Supporting Information: Peptide Retention in Hydrophilic Strong Anion Exchange Chromatography is Driven by Charged and Aromatic Residues

Sven H. Giese[1], Yasushi Ishihama[2], and Juri Rappsilber*[1,2,3]

[1]Bioanalytics, Institute of Biotechnology, Technische Universität Berlin, 13355 Berlin, Germany
[2]Graduate School of Pharmaceutical Sciences, Kyoto University, Kyoto 606-8501, Japan
[3]Wellcome Centre for Cell Biology, School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3BF, United Kingdom

Contents

List of Figures

List of Tables

*juri.rappsilber@tu-berlin.de

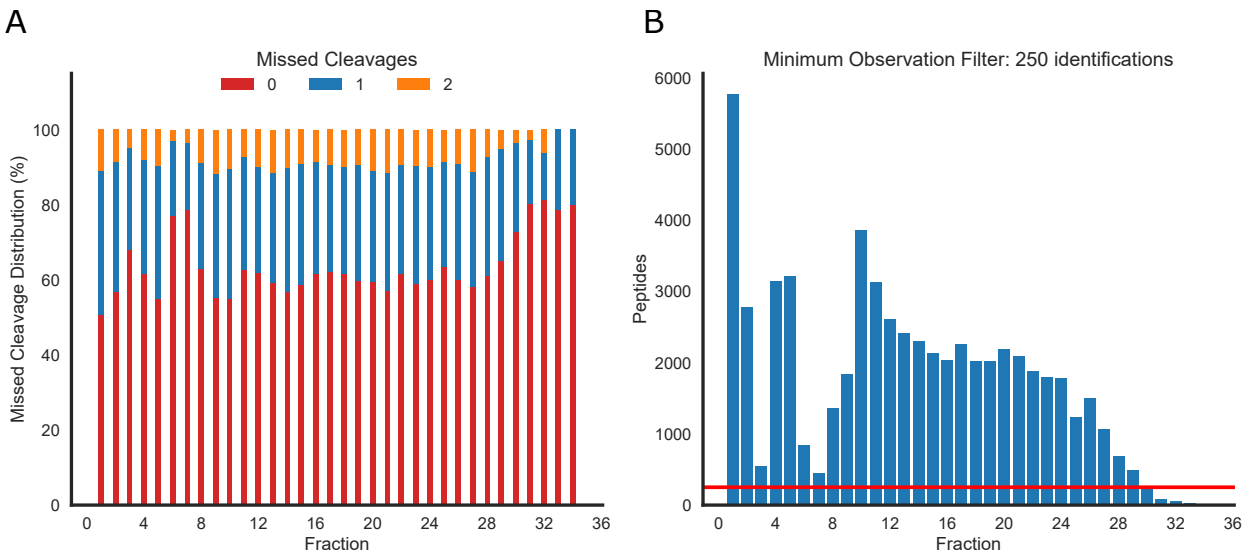

Figure S1: Data overview. (A) Missed cleavage distribution. Stacked bar charts show the number of peptides with 0, 1 or 2 missed cleavages per fraction. (B) Number of peptide identifications per fraction. The horizontal line was set as cut-off - all fractions with fewer than 300 identifications were disregarded from the analysis. A total number of 59,723 non-redundant peptides were analysed before using the cut-off.

S1 Effect Size and Retention Time Influence Differences of the Charged Amino Acids

As established in the main text the effect size of the charged amino acids is very similar. However, the distributions of peptides with 0-5 D or E residues are clearly shifted as shown in Fig. 1 of the manuscript. Table S1 shows the average mean increase of the fraction number per peptide population with 0-5 D/E counts. On average, a single D/E residue in the peptide sequence will shift the peptide 3 fractions. Since the effect size of D/E is very similar (Fig. S2 B) we assume that the estimate holds for either D or E residues. On the other hand the difference between K and R residues is more pronounced (Fig. S2 A).

Table S1: Effect of D and E residues to the retention time shift.

DE count	Mean Fraction	Difference to last Fraction
0	2.61	0
1	5.89	3.28
2	10.73	4.84
3	14.89	4.16
4	17.89	3
5	20.31	2.42

Note: The mean fraction was computed by first filtering all peptide identifications to sequences with 0-5 D or E residues. For each of the five classes the mean fraction was then computed.

