BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 14 2012, pages 1895–1901
doi:10.1093/bioinformatics/bts274

Systems biology

Advance Access publication May 10, 2012

Exhaustive database searching for amino acid mutations in proteomes

Doug Hyatt[1] and Chongle Pan[1,2,*]

[1]BioSciences Division and [2]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Amino acid mutations in proteins can be found by searching tandem mass spectra acquired in shotgun proteomics experiments against protein sequences predicted from genomes. Traditionally, unconstrained searches for amino acid mutations have been accomplished by using a sequence tagging approach that combines *de novo* sequencing with database searching. However, this approach is limited by the performance of *de novo* sequencing.

Results: The Sipros algorithm v2.0 was developed to perform unconstrained database searching using high-resolution tandem mass spectra by exhaustively enumerating all single non-isobaric mutations for every residue in a protein database. The performance of Sipros for amino acid mutation identification exceeded that of an established sequence tagging algorithm, Inspect, based on benchmarking results from a *Rhodopseudomonas palustris* proteomics dataset. To demonstrate the viability of the algorithm for meta-proteomics, Sipros was used to identify amino acid mutations in a natural microbial community in acid mine drainage.

Availability: The Sipros algorithm is freely available at http://code.google.com/p/sipros.

Contact: panc@ornl.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online

Received on January 12, 2012; revised on March 13, 2012; accepted on May 1, 2012

1 INTRODUCTION

Microbial communities drive biological processes in many natural environments, such as the rhizosphere (Berg and Smalla 2009), the human gut (Zoetendal *et al*., 2006) and acid mine drainage (AMD, Denef *et al*., 2010). In recent years, meta-proteomics has emerged as a culture-independent approach for studying the activities of microbial communities in their natural environments (Belnap *et al*. 2010; Wilmes and Bond 2006). Generally, the proteomes of a community are extracted from field samples of interest and measured using shotgun proteomics. Proteins are identified by matching tandem mass spectra to a protein database using standard database searching algorithms (Eng *et al*., 2011). The protein database is typically compiled from the meta-genomic sequences of the community. Although this approach can effectively identify peptides whose sequences are represented in the protein database, it is unable to identify peptides with amino acid mutations which differ slightly from the original sequences in the database. Characterization of amino acid mutations in meta-proteome samples will not only increase the number of identified peptides and proteins but also capture sequence polymorphisms of proteins across environmental samples. Because a microbial species in a community is generally comprised of multiple strains, and because different strains often have significant sequence polymorphisms between homologous proteins, identification of amino acid mutations informs the selection of certain strains or certain variants of a protein under different conditions of an ecosystem.

Identification of amino acid mutations is of interest for many proteomics studies. The existing methods can be classified into the following two broad categories. (i) Database-searching-based methods. This approach was first demonstrated by identification of amino acid mutations in six variants of isolated human hemoglobin using the SEQUEST-SNP algorithm (Gatlin *et al*., 2000). A total of 629 mutations were identified in a breast tumor proteome (Bunger *et al*., 2007) by searching mass spectral data against a customized protein database that contained a list of potential mutations inferred from the NCBI single-nucleotide polymorphisms database, dbSNP (Sherry *et al*., 2001). The X!Tandem algorithm can identify a subset of mutations with positive scores in the PAM (Point Accepted Mutation) substitution matrix from a targeted set of proteins that are identified in an unmodified form by a preceding search (Craig and Beavis, 2003). Because amino acid mutations can be considered as post-translation modifications, all standard database searching algorithms can be used to search for a small number of mutations. However, because there are 18 non-isobaric mutations (i.e. considering mutations to I or L as one mutation) for every residue in the protein database, all the existing methods based on database searching need to constrain their search space by limiting the number of proteins and/or the number of mutations. These constraints are necessary to reduce the running time of the search and to maintain a low false discovery rate (FDR), but, as a result of these constraints, they likely exclude many true amino acid mutations from consideration. (ii) Sequence-tagging-based methods. Sequence tags predicted by *de novo* sequencing algorithms can be used to dramatically narrow down the search space without using the pre-imposed constraints needed for database-searching-based methods. Example algorithms include GutenTag (Tabb *et al*., 2003), Inspect (Tanner *et al*., 2005), Paragon (Shilov *et al*., 2007), DirecTag (Tabb *et al*., 2008) and Vonode (Pan *et al*., 2010). The TagRecon algorithm was developed to leverage sequence tags to identify amino acid mutations in proteomic datasets (Dasari *et al*., 2010). Although much improved in recent years, *de novo* sequencing algorithms still trail behind database searching algorithms in terms of the number

[*]To whom correspondence should be addressed.

of identified spectra and the accuracy of identification (Lu *et al.*, 2009; Pan *et al.*, 2010). The sequence-tagging-based algorithms are dependent on obtaining correct sequence tags from *de novo* sequencing.

