated sequentially. To be able to compute FDRs for the results, we keep track of the sum of the weights of instances that hit the target (W_t) and decoy (W_d), as defined below. Initially $W_t = W_d = 0$. For each de novo result (a peptide P of length l and a cluster consensus spectrum S), we perform the following:

- Generate the target database $Target$ according to the number of organisms that contributed spectra to the cluster represented by a cluster. If only a single species contributed spectra to a cluster, then $Target$ contains all unique peptides of length l in the species' proteome. If $k > 1$ species contributes spectra to a cluster, then $Target$ contains all unique peptides of length l in the intersection database of the k species.

- Define the decoy database as the set of all $20^l - |Target|$ peptides of length l that do not belong to $Target$. Compute the target/decoy size ratio $r = \frac{|Target|}{20^l - |Target|}$. If $r > 1$ we ignore this de novo result since we do not have a sufficiently large decoy to evaluate against.

- Evaluate the de novo result. If $P \in Target$ then we set $W_t = W_t + 1$, otherwise we set $W_d = W_d + r$. Normalizing the target/decoy hits as $1 : r$ (as opposed to $1 : 1$ as in standard TDA) accounts for the fact that target and decoy databases have different sizes in our case.

- Compute the FDR as $\frac{W_d}{W_d + W_t}$.

By partitioning the results according to peptide length and precursor charge we ensure that each set of spectra has similar fragmentation characteristics and uses the same scoring model, so there is no bias in the de novo scores. The reason we sort the spectra in each set according to the score (rank score in the case of PepNovo) is that we assume that higher scores are correlated with the accuracy of the de novo reconstructions. Thus, analyzing the spectra in this order should give the identifications with a low FDR first. Each de novo result that gets evaluated consists of a peptide P (l amino acids long) and a cluster's consensus spectrum S. We compute a separate target and decoy database for each case. The target database T consists of all unique l-mers in the sequence database of the organism whose spectra are associated with S (or l-mers in the intersection database if more than a one organism's spectra belongs to S's cluster). For a decoy database we use all $20^l - |T|$ peptides not in the target database. If the decoy database is too small (smaller than the target database) we discard the spectrum without attempting to identify it. If the peptide $P \in T$ then we keep the result and increment the weight of the hits to the target database: $W_t = W_t + 1$. Otherwise we assume that the peptide hit the decoy, in which case we increment the weight of the hits to the decoy database with the normalized target/decoy ratio: $W_d = W_d + \frac{|T|}{20^l - |T|}$. Normalizing the decoy weight to $\frac{|T|}{20^l - |T|}$ ensures that the weight in W_d represents the typical 1:1 target/decoy ratio, so at each stage we can compute the FDR for the peptide identification as $FDR = \frac{W_d}{W_d + W_t}$.

We applied the identification procedure described above to all the clusters for which the top-scoring de novo sequence was 4-6 amino acids long. The results are described in Table 4, which lists the number of unique peptide identifications made for different peptide lengths and precursor charges. The columns of the table split the results according to different ranges of the FDRs computed for the peptides. A total of 3466 peptides of length 4, 10013 peptides of length 5, and 45252 peptides of length 6 were identified with an FDR below 0.1, and about half them had an FDR below 0.05.

Most of the peptide identifications of length 4 and 5 were derived from clusters of spectra from several organisms (with large protein databases this is the only way to obtain a sufficiently large decoy database to

Peptide length	Precursor charge	Num. peptide ids with FDR in different ranges (number of ids from different k-clusters, $k = 1, 2, 3, 4, 5^+$)		
		$0 < FDR \leq 0.01$	$0.01 < FDR \leq 0.05$	$0.05 < FDR \leq 0.1$
4	1	113	1753	1600
		(3,4,12,5,89)	(268,185,134,127,1039)	(372,180,117,102,829)
4	2	2	0	0
		(2,0,0,0,0)	(0,0,0,0,0)	(0,0,0,0,0)
5	1	179	2118	5454
		(25,27,19,19,89)	(700,452,280,178,508)	(2946,1118,570,265,555)
5	2	150	729	1383
		(48,58,28,9,7)	(354,231,108,27,9)	(867,381,97,25,13)
6	1	1710	15240	16127
		(975,367,172,80,116)	(11490,2262,789,329,370)	(14066,1516,320,123,102)
6	2	52	4685	7414
		(33,15,2,2,0)	(3665,736,181,52,51)	(6404,820,126,30,34)
6	3	1	12	11
		(1,0,0,0,0)	(12,0,0,0,0)	(10,1,0,0,0)

Table 4: Short peptide identifications. The table describes the number of short peptides (4,5 and 6 amino acids long) identified in the spectral archive created from the PNNL data sets. The table lists the number of unique peptide identifications made with varying false discovery rates for different peptide lengths and precursor charges. The table also breaks down the identifications according to the number of organisms k that contributed spectra to the identified cluster ($k = 1, 2, 3, 4$ or 5^+).

ensure accurate identifications). In contrast, for length 6 we find that most of the identifications came from clusters of spectra originating from a single species. This is feasible for organisms with small proteomes (e.g., bacteria). For example, the *Shewanella oneidensis* proteome contains ≈ 1.5 million amino acids which is much smaller than $20^6 \approx 64$ millions, the number of all peptides of length 6 (the probability of a random hit to the database is ≈ 0.0235). Even with the human proteome, there are less the 10^7 unique 6-mers in the database, which means that most of the $\approx 6.4 \cdot 10^7$ 6-mers get assigned to the decoy database.

