Predicting Retention Times of Post-translationally Modified Peptides Using Machine Learning Techniques

Kurtis Bertauche
kbertauche@unr.edu
University of Nevada, Reno
Reno, Nevada, USA

Alexander Choi
alexanderchoi@unr.edu
University of Nevada, Reno
Reno, Nevada, USA

So Young Ryu
soyoungr@unr.edu
University of Nevada, Reno
Reno, Nevada, USA

Abstract

Accurately identifying protein post-translational modifications (PTMs) is important in studying cell biology and diseases. Mass spectrometry can generate valuable information to identify proteins and their post-translational modifications. It has been previously known that retention times of peptides (substrings of proteins) can be used to improve the accuracy of peptide/protein identifications. Recognizing the needs for PTM retention time prediction models, we explored the viability of various machine-learning models (e.g., Extreme Gradient Boosting, Random Forest, Support Vector Regression) for predicting retention times of phosphorylated peptides. In addition, we evaluated retention time prediction models using various performance metrics and compared them to currently available methods. We demonstrated the retention time model performances using a large synthetic proteomics and phosphoproteomic dataset.

CCS Concepts: • **Applied computing** → **Bioinformatics**; **Computational proteomics**.

Keywords: mass spectrometry, post-translational modification, retention time, prediction, machine learning models

ACM Reference Format:
Kurtis Bertauche, Alexander Choi, and So Young Ryu. 2022. Predicting Retention Times of Post-translationally Modified Peptides Using Machine Learning Techniques. In *Proceedings of KDD Undergraduate Consortium (KDD-UC).* ACM, New York, NY, USA, 6 pages.

1 Introduction

Accurate protein identifications, especially protein PTM identifications, remain to be a challenging task [33]. Modern mass spectrometry experiments generate vast amounts of mass spectra of peptides, presenting bioinformatic challenges on how to correctly interpret this data [22, 30]. In computational mass spectrometry, it would be important to generate new but accurate information about peptides and to fully utilize the available information for accurate peptide/protein identifications.

Peptide retention times are the time points when peptides elute from the liquid chromatography column [30]. If peptide retention times are predicted accurately using peptide sequences and prediction models, they can be used to accurately identify peptides by reducing the size of candidate peptides in a peptide identification step [22, 23, 32]. Moreover, retention time information can be utilized in several areas of mass spectrometry, including targeted quantitative measurement of proteins [8, 16]. The ability to accurately predict retention times of peptides has become important in proteomics.

Developing methods that reliably and accurately identify retention times of peptides has been a topic of interest for many years. Some of the earliest retention time prediction methods use linear regression models that estimate a retention time effect of each amino acid and predict retention times by summing estimated retention time effects of amino acids in given peptide sequences [20]. With the evolution and improvement in computing power, many researchers have turned towards using deep-learning models for retention time prediction [11, 12]. However, many of these deep-learning models have focused on peptides without post-translational modifications. These existing methods may not work well in predicting datasets with post-translationally modified peptides because post-translational modifications can have significant effects on retention times [18].

As the need to accurately predict retention times of PTM peptides grows, more recent studies have sought to fill this gap and develop models capable of predicting retention times of peptides that include post-translational modifications [2, 36]. However, there are several different types of post-translational modifications and it is apparent that building models which excel at predicting peptide retention times for all types of modifications can be particularly difficult. Modifications such as phosphorylation are both physically and chemically very different from many other types of post-translational modifications, causing problems in accurately

KDD-UC, August 14–18, 2022, Washington DC, ACM, New York, NY, USA.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

predicting retention times for phosphorylated peptides using general models such as DeepLC [2]. Because phosphorylation has significant effects on retention times of peptides [4], it is important to accurately predict retention times of phosphorylated peptides and utilize the retention times of phosphorylated peptides for better phosphorylated peptide/protein identifications.

