Journal of proteome research

Article

pubs.acs.org/jpr

Terms of Use

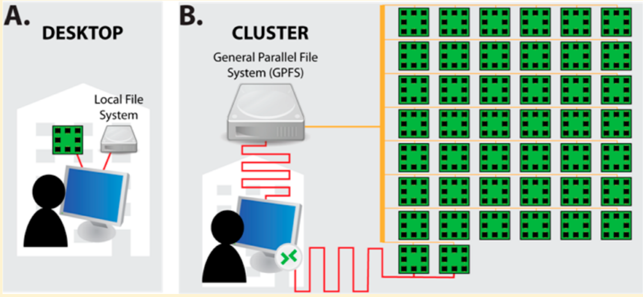
High Performance Computational Analysis of Large-scale Proteome Data Sets to Assess Incremental Contribution to Coverage of the Human Genome

Nadin Neuhauser,[†] Nagarjuna Nagaraj,[†] Peter McHardy,[‡] Sara Zanivan,[‡] Richard Scheltema,[†] Jürgen Cox,[†] and Matthias Mann*,[†]

[†]Department of Proteomics and Signal Transduction, Max-Planck Institute of Biochemistry, Am Klopferspitz 18, 82152 Martinsried, Germany

[‡]Vascular Proteomics Lab, Beatson Institute for Cancer Research, Garscube Estate, Switchback Road, Bearsden, Glasgow G61 1BD, United Kingdom

Supporting Information



ABSTRACT: Computational analysis of shotgun proteomics data can now be performed in a completely automated and statistically rigorous way, as exemplified by the freely available MaxQuant environment. The sophisticated algorithms involved and the sheer amount of data translate into very high computational demands. Here we describe parallelization and memory optimization of the MaxQuant software with the aim of executing it on a large computer cluster. We analyze and mitigate bottlenecks in overall performance and find that the most time-consuming algorithms are those detecting peptide features in the MS^1 data as well as the fragment spectrum search. These tasks scale with the number of raw files and can readily be distributed over many CPUs as long as memory access is properly managed. Here we compared the performance of a parallelized version of MaxQuant running on a standard desktop, an I/O performance optimized desktop computer ("game computer"), and a cluster environment. The modified gaming computer and the cluster vastly outperformed a standard desktop computer when analyzing more than 1000 raw files. We apply our high performance platform to investigate incremental coverage of the human proteome by high resolution MS data originating from in-depth cell line and cancer tissue proteome measurements.

KEYWORDS: *tandem mass spectrometry, shotgun proteomics, performance, analysis pipeline*

■ INTRODUCTION

The technology of mass spectrometry (MS)-based proteomics has been improving at a very fast rate during the last two decades.[1−4] Advances in instrumentation have reduced acquisition time and increased resolution and sensitivity, which in combination with the high resolving-power of current mass analyzers in both MS and MS/MS mode have led to very large data sets. For instance, in our laboratory we routinely acquire 400 000 to 650 000 data-dependent MS/MS spectra for every quadrupole−Orbitrap instrument[5] per day, requiring approximately 14 GB of storage space. The growing file size and number of raw files dramatically increases the burden on the computational tools used to analyze these data.[6] This analysis is becoming so computationally intensive that it can preclude processing on a standalone personal computer (PC) or make it so slow that it prevents researchers from trying different scenarios or hypotheses.[7]

Once the basic algorithms in the computational proteomics pipeline have been thoroughly optimized, overall performance improvements rely on better computational hardware. In addition to faster processors, developments in computer science have increasingly focused on the use of multiple processors in parallel. Parallelism as such has been employed for many years, mainly in high-performance computing (HPC). With parallel computing, computational problems are divided into smaller ones and solved concurrently, distributing the computational effort in many cases over multiple CPUs (central processing unit). This can be among multiple cores within a single processor, a multiprocessor system or a network of computers—a so-called computing cluster. However, this new hardware requires parallelized algorithms, to benefit from the increased hardware capacities. This is by no means trivial, and most existing applications cannot exploit multicore systems yet. Typically only some parts of computational problems can be completely parallelized, and overall performance is frequently limited by access to shared resources or communication between tasks. Despite these obstacles, the number of applications that use parallelization is gradually increasing. As an example from bioinformatics on next-generation sequencing

Received: February 28, 2013
Published: April 23, 2013

Table 1. Key Parameters of the Three Hardware Platforms

	standard desktop PC	high-end gaming PC	computer cluster
computing capacity	using 4 virtual cores	using 8 cores	using 336 virtual cores
computing power	3.4 GHz, 16 GB of RAM	3.4 GHz, 32 GB of RAM	2.53 GHz, 24 GB of RAM
I/O performance	HDD, SATA	SSD, PCI Express RAID	HDD, SAS + GPFS client

data, an algorithm for sequence alignment has recently been reimplemented using the principle of parallelization to use the power of a multicore environment.[8] The parallel algorithm had an analysis time 20 times faster than the nonparallelized (serial) version.

In computational proteomics, data analysis typically involves several steps and is not confined solely to peptide identification by a peptide search engine such as Mascot.[9] There are few software solutions that aim to provide data analysis from acquired raw data to final protein lists in a single environment. Examples are the Trans-Proteomic Pipeline,[10] OpenMS Proteomics Pipeline[11] or Skyline.[12] Our own laboratory has developed the MaxQuant computational proteomics framework, which is freely available to academic and commercial users and which has been widely adopted by the research community.[13,14] MaxQuant enables processing of raw MS data files, incorporates its own probabilistic search engine called Andromeda[15] and has recently been supplemented by the extensible Perseus environment for statistical and functional analysis.[16] To increase the performance of our analysis pipeline, we here adapt MaxQuant to exploit nonshared memory parallel computing, so it can be run on a high-performance computer cluster. Due to the fact that many of the component tasks are independent of each other, the MaxQuant pipeline could be substantially parallelized. This led to dramatically increased performance, which we demonstrate here by analyzing the coverage of the human genome by large-scale data sets from high resolution shotgun proteomics. We also compare performance of the cluster to intermediate hardware solutions such as high performance personal computers, which would be economically accessible to all groups using state of the art proteomics.

■ EXPERIMENTAL METHODS

Implementation of MaxQuant

Originally MaxQuant was developed to run on desktop computers with one or multiple cores, which can support a semiparallelized instance of the software. The cluster instead has a large number of nodes, consisting of multiple cores (see below). To keep MaxQuant independent from the hardware setup during parallelization, the original implementation was refactored. This step left the core algorithms for desktop and cluster versions identical, only differing in the way that the single tasks in the analysis pipeline are called from the exchangeable framework. For the desktop version the MaxQuant software itself is in charge of executing the code at the correct time in the pipeline. For the cluster this control needs to be relinquished to a job manager, requiring a new interface that uses the Job Manager provided by Windows HPC 2008 R2. MaxQuant automatically generates a job instance spanning several tasks and passes the instance to the job manager, which then distributes the tasks over all available nodes. The job manager is aware of all resources and takes care of the task queue, which can originate from different users. For this reason, the graphical interface of MaxQuant can be closed after submitting the job, in contrast to the desktop version.

Next we set out to adapt MaxQuant to efficiently use the power of the high performance cluster. The principle units of parallelization are the raw files from the project to be analyzed, and the basic structure is to allocate each raw file to a physical or virtual core. In the desktop version, we had used multiple processes, which enabled semiparallelization because the number of cores is limited on a standard PC. The challenge was therefore to correctly distribute the different tasks over several nodes.

The code in MaxQuant is structured in so-called "task groups" which are codependent and have to run one after the other. As an example, detecting the features in MS[1] scans is a task group. Each of these task groups consists of several instances, where the number of instances is dependent on the number of raw files. As these instances can run in parallel, we distribute them over the available nodes according to how many cores are available on each node. The code executed on each of the nodes is the same as on the desktop version. Implementing this basic parallelization initially led to low usage of the computing power of the nodes, because only a few of the necessary tasks truly ran concurrently.