In this study, we developed a new strategy for amino acid mutation identification. Experimentally, we acquired high-resolution Orbitrap tandem mass spectra from collision-induced dissociation (CID) for the purpose of amino acid mutation identification. Computationally, we developed an algorithm, Sipros v2.0, to perform an unconstrained database search that exhaustively enumerates and scores all single non-isobaric mutations for every residue in a protein database. The performance of Sipros was compared with an established sequence tagging algorithm, Inspect, using an *Rhodopseudomonas palustris* dataset. The new approach was used to identify amino acid mutations from a natural community in AMD. Previously, Sipros v1.0 was used to determine the stable isotope enrichment level of partially labeled peptides for proteomic stable isotope probing (Pan *et al.*, 2011).

2 SYSTEM AND METHODS

Proteome sample preparation: proteome samples were extracted from an *R.palustris* cell culture (Pan *et al.*, 2008) and an AMD biofilm sample (Ram *et al.*, 2005) as described previously. Briefly, whole-cell lysates were treated with 6 M guanidine and 10 mM dithiothreitol (DTT) (Sigma Chemical Co. St. Louis, MO, USA) at 60°C for 1 h to denature proteins and reduce disulfide bonds. Proteins were digested into peptides by overnight trypsin digestion at 37°C. A second round of trypsin digestion with additional enzyme was performed for 5 h at 37°C to achieve more complete digestion. Obtained tryptic peptides were reduced with 20 mM DTT for 1 h at 60°C and purified using C18 solid-phase extraction (Sep-Pak Plus, Waters, Milford, MA, USA).

2D-LC-MS/MS measurements: proteome samples were analyzed as described previously (Pan *et al.*, 2011). Briefly, samples were loaded offline using a pressure cell onto a strong cation exchange (SCX) column (5 cm long and 250 μm I.D.), which was then connected to an analytical C18 reverse-phase (RP) PicoFrit column (15 cm long and 150 μm I.D.) (New Objective, Woburn, MA, USA). Online 2D LC separation was performed at a flow rate of 250 nl/min using 12 salt pulses for SCX separation and 2-h continuous gradient elution for RP separation after every salt pulse. Tandem mass spectra of eluted peptides were measured using an LTQ Orbitrap XL mass spectrometer (Thermo Fisher Scientific, San Jose, CA, USA) with the following parameters: electrospray ionization with a voltage of 4 kV; 3 data-dependent MS/MS scans for every full scan; MS/MS scans acquired in Orbitrap at resolution 7500 with two-microscan averaging; full scans acquired in Orbitrap at resolution 35 000 with two-microscan averaging; 35% normalized energy and ±1.5 Da isolation window for CID; dynamic exclusion enabled with ±1.5 Da exclusion window.

Data analysis: acquired MS/MS datasets were converted from the Xcalibur Raw file format to the FT2 flat file format using the Raxport program (freely available from http://code.google.com/p/raxport/). The datasets were searched against the *R.palustris* protein database (Larimer *et al.*, 2004) and the AMD protein database (Ram *et al.*, 2005) using the Sipros algorithm on a 512-core Linux cluster (AMD Opteron processor, RAID protected NFS cluster storage, and InfiniBand data network). Reverse sequences of proteins were added to the databases to estimate forward/reverse FDR as described (Peng *et al.*, 2003). For data analysis using Inspect, MS/MS datasets were converted from the Xcalibur Raw file format to the mzXML format using the ReAdW program (Pedrioli *et al.*, 2004). The *R.palustris* dataset was searched against the concatenated forward/reverse protein database using Inspect v20100804 on a Linux workstation. Inspect was configured to process high-resolution tandem mass spectra. All amino acid mutations were explicitly provided to Inspect as potential modifications to all amino acid types. A peptide can have up to one modification.

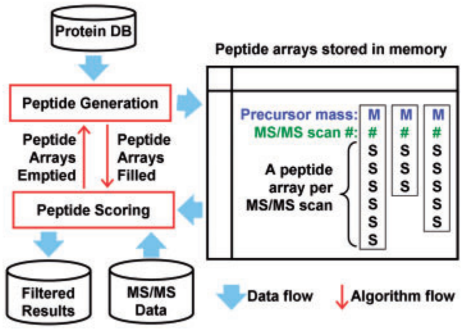

Fig. 1. Architecture of the Sipros algorithm. The algorithm iterates between the peptide generation module and the peptide scoring module as shown by the red arrows. The blue arrows indicate the input and output of the two modules. The peptide generation module predicts candidate peptides from the protein database and stores them in peptide arrays of appropriate MS/MS scans. Once the peptide arrays are filled, the peptide scoring module reads in the MS/MS data, evaluates candidate peptides for every spectrum, and prints out the results. After this, the peptide arrays are emptied to free up memory and the peptide generation module resumes

