Journal of Proteomics & Bioinformatics - Open Access

Research Article JPB/Vol.2/May 2009

Ranking Methods for the Prediction of Frequent Top Scoring Peptides from Proteomics Data

Carsten Henneges[1] *, Georg Hinselmann[1], Stephan Jung[2], Johannes Madlung[2], Wolfgang Schütz[2], Alfred Nordheim[2] and Andreas Zell[1]

[1]Wilhelm-Schickard Institute, Eberhardt Karls Universität, Tübingen, Sand 1, 72076 Tübingen, Germany
[2]Proteome Center Tübingen, Eberhardt Karls Universität, Tübingen, Auf der Morgenstelle 15, 72076 Tübingen, Germany

*Corresponding author: Carsten Henneges, Wilhelm-Schickard Institute, Eberhardt Karls Universität, Tübingen, Sand 1, 72076 Tübingen, Germany, E-mail: carsten.henneges@uni-tuebingen.de; Tel: 07071/29 –77175; Fax: 07071/29-5091

Received April 08, 2009; Accepted May 20, 2009; Published May 20, 2009

Citation: Henneges C, Hinselmann G, Jung S, Madlung J, Schütz W, et al. (2009) Ranking Methods for the Prediction of Frequent Top Scoring Peptides from Proteomics Data. J Proteomics Bioinform 2: 226-235. doi:10.4172/jpb.1000081

Abstract

Proteomics facilities accumulate large amounts of proteomics data that are archived for documentation purposes. Since proteomics search engines, e.g. Mascot or Sequest, are used for peptide sequencing resulting in peptide hits that are ranked by a score, we apply ranking algorithms to combine archived search results into predictive models. In this way peptide sequences can be identified that frequently achieve high scores. Using our approach they can be predicted directly from their molecular structure and then be used to support protein identification or perform experiments that require reliable peptide identification.

We prepared all peptide sequences and Mascot scores from a four year period of proteomics experiments on *Homo sapiens* of the Proteome Center Tuebingen for training. To encode the peptides MacroModel and DragonX were used for molecular descriptor computation. All features were ranked by ranking-specific feature selection using the Greedy Search Algorithm to significantly improve the performance of RankNet and FRank. Model evaluation on hold-out test data resulted in a Mean Average Precision up to 0.59 and a Normalized Discounted Cumulative Gain up to 0.81.

Therefore we demonstrate that ranking algorithms can be used for the analysis of long term proteomics data to identify frequently top scoring peptides.

Keywords: Peptide ranking; Feature selection; Preference learning; Mascot; Sequest; RankNet; FRank; Greedy search algorithm; SMILES; Ligprep

Abbreviations

FTICR	Fourier Transform Ion Cyclotron Resonance
GSA	Greedy Search Algorithm
HPLC	High Performance Liquid Chromatography
MAP	Mean Average Precision
MRM	Multiple Reaction Monitoring
MS	Mass Spectrometry
NDCG	Normalized Discounted Cumulative Gain
SMILES	Simplified Molecular Input Line Entry Specification
SVM	Support Vector Machine
XML	Extensive Markup Language

Introduction

Proteomics facilities archive data for documentation purposes and may wonder, whether this data can be used to improve their service. Intuitively, their data contains a lot of information about their experimental quality. The problem is that this data originates from diverse experiments and therefore cannot be analyzed as a whole by classical statistics. Statistical tools for classification and regression are often built on the assumption of independent and identically distributed random variables, which does not hold for this data. However, often all these experiments share the same experimental protocol.

In most proteomics experiments proteins are first digested using a specific protease, e.g. trypsin or LysC. After that the resulting peptides are chromatographically separated by High-Performance-Liquid-Chromatography (HPLC) and then detected using mass spectrometry (MS). Then a database search engine, as Mascot or Sequest (Perkins et al., 1999; Yates et al., 1995), identifies possible peptide sequences that could have produced the obtained spectrum. For this task, the search algorithm compares the acquired spectrum with a set of theoretical spectra that were computed from a sequence database. At least one confidence score for each comparison is computed by the search engine, indicating how likely the theoretical spectrum and its peptide have produced the detected one. Using this score to order peptide sequences, a ranking of peptides is obtained for each database search.

