nature methods

ARTICLES

https://doi.org/10.1038/s41592-021-01301-5

DeepLC can predict retention times for peptides that carry as-yet unseen modifications

Robbin Bouwmeester[1,2], Ralf Gabriels[1,2], Niels Hulstaert[1,2], Lennart Martens[1,2] ✉ and Sven Degroeve[1,2]

The inclusion of peptide retention time prediction promises to remove peptide identification ambiguity in complex liquid chromatography–mass spectrometry identification workflows. However, due to the way peptides are encoded in current prediction models, accurate retention times cannot be predicted for modified peptides. This is especially problematic for fledgling open searches, which will benefit from accurate retention time prediction for modified peptides to reduce identification ambiguity. We present DeepLC, a deep learning peptide retention time predictor using peptide encoding based on atomic composition that allows the retention time of (previously unseen) modified peptides to be predicted accurately. We show that DeepLC performs similarly to current state-of-the-art approaches for unmodified peptides and, more importantly, accurately predicts retention times for modifications not seen during training. Moreover, we show that DeepLC's ability to predict retention times for any modification enables potentially incorrect identifications to be flagged in an open search of a wide variety of proteome data.

Liquid chromatography plays a critical role in mass spectrometry (MS) analysis of bottom-up proteomics[1]. By separating peptides based on their physicochemical properties in the liquid chromatography step, the complexity of the sample presented to the MS instrument is greatly reduced. This reduction means that there is less ionization competition, improved sensitivity for data-dependent or -independent analysis and reduced chimericity in fragmentation spectra (MS^2)[2,3]. In addition to these benefits, the retention time measurement itself provides an additional dimension of information to interpret the signals generated by a peptide[4,5]. To interpret these acquired signals, they need to be matched with earlier observations of the same peptides or with a prediction of the signal.

The process by which a peptide is retained or eluted is not fully understood yet[6], which means that libraries with previously observed retention times are often used to match newly acquired signals[7]. However, these libraries are often incomplete and nontransferable between experimental setups without calibration. To fill this knowledge gap, researchers have therefore used models to predict retention times for previously unobserved peptides[5].

Many of the first methods for peptide retention time prediction relied on simulation models based on physicochemical knowledge[8]. In 1980, the first linear regression model that solely used total amino acid composition for peptide retention time was published[9]. In 2002, a method was proposed for incorporation of these predictions for increasing identification rates of proteins[10]. Improvements were then made to this modeling process, for example, by taking the positional peptide context into account[11]. Most modern approaches now use data-driven methods such as machine learning or deep learning algorithms to train a predictive model[10,12–15]. In these models, the mapping between the peptide sequence (or features derived from this sequence) and the liquid chromatography retention time apex is learned from empirical examples. After training, any of the aforementioned models can be used to generate predictions for unobserved peptides.

Such retention time prediction models have already been successfully applied for various tasks in proteomics analysis workflows, for example to improve identification confidence[16,17], to design more efficient experiments[18] and to identify chimeric fragmentation spectra[19]. Most recently, these retention time prediction models have been used in combination with fragment peak intensity prediction models to generate comprehensive, in silico chromatogram libraries for data independent acquisition (DIA) identification, effectively replacing and surpassing more limited, empirically derived data-dependent acquisition spectral libraries[20–22].

In keeping with the general trend in machine learning, there has been a switch from classical machine learning to deep learning in newly developed retention time predictors. This switch was mainly driven by recent innovations in the field of deep learning and the large amount of peptide retention time data that has become available. Because a deep learning network learns its own peptide representation, these models usually allow for more accurate predictions[23]. The types of architecture proposed by state-of-the-art deep learning retention time models include capsule convolutional neural networks (CNNs) in DeepRT(+)[15], neural networks with long short-term memory layers as used by Guan et al.[13] and an encoder–decoder principle with gated recurrent units in Prosit[14]. The architectures of these models either work with a CNN or recurrent architecture (for example, long short-term memory or gated recurrent units). CNN architectures slide a filter with a specified kernel size over the encoded peptide. In contrast, recurrent neural networks thread the sequence through a sequence of units.

All of these models share the same peptide encoding method, in which amino acids and their corresponding positions are transformed into a one-hot amino acid encoding. This encoding takes the form of a matrix in which the presence or absence of each amino acid for each position in the peptide is represented by a one or a zero, respectively. This use of one-hot encoding of amino acids restricts the models' applicability in some of the most interesting data analysis workflows, most notably in open searches where the goal is to elucidate the modification landscape of the proteome[24–27].

[1]VIB-UGent Center for Medical Biotechnology, VIB, Ghent, Belgium. [2]Department of Biomolecular Medicine, Ghent University, Ghent, Belgium. ✉e-mail: lennart.martens@vib-ugent.be

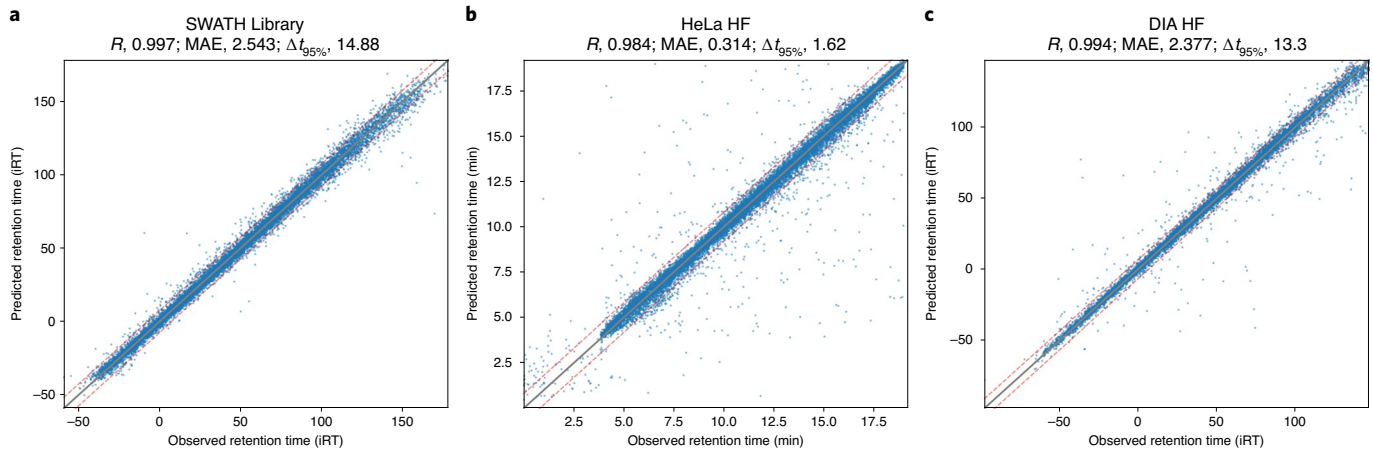

Fig. 1 | Scatter plots of predicted against observed on three of the largest data sets. **a**–**c**, Data sets are the SWATH Library (**a**), HeLa HF (**b**) and DIA HF (**c**).