3 ALGORITHM

Sipros architecture: we designed a new database searching architecture to optimize both CPU efficiency and memory efficiency. Existing database searching algorithms generally have two nested loops, one looping through every spectrum and the other one looping through every candidate peptide. Inside the two nested loops is the scoring function that compares each spectrum with each peptide. For example, the SEQUEST algorithm processes one spectrum at a time and performs *in silico* digestion to generate peptides for every spectrum (Eng *et al.*, 1994). Although this architecture is simple and memory efficient, it wastes CPU cycles by repeating the same *in silico* digestion for every spectrum. In an alternative architecture used in the DBDigger algorithm (Tabb *et al.*, 2005), spectra are first loaded into memory and every peptide generated from the protein database is compared with all the spectra within its mass tolerance window. In this way, peptide generation only needs to be performed once. However, it is not memory efficient to store in memory a large set of spectra generated from a 24-h 2D-LC-MS/MS run.

Sipros uses a peptide array queuing system as shown in Figure 1. The two core modules are the peptide generation module and the peptide scoring module. Peptide arrays are initialized by loading the precursor masses and the scan numbers of all MS/MS scans into memory. The peptide arrays are filled by the peptide generation module that generates candidate peptides from the protein database and allocates them into appropriate MS/MS scans' peptide arrays according to their precursor masses. The peptide arrays are sorted by precursor masses such that candidate peptides can be quickly allocated into arrays using a simple binary search. For large databases, it is not practical to store all the candidate peptides in memory at once. The total memory size of the peptide arrays is monitored. Once a user-defined memory upper limit is reached (default 300 MB), the algorithm dispatches the current set of peptide arrays to the peptide scoring module. First, the peptide arrays are re-sorted by FT2 filename and MS/MS scan number, such that the MS/MS data can be read from the hard drive sequentially. Then, each MS/MS scan is compared with all the peptides in its array by the scoring function. Peptides that pass a score threshold are written to standard output. Finally, all the peptide arrays are emptied to free up memory and the algorithm then returns to the peptide generation module. The algorithm switches back and forth between the two core modules until the entire protein database is processed. The output from the peptide scoring module is then parsed by the final assembly scripts. The assembly scripts are capable of compiling results from multiple instances of the Sipros program, each of which can be run on a different portion of the protein database for parallel computation.

The Sipros architecture offers a few advantages. First, candidate peptides are generated only once for a given database. This is especially important for amino acid mutation identification and post-translational modification (PTM) identification, because this step may consume a significant amount of CPU cycles. Second, it is more memory efficient to store in memory a large set of candidate peptides, each of which typically contains less than 20 residues, than to store a large set of mass spectra, each of which may contain hundreds of peaks. Mass spectral data can be efficiently accessed periodically from the hard drive by sequential read. Third, memory usage is strictly controlled by setting a maximum memory footprint for the peptide arrays, and by periodically emptying the peptide arrays. A larger memory upper limit will reduce the number of times that the peptide scoring module needs to read mass spectral data.

For simplicity, Sipros itself is a single-thread algorithm. A simple parallelism is implemented using Perl scripts to take advantage of multi-core CPUs and computer clusters. The protein database is subdivided into a specified number of partitions. Each partition is processed by an independent Sipros instance. Because there is no inter-dependency among the Sipros instances, the computation scales linearly with the number of processors. The output files from all instances are collated together in the post-processing stage, which takes an insignificant amount of time relative to the database processing and so does not interfere with the scaling of the algorithm.

Peptide generation module: the peptide generation module enumerates candidate peptides from the protein database using the following rules. First, the proteolysis follows user-defined cleavage specificities for both termini of a peptide. By default trypsin cleaves the peptide bond after residues K and R. Second, up to a certain number of missed cleavage sites are allowed in a peptide. By default a peptide can contain up to two missed cleavage sites. Third, only up to one amino acid mutation is considered in a peptide. For convenience, the isobaric amino acids I and L are considered as one amino acid type represented as J. Thus, Sipros considers a total of 19 amino acid types. Amino acid mutations complicate the *in silico* digestion task, because mutations from a K or R to a non-K-or-R residue invalidate the cleavage sites and mutations from a non-K-or-R residue to a K or R create new cleavage sites.