S2 Non-charged Amino Acid Contributions to the Retention Time

In the main text we classified the remaining amino acids as 'retaining', 'eluting' and 'other'. This classification is mainly based on investigating an isolated subset of peptides with D/E residue count of 2 and K/R residue count of 1. This subset is then used to visually and statistically infer the influence of the remaining amino acids. As shown in Fig. S3 the number of observations of DE2, KR1 peptides is still very high and distributed over 12 fractions. Based on these peptides we computed the average amino acid composition in each fraction and performed linear regression analysis with the composition as dependent variable and the fraction as target variable. Effectively, modeling the increase or decrease in the sequence composition for all 16 remaining amino acids. The magnitude of the slope can be considered as correlation between the occurrences of amino acids and

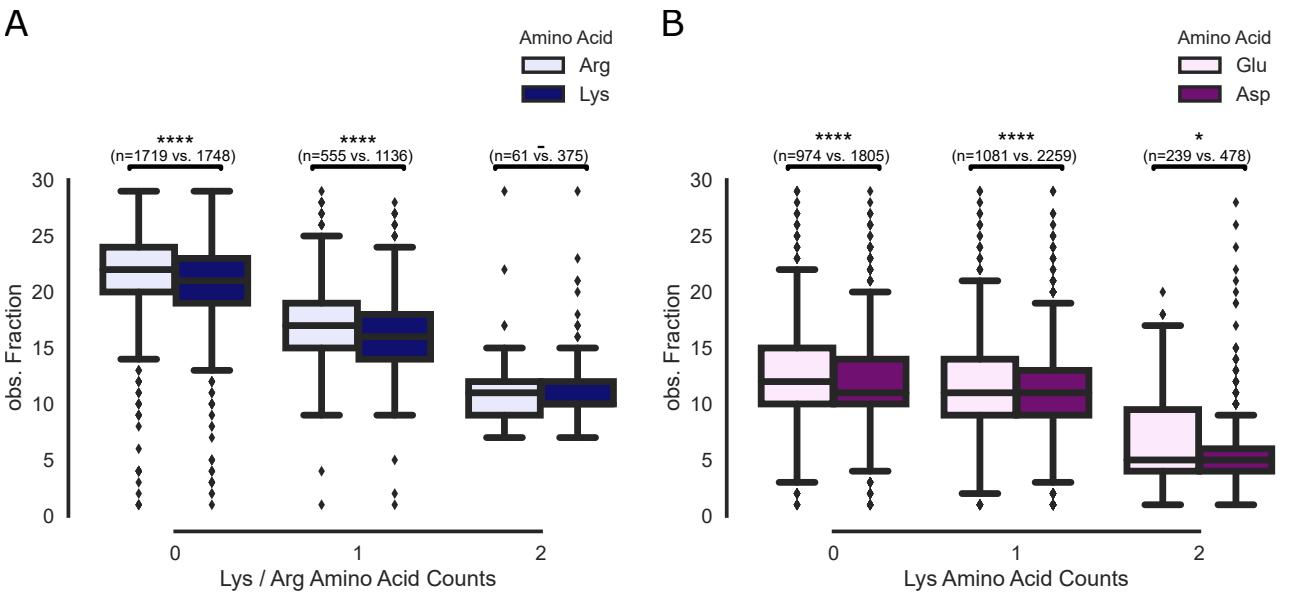

Figure S2: Observed fractions based on a peptide sequence filters. (A) Effect size of Lys and Arg residues. To compare the elution strength of Arg and Lys first all peptide identifications were filtered to only include peptides with and summed D/E residue count of 4. Then the fractions of peptides with 1, 2 and 3 K/R residues were extracted and compared. (B) Effect size of Glu and Asp residues. To compare the retaining strength of Glu and Asp first all peptide identifications were filtered to only include peptides with exactly two D or two E residues. For these two sub-populations then the peptides with 1, 2 and 3 K residues were compared. Significance tests were performed using the Mann-Whitney-U-Test.