To enable more efficient parallelization we first identified the bottlenecks in performance. In this process the poorly performing sections were iteratively identified that could safely be executed in parallel. For instance, protein group assembly consumed a disproportional amount of time (see Results and Discussion). Within this task group we identified functionality that can run in parallel and split this task group into three new task groups: "Prepare protein assembly", "Assembling protein groups" and "Finish protein assembly". Of these tasks, "Assembling protein groups" can be broken up into many small parts that can be executed in parallel (i.e., each protein group can be processed independently), whereas the other tasks cannot be performed in parallel as they consist of a single task. With this improvement, we obtained an enormous speedup in this part of the pipeline. This process was performed on the most time-consuming task groups in the pipeline. Compared to previous versions consisting of 22 task groups (Version 1.2.0.0) under default conditions, we now have 38 task groups of which 20 groups are parallelized. This division has the advantage that so-called fallback positions are created, enabling partial processing where the researcher can for example reprocess a part of the pipeline with different settings.

All of the parallelization improvements made for the cluster version also benefit the normal PC version when many CPU cores are available. As a last step, we identified the major bottleneck for these types of machine, which turned out to be the input and output (I/O) access to hard drives. To mitigate this bottleneck, we optimized the hardware as described below.

Hardware Setup

For performance benchmarking we used three different hardware setups (see Table 1). As a representative normal desktop PC we used an Intel Core i7−2600 processor with 3.4 GHz, 16 GB of RAM and 460 GB space on a conventional hard

Table 2. Summary of Data Sets Used for Mapping the Human Proteome

	name	enzyme	type	instrument	raw files	scans
1	11 cell lines[17]	trypsin	cell lines	LTQ Orbitrap XL	198	7,779,031
2	8 cell lines	trypsin	cell lines	Q Exactive	145	17,477,801
3	breast cancer[18]	trypsin	cell lines	LTQ Orbitrap XL	420	7,448,447
4	colon cancer - I[19]	trypsin	tissue	LTQ Orbitrap XL	135	4,461,151
5	colon cancer - II[20]	typsin/lys-C	tissue	Q Exactive	24	2,000,864
6	urinary proteome[21]	lys-C	body fluid	LTQ Orbitrap XL	82	1,495,293
					1004	**40,662,587**

disk drive (HDD) with a serial advanced technology attachment (SATA) connection (purchased from Dell Computers). Since such a computer is meant to still be available for normal office work, we use only 4 of the 8 virtual cores for processing the data sets. Additionally, we chose a high performance desktop computer, custom-built for advanced video gaming, which is employed in our department for highly demanding computations. This type of computer has the similar processor, but is equipped with 1 TB of solid state disks (SSD) configured in RAID 0 connected via a PCI-Express RAID controller with a battery backup unit and full cache enabled. A RAID configuration is providing a potential factor of 2 in read access speed as the data is duplicated on both drives. The I/O optimized machine also uses faster memory (DDR3 1866 MHz Quad Channel). This computer was purchased from Eclipse Computing, Ayrshire, UK and costs two to three times the amount of a desktop computer designed for typical computational tasks (for current configuration employed in our department see www.maxquant.org). We store our data on the solid state drive, which has a dramatic effect on the crucial I/O performance bottlenecks. These configurations were compared to our Windows cluster equipped with 44 nodes, two of which are exclusively used to submit MaxQuant jobs. Each node consists on an Intel Xeon E5540 processor with 2.53 GHz and 24 GB of RAM. For the global data storage, we found it advantageous to install a high performance general parallel file system (GPFS) with 10 TB of storage space on a HDD using the SAS protocol.

A 64-bit version of Windows 7 is installed on the standard desktop computer and on the I/O optimized high end computer, whereas the cluster was run with a 64-bit version of Windows HPC 2008 R2. The installation of the freely available Thermo MS FileReader and .NET Framework 4.5 is necessary for all three platforms. For more information, see www.maxquant.org/requirements.htm

We have also begun testing two rack mounted configurations with 64 logical processors, where multiple cores share the same memory. Both machines have 128 GB of memory (16 × 9 GB Dual Rank RDIMM) for 4 CPUs with 1600 MHz. The major difference for these two setups is that one is using 4 AMD processors (Opteron 6276 with 2.3 GHz) and the other is designed with 4 Intel processors (Intel Xeon E5−4640 with 2.4 GHz). Additional a RAID Controller PERC H700 or PERC H710p with 1GB NV cache is installed, respectively. The storage space is basically the same 6 × 900 GB HDD using SAS protocol. The operating system on both solutions is the Microsoft Windows Server 2012 Standard 64-bit.

Data Sets for Human Proteome Analysis

To obtain a large data set for evaluating the performance of the different hardware setups, we combined raw files from different published experiments from our group. In total we used data from in-depth proteomics studies of 30 different cell lines[17,18] resulting in 763 raw files. To cover more of the human proteome we also included raw files from two tissue[19,20] and one body fluid projects.[21] For the estimation of the measured human proteome we employed a total of 1004 raw files from the studies listed in Table 2. For an estimation of the runtime behavior of our application we tested five data sets with varying raw file numbers (6, 18, 198, 343, and 763 raw files) on the three different hardware setups.

Data Analysis

All data were processed with MaxQuant[13] version 1.3.7.4 using Andromeda[15] to search the MS/MS spectra with trypsin or LysC specificity against the complete human data set of the UniProt database[22] (release January 2013, 87638 entries) combined with 262 commonly detected contaminants. We allow for up to two missed cleavages and N-terminal acetylation and methionine oxidation were selected as variable, carbamidomethylation of cysteine was selected as fixed modification. For MS spectra an initial mass accuracy of 4.5 ppm was allowed and the MS/MS tolerance was set to 20 ppm. A sliding mass window was applied to filter the MS/MS spectra for the 10 most abundant peaks in 100 Th. For identification, the FDR at the peptide spectrum matches (PSM) and protein level was set to 0.01.

Availability

The desktop version of MaxQuant as well as the special cluster version are freely available at http://www.maxquant.org/downloads.htm

■ RESULTS AND DISCUSSION

Many of the tasks in computational proteomics place very challenging demands on the computational hardware. These demands can be thought of as a combination of three different factors: (i) processing power of the computer chips or cores employed, (ii) the number of these cores, and (iii) the speed of read and write operations. Importantly, an improvement in any one can fail to improve the analysis time when other factors still act as a bottleneck to the whole system. For example, extremely high processing speed may be practically unimportant if reading of raw data or distribution to the relevant cores is slow.

The processing power of single cores improves over the years. For setting up a computational pipeline, one typically selects the fastest and most cost-effective version of mainstream and mass produced products, such as Intel or AMD chips. The trend in high-performance computing has been to group multiple processors together, both in single chips and by connecting large numbers of chips (computing clusters). In principle, the computational capacity is multiplied by the number of chips, however, this requires efficient parallelization of the software (discussed below). Furthermore, data for processing need to be available to the cores and intermediate results need to be written out sufficiently fast so as not to slow