Here, we focus on building peptide retention time models for phosphorylation, one of the most important post-translational modifications. While there has been effort into building general purpose retention time models, we seek to build models that excel specifically for phosphorylated peptides, as current tools often struggle with this task. We built several machine learning retention time models as well as parametric models (e.g., linear regression, lasso) for unmodified and phosphorylated peptides, evaluated their performances, and compared them to the existing general models to investigate the benefit of using modification-specific models. A large synthetic proteomic and phosphoproteomic data set with nearly 100,000 peptide sequences [18] was used in this study.

2 Methods

2.1 Data

We used a synthetic proteomic and phosphoproteomic data set to evaluate retention time prediction models. The experimental details were shown in [18]. In brief, the synthetic data set contained 96 seed peptides selected from five large-scale human phosphorylation studies [7, 24–26, 29] and their synthesized variants. The synthesized variants were generated by introducing phosphorylation modifications at the phosphosite of the seed peptides and/or replacing amino acids of the seed peptides with different standard amino acids. These synthesized peptide mixtures were analyzed with an Orbitrap Velos using a beam-type collision-induced dissociation (HCD) fragmentation method. Mascot search engine (2.3.1; http://matrixscience.com) was used to identify peptides. Identified peptides were filtered with an E-value threshold of 0.01. To ensure additional reliability in the data, only identified peptides that could be matched to the sequences of synthesized peptides were considered. Finally, for each sequence, a retention time associated with the maximum intensity of the corresponding precursor ion at the time it was selected for an MS/MS fragmentation was extracted. A total of 52,960 unmodified peptides and 37,976 phosphorylated peptides were used for further analysis.

2.2 Retention Time Prediction Models

We built 8 retention time prediction models. In addition, currently available models (AutoRT [36] and DeepLC [2]) were presented for a comparison purpose. The candidate predictor variables for prediction models except for AutoRT and DeepLC were the counts of each unmodified or modified amino acid and the length of peptides. The modified amino acids included phosphorylated serine, threonine, and tyrosine amino acids as well as the oxidation of the methionine amino acid. The data set was randomly split into a training set (80%) and a test set (20%). All of the prediction models used the same training set and were evaluated on the same test set. When tuning hyperparameters was required for model building, we used the Root Mean Square Error (RMSE) metric to select the best set of hyperparameters for the model. The following sections had details about training processes for each model.

2.2.1 Proposed Machine Learning Models.

We proposed three different machine learning models for the prediction of peptide retention times, Extreme Gradient Boosting (XG Boost), Random Forest, and Support Vector Regression. All three types of models have recently gained popularity and shown impressive results in many different applications [9, 14, 34], possibly lending themselves well for the task of predicting peptide retention times.

Extreme Gradient Boosting (XG Boost). Extreme Gradient Boosting (XG Boost) [5] was implemented for peptide retention time prediction. The values of the following six parameters were selected using five-fold cross-validation: 1) a step size shrinkage used in update (η), 2) minimum loss reduction required to make a further partition on a leaf node of the tree (γ), 3) maximum depth of a tree (max_depth), 4) minimum sum of instance weight needed in a child (min_child_weight), 5) subsample ratio of the training instances (subsample), and 6) subsample ratio of predictor variables when constructing each tree (colsubsample_bytree). The R package "XGBoost" [6] was used. The implementation details for the Extreme Gradient Boosting were shown in Algorithm 1.

Algorithm 1 XG Boost Method

```
Divide data into training and testing sets
Divide training set into k = 5 subsets
for i in search space of parameter combinations do
    for j from 1 to k do
        Leave out training subset j from the training set
        Fit XG Boost model on the remaining training set
        Calculate RMSE on training subset j
    end for
    Calculate average RMSE for subsets of parameter combination i
end for
Rebuild XG Boost model on the entire training set using parameters with lowest average RMSE
Complete statistical analysis of model using the testing set
```

Random Forest. We employed a tree-based method, Random Forest [3] to build a nonlinear regression model for

peptide retention time prediction. Using OOB (out-of-bag) errors in the training set, we selected the number of trees and the number of variables. For a fair comparison among models, we used only the training set to tune the Random Forest instead of the entire dataset. The R package "ranger" was used to implement Random Forest. The implementation details for the Random Forest were shown in Algorithm 2.