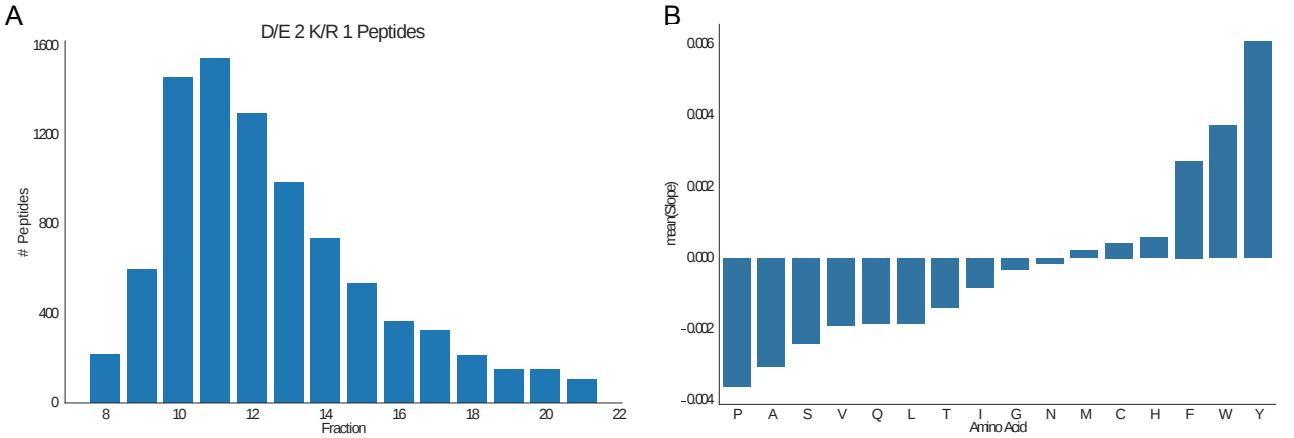

Figure S3: Sub-population of peptides with an D/E 2 and K/R 1 count. (A) The distribution of peptide occurrences is shown depending on the observed fraction. Fractions below 8 and higher than 21 all contained less than 1% (92) of the total number of observations (9,199) and were removed for further analysis. (B) Based on the average composition of the peptides from (A) 20 linear regression models (for each amino acid one) were fitted on the target variable (fraction number) and the dependent variable (average amino acid sequence composition). The slopes of the regression model are shown as bars. Amino acids that have an retaining effect are expected to have a positive slope, amino acids that have an eluting effect are expected to have a negative slope.

a shifted retention time. For large positive slopes, we expected the amino acids to have an retaining effect on the retention time. For large negative slopes, we expected the amino acids to have an eluting effect, see Fig. S3B. The linear regression model was also used to perform a significant test on the slope of the fitted model: with H_0 assuming that the slope is equal to zero and H_1 assuming that the slope is not equal to zero. The test results and fitted lines are visualized in Fig. S4. Based on this we broadly classified the remaining amino acids into retaining contributions (F, W, Y), eluting contributions (P, A, S, V, Q, T) and non-clear or other contributions (L, I, G, N, M, C, H).