Sipros performs amino acid mutation and *in silico* digestion using a series of nested loops. In the outermost loop, the algorithm traverses each residue in the protein database, using that residue as the mutation target. In the second loop, the target residue is considered for the 19 amino acid types, including its original type and the 18 alternative types that it could mutate into. In the third loop, Sipros traverses left until it either reaches the beginning of the protein, or until it reaches a cleavage site and can go no further due to exceeding the number of allowable missed cleavages. While traversing this loop, it tracks the number of missed cleavage sites on the left side of the mutated residue. In the innermost loop, the program traverses right until it either reaches the end of the protein or reaches a cleavage site and can go no further. Missed cleavage sites on the right side of the target residue are considered and added to the missed cleavage site total on the left side to know when the peptide has reached its rightmost boundary. Inside the four nested loops, peptides are generated for consideration.

Once a peptide is generated, Sipros adds it to the peptide arrays of all scans whose precursor mass lies within a user-specified mass window of that peptide's mass. The peptide addition operation is implemented efficiently by sorting all scans' peptide arrays in the ascending order of their precursor masses. Given a peptide's mass, Sipros uses a binary search to locate the scan with the closest precursor mass and then traverses backwards and forwards to the limit of the user-specified mass error. The default mass error tolerance is 0.04 Da. In testing Sipros, it was discovered that the precursor mass of an MS/MS scan of a peptide is often off from the peptide's mass by 1 Da. In order to address this problem, Sipros allows users to specify a series of discrete mass windows from $M-5$ to $M+5$, where M is the peptide mass. By default, Sipros considers M and $M-1$. The additional mass windows recover many peptide identifications at the expense of increased running time.

Peptide scoring module: the peptide generation module is executed to fill the peptide arrays up to a user-defined memory usage limit. Once the peptide arrays are full, Sipros records the break point in the protein database and switches to the peptide scoring module. First, the MS/MS spectra and their associated peptide arrays are re-sorted by FT2 filename and ascending scan number. Then, Sipros processes one spectrum at a time, reading the spectrum's peak list from the FT2 file and scoring all candidate peptides in the spectrum's peptide array. Because the spectra are stored in the ascending order of their scan number in the FT2 files as well, Sipros can read in the mass spectral data sequentially from the FT2 files as it proceeds through the spectra. Finally, the scoring results are filtered using a score cutoff and printed to standard output. Once all spectra are scored, all peptide arrays are emptied and Sipros resumes the peptide generation module to continue on from the previous break point in the protein database.

Sipros spends the majority of its running time in scoring the match quality of peptide/spectrum pairs. To quickly go through a large number of peptide/spectrum pairs, Sipros employs a fast preliminary scoring function to discard the obvious low-quality peptide/spectrum pairs. The preliminary scoring function finds all the y and b ions of a candidate peptide at a charge state of $+1$ in a MS/MS spectrum using a default mass error tolerance of 0.02 Da. The preliminary score, α, is a sum of every observed product ion's score, $\alpha=\Sigma(\omega_i+\upsilon_i)$, where ω_i is the mass accuracy score of the product ion i and υ_i is the relative intensity of this ion. ω_i is calculated based on the mass error of this ion, ε_i, and the fragment mass error tolerance, Δ (default 0.02 Da), using the formula, $\omega_i=2(1-pnorm(\varepsilon_i, 0, \Delta/2))$(Pan *et al.*, 2010). The function, *pnorm*, calculates the lower tail cumulative probability for the variable ε_i from a normal distribution function with a mean of 0 and a SD of $\Delta/2$. The mass accuracy score, ω_i, has a range of [0.05, 1.00] and increases with the lower measured mass error. υ_i is simply the relative intensity of the fragment ion's peak (relative to the highest peak's intensity) with a range of (0.00, 1.00].

Peptides that pass a preliminary score cutoff (default 12) are then scored with a primary scoring function. The primary scoring function is more computationally expensive and more accurate than the preliminary scoring function. The primary scoring function finds y and b fragment ions at charge states up to the precursor's charge

Table 1. Searching the *R palustris* proteomic dataset against two *in silico* mutated databases

Algorithm	Database mutation rate (%)	Unmutated hits	Correct mutation hits	Incorrect mutation hits	Incorrect peptide hits	Reverse hits
Inspect	2	6918	1518	755	268	38
	5	4075	2248	912	665	79
Sipros	2	6871	2199	48	76	1
	5	3972	2876	62	214	19

state. The primary score is calculated as $\beta=\Sigma(\omega_i\cdot\theta_i\cdot\pi_i)$. ω_i is the mass accuracy score as described as above. θ_i has a value of 2 if this fragment ion has a complementary fragment ion and a value of 1 otherwise. π_i is a quality measure of the isotopic envelope of the fragment ion. π_i has a value of 2.0 if the fragment's major isotopic peaks are observed at expected intensities and a value of 1.0 otherwise. Peptides that pass a primary score cutoff (default 15) are printed to standard output.