Ranking algorithms like RankNet (Burges et al., 2005) or

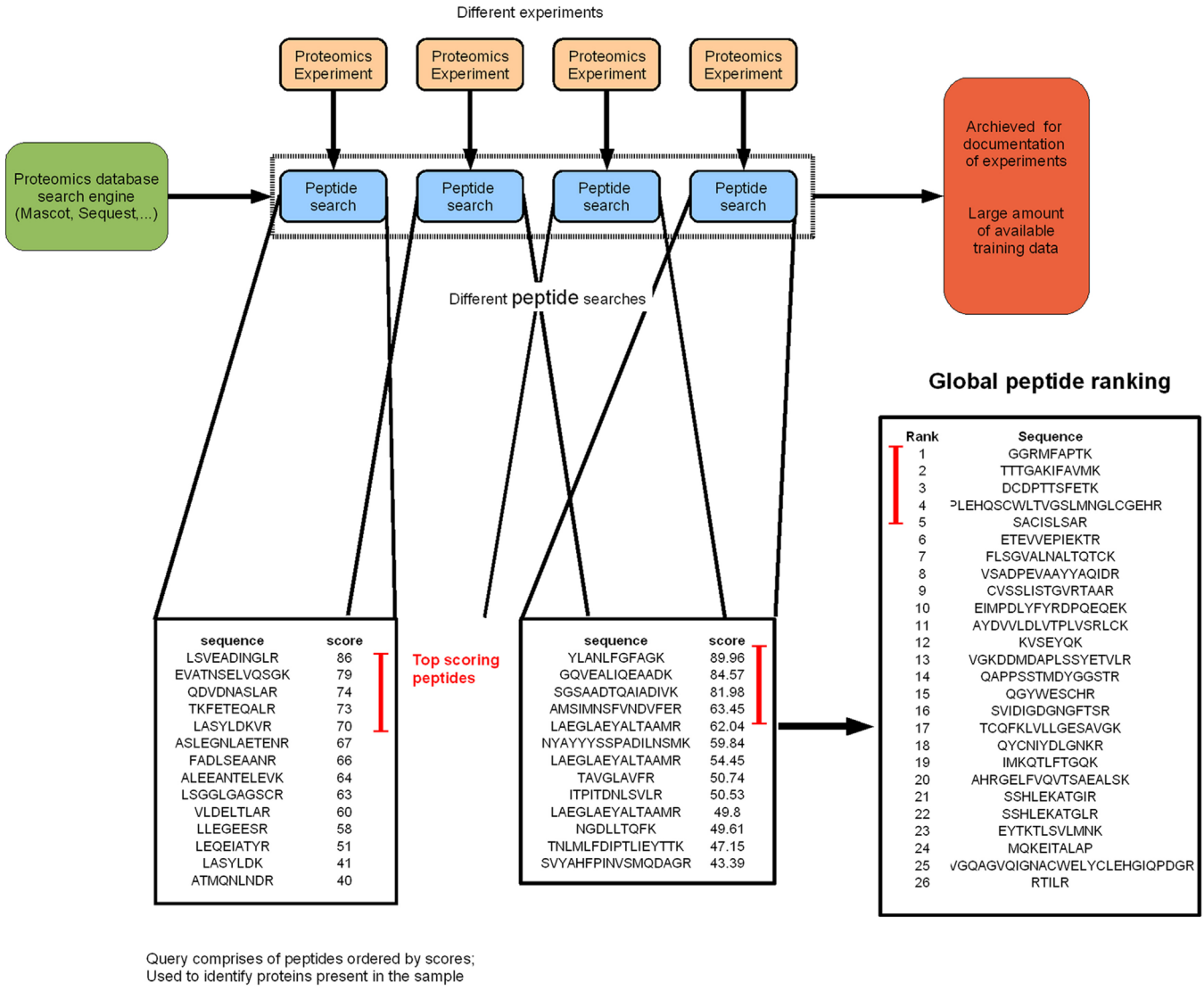

Figure 1: Using ranking algorithms to learn from search engine results

This figure illustrates the idea of our ranking approach to identify frequently top ranking peptides. During operation of a proteomics facility many database searches are performed and then archived. They represent a large data set suitable to generate ranking models for predicting top scoring peptides. To achieve this ranking algorithms construct models that perform best on all available queries.