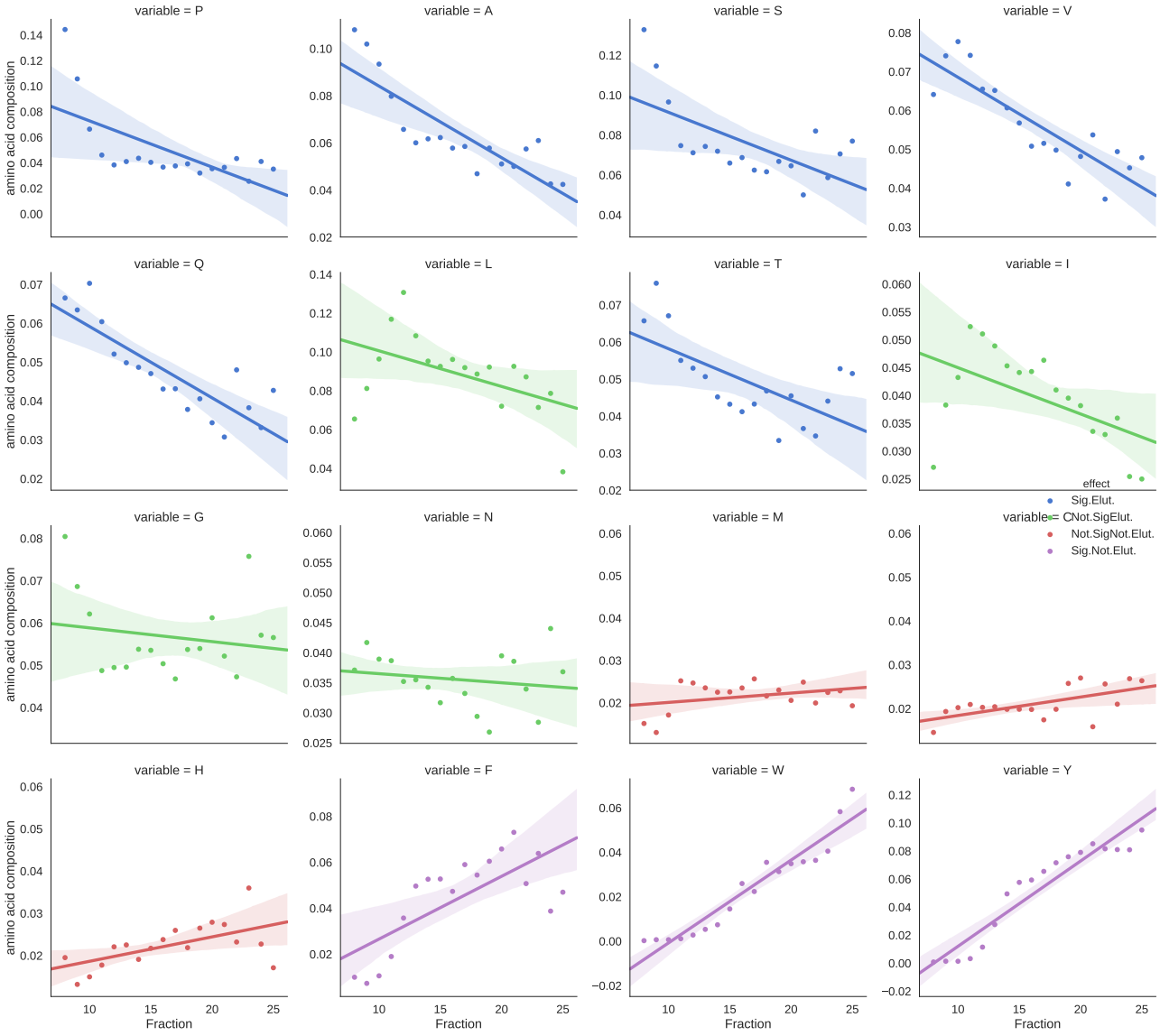

Figure S4: Classification of non-charged amino acid effects on the retention time based on linear regression. As described in Supplementary Fig. S3 linear regression models were fit to the sequence composition data from peptides with an D/E 2 and K/R 1 count. In addition, a simple test with H_0: the slope of the regression model is equal to zero was performed using SciPy. The aromatics F, W, and Y yield significant results and have a potentially large retaining effect. The amino acids P, A, S, V, Q and T also show a significant test results after Benjamini-Hochberg correction [1], but for having an eluting effect. For the amino acids L, I, G,M, N and H the slope of the regression model was not significantly different from zero and were thus classified as ambiguous ('other').