Algorithm 2 Random Forest Method

```
Divide data into training and testing sets
for i in search space of parameter combinations do
    Build RF Model using training set
    Calculate OOB MSE for current RF Model
end for
Choose the RF model with lowest OOB MSE
Complete statistical analysis of model using the testing set
```

Support Vector Regression. Support Vector Regression (SVR) [13] was used to predict retention times of peptides. Similar to the Support Vector Machine, the loss function of SVR considers only absolute values of residuals larger than some positive constant. The optimal cost parameters were chosen by employing five-fold cross-validation. The SVR model was built using the "e1071" package in R [21]. The implementation details were shown in Algorithm 3.

Algorithm 3 Support Vector Regression Method

```
Divide data into training and testing sets
Divide training set into k = 5 subsets
for i in search space of parameter combinations do
    for j from 1 to k do
        Leave out training subset j from the training set
        Fit SVR model on the remaining training set
        Calculate RMSE on training subset j
    end for
    Calculate average RMSE for subsets of parameter combination i
end for
Rebuild SVR model on the entire training set using parameters with lowest average RMSE
Complete statistical analysis of model using the testing set
```

2.2.2 Currently Available Models.

AutoRT. We also evaluated AutoRT [36], a peptide retention time prediction that uses a deep learning approach. The detailed algorithm can be found in Wen et al. [36] In brief, after encoding each peptide into a matrix using one-hot encoding [1], the algorithm ensembled the top 10 models from neural architectures based on convolutional neural networks (CNN) [27] and recurrent neural networks (RNN) [19], and these base models were further fine-tuned using reference peptides. We built two AutoRT models in our comparison study: 1) AutoRT built from scratch using our training set and 2) the published AutoRT that was further refined using our training set with a transfer learning strategy.

DeepLC. DeepLC was another deep-learning-based peptide retention time prediction algorithm [2]. It used a convolutional deep learning architecture (CNN) and predicted retention times of peptides with modifications that were not present in the training set. We allowed DeepLC to use the known retention times of the training set for calibration as well as separately using the training set as calibration data. Results from both methods were shown in the results section.

2.2.3 Baseline Models.

Linear regression models with different variable selection approaches were used in this study.

Linear Regression. We used a multiple linear regression model [35] as a baseline model. Since this baseline model did not use any variable selection procedure, the length of peptides was excluded from the model to avoid multicollinearity.

Best subset Regression. The best subset regression [35] was used as one of the linear regression-based prediction models. We employed an exhaustive stepwise procedure to find the best combination of predictor variables and the best subset size in linear regression. The R package "leaps" [17] was used and the best subset size was determined using five-fold cross-validation.

Ridge Regression. Another linear regression-based prediction model we evaluated was a ridge regression [15]. Each predictor variable was standardized with a mean of zero and a standard deviation of one. Then, the ridge regression shrank the regression coefficients by imposing a penalty on the size of the coefficients. The R package "glmnet" was used in the model building [31] and the best complexity parameter that controlled the amount of shrinkage was selected based on five-fold cross-validation. The lambda value was chosen based on its cross-validation error at a minimum.

Lasso Regression. We also investigated the performance of a lasso regression [15] in predicting peptide retention times. All the predictor variables were standardized prior to fitting the model. The tuning parameter that controlled the L1 penalty was chosen using five-fold cross-validation. The R package "glmnet" was used in this analysis. The lambda value was chosen based on its cross-validation error at a minimum.

Elastic Net Regression. Elastic net [10] is another variable selection method based on a compromise between the lasso and ridge regression penalties. We fit an elastic net

regression to predict peptide retention times after standardizing predictor variables. Using five-fold cross-validation, we selected the following two tuning parameters for the elastic net regression: 1) a complexity parameter and 2) a parameter that controlled a compromise between ridge regression and lasso regression. The R package "glmnet" was used in the model building.