Post-processing and protein assembly: the raw output from a Sipros search is simply a large table in which each row represents a peptide/spectrum pair that passes Sipros's internal filtering thresholds. If the search is done by multiple Sipros instances, the output table is stored in multiple files. The peptide/spectrum pairs are not stored in any particular order in the output table. A spectrum may have peptide hits scattered in different sections of an output table. A set of post-processing Perl scripts is used to parse the output tables and assemble the final results. All peptide identifications for a spectrum are grouped together and sorted by their scores. The highest scoring peptide that exceeds a score cutoff is considered as the peptide identification for that spectrum. Peptide identifications are then mapped onto proteins. A protein is identified if it has at least two peptide identifications.

4 RESULTS AND DISCUSSION

4.1 Performance comparison of Sipros and Inspect for amino acid mutation identification

The performance of Sipros and Inspect was benchmarked using high-resolution Orbitrap CID MS/MS data acquired from a whole-cell proteome sample of a bacterium, *R.palustris*. Amino acid mutations in the proteome sample were identified by searching the MS/MS data against a database of 4833 protein sequences predicted from the *R.palustris* genome (Larimer *et al.*, 2004). Reverse sequences of the predicted proteins were appended to the protein database as decoys. All hits matching only to the reverse sequences were assumed to be false. The forward/reverse FDR (F/R FDR) for peptide identification was calculated as twice the percentage of reverse hits out of all hits (Peng *et al.*, 2003). Figure 2 shows the number of forward hits and the F/R FDR at different score cutoffs. More stringent filtering at a higher score cutoff generally improved the F/R FDR of the two algorithms at the expense of a lower number of forward hits identified. Because it is much easier to identify unmutated peptides than mutated peptides, identification of unmutated peptides and mutated peptides were evaluated separately. Both Sipros and Inspect performed very well at identifying unmutated peptides (Fig. 2A). Sipros identified 16 542 unmutated forward hits at an F/R FDR of 0.07% using a score cutoff of 25 (Supplementary Table 1). Inspect identified 14 113 unmutated forward hits at an F/R FDR of 0.14% using a *P*-value cutoff of 0.02.

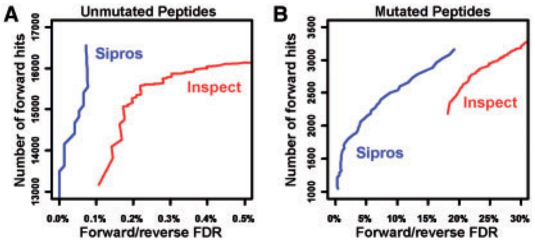

Fig. 2. Comparison of Sipros and Inspect using the *R.palustris* proteomic dataset. (**A**) Identification of unmutated peptides. (**B**) Identification of mutated peptides. Blue curves: Sipros. Red curves: Inspect. Sipros can identify more forward hits than Inspect at a given forward/reverse FDR

Filtering with these cutoffs provided substantially less reliable results for mutated peptide identification. Sipros identified 3157 mutated forward hits at an F/R FDR of 19% using a score cutoff of 25. Inspect identified 2455 mutated forward hits at an F/R FDR of 19% using a *P*-value cutoff of 0.02. The F/D FDR of 19% was clearly too high to be acceptable. Therefore, the score cutoff for Sipros was increased to 40 to obtain an F/R FDR of 1.4%, which yielded 1708 mutated forward hits (Supplementary Table 2). The Sipros score measures the quality of a match between a spectrum and a sequence. Because the sequence search space is much larger for mutated peptides, it was necessary to require a much higher quality of match for mutated peptides than unmutated peptides to obtain reasonable confidence in their identifications (i.e. extraordinary claims require extraordinary evidence). Inspect identified 2183 mutated forward hits at an F/R FDR of 18% using an inclusive *P*-value cutoff of 0. Because the best *P*-value is 0, we cannot perform a more stringent filtering for Inspect to further lower its F/R FDR. Figure 2B shows the superior performance of Sipros for mutated peptide identification, since Sipros can generate more forward hits at a given F/R FDR and better F/R FDR at a given number of forward hits.