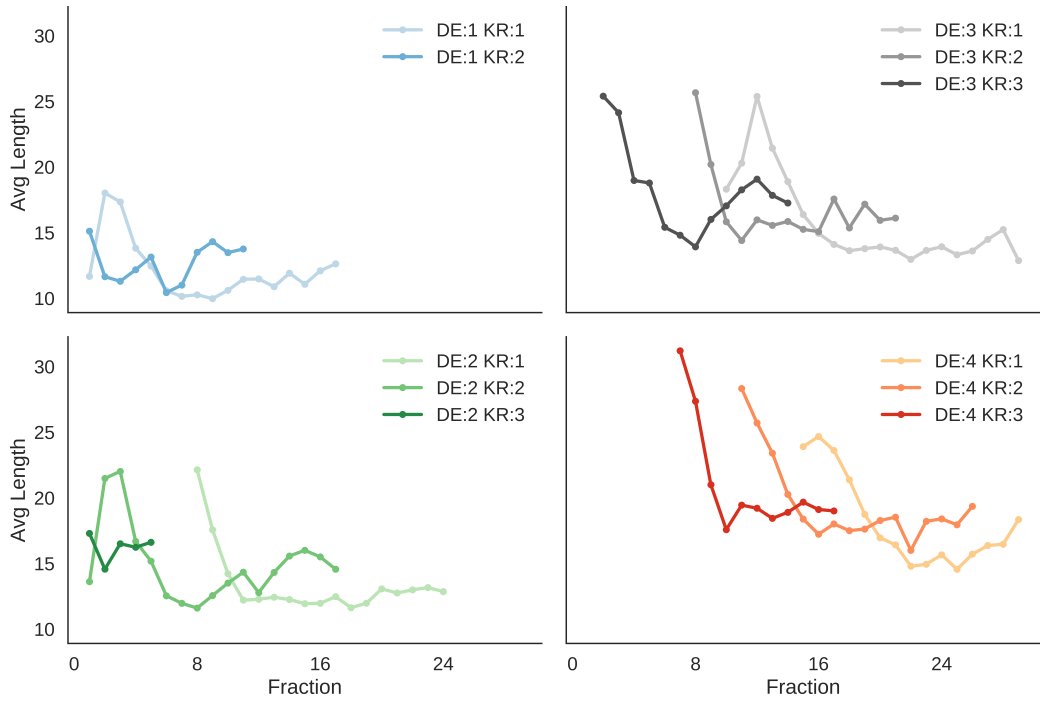

Figure S5: Peptide length influence on the retention time. In each panel a subset of peptides is first extracted (e.g. DE:1 KR:1 first filters all peptides with exactly one D or E residue and exactly one K or R residue). A minimum number of observations of 300 peptides was required per category. The Avg Length represents the mean peptide length of all peptides in a selected fraction.

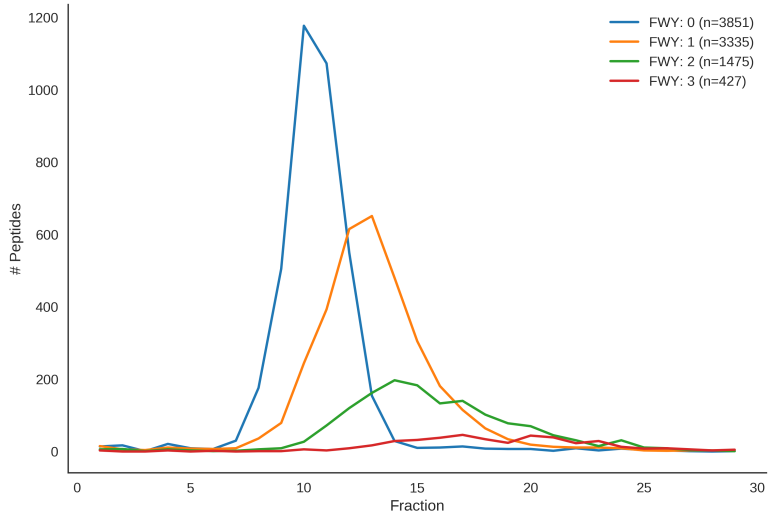

Figure S6: Peptide counts across all fractions with 0,1,2 or 3 WYF residues. In addition to the applied FYW filter all peptides have 2 D/E residues and 1 K/R residue.

S3 Machine Learning - Training, Prediction and Evaluation

Overview

We are interested in learning and predicting the interaction of peptides with the hSAX column - based on the peptide sequence we want to be able to predict when the peptide will elute. Initially, we used a set of classification algorithms in our pre-experiments. The selection of regression methods includes: simple linear regression including the length correction parameter (lcp) with only the 20 amino acids as features ('Pyteomics') [3], a linear regression model with all designed features (Supplementary Table S2), ridge regression, lasso regression, support vector machine regression (SVR) and random forest regression. The selection of classification algorithms includes: feedforward neural network (FNN, Keras implementation with the Theano backend), logistic

regression, random forest, gradient boosting (python package XGBoost[1] [2]) , a support vector machine (SVM) and ordinal logistic regression (python package MORD[2] [7]). Except for Pyteomics, the FNN, MORD and XGBoost the scikit-learn[3] [6] implementations were used.