2.3 Model Evaluation

Considering our outcome variable was a continuous variable, performance metrics for a continuous variable were used. All models were evaluated using root mean square error (RMSE), and mean absolute error (MAE). The root-mean-square error (1) measured how much difference there was between predicted and observed retention times. The smaller RMSE was the better a model performed.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{1}$$

where y_i was an observed retention time for peptide i, $\hat{y}_i$ was a predicted (or fitted) retention time for peptide i, and N was a total number of peptides.

The mean absolute error also measured how much difference there was between predicted and observed retention times. Different from RMSE, MAE used absolute errors as shown in (2).

$$MAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N} \tag{2}$$

We also reported an average window size of 95% of the difference between observed and fitted retention times (denoted as 95% Window). The smaller the window size was the better a model performed in predicting retention times.

Finally, we measured Pearson correlations between fitted vs. actual retention times as shown in (3). The larger the correlation was the better a retention time model performed.

$$r = \frac{\sum_{i=1}^{N}(y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2(\hat{y}_i - \bar{\hat{y}})^2}} \tag{3}$$

where $\bar{y}$ and $\bar{\hat{y}}$ were a mean of observed retention times and a mean of predicted retention times, respectively.

2.4 Software and Reproducibility

All the models in Section 2.2.1 were written in R [28] using the following R packages: "XGBoost" [6], "ranger" [3], "e1071" [21], "leaps" [17], and "glmnet" [31]. We also used the existing software tools AutoRT [36] and DeepLC [2] in section 2.2.2, which were written in Python. Codes, scripts, and datasets were posted in a public GitHub repository at the following address: https://github.com/soyoungryu/NSF-CAREER-PTM/tree/main/RTPrediction_v1. In the repository, we included the readme document that explained the codes and data.

In any process in which randomness was required, such as splitting data into training and test sets, the random seed was set to 37. Thus, users can reproduce the results shown in this paper. In addition, the same test set was used for all the models for a fair comparison.

3 Results and Discussion

The RMSE and MAE performance measures of all the models were shown in Table 1. The Extreme Gradient Boosted model had the best performance with an RMSE of 4.94, and an MAE of 2.55. The Random Forest model performed slightly worse than the Extreme Gradient Boosted model, with an RMSE of 5.23 and an MAE of 3.07. Support Vector Regression did not perform as well as the other two machine learning approaches. Furthermore, SVR performed worse than the baseline models in terms of RMSE values.

Table 1. The performance of all the models using a test dataset. All models used the same test dataset. The RMSEs (Root Mean Square Error) and MAEs (Mean Absolute Error) in minutes were shown. The best performance measures were highlighted in bold for each performance metric.

Method	RMSE	MAE
Extreme Gradient Boosting (XG Boost)	**4.94**	**2.55**
Random Forest	5.23	3.07
Support Vector Regression	8.32	5.42
AutoRT	6.15	3.20
AutoRT Transfer Learning	6.10	3.18
DeepLC	9.33	6.42
DeepLC with Calibration	9.34	6.42
Linear Regression	8.18	5.59
Best subset Regression	8.18	5.59
Ridge Regression	8.20	5.69
Lasso Regression	8.18	5.60
Elastic Net Regression	8.18	5.60

All of the baseline models produced very similar results, except Ridge Regression, although the difference in performance is very small with only 0.02 minutes difference in RMSE value. The range of RMSE for linear models was between 8.18 and 8.20. Between baseline models, chosen coefficients were very consistent, lending well to the similar performance of all the baseline models.

All the linear models produced results significantly worse than those of the Random Forest model and the Extreme Gradient Boosted model. The baseline models performed worse than the proposed machine learning models in terms of MAE values, too. Based on our residual analyses (not shown here), we observed that linear assumptions were violated for

Table 2. The performance of all the models using a test dataset. All models used the same test dataset. The 95% Window and Pearson Correlation (Corr) were shown. The 95% Window represented an average window size of 95% of the difference between observed and fitted retention times. The best performance measures were highlighted in bold for each performance metric.