F/R FDR is a widely used estimate of the true FDR of peptide identification. The key assumption of F/R FDR is that the number of false forward hits is equal to the number of reverse hits. This assumption is flawed for identification of mutations or PTMs, because it is more likely to add a false modification onto a forward sequence than a reverse sequence as shown below. Furthermore, F/R FDR can be misleading due to the lack of true positive identification

Table 2. Identification of amino acid mutations in an AMD metaproteome

Microorganism	Unique unmutated peptides	Non-unique unmutated peptides	Unique mutated peptides	Non-unique mutated peptides
Lepto III	879	456	156	87
Lepto II UBA	426	1878	55	328
Lepto II 5wayCG	25	1864	31	336
E-plasma	162	59	4	5
Others	148	523	100	98
Reverse	8		65	

control and data overfitting (Barboza *et al.*, 2011; Cooper, 2011). To better benchmark the performance of mutation identification, a set of known mutations was created *in silico* for database searching. Random mutations were introduced to the protein sequences in the predicted *R.palustris* database at two mutation rates, 2% and 5%. 2% of the residues in the 2%-mutated database and 5% in the 5%-mutated database were randomly selected and mutated to another amino acid type. The MS/MS data were searched against the two mutated databases using Sipros and Inspect. Obtained identifications were examined for the set of shared 10 293 spectra that were successfully identified as unmutated forward hits by both Sipros and Inspect in the original database searching. Because both algorithms have very low F/R FDRs for unmutated peptide identification, the peptide identifications from the original database searching were considered to be true positive identifications and were used to check the correctness of mutation identifications from the mutated database searches. Since these spectra were identified by both algorithms, selecting these spectra would not have any bias against either algorithm.

Table 1 lists the numbers of hits from the two mutated databases by the two algorithms in the following five categories. (i) The *unmutated hits* from peptides without any mutation in the mutated databases were identified correctly as such. More unmutated peptides were identified in the 2%-mutated database than the 5%-mutated database as expected. The two algorithms found similar numbers of unmutated peptides. (ii) The *correct mutation hits* changed the mutated peptides in the database back to the correct peptides in the sample. Sipros identified more correct mutation hits than Inspect in both databases. (iii) The *incorrect mutation hits* were identified from the correct peptides, but were not the mutations added to those peptides. The identified peptides in this category were two mutations away from the correct peptides in the sample. The mutation created in the database and the mutation identified by proteomics cancel out each other's mass change. Inspect identified many more incorrect mutation hits than Sipros. Inspect uses *de novo* sequencing to guide the subsequent database searching. We posit that many incorrect mutation hits probably stemmed from almost, but not exactly, correct sequence tags generated by *de novo* sequencing. (iv) The *incorrect peptide hits* were identified to be peptides different from the correct ones in the sample. The two algorithms identified many more incorrect peptide hits in the 5%-mutated database than the 2%-mutated database. Sipros identified less incorrect peptide hits than Inspect in both databases. (v) The *reverse hits* were identified from the reverse sequences in the database. The number of reverse hits was much less than the numbers of incorrect mutation hits and incorrect peptide hits. This supports the proposition that it is much more likely to identify a false modification from the forward sequences than from the reverse sequences and it is misleading to assess the confidence of mutation and PTM identification by F/R FDR alone.

The benchmarking using the *R.palustris* dataset indicated that Sipros can identify more correct mutations at a higher level of accuracy than Inspect. However, Inspect has the following two advantages over Sipros. First, Inspect can identify multiple mutations on a peptide, whereas Sipros is restricted to one mutation per peptide. Second, Inspect used much less computing time than Sipros. Analysis of the *R.palustris* dataset took <10 CPU hours using Inspect and ∼100 CPU hours using Sipros. This reflects the pros and cons of the different approaches used by the two algorithms. Inspect can search a larger sequence space more quickly by using sequence tags to reduce the search space, but it depends on correct sequence tags. To identify mutations by exhaustive searching, Sipros must consider 18 non-isobaric alternatives for every residue in the protein database. The average length of peptides generated from *in silico* digestion of the *R.palustris* protein database is approximately 20 residues. Mutating each residue in these peptides to 18 alternatives would increase the size of the search space ∼360-fold. The vast majority of the added search space is populated by false sequences, which requires a very selective scoring function coupled with high-resolution tandem mass spectral data to control the false discovery rate of identification. The efficient architecture of Sipros allowed searching such a large sequence space using a moderate amount of computing time on a small cluster.