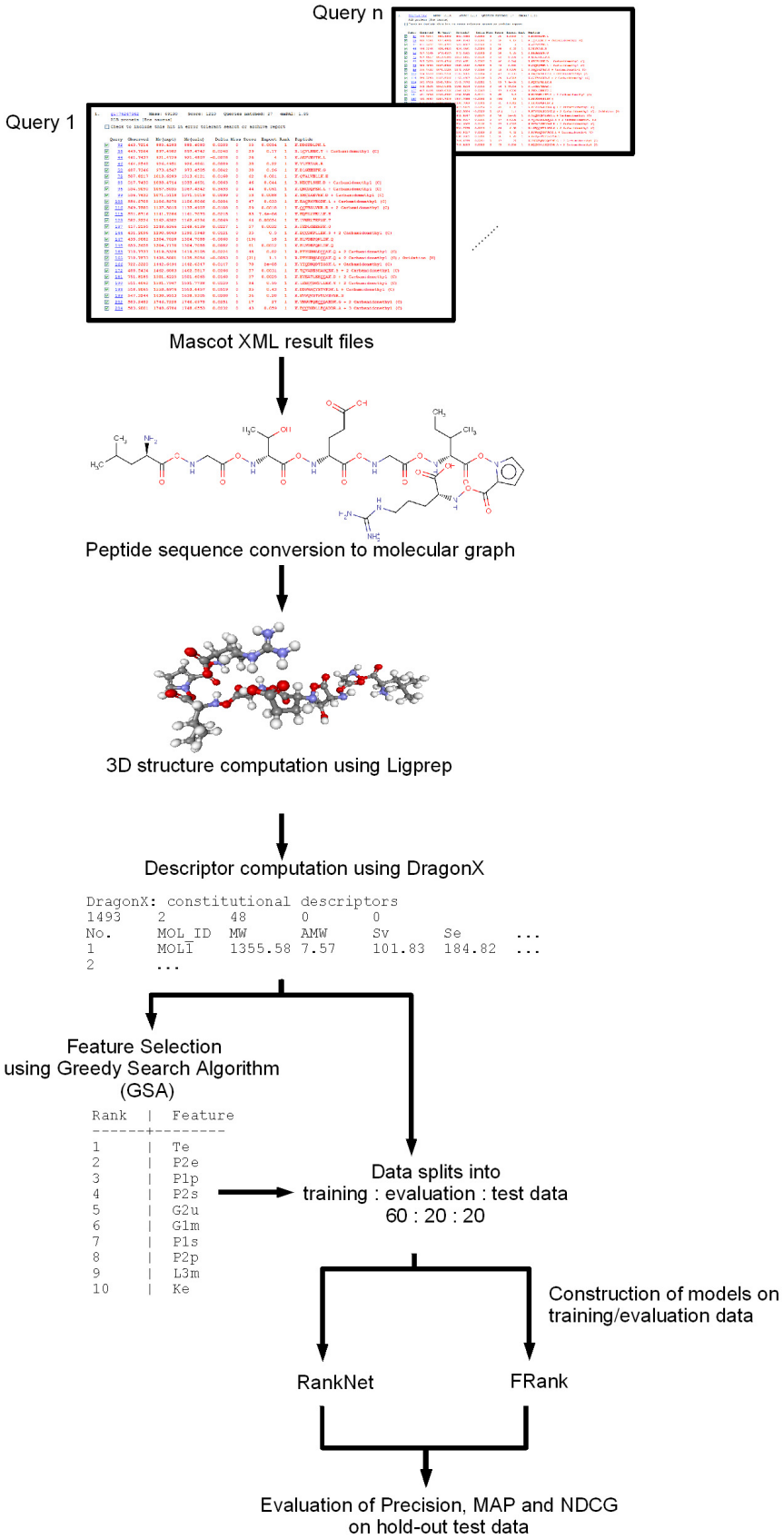

Figure 2: Data set preparation

This graphic illustrates the training workflow. After extraction from the Mascot XML files, peptide sequences are converted into molecular descriptor vectors using SMILES, MacroModel and DragonX. These encodings are combined with the extracted queries for subsequent feature selection. Selected features were used for the training of RankNet and FRank. Finally all generated models were evaluated on a hold-out test data set.

FRank (Tsai et al., 2007) allow to generate predictive models from a collection of rankings. They optimize different loss functions to obtain a ranking model that performs best in reproducing the ordering of all given training input. The input of ranking algorithms consists of a set of rankings, referred to as queries. During training all elements of a query can be directly compared with each other and are examples for the ordering that is learned.

By learning ranking models peptides that frequently achieve top scores can be identified and predicted. These peptides then can be considered as being reliably detectable with high confidence scores. This property was also requested by other prediction approaches as described in other studies (Mallick et al., 2006; Sanders et al., 2007, Webb-Robertson et al., 2008). Since these methods used algorithms for classification, they required training on a curated data set from different facilities, on data obtained from a standardized experiment or on data that was verified by extremely sensitive mass spectrometry. Consequently, the obtained predictors were either not specific for a proteomics facility or required extra costs for calibration and verification experiments. By applying ranking algorithms predictors can be obtained from data that is already available (see Figure 1). These predictive models can help to select those peptides for a focused analysis of proteins, i.e. MRM experiments, or support protein identification within an experimental setting. Also measuring the prediction error over time periods could serve as quality control, since a reliable experimental setup should yield predictable results. To the best of our knowledge ranking algorithms have not been applied for this kind of analysis so far.

In this paper we demonstrate how ranking algorithms can be used to predict top scoring peptides from a long term data set of Mascot (Perkins et al., 1999) queries (see Figure 2). In our approach peptides are encoded as SMILES (Weininger, 1988) and prepared by MacroModel to compute molecular descriptors using DragonX (Tetko et al., 2005). Then we perform ranking specific feature selection using the Greedy Search Algorithm (Geng et al., 2007) and train RankNet (Burges et al., 2005) and FRank (Tsai et al., 2007) models. Finally the generalization performance is tested on hold-out data sets using the ranking specific evaluation measures precision, NDCG and MAP.