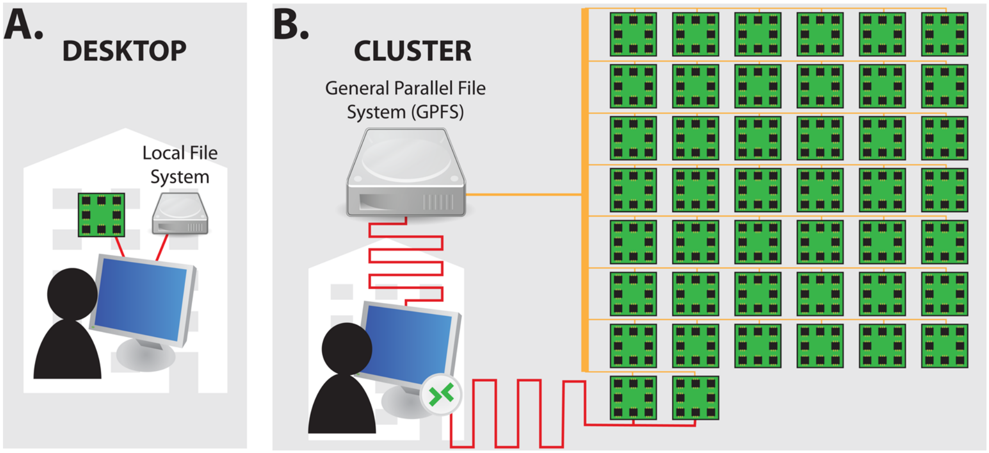
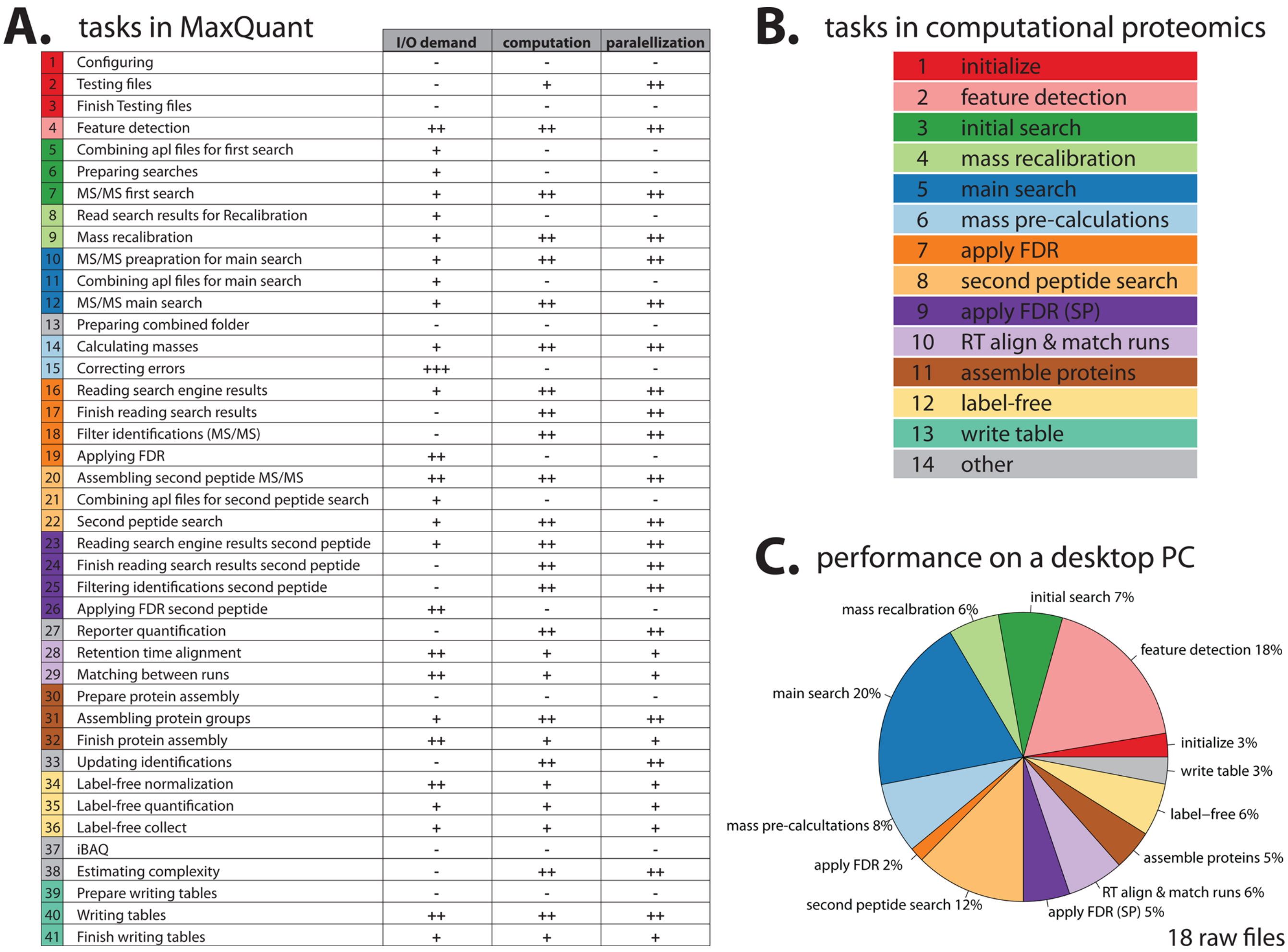


Figure 1. Distinct hardware setups. (A) In the field of proteomics, desktop computers with a single (multicore) processor and data located on a local file system are generally used. (B) In contrast, a computer cluster has multiple nodes, where one node represents one desktop computer. Additionally a global file system is required where the raw and meta data are stored and is accessible from all nodes. Usually, the cluster has a remote location, such as in a large computing center. In the figure, we compare a quad core desktop computer with a cluster consisting of 42 nodes and 336 cores, both running the Windows operating system.



		I/O demand	computation	paralellization
1	Configuring	-	-	-
2	Testing files	-	+	++
3	Finish Testing files	-	-	-
4	Feature detection	++	++	++
5	Combining apl files for first search	+	-	-
6	Preparing searches	+	-	-
7	MS/MS first search	+	++	++
8	Read search results for Recalibration	+	-	-
9	Mass recalibration	+	++	++
10	MS/MS preapration for main search	+	++	++
11	Combining apl files for main search	+	-	-
12	MS/MS main search	+	++	++
13	Preparing combined folder	-	-	-
14	Calculating masses	+	++	++
15	Correcting errors	+++	-	-
16	Reading search engine results	+	++	++
17	Finish reading search results	-	++	++
18	Filter identifications (MS/MS)	-	++	++
19	Applying FDR	++	-	-
20	Assembling second peptide MS/MS	++	++	++
21	Combining apl files for second peptide search	+	-	-
22	Second peptide search	+	++	++
23	Reading search engine results second peptide	+	++	++
24	Finish reading search results second peptide	-	++	++
25	Filtering identifications second peptide	-	++	++
26	Applying FDR second peptide	++	-	-
27	Reporter quantification	-	++	++
28	Retention time alignment	++	+	+
29	Matching between runs	++	+	+
30	Prepare protein assembly	-	-	-
31	Assembling protein groups	+	++	++
32	Finish protein assembly	++	+	+
33	Updating identifications	-	++	++
34	Label-free normalization	++	+	+
35	Label-free quantification	+	+	+
36	Label-free collect	+	+	+
37	iBAQ	-	-	-
38	Estimating complexity	-	++	++
39	Prepare writing tables	-	-	-
40	Writing tables	++	++	++
41	Finish writing tables	+	+	+

1	initialize
2	feature detection
3	initial search
4	mass recalibration
5	main search
6	mass pre-calculations
7	apply FDR
8	second peptide search
9	apply FDR (SP)
10	RT align & match runs
11	assemble proteins
12	label-free
13	write table
14	other

Figure 2. Time spent on tasks and groups of tasks in the MaxQuant pipeline. (A) Detailed list of task groups that are performed on the raw data in the course of a complete MaxQuant analysis. For each of them the demands or suitability to I/O, computational power and parallelization are indicated. (B) Task groups from A grouped into larger procedures. (C) Proportional times consumed by the procedures from B. during a typical analysis of 18 raw files of approximately 20 GB per data set (total of 133 110 high resolution MS scans and 514 912 high resolution MS/MS scans).

down overall performance. This may require equipping the cores with large individual memory stores and advanced overall memory management.

Given efficient hardware for computationally intensive tasks, the software needs to be structured to take optimal advantage of the resources. In general, one tries to divide the computational workflow for one proteomic analysis (a "job") into largely self-contained units of 'tasks' that run independently as separate processes. With this strategy, tasks normally do not communicate with each other. Designing a parallel workflow therefore involves "decomposition", which entails breaking down a complex system into smaller pieces, to find tasks that can run concurrently in parallel applications. There are two major decomposition methods in parallel programming, functional and data decomposition.[23] Functional decomposition requires a restructuring of the algorithms into independent units, which can be very challenging. Data decomposition is used more often, because it only requires a solid understanding of the data and how the algorithms process it. In the context of computational proteomics data, decomposition can take the form of processing each of the raw files on a different core.

Once a significant fraction of the proteomic analysis pipeline is separated into independent tasks executed on different cores, it is crucial to minimize communication between the cores. Likewise, the tasks of the different cores must be balanced, so that ideally no single core does more work than the others. Furthermore, when working with large amounts of data on a distributed computing system, the speed and latency of the network can be a bottleneck. (This largely makes cloud computing solutions impractical in current computational proteomics.)