4.2 Identification of amino acid mutations from a microbial community proteome

The new method was used to identify amino acid mutations in a meta-proteome sample of a natural microbial community from AMD. The AMD community is composed of a few dominant bacterial species, including *Leptospirillum* group II (*Lepto* II), *Leptospirillum* group III (*Lepto* III) and a few less abundant archaeal species (A-plasma, G-plasma, E-plasma, etc.). *Lepto* II consists of the 5wayCG strain, the UBA strain, and their recombinants. The abundance of different microorganisms varies depending on environmental conditions. An AMD community dominated by the UBA strain of *Lepto* II was measured by proteomics. High-resolution Orbitrap MS/MS data was searched against a protein database containing 57 001 proteins predicted from the community metagenome. We identified a total of 3842 unmutated peptides from 9804 spectra (Supplementary Table 3). A peptide may be identified by multiple hits from different MS/MS spectra. Table 2 shows the unique and non-unique peptides identified from the major microorganisms. The non-unique peptides can be assigned to multiple microorganisms. As expected, most of the unique unmutated *Lepto* II peptides were identified from the dominant *Lepto* II strain—*Lepto* II UBA. A total of 755 mutated peptides were identified from 1683 spectra (Supplementary Table 4). The mutated peptides are grouped based on the microorganism of the source peptides of Sipros searches in Table 2, but it is possible that these 'mutated' peptides could actually originate from microorganisms not represented in the metagenome database. The frequency of all 360 types of mutations (i.e. mutation of 20 amino acid types to 18 non-isobaric alternative types) is shown in Supplementary Table 5.

Table 3. Recovery of unique peptides of a removed community member by amino acid mutation search

Removed microorganism	Lost unique peptides	Recovered peptides	Source of recovered peptides					Unrecovered peptides	
			Lepto II 5wayCG	*Lepto* II UBA	*Lepto* III	E-plasma	Others	Scores∈[25, 40)	No source
Lepto II UBA	426	150	149	N/A	7	2	22	139	137
Lepto III	879	94	38	35	N/A	0	68	99	686
E-plasma	162	18	0	0	0	N/A	18	25	119

A proteome sample of a natural microbial community may have a new species or a different strain from the members in the sequenced metagenome of the community. Unique peptides from a new member would be missed by regular database searching, but they may be recovered by an amino acid mutation search if they are different from related peptides in the sequenced members by one amino acid mutation. To simulate this scenario, we removed the genome of a dominant microorganism from the AMD protein database and performed database searching using Sipros. The results were shown in Table 3. When *Lepto* II UBA was removed from the protein database, 426 peptides unique to this microorganism were lost in the list of unmutated peptide identifications. Of these peptides, 150 were recovered as mutated peptides by Sipros. Of the 150, 149 recovered peptides originated from a source peptide in the *Lepto* II 5wayCG genome, which was expected because the two strains were closely related. Some of the recovered peptides also originated from other microorganisms. 139 unique *Lepto* II UBA peptides were identified correctly as mutated peptides with a score above 25; however, they failed to pass the more stringent filtering for mutation identification with a higher score cutoff of 40. 137 unique *Lepto* II UBA peptides were not identified because there was no source peptide within one amino acid mutation of these peptides. Simulation results with the removal of *Lepto* III and E-plasma were also shown in Table 3. Most of the unique peptides from these microorganisms were not recovered because there is no closely related microorganism in the database.

There are two caveats for the biological interpretation of identified ‘mutations’. First, these mutations are computational mutations from predicted protein sequences in the database, which do not necessarily correspond to biological mutations that have actually occurred in the real world. In the end Sipros only identifies a peptide based on a spectrum. The mutation and the source microorganism of that peptide are only inferred based on the assumptions of comprehensive database coverage and minimum edit distance. Second, the identified mutations can stem from chemical modifications. The changes of Q to E and N to D can be caused by a common chemical modification, deamidation. Chemical modification, instead of genetic mutation, may explain why these two types of mutations were identified much more frequently than the others (Supplementary Table 5). Other types of mutations could also be attributed to chemical mutations with the same mass change. Additional experiments will be needed to revolve the uncertainty associated with the two caveats. For example, genome re-sequencing (Bentley, 2006) and RNA sequencing (Majewski and Pastinen, 2011; Ozsolak and Milos, 2011) can be used to differentiate genetic mutations from chemical modifications, resolve the I/L ambiguity, and determine the single-nucleotide polymorphism responsible for an amino acid mutation.

ACKNOWLEDGEMENTS

We would like to thank Robert L. Hettich for technical discussions and instrument support, Jillian F. Banfield for technical discussions and the AMD sample, Dale A. Pelletier for the *R.palustris* sample, and Edward C. Uberbacher for helpful discussions and support.

Funding: This work was funded by the US Department of Energy, Office of Biological and Environmental Research, and Office of Advanced Scientific Computing Research SciDAC program. Oak Ridge National Laboratory is managed by University of Tennessee-Battelle LLC for the Department of Energy under contract DOE-AC05-00OR22725.

Conflict of Interest: none declared.

REFERENCES

Barboza,R. *et al.* (2011) Can the false-discovery rate be misleading? *Proteomics*, **11**, 4105–4108.