Method	95% Window	Corr
Extreme Gradient Boosting (XG Boost)	**19.99**	**0.94**
Random Forest	21.44	0.93
Support Vector Regression	35.76	0.81
AutoRT	26.36	0.90
AutoRT Transfer Learning	26.56	0.90
DeepLC	37.29	0.76
DeepLC with Calibration	37.39	0.76
Linear Regression	34.24	0.81
Best subset Regression	34.21	0.81
Ridge Regression	34.07	0.81
Lasso Regression	34.16	0.81
Elastic Net Regression	34.16	0.81

the linear models. Therefore, using linear regression models for retention time prediction may not be appropriate, thus it is expected that nonlinear models perform better than linear models.

Among existing retention time prediction methods (AutoRT, AutoRT Transfer Learning, DeepLC, DeepLC with Calibration), AutoRT with Transfer Learning performed the best with an RMSE of 6.10 and an MAE of 3.18. The superior performance of AutoRT with Transfer learning compared to AutoRT was expected because it utilized our training set as well as the dataset with which AutoRT was initially built. However, it did not perform better than our proposed machine learning models which utilized only our training set. Both DeepLC and DeepLC with Calibration performed worse than linear regression models in our dataset.

Table 2 showed the performance of all the methods in terms of 95% Window and Pearson Correlation. Consistent with performance measured using RMSE and MAE values, the Extreme Gradient Boosted model performed the best with a 95% window of 19.99 minutes and a correlation of 94%. Similarly, the Extreme Gradient Boosting and the Random Forest performed better than the baseline models (e.g., linear regression). AutoRT and AutoRT with transfer learning performed better than DeepLC methods. However, these existing retention time methods for PTM peptides performed worse than our proposed models in phosphorylated peptides. Additionally, our results are limited by the usage of only a single data set. In the future, it is necessary to further include additional data sets to verify the results that we have shown.

4 Conclusions and Future Directions

In summary, our proposed machine models using the Extreme Gradient Boosted model and the Random Forest model performed better than the existing deep-learning-based approaches in predicting retention times of phosphorylated peptides in terms of RMSE, MAE, 95% Window, and correlations. In the future, we will investigate the performance of our Extreme Gradient Boosting using other phosphoproteomic datasets to validate the results found here.

With the knowledge that phosphorylation modifications can vary significantly from other modifications both chemically and physically [2], it is plausible that a deep-learning model that specializes in phosphorylated peptides could outperform the currently available general deep-learning models in predicting retention times of phosphorylated peptides. Thus, we also want to develop phosphorylation-specific deep learning models utilizing phosphoproteomic datasets better in our future studies.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2046122 for KB and SR and the National Institute of General Medical Sciences/National Institutes of Health under Grants GM103440 and 1R15GM126562–01 for SR. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies. The authors thank other Ryu Computational Group members (Riley Brenner, Gabriella Goodwin, Sijia Qiu, Jacob Tucker, Korben DiArchangel) for helpful discussion and support.

References

[1] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. 2016. Deep learning for computational biology. *Molecular systems biology* 12, 7 (2016), 878.

[2] Robin. Bouwmeester, Ralf. Gabriels, Niels. Hulstaert, et al. 2021. DeepLC can predict retention times for peptides that carry as-yet unseen modifications. *Nat Methods* 18 (October 2021), 1363–1369. https://doi.org/10.1038/s41592-021-01301-5

[3] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[4] Riley Brenner, Kurtis Bertauche, Alexander Choi, and So Young Ryu. 2021. VA-PRT: A Visualization Tool for Analyzing Post-translational Modification Retention Times. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 1643–1645.

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[6] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Li. 2021. *xgboost: Extreme Gradient Boosting*. https://CRAN.R-project.org/package=xgboost R package version 1.4.1.1.