These open search workflows are gaining popularity in the field of proteomics as they make it possible to search for a large variety of peptide modifications simultaneously. Unfortunately, current retention time prediction methods cannot be directly applied in open searches because of the vast number of potential modifications[28]. With one-hot amino acid encoding, each potential modification must be represented by a binary feature indicating the presence of this modification. Additionally, sufficient training examples are required for each modification for the machine learning algorithm to learn the hidden impact of every one of these modifications on the peptide retention time.

Here, we solve this fundamental issue with DeepLC, our retention time predictor that is able to accurately predict the retention time for all peptides and their modifications, even when these modifications have not been seen during training. DeepLC achieves this by encoding peptides and modifications at the atomic composition level, allowing generalization of the patterns learned from the modifications seen during training.

Results

The results section is split in two main parts. We first evaluate the performance of DeepLC on retention time prediction for unmodified peptides. We then proceed to evaluate DeepLC's unique ability to predict retention times for modified peptides. We rely on two distinct ways of evaluating DeepLC's performance on these modified peptides: (1) evaluate performance on unseen modifications, and (2) evaluate performance by treating unmodified amino acids as modified glycines. These evaluations show that DeepLC is not only competitive with state-of-the-art retention time prediction algorithms for unmodified peptides, but can also achieve similar performance for unseen modified peptides. Finally, we illustrate the unique capability of DeepLC by flagging potential false positive identifications in open searches of a variety of human tissue data sets.

Evaluation on unmodified peptides. Our approach to model amino acids by their atomic composition provides accurate predictions of liquid chromatography retention times for unmodified peptides (including carbamidomethylation of cysteine and oxidation of methionine), with similar performance to current state-of-the-art retention time prediction models DeepRT[15], Prosit[14] and Guan et al.[13] that model amino acids directly.

DeepLC test set predictions for the three selected data sets are plotted in Fig. 1. We observe very high prediction accuracy, with Pearson correlations larger than 0.98 for all three data sets. Hela HF data show a slightly worse performance, but here the liquid chromatography gradient is substantially shorter, indicating that retention times become less predictable. Indeed, this negative effect of shorter gradients on resolution and peak capacity are well known[29], making apex peptide elution times less predictable. Figure 1 also reveals a small but substantial number of peptides with high prediction errors. These are potentially wrong identifications or wrongly determined elution apexes. Most of these outliers fall within the worst 1% of predictions (Supplementary Fig. 1) and we expect up to 1% incorrect identifications due to the same set false discovery rate (FDR). The same plots for the other 17 data sets can be found in Supplementary Figs. 2 and 3 where we make very similar observations to those in Fig. 1.

Supplementary Table 1 summarizes the test set performance for all 20 data sets described in the data sets and evaluation section of the Methods. The atomic composition encoding approach of DeepLC is able to learn accurate prediction models with high R values for all data sets. Correlation provides a measure for how much variance is, and is not, explained by these predictions and allows for comparison between different liquid chromatography setups. For most data sets, DeepLC achieves an R above 0.98, with four data sets even achieving correlations above 0.995. This R value is highly comparable to the other models. Nearly all data sets were obtained with reverse phase columns, yet even though there are fewer data sets with hydrophilic interaction chromatography and strong ion exchange chromatography, DeepLC also performs very well on these data sets with relative mean absolute error (MAE) errors below 1.5%.

For the $\Delta t_{95\%}$ metric the differences are more pronounced. This metric describes the error for a retention time window that contains 95% of the peptides in the error distribution and is thus very sensitive to outliers. Here we observe that DeepLC performs consistently worse than the other models. It is, however, unclear whether these differences should be attributed to the atomic composition encoding, a different deep learning architecture, a difference in train-validation-test split (note that for the other prediction models, the paper does not mention the use of a validation set) or a combination of these. As we want to focus on the capability of DeepLC to predict retention times for modified peptides we leave this question open for further research.

The trained models are also highly transferrable between different data sets. This transferability is especially useful when applying models trained on larger data sets to smaller ones and application of the models without retraining. Only a simple calibration is required to transfer the predictions between liquid chromatography