Implementing MaxQuant on a Cluster

When MaxQuant was released in 2008, it was designed to be executed on conventional desktop PCs. The requirements to run MaxQuant efficiently were to have sufficient processing power and space on a local disk (see Figure 1A). Already in the original release, the program was semiparallelized using multiple processes. The user had to enter the maximum number of threads to be used, depending on available virtual cores and other uses of the computer[14] (when the number of threads selected is the same or higher than the number of available computing cores, the computer will become unresponsive). In the new release, MaxQuant was extended to use a computer cluster, where the processes are distributed over several computational nodes. A typical computer cluster contains many nodes, ideally with the same configuration and a global file system that is accessible from each of the nodes and where the data is stored (see Figure 1B).

As shown in Figure 2A the computational pipeline, which appears as a single and unified whole to the user, can conceptually be broken down into consecutive "task groups" were some can be parallelized and others not. Limiting factors for the performance are (i) their demand on I/O speed, (ii) the CPU load of the particular computations and (iii) the degree to which the task groups can be parallelized.

The computational task groups of MaxQuant can be conceptualized as 14 fundamental groups, which we briefly summarize below (see Figure 2B). In the "initialize" phase the raw files are verified for intactness and readability, additionally an index file for each raw file is created containing scan metadata that is accessed many times. The next step—"feature detection"—extracts the peptide features present at the full scan (MS^1) level, typically peptide precursor masses. The detection of the 3D peaks (m/z over the retention time) and of label pairs (typically SILAC-pairs) is also performed here. In the following section we perform an initial search using the Andromeda search engine[15] to get a first list of identified peptides, which can be used in later steps. To correct for any systematic mass errors occurring during data acquisition the initial list of identifications is used to calculate mass calibration curves over time and m/z.[24] The search is now repeated with the updated m/z values resulting in the final list of peptide identifications, where for each fragmentation spectrum the up to 10 best scoring peptide sequences are retained. After peptide identification, the peptide spectrum matches (PSM) whose measured mass differences exceed the individually calculated mass tolerance are removed and the peptide identification with the next best score falling within the mass tolerance is retained.[13] The peptide false discovery rate (FDR) is calculated on this prefiltered peptide list and peptide identifications below a specified threshold are discarded. Since different peptide species resulting in very similar m/z values can elute at the same time, multiple precursors can occur in the same selection window, leading to fragmentation of several peptides in a single MS/MS spectrum. In cases where we identified the intended precursor, a third Andromeda search attempts to also identify the coeluting and cofragmented peptide. These additional peptides require a separate FDR correction.[15] For replicates or comparison of different runs, a sophisticated tree-based retention time alignment is performed. This alignment is used to transfer peptide identifications to raw files were a particular feature is observed but not identified ("match between runs" option in MaxQuant), increasing the number of identified peptides per raw file and reducing missing values for quantification.[17] The next step is to assemble the identified peptides to proteins. For this purpose we group proteins that are identified by the same peptides in a user-configurable manner. Depending on the user settings, label-free quantification is performed, correcting for systematic differences in quantification between the raw-files. The last step is to write the results to output tables, which can then be used for downstream analysis or loaded into the "Viewer" part of MaxQuant for detailed visual inspection of the data. Notably, the output of the calculations themselves can reach gigabytes.

The time spent on each of these task groups is strongly dependent on the number and the size of the raw files. Figure 2C illustrates the percentages of the total time spent on each task group for a typical project using a standard desktop computer (see Experimental Methods). The data set contains measurements of a fractionated cell line in triplicate, giving rise to 18 raw MS files. Although most computation time is required for the peptide identification by the Andromeda search engine (initial search, main search and second peptide search), this takes less than half of the total (39%). The next largest item is the feature detection in these large data files (18%). Tasks like mass recalibration (6%), applying the peptide FDR (7%), match-between runs including retention time alignment (6%), label-free quantification (6%) and protein group assembly (5%) are also time-consuming. If the number of cores is limited, many of these computational times grow directly with the size of the data set, quickly becoming impractical.

We next illustrate the benefits of parallelization on the main peptide search procedure. After preprocessing of the MS/MS spectra in MaxQuant the resulting fragment spectra are sorted

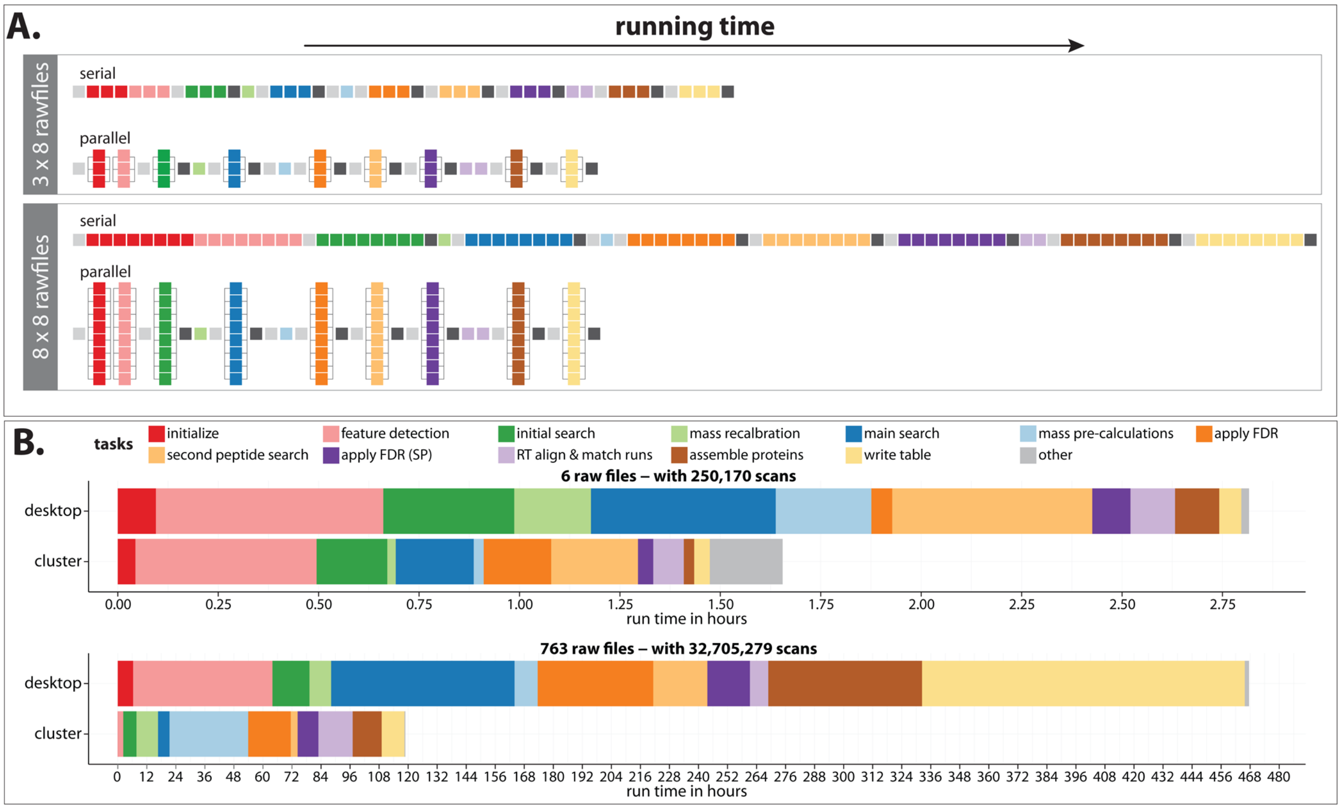


Figure 3. Comparison of the improvements by parallelization of different task groups. (A) Conceptual visualization of the effects of parallelization of some but not all task groups on total computing time. When only a few raw files need to be analyzed the gain is minimal, but with extension of the data set the time savings become dramatic. (B) Total run times of two different data sets (6 vs 763 raw files) with color coding of the different task groups.

by their precursor peptide mass, which has important advantages later on in the search. This task is not trivial, since the work should be distributed in a way that all processes finish at the same time, to avoid slowing down the overall pipeline with a single peak list file taking a disproportional amount of time to finish. For this reason, we make the number of spectra in each peak list file dependent on the peptide mass range to counterbalance the increasing combinatorial calculation time for peptides with higher mass. The first step in the Andromeda search is the creation of the peptide sequence database with its associated peptide masses. The *in silico* digestion of the proteins and the creation of database search indices is parallelized using multithreading of only one processor. Since we have split all spectra into independent peak list files at this point, the following peptide search can be executed in separate, parallel processes. This data decomposition is done in a similar manner for the initial and second peptide search. We compared serial and parallel execution by running either a single or 18 CPUs on the cluster. For the initial, main and second peptide search of a data set of 547 900 MS/MS spectra that are distributed into 18 peak list files we decreased the run time almost 7-fold (1.1 h for the parallel and 7.9 h for the serial search, respectively).