Input Features

An essential part of classical machine learning algorithms is the engineering of features. Based on initial observations and by investigating the literature we came up with 218 features to summarize the properties of a peptide that might govern retention. These features are summarized in Table S2. Similar to ELUDE and SSRCalc we used hydrophobicity features, consecutive occurrences of amino acids [5] and position specific features for the 20 amino acids [4].

Hyper-parameter Optimization

The above mentioned machine learning algorithms all require a fine tuning of their parameters to achieve the best possible performance (hyper-parameter optimization). Our workflow for optimization, testing and validation the best parameters was as follows: (1) grid search for optimal hyper-parameters with 5-fold cross-validation (CV), (2) selection of the best set of parameters for each classifier based on the achieved accuracy on the test data and (3) validating the best performing classifier on a hold-out validation set that was never used for training. Table S3 gives and overview of the grid search for hyper-parameter optimization. Table S4 summarizes the results for each classifier with the best set of parameters. The best performing classifier was a a feedforward neural network implementation with an CV accuracy of $70 \pm 0.81\%$ (mean $\pm$ standard error of the mean). The linear regression models achieved the lowest accuracy on the test sets with $19\% \pm 0.002$. Pyteomics and the corresponding linear model (lcp) with a minimal set of features were not included in the grid search.

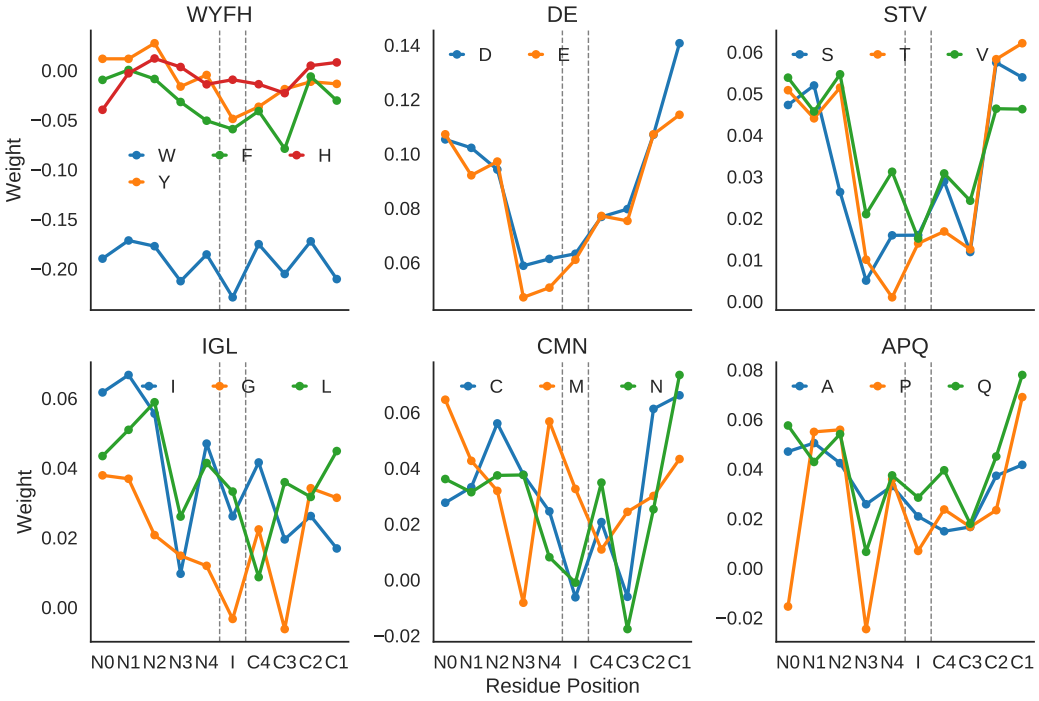

Figure S7: Mean weights from the input layer to the first hidden layer in the FNN. The X-axes indicates the position of an residue in the peptide sequence. The weights are derived from training the FNN on the complete training data (Accuracy: 0.74, Correlation: 0.95). Abbreviations: N - peptide N-terminal, C - peptide C-terminal, I - internal. The numbers indicate the distance to the respective termini.