[7] Henrik Daub, Jesper V. Olsen, Michaela Bairlein, Florian Gnad, Felix S. Oppermann, Roman Körner, Zoltán Greff, György Kéri, Olaf Stemmann, and Matthias Mann. 2008. Kinase-Selective Enrichment Enables

Quantitative Phosphoproteomics of the Kinome across the Cell Cycle. *Molecular Cell* 31, 3 (2008), 438–448. https://doi.org/10.1016/j.molcel.2008.07.007

[8] Claudia Escher, Lukas Reiter, Brendan MacLean, Reto Ossola, Franz Herzog, John Chilton, Michael J MacCoss, and Oliver Rinner. 2012. Using i RT, a normalized retention time for more targeted measurement of peptides. *Proteomics* 12, 8 (2012), 1111–1121.

[9] Junliang Fan, Xiukang Wang, Lifeng Wu, Hanmi Zhou, Fucang Zhang, Xiang Yu, Xianghui Lu, and Youzhen Xiang. 2018. Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China. *Energy Conversion and Management* 164 (2018), 102–111. https://doi.org/10.1016/j.enconman.2018.02.087

[10] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software* 33, 1 (2010), 1.

[11] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, et al. 2019. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nat Methods* 16 (June 2019), 509–518. https://doi.org/10.1038/s41592-019-0426-7

[12] S. Guan, M. F. Moran, and B. Ma. 2019. Prediction of LC-MS/MS Properties of Peptides from Sequence by Deep Learning. *Molecular & cellular proteomics : MCP* 18, 10 (June 2019), 2099–2107. https://doi.org/10.1074/mcp.TIR119.001412

[13] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

[14] Jiun-Chi Huang, Yi-Chun Tsai, Pei-Yu Wu, Yu-Hui Lien, Chih-Yi Chien, Chih-Feng Kuo, Jeng-Fung Hung, Szu-Chia Chen, and Chao-Hung Kuo. 2020. Predictive modeling of blood pressure during hemodialysis: a comparison of linear model, random forest, support vector regression, XGBoost, LASSO regression and ensemble method. *Computer Methods and Programs in Biomedicine* 195 (2020), 105536. https://doi.org/10.1016/j.cmpb.2020.105536

[15] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Springer, New York. https://doi.org/10.1007/978-1-4614-7138-7

[16] Vinzenz Lange, Paola Picotti, Bruno Domon, and Ruedi Aebersold. 2008. Selected reaction monitoring for quantitative proteomics: a tutorial. *Molecular systems biology* 4, 1 (2008), 222.

[17] Thomas Lumley and based on Fortran code by Alan Miller. 2020. *leaps: Regression Subset Selection*. https://CRAN.R-project.org/package=leaps R package version 3.1.

[18] Harald Marx, Simone Lemeer, Jan Erik Schliep, et al. 2013. A large synthetic peptide and phosphopeptide reference library for mass spectrometry–based proteomics. *Nat Biotechnol* 31 (May 2013), 557–564. https://doi.org/10.1038/nbt.2585

[19] Larry Medsker and Lakhmi C Jain. 1999. *Recurrent neural networks: design and applications*. CRC press.

[20] James L. Meek. 1980. Prediction of peptide retention times in high-pressure liquid chromatography on the basis of amino acid composition. *Proceedings of the National Academy of Sciences* 77, 3 (1980), 1632–1636. https://doi.org/10.1073/pnas.77.3.1632 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.77.3.1632

[21] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. 2021. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. https://CRAN.R-project.org/package=e1071 R package version 1.7-9.

[22] Luminita Moruz and Lukas Käll. 2017. Peptide retention time prediction. *Mass Spectrometry Reviews* 36, 5 (2017), 615–623. https://doi.org/10.1002/mas.21488

[23] Luminita Moruz, Daniela Tomazela, and Lukas Käll. 2010. Training, Selection, and Robust Calibration of Retention Time Models for Targeted Proteomics. *Journal of Proteome Research* 9, 10 (2010), 5209–5216. https://doi.org/10.1021/pr1005058 arXiv:https://doi.org/10.1021/pr1005058 PMID: 20735070.