Similarly to the peptide search tasks, we particularly concentrated our parallelization efforts on feature detection, mass recalibration, FDR application, protein group assembly and writing tables, constituting the major remaining bottlenecks.

Performance of Desktop vs Cluster for Data Sets of Variable Size

In modern proteomics, large numbers of files are often analyzed together. These could for instance be generated during in-depth analysis of a proteome with many fractionation steps across several conditions. Furthermore, all files associated with a given project spanning many months or even years are best analyzed together in MaxQuant to guarantee overall comparability of results and to avoid inflation of the FDR.[25] Ideally, the computational proteomics infrastructure should not pose a limitation to such analyses. Here we investigate the gains of our optimization efforts using a computer cluster consisting of 42 nodes with 8 virtual cores each, resulting in the potential for 336 parallel operations. We compare this setup to a conventional desktop PC with a comparable processor configuration, in which 4 parallel cores are dedicated to MaxQuant. In Figure 3A the advantages of parallelization in terms of analysis time are visualized by horizontal bars, consisting of individual tasks that represent the processing time for each task group. If the task group can be parallelized, the bar is rotated vertically since a group of files is now analyzed as fast as a single file. In cases with only few raw files rotating these task groups vertically does not shorten the entire processing time appreciably. However, for larger number of files, the savings become dramatic. To test this on a specific example, we analyzed a small data set with 6 raw files and a large data set with 763 raw files on both the desktop and the computer cluster (Figure 3B). For the small data set, the saving in computation time were 41% (2.8 h vs 1.7 h). However, for the very large data set, processing time on the cluster was 5 day

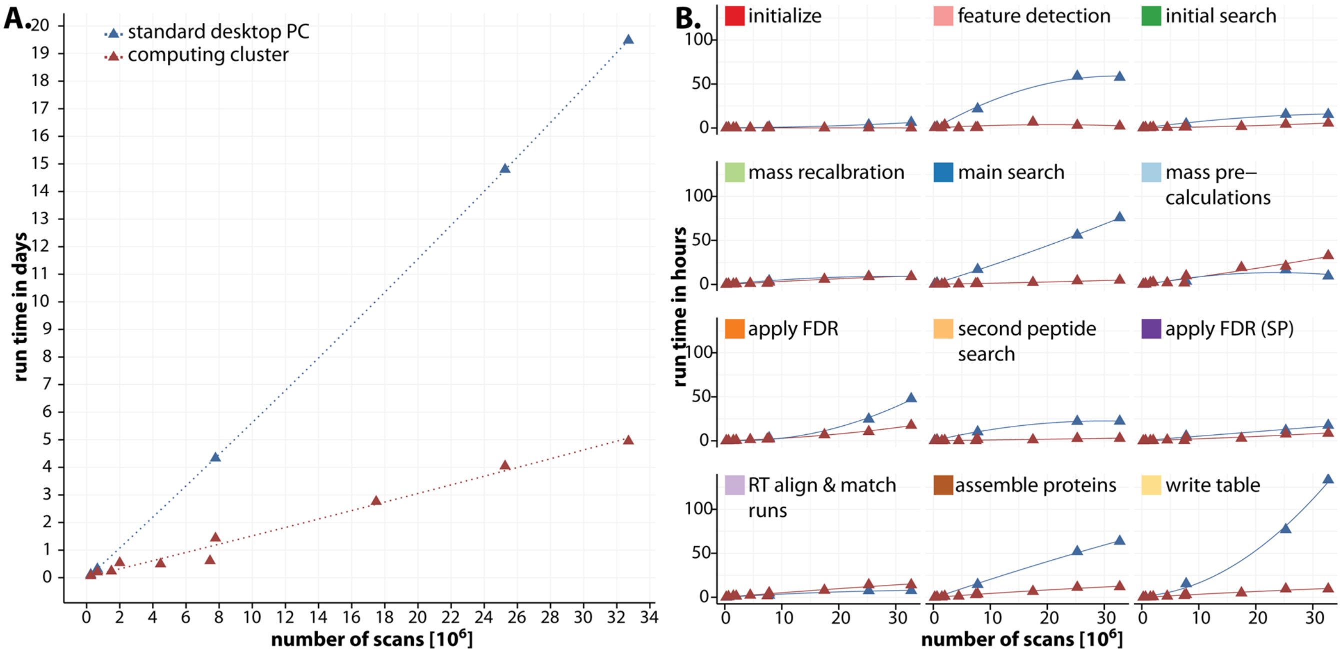


Figure 4. (A) Computational time as a function of data to be processed. The *x*-axis is in units of added MS scans in the individual raw files. (B) Same as in panel A, but for each task group separately.

whereas the desktop calculation took almost 20 days (Suppl. Table 1).

Next, we systematically investigated the advantages of the computer cluster over the desktop computer for increasing number of raw files. Because different raw files can contain very different amounts of data, we scaled the *x*-axis in Figure 4A in scans instead of raw files (one raw file contains generally between 3,500 and 21,000 scans, depending on the instrument, gradient length and the chosen topN method). Recapitulating the results described above, a clear trend emerged, in which the saved computing time was negligible for small data sets and increased drastically at very large data sizes. We also plotted processing times for the different task groups separately (Figure 4B). This revealed that feature detection benefited most from parallelization, followed by the main peptide search. However, tasks like write out of the large output tables also profit extensively from parallelization (in this case because MaxQuant needs to access all the raw files in this task group, which is much faster in parallel mode).

Performance of an I/O Optimized Desktop Computer

Given the time expenditure of the MaxQuant task groups on the large data sets, it appeared that memory constraints during access to raw and intermediate data might play just as large a role as total processing power (Figure 5). We tested this notion using a custom-built computer that was optimized for applications such as high end gaming, equipped with 1 TB of solid state disks configured in RAID mode (see Experimental Methods). On this computer we used all 8 cores and processed small, medium and very large data sets. As can be seen in Figure 5, the processing time per raw file was very similar to that of the cluster, even for the very large data set. As the cluster has 336 cores and the I/O optimized high end desktop computer only 8, we conclude that the benefits of parallelization accrue mainly from better I/O access, whereas computing power is less of a limiting resource under these circumstances. In terms of expenditure, this makes high-end computational resources readily accessible to a large number of research groups lacking access to cluster computing facilities.

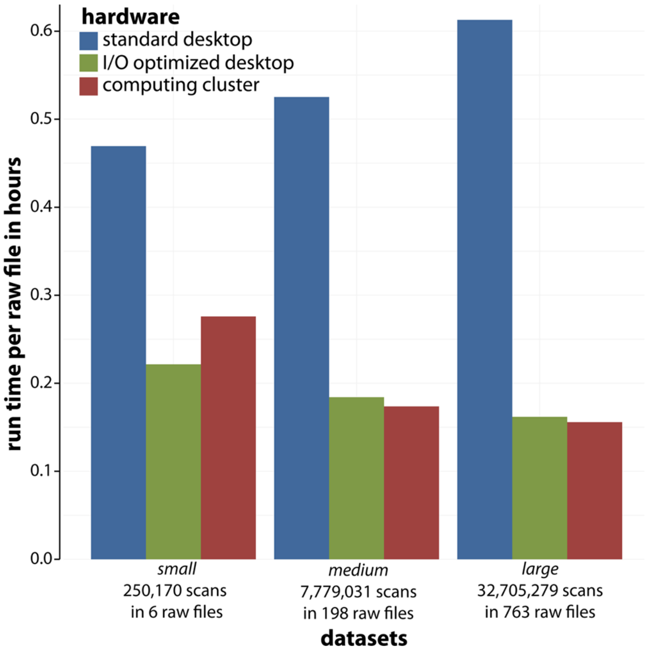


Figure 5. Comparison of total analysis time for a small, medium and very large data set using a desktop, I/O optimized, high end desktop and a computer cluster.