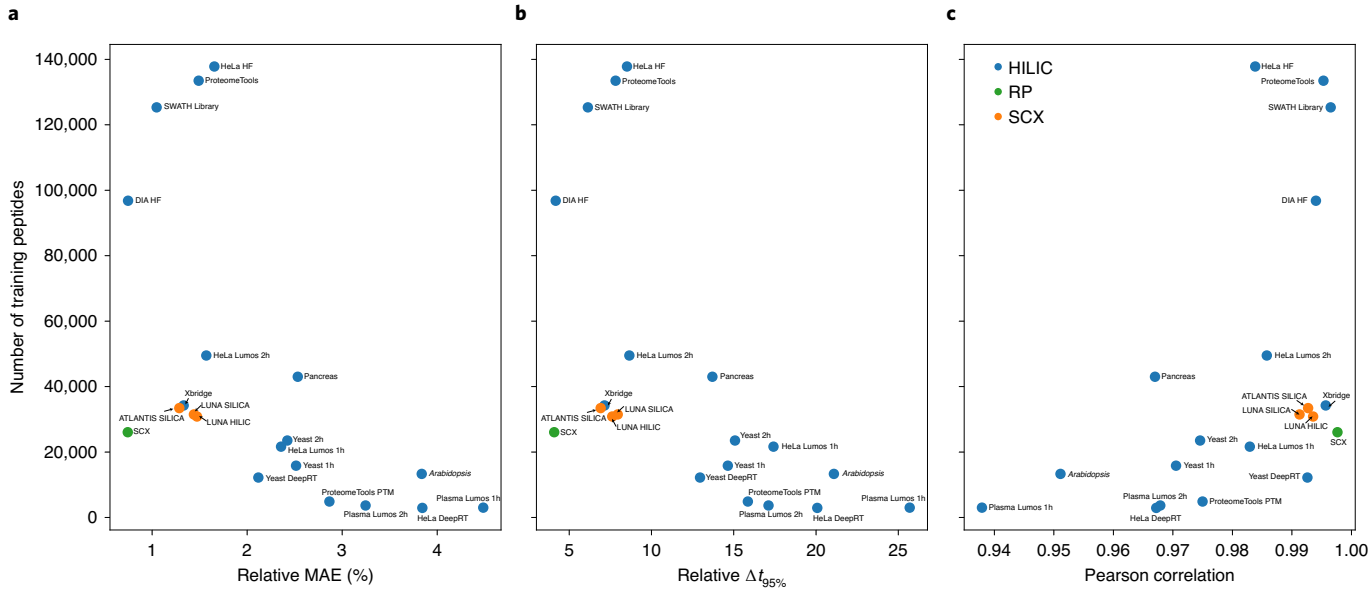

Fig. 2 | Prediction performance in terms of three metric for all data sets. a–c, For the 20 data sets, the number of training peptides (*y* axis) is plotted against the relative MAE (**a**), relative $\Delta t_{95\%}$ (**b**) and Pearson correlation (**c**). HILIC, hydrophilic interaction chromatography; RP, reverse phase; SCX, strong ion exchange chromatography.

setups. Supplementary Fig. 4 shows that models that achieve high performance on a given data set also show high Spearman correlation when applied to different data sets. The only exception is when different stationary phases are used, for example hydrophilic interaction liquid chromatography that retains hydrophilic peptides instead of hydrophobic peptides for reverse phase.

DeepLC builds on a deep learning approach that greatly benefits from a large number of training peptides, and we can show that large data sets indeed do have a positive influence on the performance of DeepLC. The performance on each individual data set in relation to the number of training peptides is shown in Fig. 2 and Supplementary Table 1. Data sets with a very small number of training peptides (<10,000) tend to have a performance between 2 and 4.5% relative MAE. For medium sized data sets (>10,000 and <75,000 peptides), the performance can vary widely, with relative MAE's ranging from 0.9 to 4.5%. For larger data sets (>75,000 peptides) the performance tends to be below 2% relative MAE. These larger data sets start to converge in terms of performance.

To further evaluate the relationship between the number of training peptides and prediction performance we computed learning curves for the three selected data sets (Fig. 3). These curves show a sharp improvement for the first four to five steps (comprising up to 50% of the total number of training peptides). Beyond these steps, prediction performance improves only linearly for the SWATH Library and HeLa HF, while showing smaller improvements in the last step for DIA HF. For two of these data sets, the performance continues to improve right to the last step of the learning curve. This ability to continuously improve performance suggests that DeepLC, like most other deep learning approaches, is capable of fitting even more complex relations than classical machine learning when provided with sufficient data. The same observation of increasing performance for larger training sets can be made for 15 of the remaining 17 data sets (Supplementary Figs. 5 and 6).

Evaluation on modified peptides. DeepLC is able to generalize effectively for unmodified peptides as well as extend its retention time predictions to modifications that were not included in the training set. We can thus show that the DeepLC models have not just learned the general shift in retention time caused by modifications, but also how this shift depends on the context of the modification in the peptide.

Prediction performance for modified peptides would ideally be evaluated on a large data set with a variety of modifications. Indeed, as shown in Fig. 3, the full performance potential of DeepLC is achieved by the largest possible data set size. However, such large data sets with many modifications are currently not available in the public domain.

Instead, we show DeepLC's prediction performance here for modified peptides on a recently published smaller data set (ProteomeTools PTM[30]). Furthermore, we introduce an evaluation procedure that allows the use of larger data sets based on the fact that any amino acid can be considered a modified glycine.

We first evaluate DeepLC on all 14 modifications in the ProteomeTools PTM data set. We trained and optimized 14 DeepLC models where each model only saw peptides that did not contain a specific modification. Each model was then evaluated on the remaining peptides, which all contained the modification that was excluded during training. We created two test sets from these remaining peptides to evaluate predictions: one where the excluded modification was encoded and one where it was not. Prediction performance for both test sets were then evaluated and compared. This comparison thus allows performance to be assessed on a modification that is not included in training in terms of the improvement that DeepLC offers over a baseline of simply ignoring the presence of the modification.

Figure 4 and Supplementary Fig. 7 show the prediction errors for each of the left-out modifications for training. Figure 4 shows the performance when a given modification was not present in the training set for the model and afterward was either not encoded (red boxplots, baseline) or encoded (blue boxplots) during the predictions. It should be noted that many modifications did not cause a substantial change in terms of predicted retention time, as was also observed in the original paper for this data set[30]. Examples of such modifications with limited impact are methyl, dimethyl, trimethyl and deamidation. In contrast, the acyl modifications (including propionyl, succinyl, malonyl, crotonyl and acetyl) showed a clear