[1] https://github.com/dmlc/xgboost
[2] https://pythonhosted.org/mord/
[3] http://scikit-learn.org/stable/

Table S2: Extracted Features and their description.

Feature	Description	Total Features
AAcount	Amino acid counts of all 20 amino acids	20
N[AA]1-5	Amino acid counts encoding the n-terminal positions from 1-5.	100
C[AA]2-5	Amino acid counts encoding the c-terminal positions from 2-5. N-term was excluded as mostly R/K was observed.	80
CtermK/R	Indicator if Lysine is the C-Terminal Residue	2
Patterns	Counts the number of coherent amino acid patterns in the peptide sequence of different classes: acidic, basic, aromatics, mixed (acidic+basic) patterns. For example DD, KR, WW, DK.	4
Structural Features	Percentage of amino acids from the sequence that are preferably in the following secondary structure elements: Helix: V, I, Y, F, W, L. Turn: N, P, G, S. Sheet: E, M, A, L.	3
Gravy	Gravy according to Kyte and Doolittle.	1
pI	Isoelectric point of the peptide sequence.	1
loglength	Natural logarithm of the peptide length.	1
Netcharge	Defined as sum of the acidic residues (-1 each), basic residues (+1) and the aromatics F (0.3), W (0.8) and Y (0.6) in a peptide sequence.	1
N-/C-Term distance	Shortest distance of E/D to the C-term and shortest distance of K/R to the N-term.	2
TurnIndicator	Average distance between Proline residues in the sequence.	1
Sandwich	Aromatic patterns that are separated in sequence by one amino acid, e.g. WXY.	1
Aromaticity	Percentage of amino acids belonging to WFY.	1
	Total number of features	218

Note: The count features were scaled with a length correction parameter (lcp).

Table S3: Initial parameter grid for hyper-parameter optimization

Classifier	Parameter Grid
ORL IT	'alpha': [0.1, 0.3, 0.5, 0.7, 0.9]
Lasso	'fit_intercept': [True, False], 'alpha': [0.1, 0.3, 0.5, 0.7, 1], 'normalize': [True, False]
LinearRegression	'fit_intercept': [True, False], 'normalize': [True, False]
Ridge	'fit_intercept': [True, False], 'alpha': [0.1, 0.3, 0.5, 0.7, 0.9], 'normalize': [True, False]
SVM	['C': [0.1, 1, 10], 'kernel': ['linear'], 'class_weight': [None, 'balanced'], 'C': [0.1, 1, 10], 'gamma': [0.001, 0.0001], 'kernel': ['rbf'], 'class_weight': [None, 'balanced']]
OLR AT	'alpha': [0.1, 0.3, 0.5, 0.7, 0.9]
RandomForestClassifier	'n_jobs': [20], 'n_estimators': [100, 500], 'max_features': ('log2', 'auto'), 'max_depth': (None, 4, 7), 'min_samples_split': (2, 15)
XGB	'reg_alpha': [0.01, 0.5, 1], 'n_estimators': [300, 500], 'gamma': [0, 0.1, 1], 'max_depth': [3, 5, 9], 'reg_lambda': [0.01, 0.5, 1], 'nthread': [20], 'learning_rate': [0.1, 0.05]
RandomForestRegressor	'n_jobs': [20], 'n_estimators': [100, 500], 'max_features': ('log2', 'auto'), 'max_depth': (None, 5, 15), 'min_samples_split': (2, 15)
LogisticRegression	'C': [0.01, 0.1, 1, 10], 'multi_class': ['ovr', 'multinomial'], 'n_jobs': [20], 'solver': ['newton-cg'], 'class_weight': [None, 'balanced']

Note: The parameter grid was searched exhaustively with all combinations. The definition of each parameter is available via the documentations of scikit-learn, MORD and XGBoost. The neural network architecture was optimized manually.