Belnap,C.P. *et al.* (2010) Cultivation and quantitative proteomic analyses of acidophilic microbial communities. *Isme J.*, **4**, 520–530.

Bentley,D.R. (2006) Whole-genome re-sequencing. *Curr. Opin. Genet. Dev.*, **16**, 545–552.

Berg,G. and Smalla,K. (2009) Plant species and soil type cooperatively shape the structure and function of microbial communities in the rhizosphere. *FEMS Microbiol. Ecol.*, **68**, 1–13.

Bunger,M.K. *et al.* (2007) Detection and validation of non-synonymous coding SNPs from orthogonal analysis of shotgun proteomics data. *J. Proteome Res.*, **6**, 2331–2340.

Cooper,B. (2011) The problem with peptide presumption and low Mascot scoring. *J. Proteome Res.*, **10,** 1432–1435.

Craig,R. and Beavis,R.C. (2003) A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid Commun. Mass Spectrom.*, **17**, 2310–2316.

Dasari,S. *et al.* (2010) TagRecon: high-throughput mutation identification through sequence tagging. *J. Proteome Res.*, **9**, 1716–1726.

Denef,V.J. *et al.* (2010) AMD biofilms: using model communities to study microbial evolution and ecological complexity in nature. *Isme J.*, **4**, 599–610.

Eng,J.K. *et al.* (1994) An approach to correlate tandem mass-spectral data of peptides with amino-acid-sequences in a protein database. *J. Am. Soc. Mass Spectrom.,* **5**, 976–989.

Eng,J.K. *et al.* (2011) A face in the crowd: recognizing peptides through database search. *Mol. Cell Proteomics*, **10**, R111 009522.

Gatlin,C.L. *et al.* (2000) Automated identification of amino acid sequence variations in proteins by HPLC/microspray tandem mass spectrometry. *Anal. Chem.*, **72**, 757–763.

Larimer,F.W. *et al.* (2004) Complete genome sequence of the metabolically versatile photosynthetic bacterium Rhodopseudomonas palustris. *Nat. Biotechnol.*, **22**, 55–61.

Lu,B. *et al.* (2009) Shotgun protein identification and quantification by mass spectrometry. *Methods Mol. Biol.*, **564**, 261–288.

Majewski,J. and Pastinen,T. (2011) The study of eQTL variations by RNA-seq: from SNPs to phenotypes. *Trends Genet.*, **27**, 72–79.

Ozsolak,F. and Milos,P.M. (2011) RNA sequencing: advances, challenges and opportunities. *Nat. Rev. Genet.*, **12**, 87–98.

Pan,C. *et al.* (2008) Characterization of anaerobic catabolism of p-coumarate in Rhodopseudomonas palustris by integrating transcriptomics and quantitative proteomics. *Mol. Cell Proteomics*, **7,** 938–948.

Pan,C. *et al.* (2010) A high-throughput de novo sequencing approach for shotgun proteomics using high-resolution tandem mass spectrometry. *BMC Bioinformatics*, **11,** 118.

Pan,C. *et al.* (2011) Quantitative tracking of isotope flows in proteomes of microbial communities. *Mol. Cell Proteomics*, **10**, M110 006049.

Pedrioli,P.G. *et al.* (2004) A common open representation of mass spectrometry data and its application to proteomics research. *Nat. Biotechnol.*, **22**, 1459–1466.

Peng,J. *et al.* (2003) Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome. *J. Proteome Res.*, **2,** 43–50.

Ram,R.J. *et al.* (2005) Community proteomics of a natural microbial biofilm. *Science*, **308**, 1915–1920.

Sherry,S.T. *et al.* (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, **29**, 308–311.

Shilov,I.V. *et al.* (2007) The Paragon Algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Mol. Cell Proteomics*, **6**, 1638–1655.

Tabb,D.L. *et al.* (2003) GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model. *Anal. Chem.*, **75**, 6415–6421.

Tabb,D.L. *et al.* (2005) DBDigger: reorganized proteomic database identification that improves flexibility and speed. *Anal. Chem.*, **77**, 2464–2474.

Tabb,D.L. *et al.* (2008) DirecTag: accurate sequence tags from peptide MS/MS through statistical scoring. *J. Proteome Res.*, **7**, 3838–3846.

Tanner,S. *et al.* (2005) InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal. Chem.*, **77**, 4626–4639.

Wilmes,P. and Bond,P.L. (2006) Metaproteomics: studying functional gene expression in microbial ecosystems. *Trends Microbiol.,* **14**, 92–97.

Zoetendal,E.G. *et al.* (2006) A microbial world within us. *Mol. Microbiol.*, **59**, 1639–1650.