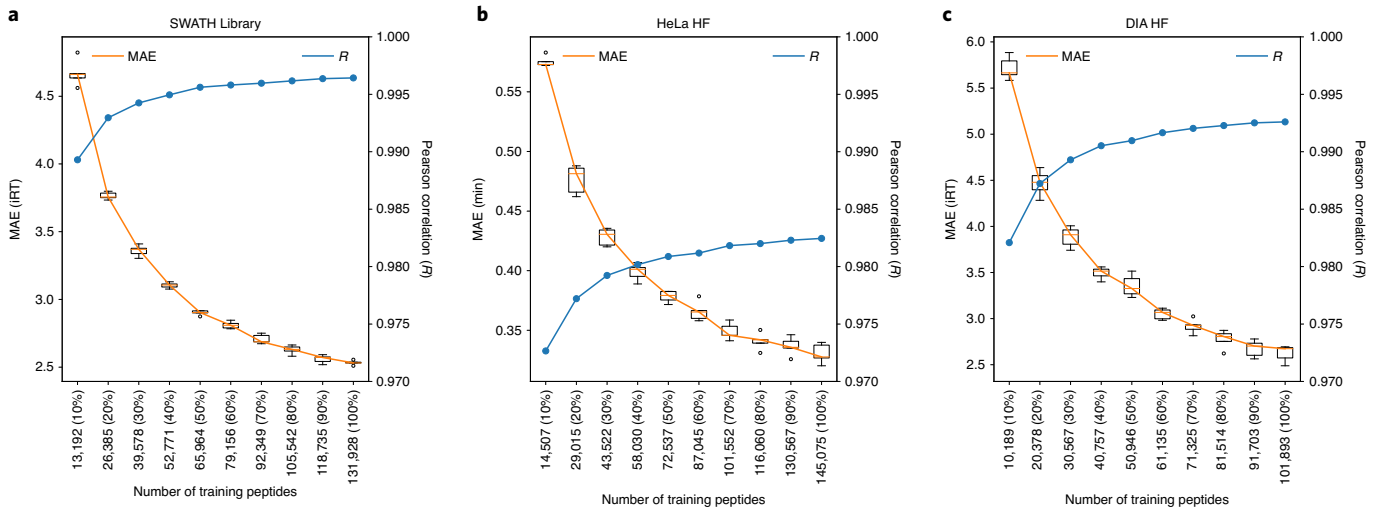

Fig. 3 | Learning curves for each of the three selected data sets. Prediction performances (*R* and MAE for five random subsamples at each step) for models trained on different training set sizes (*x* axis) are computed for a fixed test set. Boxplots show the median, Q1 (25%), Q3 (75%) and whiskers at Q ± 1.5 (Q3–Q1). **a**–**c**, SwathLibrary (**a**), HeLa HF (**b**) and DIA HF (**c**).

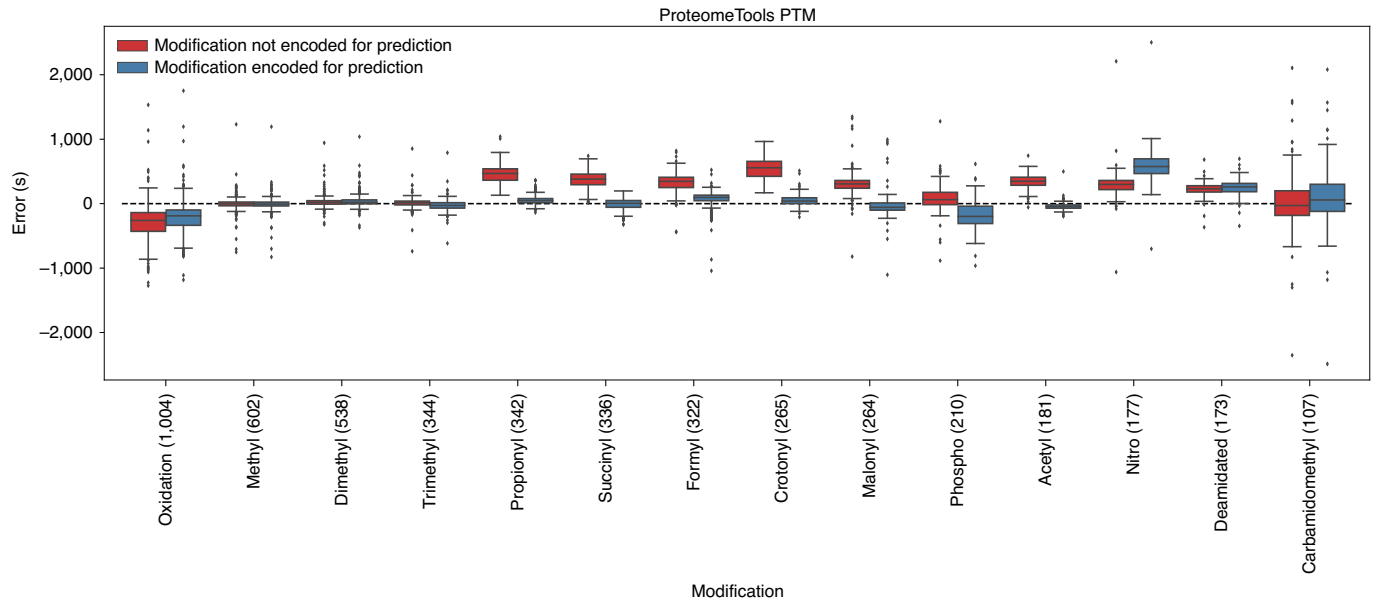

Fig. 4 | The modification that was excluded for training is shown on the horizontal axis, and the vertical axis shows the retention time error (experimental − predicted) when the modification was either not encoded (red) or encoded during the predictions (blue). Each modification name is followed by the number of peptides containing that modification. Boxplots show the median, Q1 (25%), Q3 (75%) and whiskers at Q ± 1.5 (Q3–Q1).

performance increase when these modifications were encoded during the predictions. For instance, Supplementary Fig. 7 shows that the MAE was improved by 700% (from 462 to 66 s) for propionyl. These improvements were mainly due to the correct prediction by DeepLC of the shift in retention time caused by the modification. Most importantly, besides a substantially decreased MAE, the correlation *R* also showed a substantial improvement. This is shown in Fig. 4 through the substantially smaller variance for the blue boxplots. For crotonyl, for instance, Supplementary Fig. 7 presents an increase of *R* from 0.975 to 0.990 when encoding the modification in the test set. This means that the DeepLC models did not just learn the general shift in retention time caused by modifications, but also how this shift depended on the context of the modification in the peptide.

Only nitrotyrosine and phosphorylation modifications show a substantially lower performance when encoded, but these modifications can be classified as physicochemically very different from the other modifications. This inability of DeepLC to accurately predict retention times for modifications that are chemically very different from anything encountered the training set indicates that even DeepLC requires some relevant training data for a given class of modifications.

In the second evaluation procedure, we used the larger DIA HF and the smaller HeLa HF data sets to train and optimize 19 DeepLC