By analyzing the spectral archive, we were able to annotate the largest currently available set of spectra of short peptides. This collection of annotated spectra can be used both as a spectral library for peptide identification, and as a training set for creating more accurate scoring models for short peptides.

	Set of clusters searched			
Search type	10K random	10K largest	100K random	100K largest
DB search - no PTMs	1127	3488	8889	11840
DB search 15 PTMs	769	2717	6059	8941
"Blind" DB search	126	721	1412	2173

Table 5: Search results of different subsets of clusters from the HEK dataset. Close to 800,000 spectra from the human HEK cell line were clustered into 380,000 clusters. From this set of clusters, 4 subsets were selected and searched in a variety of methods: 10,000 randomly selected clusters, 10,000 of the largest clusters, 100,000 randomly selected clusters, and 100,000 of the largest clusters. The cluster's consensus spectra where searched against the human IPI protein database in 3 modes: regular database search, a search considering 15 common PTMs, and a "blind" search that considers arbitrary mass gaps. For each search, the table reports the number of unique peptide identifications that were made at a false discovery rate of 2%.

Supplementary Note 6 - Searches for Mutations and Unexpected Modifications: Large vs. Small clusters

While the algorithms for identification of peptides with mutations and unexpected modifications (blind MS/MS search) are available [15], they remain rather slow. As a result, blind MS/MS searches of large spectral datasets remain a luxury that very few mass spectrometry labs can afford. Other complex MS/MS searches [1, 10] face a similar computational bottleneck. Clustering can help focus the attention on the spectra that are more likely to be identified with blind and other complex MS/MS searches.

Table 5 describes results of experiments which compared the number of unique peptide ids made searching clusters from a HEK 293 dataset ($\approx 800,000$ spectra grouped into 380,000 clusters), in various search modes. When a limited number of clusters is searched (subsets of 10,000 and 100,000 clusters), searching the largest clusters has a clear advantage when compared to the results obtained by searching an equally sized set of randomly selected clusters. Searching the 10,000 largest clusters gave 3 (regular DB) to 5.5 (blind search) times more unique peptide ids than searching an equally sized set of 10,000 randomly selected clusters. When subsets of 100,000 clusters were considered the increase in identifications was by a factor of between 1.33 (regular DB search) and 1.9 (blind search). These results demonstrate that when the resources are limited one is better off searching the large clusters since they are more likely to be identifiable.

References

[1] N.E. Castellana, S.H. Payne, Z. Shen, M. Stanke, V. Bafna, and S.P. Briggs. Discovery and revision of arabidopsis genes by proteogenomics. *Proc. Natl. Acad. Sci. U S A.*, 105:21034–8, 2008.

[2] R. Craig and R.C. Beavis. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, 20:1466–1467, 2004.

[3] J.E. Elias and S.P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods*, 4:207–214, 2007.

[4] A. Frank and P. Pevzner. Pepnovo: De novo peptide sequencing via probabilistic network modeling. *Anal. Chem.*, 77:964–973, 2005.

[5] A.M. Frank. A ranking-based scoring function for peptide-spectrum matches. *J. Proteome Res.*, 8:2241–2252, 2009.

[6] A.M. Frank, N. Bandeira, Z. Shen, S. Tanner, S.P. Briggs, R.D. Smith, and P.A. Pevzner. Clustering millions of tandem mass spectra. *J. Proteome Res.*, 7:113–122, 2008.

[7] L. Käll, J.D. Storey, M.J. MacCoss, and W.S. Noble. Posterior error probabilities and false discovery rates: Two sides of the same coin. *J. Proteome Res.*, 7:40–44, 2008.

[8] H. Lam, E. W. Deutsch, and R. Aebersold. Artificial decoy spectral libraries for false discovery rate estimation in spectral library searching in proteomics. *J. of proteome res.*, 9:605–610, 2010.

[9] J. Liu, A.W. Bell, J.J. Bergeron, C.M. Yanofsky, B. Carrillo, C.E. Beaudrie, and R.E. Kearney. Methods for peptide identification by spectral comparison. *Proteome Sci.*, 5:3, 2007.

[10] J. Ng and P.A. Pevzner. Algorithm for identification of fusion proteins via mass spectrometry. *J. Proteome Res.*, 7:89–95, 2008.

[11] S.E. Stein and P.A. Rudnick. NIST Peptide Tandem Mass Spectral Libraries. Human Peptide Mass Spectral Reference Data, H. sapiens, ion trap, Official Build Date: Feb. 4, 2009. National Institute of Standards and Technology, Gaithersburg, MD, 20899, 2009.

[12] S.E. Stein and D.R. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *J. Am. Soc. Mass. Spectrom.*, 5:859–866, 1994.

[13] S. Tanner, Z. Shen, J. Ng, L. Florea, R. Guig, S.P. Briggs, and V. Bafna. Improving gene annotation using peptide mass spectrometry. *Genome Res.*, 17:231–239, 2007.

[14] S. Tanner, H. Shu, A. Frank, M. Mumby, P. Pevzner, and V. Bafna. Inspect: Fast and accurate identification of post-translationally modified peptides from tandem mass spectra. *Anal. Chem.*, 77:4626–4639, 2005.

[15] D. Tsur, S. Tanner, E. Zandi, V. Bafna, and P.A. Pevzner. Identification of post-translational modifications via blind search of mass-spectra. *Nature Biotech.*, 23:1562–2567, 2005.