[24] Jesper V. Olsen, Blagoy Blagoev, Florian Gnad, Boris Macek, Chanchal Kumar, Peter Mortensen, and Matthias Mann. 2006. Global, In Vivo, and Site-Specific Phosphorylation Dynamics in Signaling Networks. *Cell* 127, 3 (2006), 635–648. https://doi.org/10.1016/j.cell.2006.09.026

[25] Jesper V. Olsen, Michiel Vermeulen, Anna Santamaria, Chanchal Kumar, Martin L. Miller, Lars J. Jensen, Florian Gnad, Jürgen Cox, Thomas S. Jensen, Erich A. Nigg, Søren Brunak, and Matthias Mann. 2010. Quantitative Phosphoproteomics Reveals Widespread Full Phosphorylation Site Occupancy During Mitosis. *Science Signaling* 3, 104 (2010), ra3–ra3. https://doi.org/10.1126/scisignal.2000475 arXiv:https://www.science.org/doi/pdf/10.1126/scisignal.2000475

[26] Felix S. Oppermann, Florian Gnad, Jesper V. Olsen, Renate Hornberger, Zoltán Greff, György Kéri, Matthias Mann, and Henrik Daub. 2009. Large-scale Proteomics Analysis of the Human Kinome. *Molecular & Cellular Proteomics* 8, 7 (2009), 1751–1764. https://doi.org/10.1074/mcp.M800588-MCP200

[27] Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. https://doi.org/10.48550/ARXIV.1511.08458

[28] R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[29] Klarisa Rikova, Ailan Guo, Qingfu Zeng, Anthony Possemato, Jian Yu, Herbert Haack, Julie Nardone, Kimberly Lee, Cynthia Reeves, Yu Li, Yerong Hu, Zhiping Tan, Matthew Stokes, Laura Sullivan, Jeffrey Mitchell, Randy Wetzel, Joan MacNeill, Jian Min Ren, Jin Yuan, Corey E. Bakalarski, Judit Villen, Jon M. Kornhauser, Bradley Smith, Daiqiang Li, Xinmin Zhou, Steven P. Gygi, Ting-Lei Gu, Roberto D. Polakiewicz, John Rush, and Michael J. Comb. 2007. Global Survey of Phosphotyrosine Signaling Identifies Oncogenic Kinases in Lung Cancer. *Cell* 131, 6 (2007), 1190–1203. https://doi.org/10.1016/j.cell.2007.11.025

[30] So Young Ryu. 2014. Bioinformatics tools to identify and quantify proteins using mass spectrometry data. *Advances in Protein Chemistry and Structural Biology* 94 (2014), 1–17.

[31] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. 2011. Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software* 39, 5 (2011), 1–13. https://www.jstatsoft.org/v39/i05/

[32] Eric F Strittmatter, Lars J Kangas, Konstantinos Petritis, Heather M Mottaz, Gordon A Anderson, Yufeng Shen, Jon M Jacobs, David G Camp, and Richard D Smith. 2004. Application of peptide LC retention time information in a discriminant function for peptide identification by tandem mass spectrometry. *Journal of Proteome Research* 3, 4 (2004), 760–769.

[33] Dávid Virág, Borbála Dalmadi-Kiss, Károly Vékey, László Drahos, Imre Klebovich, István Antal, and Krisztina Ludányi. 2020. Current trends in the analysis of post-translational modifications. *Chromatographia* 83, 1 (2020), 1–10.

[34] Xiaocheng Wang, DaFang Fu, Yajun Wang, Ying Guo, and Yunfei Ding. 2021. The XGBoost and the SVM-based prediction models for bioretention cell decontamination effect. *Arabian Journal of Geosciences* 14, 8 (06 Apr 2021), 669. https://doi.org/10.1007/s12517-021-07013-6

[35] Sanford Weisberg. 2005. *Applied linear regression*. Vol. 528. John Wiley & Sons.

[36] Bo Wen, Kai Li, Yun Zhang, et al. 2020. Cancer neoantigen prioritization through sensitive and reliable proteogenomics analysis. *Nat Commun* 11 (April 2020), 1759. https://doi.org/10.1038/s41467-020-15456-w