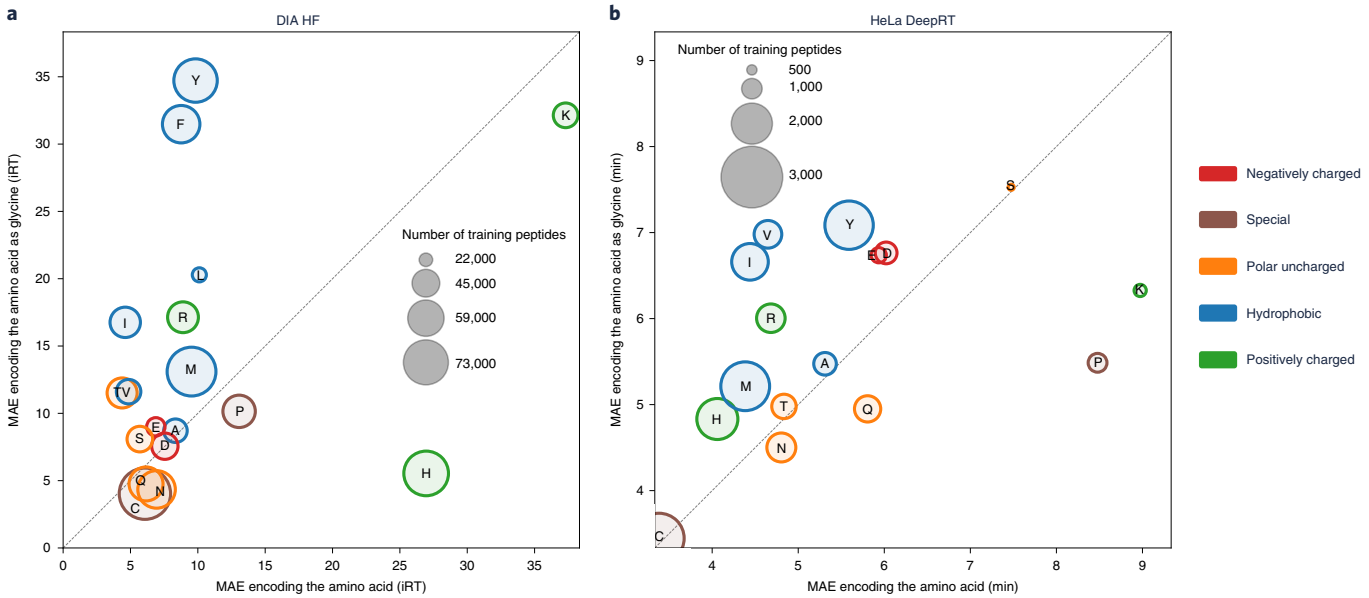

Fig. 5 | Each amino acid that was excluded for training is shown as a circle, where the size of the circle and color indicates the remaining training peptides and chemical property, respectively. The amino acid is either encoded as glycine (vertical axis) or as its own atomic composition (horizontal axis) and its position depicts the MAE for all amino acid containing peptides. This means that everything above the diagonal line is predicted with a higher accuracy when the amino acid is encoded as itself, while the reverse is true if it is below the diagonal line. **a**,**b**, DIA HF (**a**) and HeLa DeepRT (**b**).

models, where each model only saw peptides that did not contain a specific amino acid. The nomenclature was as above, in which a model was trained and optimized on peptides that did not contain a specific amino acid. Next, each model was evaluated on peptides that contained the amino acid excluded from training. For this, we again created two test sets from these remaining peptides: one where the excluded amino acid was encoded as the composition of glycine only and one with its actual composition.

We show that encoding an amino acid as itself instead of as glycine improves the MAE for most amino acids (Fig. 5). DeepLC performed very well when modeling large hydrophobic residues as modified glycines, and slightly less well when modeling polar uncharged and negatively charged residues. Finally, for the positively charged amino acids only arginine showed an improvement, while lysine and histidine decreased in performance. The poor performance for lysine can be explained by the difference between the amino acid and the closest atomic composition. For lysine, the closest atomic compositions are arginine and leucine (or isoleucine), which are substantially less hydrophobic or more hydrophobic, respectively. As shown previously, DeepLC was unable to extrapolate to unseen modifications that were very different in composition.

This nonmodified amino acid evaluation shows that performance is slightly worse in comparison to including the amino acid in the training set, with DIA HF and HeLa DeepRT having MAEs of 2.37 and 3.2 min, respectively. The MAE errors shown in Fig. 5 are about 1.5 to 2.5 times higher.

It is important to note that this evaluation is harsh because the trained model has never seen a given amino acid and, moreover, because peptides that are similar to each other are likely to all be excluded from training due to these peptides having a higher likelihood of also containing the removed amino acids. This can create biased training sets, especially for lysine and arginine as most peptides are tryptic. However, the model is still able to predict retention times very accurately for amino acids that were not used in training.

Evaluation of open identifications. Predicted retention times have the potential to overcome the identification ambiguity issue[23,31]. Because of DeepLC's unique capability to accurately predict retention times of (unseen) modifications, these predictions can be applied to open searches, where identification ambiguity is a key problem[25]. Indeed, open searches introduce considerable ambiguity through the very large number of possible modifications considered, which can be reduced through orthogonal measurements such as retention time.

DeepLC was applied to the results of an open search of human tissue data[32] using Open-pFind[26]. Figure 6a shows observed retention time plotted against predicted retention time for the resulting peptide spectrum matches (PSMs). While the retention time was accurately predicted for PSMs with a Q value <0.01, much higher retention time errors were observed for PSMs with $Q \geq 0.01$. The group of peptides with a $Q \geq 0.01$ also showed clustering around the predicted retention time of 1,000. These are PSMs for very hydrophilic peptides that were predicted to be nonretained. Figure 6b shows that PSMs with a $Q \geq 0.01$ did have a higher error, but the mode was still around zero. There is no substantial difference observed in the error distributions for unmodified and modified peptides (Supplementary Fig. 8). This indicates that we mostly expect false identifications to have their mode around zero with a large deviation from this mode.

The error distribution of filtered PSMs ($Q \geq 0.01$) is now compared to distributions of selected modifications to flag suspect modification groups, as the error distribution of suspect modifications is expected to be similar to filtered PSMs. Boxplots of the error distributions for four subsets of modifications are shown in Fig. 6c (see Supplementary Figs. 9–13 for all modifications). The subset containing the ten most found modifications all show a low error spread that is within 5% of the maximum elution time (300 s) and is generally centered around zero. The subset of modifications with the largest errors are in the range of 25% of the maximum elution time (1,500 s) and are widely spread around 0 s. A notable exception is dethiomethyl with a shifted median retention time error of 1,000 s. This shift can be explained by in-source fragmentation, which causes oxidized methionine to lose its side chain[33]. If in-source fragmentation is the cause, then the observed retention