Incremental coverage of the human genome by large-scale data sets

Although the human genome has been sequenced more than ten years ago, it is still not clear how many different gene products it specifies. Estimates for the number of protein coding genes have been shrinking over the last ten years, from initial values of over 40,000 to a recent one that finds 20,225 open reading frames with at least some associated experimental

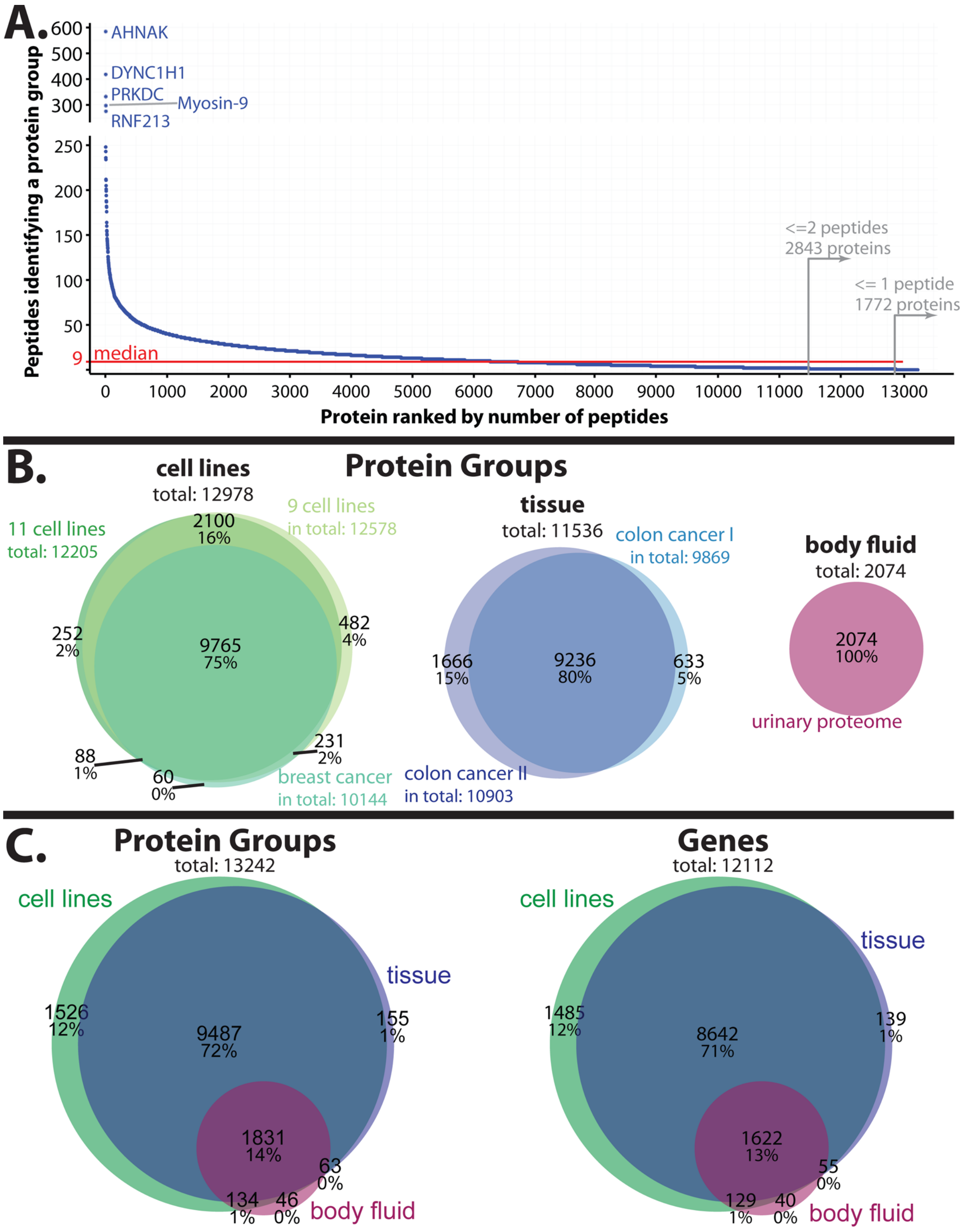


Figure 6. (A) Number of peptides identifying each protein group. Proteins are ranked by highest to lowest number of peptides. AHNAK, or desmoyokin an abundant and exceptionally large protein (700 kDa), is identified the largest number of unique peptides. Interestingly the relatively uncharacterized E3 ligase RNF213 is also among the top 5 proteins. (B) Number of proteins identified in the three cell line projects, two colon cancer projects and in the body fluid proteome. (C) Total number of protein groups identified from the cell line, colon cancer and body fluid proteomes shown individually in A (left). Same Venn diagram but showing different protein coding genes instead of protein groups (right).

or bioinformatics evidence.[26] Definitive proof of protein coding potential would be provided by solid data obtained by MS based proteomics. Accordingly, one of the goals of the chromosome centric Human Proteome Project is to map the entire human protein set to the set of protein-coding genes.[27] Currently, unambiguous protein level information is missing for up to 30% of human genes.[28] Here we employ a large collection of high resolution mass spectrometric data to determine the increase of coverage of the human genome as more and more experiments spanning multiple human sample types for different conditions are combined. Due to the large number of raw files involved, this task could not be carried out with a standard desktop configuration but rather required computational advances as described above.

We have previously found that in-depth proteomic sequencing of human cancer cell lines allows unambiguous identification of about 10,000 different protein groups using currently available technology[17,29] and other groups have reported similar results.[30,31] We therefore collected raw files from our laboratory from three deep cell line proteome projects

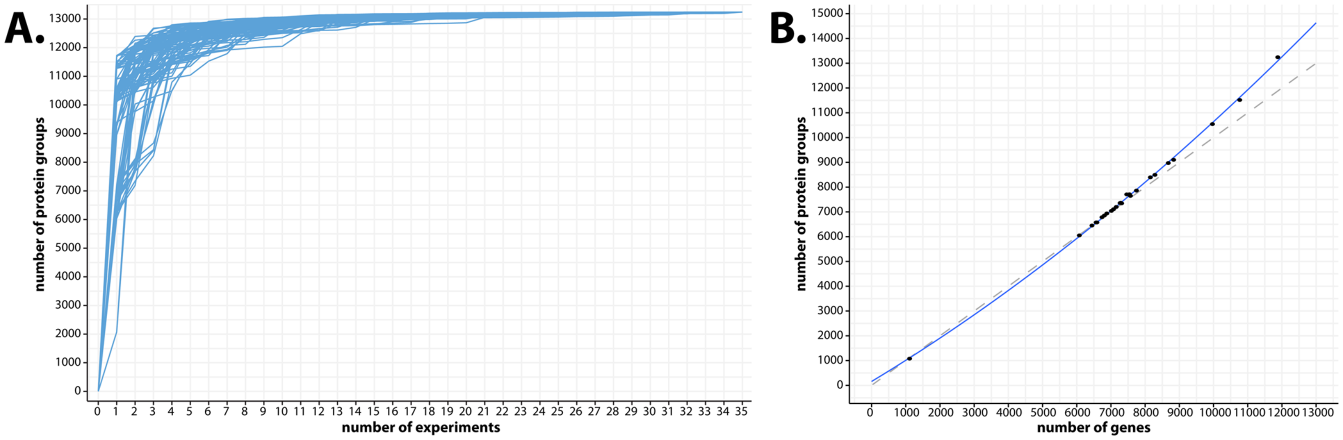


Figure 7. (A) Saturation curves of number of proteins identified when incrementally adding experiments. The traces represent 100 different simulations. (B) Number of identified and distinguishable protein groups as a function of the number of identified protein coding genes (see text for details).

together covering 30 cell lines[17,18] (Experimental Methods). Furthermore, we added data from two recent studies of colon cancer tissues[19,20] as well as a representative of a body fluid proteome.[21] Together these data comprised a collection of 1004 raw files, analyzed together on the cluster in a run time of only 5.5 days. At a 1% percent FDR at both peptide and protein levels, MaxQuant found a total of 13,242 protein groups in the UniProt database. We identify 255,432 different tryptic or LysC peptides, whereas 61,583 peptide sequences are unique within all proteins in the fasta file. A protein group contains on average 14.9 unique peptides, whereas the median is 9 (Figure 6A, Suppl. Table 2b). Just 1.9 or 3.6% of the proteome was identified with only one or only two peptides, respectively (Figure 6A). Sequence coverage was on average 42% (41.2% median). To our knowledge, this is the largest collection of unique peptides reported so far. For the expressed proteome that was identified here, close to half of the primary structure was therefore verified on average in this data set. Matching all separately identified protein groups to the human genome yielded 12,112 genes, which is almost 60%, assuming a total number of 20,225.