Table S4: Best Results after hyper-parameter optimization with 5-fold cross-validation.

Classifier	Best Parameters	Train Accuracy (%)	Test Accuracy (%)
FNN	'layer': 4, 'neurons': [50, 40, 35, 29], 'activation':['relu', 'tanh', 'relu', 'softmax'], 'batch_size':512, 'epochs': 100	79 ± 1.27	70 ± 0.81
SVC	'class_weight': None, 'C': 10, 'kernel': 'linear'	64 ± 0.17	53 ± 0.19
SVR	'C': 10, 'kernel': 'rbf', 'gamma': 'auto', 'epsilon': 0.1	52 ± 0.06	50 ± 0.26
XGBClassifier	'n_estimators': 300, 'learning_rate': 0.1, 'reg_lambda': 0.01, 'reg_alpha': 1, 'max_depth': 9, 'nthread': 25, 'gamma': 0.1	100 ± 0.0	47 ± 0.32
XGBRegressor	'n_estimators': 300, 'learning_rate': 0.1, 'reg_lambda': 0.01, 'reg_alpha': 0.01, 'max_depth': 9, 'nthread': 25, 'gamma': 0.1	67 ± 0.17	46 ± 0.31
RF-Classifier	'max_features': 'auto', 'n_jobs': 20, 'n_estimators': 500, 'min_samples_split': 2, 'max_depth': None	100 ± 0.0	43 ± 0.33
LogisticAT	'alpha': 0.5	43 ± 0.14	43 ± 0.18
RF-Regressor	'max_features': 'auto', 'n_jobs': 20, 'n_estimators': 500, 'min_samples_split': 2, 'max_depth': None	77 ± 0.04	42 ± 0.19
LogisticRegression	'solver': 'newton-cg', 'multi_class': 'multinomial', 'C': 10, 'class_weight': None, 'n_jobs': 20	48 ± 0.07	40 ± 0.2
LinearRegression	'fit_intercept': True, 'normalize': False	19 ± 0.15	19 ± 0.36
Ridge	'alpha': 0.1, 'normalize': False, 'fit_intercept': True	19 ± 0.15	19 ± 0.34
Lasso	'alpha': 0.1, 'normalize': False, 'fit_intercept': True	14 ± 0.07	14 ± 0.06

Note: The grid search results are based on 5-fold cross-validation and sorted after the test accuracy in descending order. Values in the accuracy column represent the mean and standard error of the mean from the CV. A full explanation of the parameters is available through the scikit-learn documentation. Abbreviations: SVC - Support Vector machine Classification, OLR - Ordinal Logistic Regression, AT - All-Threshold, IT - Immediate-Threshold, RF - Random Forest, FNN - Feedforward Neural Network.

References

[1] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing, 1995.

[2] Tianqi Chen and Carlos Guestrin. XGBoost : Reliable Large-scale Tree Boosting System. *arXiv*, pages 1–6, 2016.

[3] Anton A. Goloborodko, Lev I. Levitsky, Mark V. Ivanov, and Mikhail V. Gorshkov. Pyteomics - a Python Framework for Exploratory Data Analysis and Rapid Software Prototyping in Proteomics. *J. Am. Soc. Mass Spectrom.*, 24(2):301–304, feb 2013.

[4] Oleg V. Krokhin. Sequence-specific retention calculator. Algorithm for peptide retention prediction in ion-pair RP-HPLC: application to 300- and 100-A pore size C18 sorbents. *Anal. Chem.*, 78(22):7785–95, nov 2006.

[5] Luminita Moruz, An Staes, Joseph M. Foster, Maria Hatzou, Evy Timmerman, Lennart Martens, and Lukas Käll. Chromatographic retention time prediction for posttranslationally modified peptides. *Proteomics*, 12(8):1151–9, apr 2012.

[6] Fabian Pedregosa and G Varoquaux. *Scikit-learn: Machine learning in Python*, volume 12. 2011.

[7] Fabian Pedregosa-Izquierdo. *Feature extraction and supervised learning on fMRI : from practice to theory.* Theses, Universit{é} Pierre et Marie Curie - Paris VI, 2015.