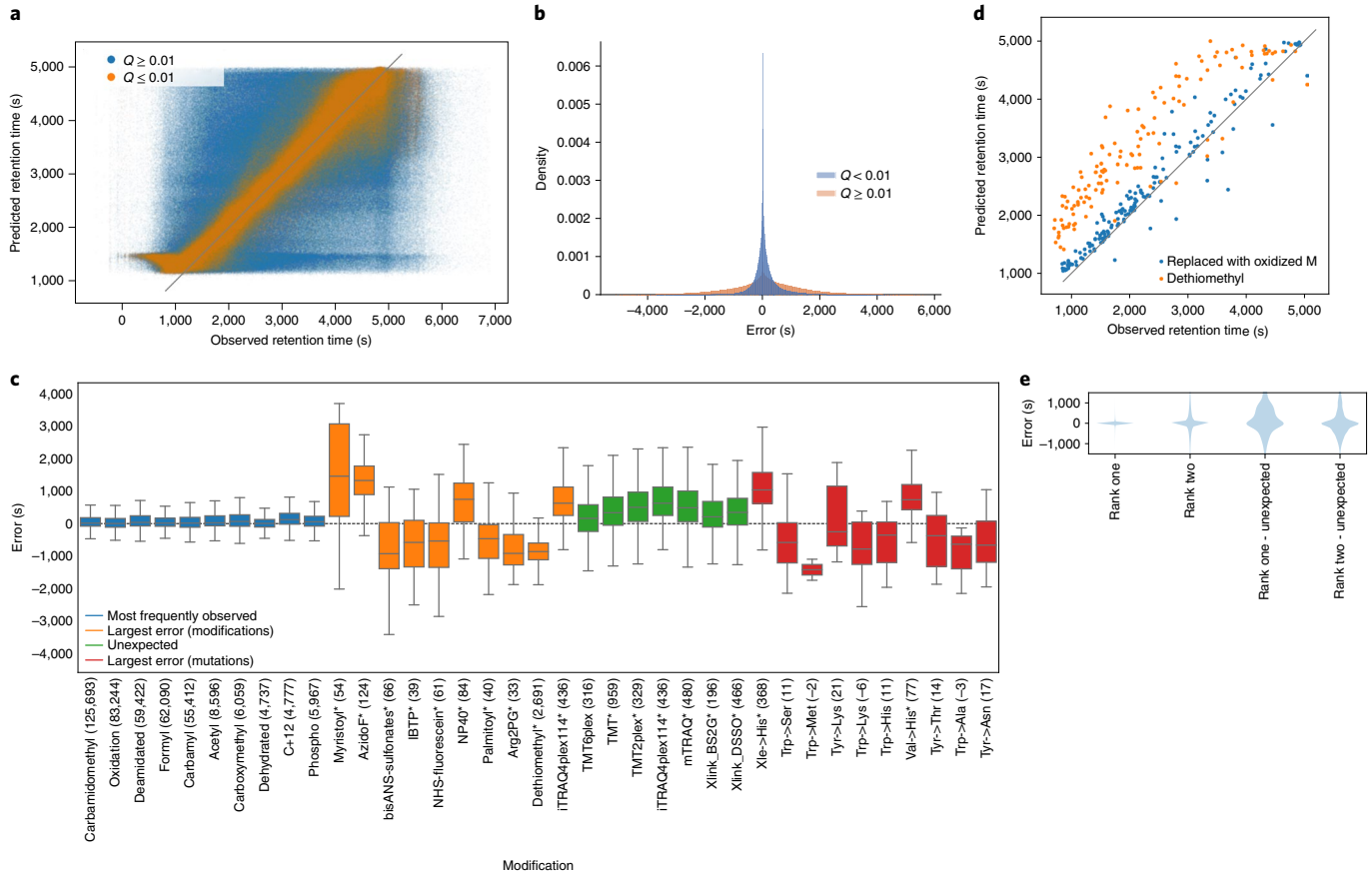

Fig. 6 | Predicted retention time analysis for open results of human tissue data. a, Predicted and observed retention times split by Q value, with 1,813,404 PSMs with a Q value equal to or above 0.01 and 689,438 PSMs with a Q value below 0.01. Q values are calculated with a FDR target-decoy approach. **b**, The error distribution split by Q value and any points that are higher than 1.5× the interquartile range plus the relevant quartile range are excluded from the plot. Q values are calculated with a FDR target-decoy approach. **c**, The PSMs with $Q < 0.01$. Colors indicate four different subsets of detected modifications. (1) PSMs carrying the top ten most abundant modifications, (2) PSMs carrying the ten modifications with the largest absolute mean error, (3) PSMs carrying modifications that are not expected to occur in the sample and (4) PSMs carrying the top ten mutations with the largest absolute mean error. Error distributions that are substantially different from the expected error distribution of carbamidomethyl are marked with a '*'-symbol. This difference is calculated by subtracting 20 equidistant percentiles from 5 to 100% of both distributions and calculating the summed absolute difference. The 25% error distributions with largest distance are marked as substantially different. Modifications are followed by the number of peptides identified with the modification in brackets. Boxplots show the median, Q1 (25%), Q3 (75%) and whiskers at $Q \pm 1.5$ (Q3–Q1). **d**, PSMs identified with a dethiomethyl modification in CD8 T cell data, and predictions for these but with dethiomethyl replaced with oxidation. **e**, Error distributions as violin plots for all rank one PSMs ($Q < 0.01$) and their corresponding second ranked PSMs. The same rank one and rank two PSMs are visualized for rank one identifications with unexpected modifications.

times are expected to be based on the oxidized methionine equivalent. To verify whether this is the case, PSMs with dethiomethyl were replaced with their oxidized methionine precursors. Figure 6d shows that replacing detiomethyl with oxidized methionine reduced the predicted retention time error to around the same level as expected for oxidized methionine peptides in Fig. 6c.

The subset of modifications with large errors shows very similar patterns to the next subset, which contains modifications that are not expected to occur in the sample, as these are experimentally induced and thus should not be found in the untreated biological sample. In effect, the similarity between these two subsets of modifications indicates that modifications with large retention time errors according to DeepLC can be flagged as highly suspect. These PSMs with unexpected modifications were then inspected for their associated second rank PSM in Fig. 6e. For all PSMs, first ranked PSMs are shown to have the narrowest error distribution. In contrast, when considering only PSMs with unexpected modifications, the second ranked PSMs display the narrower error distribution. This difference indicates that DeepLC might well be able to select better alternatives for these unexpected modifications, as judged by the generally better fit of these alternatives' retention times.

Finally, the last subset singles out presumed detection of mutations that show similar or worse error distributions than the previous two subsets. The observation that presumed mutations are among the most problematic corresponds to the known nontrivial nature of reliably identifying such sequence changes[31,34].

These results thus demonstrate that the unique capabilities of DeepLC allow it to be used as an orthogonal measure to flag suspect identifications in open searches. As a proof-of-concept, we here show that a comparison of error distributions between expected and potentially falsely identified modifications can select those modification distributions that are likely the result of incorrect identifications. This method allows the most suspect modifications to be selected, and can be configured to be more conservative or more lenient based on the needs of the analysis.