Material and Methods

Our goal was the prediction of the search engine ranking directly from the peptide sequences. Therefore, we first extracted the search results from the archived proteomics data, followed by encoding the sequences as molecular graphs. Then feature selection, model training and performance evaluation were carried out. In this order each of these steps is described in the following sections.

Data Extraction

The first step in our ranking experiment was to extract and prepare the training data into queries. This data consisted of mass spectra acquired by NanoHPLC-MS/MS on an Ultimate-LC (Dionex, Idstein, Germany) coupled online to a QStar MS (QStar Pulsar i, Applied Biosystems, Darmstadt, Germany). All experiments were carried out for *Homo sapiens* throughout a four year period by the Proteome Center Tübingen following a similar protocol as described in (Hala et al., 2008). During this time Mascot 2.2 (Perkins et al., 1999) was used to generate the protein identifications. We extracted all database searches encoded as XML files from the Mascot result page using the "Save as XML" option and setting the threshold to 0.0 to avoid filtering. Next, we extracted all peptide sequences together with their corresponding Mascot score from each XML file. Each pair of peptide sequence and score was assigned a unique query identifier to distinguish its origin for the ranking algorithm.

Peptide Encoding

For training ranking predictors each peptide sequence was converted into a vector of numerical descriptors representing its molecular graph by DragonX (Tetko et al., 2005). DragonX allows for the computation of 1,664 molecular descriptors for each input structure. These are organized in 20 blocks of different sizes (shown in Table 1) and each block requires a specific level of detail of the input. For example, while the [Constitutional] descriptors count only atom and bond types, the [Geometrical] block requires a refined molecule description including 3D atom coordinates.[1] To compare all the different descriptor blocks, each sequence had to be converted into a 3D representation of the peptide.

This was accomplished by encoding each sequence into a linear Simplified Molecular Input Line Entry Specification (Weininger, 1988) of the molecular graph and then computing atom coordinates using MacroModel from the Schödinger Molecular Modeling Suite. The first step was achieved by replacing each amino acid letter by its substitution SMILES from Table 2. The strings are designed such that a concatenation forms the peptide bond and in this way a valid molecular graph is obtained. Next we applied MacroModel to convert all generated SMILES into the MDL

[1]We embrace DragonX block names by brackets [...].

Descriptor Block	Block Size	Required Information
[Constitutional]	48	0D
[Atom-centered Fragments]	120	1D
[Functional Group]	154	1D
[2D Autocorrelation]	96	2D
[Burden]	64	2D
[Connectivity]	33	2D
[Edge-adjacency]	107	2D
[Eigenvalue]	44	2D
[Information]	47	2D
[Topological Charge]	21	2D
[Topological]	119	2D
[Walk and Path Counts]	47	2D
[3D MoRSE]	160	3D
[Geometrical]	74	3D
[GETAWAY]	197	3D
[Randic]	41	3D
[RDF]	150	3D
[WHIM]	99	3D
[Charge]	14	Other
[Molecular Properties]	29	Other

Table 1: DragonX descriptor blocks

Overview of DragonX descriptor blocks and the number of corresponding features and dimension of the required information.

SD file format (Dalby et al., 1992) containing appropriate atom coordinates. For this reason MacroModel was run using the OPLS2005 force field to compute energetically optimized atom positions. After that the SD files were processed by DragonX to compute the molecular descriptor blocks encoding the peptides. Using the generated numerical descriptors, we encoded each peptide sequence within each query for feature selection and training.

Feature Selection

To facilitate training and assess the impact of each descriptor, we conducted ranking specific feature selection. We applied a method proposed by Geng et al. (Geng et al., 2007), which is named the Greedy Search Algorithm (GSA). Using this method 10 features from each block were selected, since the [Charge] descriptor block consists only of 14 descriptors. The GSA searches a set of features that is maximal with respect to a scoring function, but also maximizes dissimilarity between the selected features. As scoring function the Mean Average Precision (MAP) with the top 30% of a query to be hits was used. The trade-off parameter c between maximum score and maximal dissimilarity was set to 0.8. To simplify this section, MAP is described in section "performance evaluation".

Training Ranking Models

We compared the established ranking models RankNet (Burges et al., 2005) and FRank (Tsai et al., 2007) to set up a performance baseline.

Memory requirements became also an issue when training on the large set of queries and using hundreds of descriptors. So experiments using RankSVM (Joachims, 2002) failed due to the memory requirements of this method, which prevented loading the complete data set into memory. In contrast to RankSVM, RankNet as well as FRank are online learning algorithms that require only scanning through the data without the need to load it completely. Therefore they are capable of handling large data sets.

Both algorithms are iterative gradient descent methods that require a stopping criterion while processing the training data. We implemented both algorithms to control the generalization performance on a hold-out evaluation data set. If the optimized loss function was not decreased on this data under its minimal value within k subsequent steps, the iteration was stopped. For our experiments we chose k to be 30.