The data analyzed above originated from three main sources – three different cell line projects, two colon cancer tissue investigations and a study of the variability of the urinary proteome (Table 2). The three cell line studies together identified more than 13,000 different protein groups (Figure 6B). The depth of coverage in each project depended on the technology used (long columns and Q Exactive, vs shorter columns and LTQ Orbitrap Velos mass spectrometers), but the main finding is that there is a very large overlap among the cell line proteomes. Notably, despite a very large number of raw files (420), the breast cancer cell line study added only 3% unique proteins to the other two cell line projects. This reflects the advance in shotgun proteomics technology and illustrates a general finding that accumulating large number of measurements by itself does not necessarily lead to larger identified proteomes.

The large overlap in cell line proteomes agrees with previous findings that found remarkably similarity in the identity – if not the abundance – of the expressed proteins.[17,32] Naturally, the two cancer tissue proteomes have large overlap but interestingly the number of proteins identified in this single *in vivo* source was 11,536 – not much smaller than the total number from the different cell lines. The body fluid proteome identified about 2000 protein groups, partially reflecting the higher dynamic range of this proteome and the absence of fractionation.

Next we compared the cell line projects, colon cancer study and the body fluid study (Figure 6C). Again we found a large overlap, and intriguingly the in-depth colon cancer proteome only added 1% to the total identified proteome. This may reflect the fact that nearly all these proteomes are of cancer origin, but it also highlights the fact that the addition of tissue, per se, does not necessarily add many unique protein identifications. Likewise, the urinary proteome only added very few new proteins, indicating that body fluids may also not necessarily add substantially to overall coverage. When considering protein groups mapped to genes, we observe slightly smaller overall numbers, but the proportional contribution of the individual proteome sources remains largely unchanged (Figure 6C).

To study saturation properties of proteome coverage in large data sets in more detail, we investigated how quickly proteome coverage was reached as a function of the number of experiments used as input. Since this depends on the order in which the projects are added, we simulated the additional coverage from the analysis of the 1000 raw files. Using 100 different combinations yielded the saturation curves shown in Figure 7A. In some of the simulations, the final proteome coverage was essentially reached with less than 5% of the total experiments. For instance, measurements with the Q Exactive and long columns reached already 95% of the total protein identifications with only 91 raw files. In each of the 100 simulations, nearly the final number of identified proteins was obtained after a third to half of the experiments had been added. This analysis again underscores that reaching a given depth depends more on the technology used than on the cumulative number of analyses.

Finally, we investigated the relationship between identified protein groups and genes. As can be seen in Figure 7B, at low numbers of identified genes the ratio between protein groups and genes is about one to one. Starting from 7000 genes, a larger number of isoforms is added as a function of additional genes. This is because the increasing depth of coverage necessary to identify many genes also concomitantly leads to increasing sequence coverage.

CONCLUSION AND OUTLOOK

Here we have analyzed the different task groups comprising the computational pipeline in the MaxQuant environment. This revealed specific bottlenecks, which were removed as far as possible. The resulting MaxQuant version is highly parallelized and memory optimized. For small sets of MS raw files it performs very fast on both the desktop or on a large cluster. For instance, in our laboratory we often analyze proteomes with six fractions, each measured in 4 h gradients, which in triplicate experiments results in 18 large raw files. These are processed in 7.6 h on a standard desktop (using 4 virtual cores) and 4.9 h on the cluster (Suppl. Table 1). For very large data sets, however, the cluster massively outperforms the desktop computer, to the extent that some analyses are only practical on the cluster.

In the course of improving the computational speed of MaxQuant, we also tested an I/O optimized high end desktop PC. Surprisingly, this configuration performed essentially as well as the large cluster, at a small fraction of the costs and with much less administration overhead. Therefore, our recommendation at this point is to invest in this or similar configuration for laboratories or facilities with medium to large data production. Close to 1000 raw files can still be efficiently processed in the standard workflow in a matter of a few days.

What do these findings imply for potential bottlenecks in the computational analysis of deep proteome data? As we have shown here, current data sets can easily be handled on relatively inexpensive hardware. For the future, both the power of computational hardware and the size of the data acquired in proteomic investigations will increase. For instance, the number of MS and MS/MS scans used in standard acquisitions could increase several fold over the next few years, just as it has over the last several years. Countering this additional computational load, current desktop chips with 12 virtual cores already exist, rather than the 8 cores employed here, and chips with 16 cores are to be released shortly. Similarly, after initial submission of this manuscript, we have installed two rack mounted solutions with 64 logical processors, from Intel and AMD, respectively (see Experimental Methods). Both systems are essentially as easily administered as PCs, but combine improvements due to fast and local memory with increased number of computation units, and are still quite economical. In our initial tests, they performed equally well to the I/O optimized PCs on small numbers of files but were able to handle larger files sets without slow down.

Based on these trends we expect that the computational demands of the standard workflow for in depth shotgun proteomics can be comfortably handled for the foreseeable future. However, specialized tasks, such as searches in six frame translations of large genomes, and other extremely computing intensive tasks may benefit from large clusters.

We applied the improvements in software and hardware to investigate the incremental contribution to coverage of the human genome from large-scale data sets generated in our laboratory. This revealed that there is a large overlap in the identity of proteins in different cell line proteomes as well as an in-depth measured human tissue proteome, consistent with earlier findings. Together, the analysis of more than 1000 raw files identified more than 13,000 different protein groups, mapping to more than 12,000 of the roughly human 20,000 protein coding genes. Interestingly, this depth could be reached with a small subset of the raw MS data, namely the ones using the latest technology. In contrast, hundreds of raw files obtained with a workflow from just a few years ago made essentially no contribution to total identifications. The implications for current efforts to map the entire proteome would be to focus on technology development for in-depth measurements rather than predominantly on accumulation of large numbers of data sets.

ASSOCIATED CONTENT

Supporting Information

Supplementary tables. This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author

*E-mail: mmann@biochem.mpg.de. Phone: +49 (89) 8578 - 0. Fax: +49 (89) 8578 - 37 77.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We thank our colleagues at the Max Planck Institute of Biochemistry for help and fruitful discussions, in particular excellent hardware and configuration assistance by Mario Oroshi. We thank Martin Hoffmann and Bernhard Busch of our institute's computing center for help with the compute cluster. The research leading to these results has received funding from the European Commission's seventh Framework Programme (grant agreement HEALTH-F4-2008-201648/PROSPECTS).

ABBREVIATIONS:

MS, mass spectrometry; MS/MS, tandem mass spectrometry; PC, personal computer; HPC, high performance computing; HDD, hard disk drive; SSD, solid state disk; GPFS, general parallel file system; FDR, false discovery rate; ATA, advanced technology attachment; SATA, serial ATA; SCSI, small computer system interface; SAS, serial attached SCSI; Th, Thomson; PSM, peptide spectrum match; CPU, central processing unit; I/O, input/output

REFERENCES

(1) Aebersold, R.; Mann, M. Mass spectrometry-based proteomics. *Nature* **2003**, *422* (6928), 198−207.

(2) Mallick, P.; Kuster, B. Proteomics: a pragmatic perspective. *Nat. Biotechnol.* **2010**, *28* (7), 695−709.

(3) Cox, J.; Mann, M. Quantitative, high-resolution proteomics for data-driven systems biology. *Annu. Rev. Biochem.* **2011**, *80*, 273−99.

(4) Altelaar, A. F.; Munoz, J.; Heck, A. J. Next-generation proteomics: towards an integrative view of proteome dynamics. *Nat. Rev. Genet.* **2012**, *14* (1), 35−48.