Discussion

Our evaluation shows that DeepLC performed similarly to current state-of-the-art models for unmodified peptides, but DeepLC could accurately predict the retention time of modified peptides, even for modifications that were not included in the training set. This ability to predict for unseen modifications was evaluated with a two-pronged evaluation strategy using both unmodified peptides as well as synthetic, modified peptides. For both evaluations, encoding modifications for prediction improved performance, while performance was reduced only for specific modifications that were very different from any other structure in the data set. Finally, the potential of this unique capability of DeepLC was illustrated through its ability to flag suspect PSMs in an open search. Crucially, DeepLC showed much larger prediction errors for PSMs that carried modifications that were certain to be absent from the sample.

Future development of models that can predict the retention time for unseen modifications could focus on the structural aspects of modifications. DeepLC is currently limited in differentiating between isomeric structures that are physicochemically different. Indeed, the observation that structure, not just atomic composition, leads to the physicochemical properties of molecules has already been observed for small molecules[35,36]. Here, the decision was made to work with atomic composition because of the ready availability of the composition in databases such as Unimod, and the greater ease of integration when compared to more complex structural descriptors.

DeepLC enables the field to generate predictions for a wide landscape of modification. To improve the availability to researchers and their use cases, DeepLC is freely available online and has a user-friendly graphical user interface (GUI). Furthermore, the tool is available in code repositories that enable easy incorporation in workflows and pipelines for automatic predictions.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-021-01301-5.

Received: 15 April 2020; Accepted: 13 September 2021;
Published online: 28 October 2021

References

1. Aebersold, R. & Mann, M. Mass spectrometry-based proteomics. *Nature* **422**, 198–207 (2003).
2. Shishkova, E., Hebert, A. S. & Coon, J. J. Now, more than ever, proteomics needs better chromatography. *Cell Syst.* **3**, 321–324 (2016).
3. Michalski, A., Cox, J. & Mann, M. More than 100,000 detectable peptide species elute in single shotgun proteomics runs but the majority is inaccessible to data-dependent LC–MS/MS. *J. Proteome Res.* **10**, 1785–1793 (2011).
4. Bruderer, R. et al. Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues*[S]. *Mol. Cell. Proteom.* **14**, 1400–1410 (2015).
5. Moruz, L. & Käll, L. Peptide retention time prediction. *Mass Spectrom. Rev.* **36**, 615–623 (2017).
6. Reimer, J., Spicer, V. & Krokhin, O. V. Application of modern reversed-phase peptide retention prediction algorithms to the Houghten and DeGraw dataset: peptide helicity and its effect on prediction accuracy. *J. Chromatogr. A.* **1256**, 160–168 (2012).
7. Searle, B. C. et al. Chromatogram libraries improve peptide detection and quantification by data independent acquisition mass spectrometry. *Nat. Commun.* **9**, 5128 (2018).
8. Guo, D., Mant, C. T., Taneja, A. K. & Hodges, R. S. Prediction of peptide retention times in reversed-phase high-performance liquid chromatography II. Correlation of observed and predicted peptide retention times factors and influencing the retention times of peptides. *J. Chromatogr. A.* **359**, 519–532 (1986).
9. Meek, J. L. Prediction of peptide retention times in high-pressure liquid chromatography on the basis of amino acid composition. *Proc. Natl Acad. Sci. USA* **77**, 1632–1636 (1980).
10. Palmblad, M., Ramström, M., Markides, K. E., Håkansson, P. & Bergquist, J. Prediction of chromatographic retention and protein identification in liquid chromatography/mass spectrometry. *Anal. Chem.* **74**, 5826–5830 (2002).
11. Moruz, L., Tomazela, D. & Käll, L. Training, selection, and robust calibration of retention time models for targeted proteomics. *J. Proteome Res.* **9**, 5209–5216 (2010).
12. Moruz, L. et al. Chromatographic retention time prediction for posttranslationally modified peptides. *Proteomics* **12**, 1151–1159 (2012).
13. Guan, S., Moran, M. F. & Ma, B. Prediction of LC-MS/MS properties of peptides from sequence by deep learning. *Mol. Cell. Proteom.* **18**, 2099–2107 (2019).
14. Gessulat, S. et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nat. Methods* **16**, 509–518 (2019).
15. Ma, C. et al. Improved peptide retention time prediction in liquid chromatography through deep learning. *Anal. Chem.* **90**, 10881–10888 (2018).
16. MacLean, B. et al. Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics* **26**, 966–968 (2010).
17. C Silva, A. S. et al. Accurate peptide fragmentation predictions allow data driven approaches to replace and improve upon proteomics search engine scoring functions. *Bioinformatics* **35**, 1401–1403 (2019).
18. Bertsch, A. et al. Optimal de novo design of MRM experiments for rapid assay development in targeted proteomics. *J. Proteome Res.* **9**, 2696–2704 (2010).
19. Dorfer, V., Maltsev, S., Winkler, S. & Mechtler, K. CharmeRT: boosting peptide identifications by chimeric spectra identification and retention time prediction. *J. Proteome Res.* **17**, 2581–2589 (2018).
20. Van Puyvelde, B. et al. Removing the hidden data dependency of DIA with predicted spectral libraries. *Proteomics* **20**, 1900306 (2020).
21. Yang, Y. et al. In silico spectral libraries by deep learning facilitate data-independent acquisition proteomics. *Nat. Commun.* **11**, 146 (2020).
22. Searle, B. C. et al. Generating high quality libraries for DIA MS with empirically corrected peptide predictions. *Nat. Commun.* **11**, 1548 (2020).
23. Bouwmeester, R., Gabriels, R., Van Den Bossche, T., Martens, L. & Degroeve, S. The age of data-driven proteomics: how machine learning enables novel workflows. *Proteomics* **20**, 1900351 (2020).
24. Bittremieux, W., Meysman, P., Noble, W. S. & Laukens, K. Fast open modification spectral library searching through approximate nearest neighbor indexing. *J. Proteome Res.* **17**, 3463–3474 (2018).
25. Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D. & Nesvizhskii, A. I. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. *Nat. Methods* **14**, 513–520 (2017).
26. Chi, H. et al. Comprehensive identification of peptides in tandem mass spectra using an efficient open search engine. *Nat. Biotechnol.* **36**, 1059–1066 (2018).
27. Na, S., Bandeira, N. & Paek, E. Fast multi-blind modification search through tandem mass spectrometry. *Mol. Cell Proteomics* **11**, M111.010199 (2012).
28. Creasy, D. M. & Cottrell, J. S. Unimod: protein modifications for mass spectrometry. *Proteomics* **4**, 1534–1536 (2004).
29. Wren, S. A. C. Peak capacity in gradient ultra performance liquid chromatography (UPLC). *J. Pharm. Biomed. Anal.* **38**, 337–343 (2005).
30. Paul Zolg, D. et al. Proteometools: systematic characterization of 21 post-translational protein modifications by liquid chromatography tandem mass spectrometry (LC-MS/MS) using synthetic peptides. *Mol. Cell. Proteom.* **17**, 1850–1863 (2018).
31. Colaert, N., Degroeve, S., Helsens, K. & Martens, L. Analysis of the resolution limitations of peptide identification algorithms. *J. Proteome Res.* **10**, 5555–5561 (2011).
32. Kim, M. S. et al. A draft map of the human proteome. *Nature* **509**, 575–581 (2014).
33. Müller, T. & Winter, D. Systematic evaluation of protein reduction and alkylation reveals massive unspecific side effects by iodine-containing reagents. *Mol. Cell. Proteom.* **16**, 1173–1187 (2017).
34. Salz, R. et al. Personalized proteome: comparing proteogenomics and open variant search approaches for single amino acid variant detection. *J. Proteome Res.* **20**, 3353–3364 (2021).
35. Aicheler, F. et al. Retention time prediction improves identification in nontargeted lipidomics approaches. *Anal. Chem.* **87**, 7698–7704 (2015).
36. Creek, D. J. et al. Toward global metabolomics analysis with hydrophilic interaction liquid chromatography–mass spectrometry: improved metabolite identification by retention time prediction. *Anal. Chem.* **83**, 8703–8710 (2011).