To compute the required training and evaluation data sets as well as a hold-out test data set for the model evaluation, we randomly split our encoded queries into 60% training data, 20% evaluation data and 20% test data. The random splits were computed such that no query was divided during this procedure and the available peptide sequences approximated the 60:20:20 distribution.

The neural network RankNet was trained using the configuration in (Burges et al., 2005) having $m = 10$ hidden neurons for a maximum of 1000 epochs each composed of 100 backpropagation iterations. The step-size parameter η was initially set to 1.0 and was scaled by 1.1 when evaluation performance increased and by 0.9 in the case of a decrease. In this way a significant speed up in training time was achieved. The probability $\overline{P}$ that the trained order is correct was set to 1.0.

The boosting method FRank (Tsai et al., 2007) was run to select at least $t = 10$ binary Weak-Learners. Then, further iterations were allowed until the stopping criterion was fulfilled. FRank was initialized with 50 precomputed Weak-Learners per feature that were obtained by selecting equally distributed thresholds over the feature value range.[2] The probability $\overline{P}$ was set to 1.0 as in the case of RankNet.

[2]This procedure is based on a private communication of CH with the author of FRank.

Because of the many combinations and long training times, which were on the order of days, we trained five models for each combination of descriptor block, feature selection and ranking algorithm. Training was performed on a computing cluster comprising 24 nodes each composed of AMD Opteron 250 dual-core processors having 2.4 GHz and 6 GB RAM. All ranking algorithms (RankNet/FRank) as well as the feature selection method (GSA) were implemented in Java 1.5. For data preparation a set of PERL scripts was developed. Our software is available for free download at www.ra.cs.uni-tuebingen.de/software/.

Performance Evaluation

We evaluated the precision, the Mean Average Precision (MAP) and the Normalized Discounted Cumulative Gain (NDCG) as ranking-specific performance measures. The NDCG was computed according to (Järvelin and Kekäläinen, 2002) using $b = 2$ as basis function. Using a gain function, the NDCG measures the fraction of the optimal gain, which is normalized to 1.0, obtained when examining the query at a specific position. Since the queries have different length this position was specified relative to the top of a query. We evaluated the NDCG at the 10%, 20%, 30%, 40% and 50% position of each hold-out test query. Additionally to a specified position in a query, the precision requires also the definition of hits. Hits are the samples of a query that are considered relevant and therefore should be ranked at the top. Similarly to the NDCG positions, we defined the hits to be the top 10%, 20%, 30%, 40% and 50% of a query. Using these position and hit specifications we evaluated the precision at position n of a query, which is defined as

$$P@n = \frac{|\text{hits within top } n|}{n} \quad (1)$$

The precision reports the fraction of hits, defined as the top fraction of a query, within the top fraction of the predicted ranking. Thus for the precision only positions within the range of hits were considered. Based on the precision the most commonly used performance measure for ranking is the MAP. It averages all precisions with respect to hits and is defined as

$$MAP = \frac{1}{|\text{number of hits}|} \sum_{\text{hit at position } i} P@i \quad (2)$$

MAP is a measure of the spreading of hits within the prediction.

One Letter Code	Substitution SMILES	Amino Acid Name
A	N[C@@H](C)C(=O)O	Alanine
V	N[C@@H](C(C)C)C(=O)O	Valine
F	N[C@@H](CC1=CC=CC=C1)C(=O)O	Phenylalanine
I	N[C@@H]([C@@H](C)CC)C(=O)O	Isoleucine
L	N[C@@H](CC(C)C)C(=O)O	Leucine
P	N1CCC[C@H]1C(=O)O	Proline
M	N[C@@H](CCSC)C(=O)O	Methionine
D	N[C@@H](CC(O)=O)C(=O)O	Aspartic acid
E	N[C@@H](CCC(O)=O)C(=O)O	Glutamic acid
K	N[C@@H](CCCCN)C(=O)O	Lysine
R	N[C@@H](CCCNC(N)=N)C(=O)O	Arginine
S	N[C@@H](CO)C(=O)O	Serine
T	N[C@@H]([C@H](O)C)C(=O)O	Threonine
Y	N[C@@H](CC1=CC=C(C=C1)O)C(=O)O	Tyrosine
C	N[C@@H](CS)C(=O)O	Cysteine
N	N[C@@H](CC(N)=O)C(=O)O	Asparagine
Q	N[C@@H](CCC(N)=O)C(=O)O	Glutamine
H	N[C@@H](CC1=CNC=N1)C(=O)O	Histidine
W	N[C@@H](CC1=CNC2=CC=CC=C12)C(=O)O	Tryptophan
G	NCC(O)=O	Glycine

Table 2: Substitution table

Substitution table used to convert one letter code into valid SMILES for MacroModel preparation. Each substitution SMILES are designed such that a concatenation yields a valid peptide molecular graph. Extending this table could facilitate the descriptor generation for modified peptides, e.g. phosphorylations.