(5) Michalski, A.; Damoc, E.; Hauschild, J. P.; Lange, O.; Wieghaus, A.; Makarov, A.; Nagaraj, N.; Cox, J.; Mann, M.; Horning, S. Mass spectrometry-based proteomics using Q Exactive, a high-performance benchtop quadrupole Orbitrap mass spectrometer. *Mol. Cell. Proteomics* **2011**, *10* (9), M111 011015.

(6) MacCoss, M. J. Computational analysis of shotgun proteomics data. *Curr. Opin. Chem. Biol.* **2005**, *9* (1), 88−94.

(7) Mueller, L. N.; Brusniak, M. Y.; Mani, D. R.; Aebersold, R. An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data. *J. Proteome Res.* **2008**, *7* (1), 51−61.

(8) Galvez, S.; Diaz, D.; Hernandez, P.; Esteban, F. J.; Caballero, J. A.; Dorado, G. Next-generation bioinformatics: using many-core processor architecture to develop a web service for sequence alignment. *Bioinformatics* **2010**, *26* (5), 683−6.

(9) Perkins, D. N.; Pappin, D. J.; Creasy, D. M.; Cottrell, J. S. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **1999**, *20* (18), 3551−67.

(10) Keller, A.; Eng, J.; Zhang, N.; Li, X. J.; Aebersold, R. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol. Syst. Biol.* **2005**, *1*, 2005 0017.

(11) Kohlbacher, O.; Reinert, K.; Gropl, C.; Lange, E.; Pfeifer, N.; Schulz-Trieglaff, O.; Sturm, M. TOPP–the OpenMS proteomics pipeline. *Bioinformatics* **2007**, *23* (2), e191−7.

(12) MacLean, B.; Tomazela, D. M.; Shulman, N.; Chambers, M.; Finney, G. L.; Frewen, B.; Kern, R.; Tabb, D. L.; Liebler, D. C.; MacCoss, M. J. Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* **2010**, *26* (7), 966−8.

(13) Cox, J.; Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.* **2008**, *26* (12), 1367−72.

(14) Cox, J.; Matic, I.; Hilger, M.; Nagaraj, N.; Selbach, M.; Olsen, J. V.; Mann, M. A practical guide to the MaxQuant computational platform for SILAC-based quantitative proteomics. *Nat. Protoc.* **2009**, *4* (5), 698−705.

(15) Cox, J.; Neuhauser, N.; Michalski, A.; Scheltema, R. A.; Olsen, J. V.; Mann, M. Andromeda: a peptide search engine integrated into the MaxQuant environment. *J. Proteome Res.* **2011**, *10* (4), 1794−805.

(16) Cox, J.; Mann, M. 1D and 2D annotation enrichment: a statistical method integrating quantitative proteomics with complementary high-throughput data. *BMC Bioinform.* **2012**, *13* (Suppl 16), S12.

(17) Geiger, T.; Wehner, A.; Schaab, C.; Cox, J.; Mann, M. Comparative proteomic analysis of eleven common cell lines reveals ubiquitous but varying expression of most proteins. *Mol. Cell. Proteomics* **2012**, *11* (3), M111 014050.

(18) Geiger, T.; Madden, S. F.; Gallagher, W. M.; Cox, J.; Mann, M. Proteomic portrait of human breast cancer progression identifies novel prognostic markers. *Cancer Res.* **2012**, *72* (9), 2428−39.

(19) Wisniewski, J. R.; Ostasiewicz, P.; Dus, K.; Zielinska, D. F.; Gnad, F.; Mann, M. Extensive quantitative remodeling of the proteome between normal colon tissue and adenocarcinoma. *Mol Syst Biol* **2012**, *8*, 611.

(20) Wisniewski, J. R.; Dus, K.; Mann, M. Proteomic workflow for analysis of archival formalin fixed and paraffin embedded clinical samples to a depth of 10,000 proteins. *Proteomics Clin. Appl.* **2013**, *7* (3−4), 225−33.

(21) Nagaraj, N.; Mann, M. Quantitative analysis of the intra- and inter-individual variability of the normal urinary proteome. *J. Proteome Res.* **2011**, *10* (2), 637−45.

(22) UniProt Consortium. Reorganizing the protein space at the Universal Protein Resource (UniProt). *Nucleic Acids Res.* **2012**, *40* (Database issue), D71−5.

(23) Mohammed, Y.; Shahand, S.; Korkhov, V.; Luyf, A. C. M.; van Schaik, B. D. C.; Caan, M. W. A.; van Kampen, A. H. C.; Palmblad, M.; Olabarriaga, S. D. In *Data Decomposition in Biomedical e-Science Applications*, e-Science Workshops (eScienceW), 2011 IEEE Seventh International Conference on, 5−8 Dec. 2011; 2011; pp 158−65.

(24) Cox, J.; Michalski, A.; Mann, M. Software lock mass by two-dimensional minimization of peptide mass errors. *J. Am. Soc. Mass Spectrom.* **2011**, *22* (8), 1373−80.

(25) Schaab, C.; Geiger, T.; Stoehr, G.; Cox, J.; Mann, M. Analysis of high accuracy, quantitative proteomics data in the MaxQB database. *Mol. Cell. Proteomics* **2012**, *11* (3), M111 014068.

(26) Clamp, M.; Fry, B.; Kamal, M.; Xie, X.; Cuff, J.; Lin, M. F.; Kellis, M.; Lindblad-Toh, K.; Lander, E. S. Distinguishing protein-coding and noncoding genes in the human genome. *Proc. Natl. Acad. Sci. U.S.A.* **2007**, *104* (49), 19428−33.

(27) Paik, Y. K.; Jeong, S. K.; Omenn, G. S.; Uhlen, M.; Hanash, S.; Cho, S. Y.; Lee, H. J.; Na, K.; Choi, E. Y.; Yan, F.; Zhang, F.; Zhang, Y.; Snyder, M.; Cheng, Y.; Chen, R.; Marko-Varga, G.; Deutsch, E. W.; Kim, H.; Kwon, J. Y.; Aebersold, R.; Bairoch, A.; Taylor, A. D.; Kim, K. Y.; Lee, E. Y.; Hochstrasser, D.; Legrain, P.; Hancock, W. S. The Chromosome-Centric Human Proteome Project for cataloging proteins encoded in the genome. *Nat. Biotechnol.* **2012**, *30* (3), 221−3.

(28) Legrain, P.; Aebersold, R.; Archakov, A.; Bairoch, A.; Bala, K.; Beretta, L.; Bergeron, J.; Borchers, C.; Corthals, G. L.; Costello, C. E.; Deutsch, E. W.; Domon, B.; Hancock, W.; He, F.; Hochstrasser, D.; Marko-Varga, G.; Salekdeh, G. H.; Sechi, S.; Snyder, M.; Srivastava, S.; Uhlen, M.; Hu, C. H.; Yamamoto, T.; Paik, Y. K.; Omenn, G. S. The human proteome project: Current state and future direction. *Mol. Cell. Proteomics* **2011**, *10* (7), No. M111.009993.

(29) Nagaraj, N.; Wisniewski, J. R.; Geiger, T.; Cox, J.; Kircher, M.; Kelso, J.; Paabo, S.; Mann, M. Deep proteome and transcriptome mapping of a human cancer cell line. *Mol. Syst. Biol.* **2011**, *7*, 548.

(30) Beck, M.; Schmidt, A.; Malmstroem, J.; Claassen, M.; Ori, A.; Szymborska, A.; Herzog, F.; Rinner, O.; Ellenberg, J.; Aebersold, R. The quantitative proteome of a human cell line. *Mol. Syst. Biol.* **2011**, *7*, 549.

(31) Munoz, J.; Low, T. Y.; Kok, Y. J.; Chin, A.; Frese, C. K.; Ding, V.; Choo, A.; Heck, A. J. The quantitative proteomes of human-induced pluripotent stem cells and embryonic stem cells. *Mol. Syst. Biol.* **2011**, *7*, 550.

(32) Lundberg, E.; Fagerberg, L.; Klevebring, D.; Matic, I.; Geiger, T.; Cox, J.; Algenas, C.; Lundeberg, J.; Mann, M.; Uhlen, M. Defining the transcriptome and proteome in three functionally different human cell lines. *Mol. Syst. Biol.* **2010**, *6*, 450.