Since every ranking method has to deal with the problem of ties, which are samples assigned to the same rank in a query or prediction. To solve this problem we averaged each measure 100 times for randomized orderings of each tie. Finally we averaged all five training replicates on the different test data sets to obtain the final ranking performance.

Results and Discussion

The aim of this study was to apply ranking methods to identify and predict frequently top scoring peptides for mass spectrometry based proteomics (see Figures 1 and 2). This was achieved by training the ranking models RankNet and FRank on proteomics search engine rankings. To optimize prediction performance we also conducted ranking-specific feature selection using the GSA algorithm.

We used a *Homo sapiens* experimental dataset and extracted 9,967 peptides as well as the corresponding search engine scores from 915 Mascot XML files containing search results to form queries. Each peptide sequence was converted into a numerical descriptor vector using DragonX (Table 1). This step required the computation of atom coordinates by MacroModel and the conversion of peptide sequences into a molecular graph representation. A flexible method for the latter task was using the substitution Table 2. To reduce computational effort long peptides were filtered out by Ligprep, a wrapper for MacroModel, removing 20% of all peptides and resulting in 7,973 distinct sequences for the remaining experiments. This filtering does only minimally affect our training because long peptide sequences achieve in most cases low Mascot scores due to the limited mass detection range of the QStar MS (m/z < 900).

Next we performed ranking specific feature selection using the GSA algorithm. The results are summarized in Table 3 in which only those blocks are listed, for which subsequently trained ranking models with an optimal generalization error for any configuration of feature selection, ranking algorithm and performance measure were obtained. The column "block usage" counts how many optimal models are based on a specific descriptor block. In this statistic each model of RankNet and FRank that was either trained with or without feature selection is considered. Descriptors that are related to electro-chemistry and polarizability are written in bold font, because in former studies these properties were considered to be relevant for mass spectrometry (Mallick et al., 2006; Sanders et al., 2007; Webb-Robertson et al., 2008).

As listed in Table 3 our results show, that three descriptor blocks performed best to train optimal models as indicated by the number in the block usage column. (1) The [Edge-adjacency] descriptor block was best to train optimal models regardless of ranking method and feature selection (block usage 35). This block was followed by (2) the [Topological] block and the (3) the [Functional Group] block with a block usage of 25 and 18, respectively. The [Edge-adjacency] block consists of descriptors that are computed from the adjacency matrix and therefore encode the topology, molecular constitution and connectivity, and they also in-

		Selection Rank									
Descriptor Block	**Block Usage**	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	**10.**
[Edge-adjacency]	35	EEig15r	EEig01r	EEig03r	EEig11r	EEig02x	**ESpm01d**	EEig01x	**EEig08d**	EEig02r	EEig12r
[Topological]	25	**T(O..O)**	Xt	SNar	BLI	TI1	**Jhetp**	MSD	STN	**Jhete**	PW2
[Functional Group]	18	nCp	**nCar**	nCs	nCt	nCq	nCrs	nCrt	nCrq	**nCbH**	**nCb-**
[Charge]	6	**LDI**	**qpmax**	**qnmax**	**Qpos**	**Qneg**	**Qtot**	**Qmean**	**Q2**	**RPCG**	**RNCG**
[Geometrical]*	4	**G(O..O)**	**RCI**	J3D	FDI	**HOMT**	DISPm	PJI3	**QXXp**	SPAM	H3D
[Topological Charge]	4	**GGI1**	**JGI3**	**JGI4**	**JGI5**	**JGI8**	**JGI7**	**GGI9**	**JGI2**	**GGI10**	**JGI1**
[WHIM]*	4	Tm	**P2e**	**Gs**	**P1p**	P2s	L3v	P1s	**P2p**	P1u	G1u
[Atom-centered Fragments]	1	**N-067**	C-024	C-003	**O-062**	H-047	H-046	H-051	H-052	C-001	H-050
[Randic]	1	DP01	SHP2	SP01	SP20	DP20	DP02	DP09	SP02	SP19	DP03

Table 3: Feature selection results

This table shows the selected features from each DragonX descriptor block. Only blocks that were used by best-performing models for at least one evaluation measure are listed. The values in the block usage column count how often a descriptor block is used within Table 4 indicating its importance for model performance. Electro-physical properties are marked by **bold font**. Asterisks (*) indicates 3D descriptor blocks.

clude information about the molecular dipole moments. The [Topological] block provides information about quantifying molecular topology, which is based on the molecular graph representation encoding chemical information about atom types or bond multiplicity. These features have been designed to be sensitive to size, shape, symmetry, branching as well as cyclicity. Finally, the [Functional Group] contains defined molecule fragments together with information about their hybridization state.

A closer look at the selected features reveals that ranking methods require two kinds of descriptors for good perfor-

(4a) RankNet without feature selection						
Position/Relevant	**10%**	**20%**	**30%**	**40%**	**50%**	**NDCG**
10%	0.2607	0.3981	0.4935	0.5645	0.6283	0.6344
20%		**0.3700**	0.4681	0.5398	0.5968	0.6842
30%			0.4390	0.5159	0.5866	0.7193
40%				0.5079	0.5763	0.7558
50%					0.5663	0.7896
MAP	0.2155	0.3360	0.4271	0.5015	0.5682	
(4b) FRank without feature selection						
Position/Relevant	**10%**	**20%**	**30%**	**40%**	**50%**	**NDCG**
10%	0.1941	0.3279	0.4331	0.5134	0.5765	0.5956
20%		0.2984	0.4050	0.4892	0.5618	0.6449
30%			0.3821	0.4747	0.5685	0.6893
40%				0.4606	0.5695	0.7350
50%					0.5654	0.7756
MAP	0.1481	0.2700	0.3680	0.4607	0.5504	
(4c) RankNet with feature selection						
Position/Relevant	**10%**	**20%**	**30%**	**40%**	**50%**	**NDCG**
10%	**0.2721**	**0.4086**	**0.5125**	**0.5754**	**0.6436**	**0.6526**
20%		0.3593	**0.4758**	**0.5476**	**0.6147**	**0.6885**
30%			**0.4493**	**0.5337**	**0.6043**	**0.7280**
40%				**0.5277**	**0.5990**	**0.7682**
50%					**0.5937**	**0.8081**
MAP	**0.2221**	**0.3404**	**0.4444**	**0.5188**	**0.5874**	
(4d) FRank with feature selection						
Position/Relevant	**10%**	**20%**	**30%**	**40%**	**50%**	**NDCG**
10%	0.2255	0.3586	0.4764	0.5676	0.6364	0.6270
20%		0.3317	0.4351	0.5173	0.6087	0.6669
30%			0.4082	0.5029	0.5961	0.7063
40%				0.5000	0.5893	0.7501
50%					0.5824	0.7871
MAP	0.1938	0.3039	0.3913	0.4833	0.5699	

Table 4a to 4d: Prediction results for RankNet and FRank

These tables shows the prediction results for RankNet (4a, 4c) and FRank (4b, 4d) with and without feature selection for all evaluation measures. The headings of each table denotes the query part considered as hits. The first column contains the evaluation position. The rightmost column shows values for NDCG evaluation, while the bottom row reports the hit-based MAP values. The center of each table lists the precision values for a given query position and top fraction of hits. All overall optimal values for a given measure is marked by bold fonts. Each table shows only the optimal performance for a ranking algorithm trained with or without feature selection.

mance: (1) descriptors that are related to the target variable, which in this case are the electro-chemistry related properties. This is supported by the observation that two to three of ten features belong to this category. And (2) descriptors that discriminate between different molecules, which is corroborated by the fact that blocks containing descriptors being sensitive to molecular constitution, e.g. the [Edge-adjacency] block or the [Topological] block, often generate optimal ranking models (see also Tables 3 and 4).

We only found two 3D descriptor blocks to be useful for model training, the [Geometrical] and the [Whim] block (Table 3). Interestingly, four of ten selected features are related to electro-chemistry, which is similar to the blocks discussed above. To summarize our results we assume that 3D information might improve the training of ranking models to some extend but is neither crucial nor sufficient.

Tables 4a-d summarize the evaluation on hold-out data of the 400 ranking models. The use of RankNet in combination with feature selection achieved the best prediction performance for nearly all measures (Table 4c). Interestingly, all these results were obtained by training on the [Edge-adjacency] descriptor block rendering this combination as most useful. We compared our NDCG and MAP values to those achieved on other ranking problems. In (Tsai et al., 2007) MAP values range between 0.13 to 0.25, NDCG values range up to 0.55 and the precision is below 0.4. In comparison we achieve MAP values between 0.14 to 0.59, NDCG values up to 0.81 and our precision is below 0.6. Therefore, our results are 1.5 to 2.5fold better than those achieved in another study on a web-learning problem (Tsai et al., 2007).

Feature selection using GSA significantly improved the prediction performance of both ranking methods. RankNet (Table 4c) as well as FRank (Table 4d) achieved better generalization errors when trained with feature selection for nearly all measures. This improvement achieved by feature selection ranges up to 5% for RankNet and up to 31% for FRank.

Summarizing our results, we have demonstrated that using a selection of ten [Edge-adjacency] descriptors enables RankNet to efficiently predict top scoring peptides learned from Mascot search results.

Similar studies with alternative experiments were performed and published. A predictor employing Gaussian-Mixture models were used to classify proteotypic peptides from publicly available data sets (Mallick et al., 2006). Their predictor identified reliably detectable peptides for up to 60% of all proteins, but was not specific for an experimental setting. Another approach trains on a specially created data set using a known protein mixture (Sanders et al., 2007). They set a threshold on the identification score to define flyable peptides, being reliably detectable, within this data set to obtain a balanced data set for neural network classification. Their predictor is specific for an experimental setting, but has the drawback to require a standardized experiment for training purposes each time the instrumental setup changes. In a similar way Webb-Robertson et al. train SVM classifiers on a large set of MS identifications that were verified using FTICR MS (Webb-Robertson et al., 2008). This method yields also facility specific classifiers, but requires an additional verification of the training data.

In contrast to all these published methods, our new approach has the clear advantage to analyze the protein identifications made during a time period for learning a ranking function that ranks top scoring peptides. In this way, our approach does not require any kind of additional experiments, while resulting in predictors that are specific for a certain facility. Since these advantages come in with the usage of ranking methods, a direct comparison to classification algorithms are prohibitive due to differences of the learning problem.

Conclusion

This paper applies ranking methods to predict peptide sequences that achieve frequently higher search engine scores without performing additional experiments or needing expensive hardware. By using SMILES substitution table each peptide sequence was encoded into a molecular graph representation. We evaluated diverse molecular descriptors sets in combination with feature selection and showed that RankNet trained on the [Edge-adjacency] block obtained optimal prediction performance.

Because our approach is independent of the employed proteomics database search engine a training on queries obtained from different search engines or eventually consensus scores could easily be integrated and may result in decreased false-positive rates. Also, our ranking method would not only increase confidence in normal protein identification but would also conduct targeted peptide based experiments, e.g. MRM.

Also, our approach is well suited to address the task of ranking proteotypic peptides having post-translational modifications by incorporation of phosphorylations or oxidations into the substitution table.

Future work should compare other methods for peptide encoding as well as other ranking methods. Additionally, analysis methods could be developed that exploit the struc-

ture immanent in rankings, e.g. align ranking predictions with search engine queries to deduce quality statements.

Acknowledgements

CH designed and implemented the experiment. GH created the SMILES conversion table and supported the usage of MacroModel and DragonX. SJ prepared Mascot results for further processing. JM performed the MS experiments. WS, SJ, AN and AZ critically read the manuscript and supervised experimental design.

References

1. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, et al. (2005) Learning to rank using gradient descent ICML '05: Proceedings of the 22nd international conference on Machine learning. 89-96.
2. Dalby A, Nourse JG, Hounshell WD, Gushurst AKI, Grier DL, et al. (1992) Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. J Chem Inf Comput Sci 32: 244-255. » CrossRef » Google Scholar
3. Geng X, Liu TY, Qin T, Li H (2007) Feature selection for ranking SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 407-414. » CrossRef » Google Scholar
4. Hala M, Cole R, Synek L, Drdova E, Pecenkova T, et al. (2008) An exocyst complex functions in plant cell growth in Arabidopsis and tobacco. Plant Cell 20: 1330-1345. » CrossRef » Pubmed » Google Scholar
5. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques ACM Transactions on Information Systems. 20: 422-446. » CrossRef » Google Scholar
6. Joachims T (2002) Optimizing search engines using clickthrough data KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. 133-142. » CrossRef » Google Scholar
7. Mallick P, Schirle M, Chen SS, Flory MR, Lee H, et al. (2006) Computational prediction of proteotypic peptides for quantitative proteomics. Nat Biotechnol 25: 125-131. » CrossRef » Pubmed » Google Scholar
8. Perkins DN, Pappin DJC, Creasy DM, Cottrell JS (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20: 3551-3567. » CrossRef » Pubmed » Google Scholar
9. Sanders W, Bridges S, Mccarthy F, Nanduri B, Burgess S (2007) Prediction of peptides observable by mass spectrometry applied at the experimental set level. BMC Bioinformatics 8: S23. » CrossRef » Pubmed » Google Scholar
10. Tetko IV, Gasteiger J, Todeschini R, Mauri A, Livingstone D, et al. (2005) Virtual computational chemistry laboratory - design and description. J Comput Aid Mol Des 19: 453-63. » CrossRef » Pubmed » Google Scholar
11. Tsai MF, Liu TY, Qin T, Chen HH, Ma WY (2007) FRank: a ranking method with fidelity loss SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 383-390. » CrossRef » Google Scholar
12. Webb-Robertson BJM, Cannon WR, Oehmen CS, Shah AR, Gurumoorthi V, et al. (2008) A support vector machine model for the prediction of proteotypic peptides for accurate mass and time proteomics. Bioinformatics 24: 1503-1509. » CrossRef » Pubmed » Google Scholar
13. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci. 28:31-36. » CrossRef » Google Scholar
14. Yates JRI, Eng J, McCormack A, Schieltz D (1995) Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. Anal Chem 67: 976-989. » CrossRef » Pubmed » Google Scholar