Methods

Architecture. DeepLC uses a convolutional deep learning architecture with four different paths for a given encoded peptide. The same peptide acts as the input for the four paths, which have multiple separated layers, as shown in Supplementary Fig. 14. Three of the initial paths use a combination of convolutional[37] and maximum pooling layers[38]. The paths with convolutional layers use a sliding window filter approach to encode local structure in the peptide. The maximum pooling layers further generalize the encoding of the convolutional filters by only propagating the maximum activations. The remaining path, which propagates global features, consists of densely connected layers. These densely connected layers do not take local structures into account, but this is not required as this path is meant to encode global structure only. The results of all initial four paths are flattened and concatenated to provide an input for the final combined path that consists of six connected dense layers. A detailed visualization of the architecture is available in Supplementary Fig. 15.

The input matrix for the amino acids composition path has a dimension of 60 for the peptide sequence by six for the atom counts (C, H, N, O, P and S). Not every peptide is 60 amino acids long, thus 'X'-characters without atomic composition are padded to reach 60 amino acids. This indicates that encoding modified amino acids becomes straightforward, as computing their atomic composition is trivial. Note that for modified amino acids, the atomic composition of the modification is added to the atomic composition of the unmodified residue. This encoding allows the model to learn patterns that generalize to unseen modifications.

The diamino acid path is added to further improve the generalization capability of the model. In this layer, the peptide is divided into diamino acids without overlap. This improves the generalization capability, as the input values for each position are more thoroughly represented. Otherwise there would only be 20 unmodified amino acid representations, combined with a limited amount of modifications. Besides interpreting the amino acids in pairs, the diamino acid path uses the same logic as the amino acids composition path, leading to an input matrix of 30 paired positions by six atoms.

Encoding amino acids and their modifications by strictly using the atomic composition does, however, not allow for comprehensively capturing all molecular information. Indeed, the structure of isomers can play an important role in the physicochemical properties of amino acids, as is exemplified by structural isomers isoleucine and leucine[39]. This is the reason that one-hot encoding of unmodified amino acids is still used in DeepLC as an input for the One-hot encoding path. However, to reduce the impact of this layer, the number of filters for this path are limited to two. The dimensions of this input matrix are 60 positions by 20 amino acids.

In addition to all paths that encode position specific information, the Global features path takes global information of the peptide into account. These global features include the length and total atomic composition of the peptide. In addition to these global counts and length, the atomic composition of the first and last four positions of the peptide are encoded. This adds a 6×8 feature matrix, or a flattened feature vector of 48. The dimension of this input vector is 55.

Three versions of the model were trained, solely differing in kernel sizes (of 2, 4 and 8) for the amino acids composition path. These three models were combined in an ensemble by averaging their predictions. This strategy is similar to the ensemble used in DeepRT+ (ref. [15]) and ensures adaptability to different data sets that might require encoding of longer local peptide structures.

The paths were optimized on the validation set of the SWATH Library data set. This optimization consisted of selecting the number of convolutional and maximum pooling blocks for the amino and diamino acids composition paths that yielded a lower MAE. For the diamino acids composition path, we chose to not encode redundant information and thus the encoding was nonoverlapping. The rationale was to limit the already redundant information within and between the diamino and amino acids composition paths. However, as with many of the architecture decisions, there is no guarantee that the chosen hyperparameters of the architecture provide a global or even local optimum. Inspection of the weights of the dense layer after concatenation shows that all paths propagate activations and thus contribute to the predictions (Supplementary Fig. 16).

Finally, the other hyperparameters of each layer in DeepLC are consistent for all versions with different kernel sizes. All layers, except the output layer and the one-hot encoding path, use L1 regularization with $\alpha=2.5\times10^{-7}$ and a leaky ReLU[40] with a maximum activation value of 20. The one-hot encoding path uses the tanh activation function, as within this path we are only interested in the ability to separate unmodified amino acid isomers.

Data sets and evaluation. To evaluate the generalization performance of DeepLC, we selected 20 data sets from a wide variety of organisms and experimental setups (Supplementary Table 2). We further selected three data sets (SWATH Library[41], HeLa HF[42] and DIA HF[43]) for detailed result reporting, with the results for the other 17 data sets described in the Supplementary Information. The data sets SWATH Library and DIA HF were selected based on their previous use by Ma et al. for DeepRT[15] and by Guan et al.[13], respectively. A third data set, HeLa HF was selected because of its use of short (compared to other used data sets) gradients of 15 min and the large number of training peptides. Only unique peptidoforms (peptide modifications combination) were used for training or, as indicated by the reference, the previously published data set was used.

The variety in experimental setups and protocols means that the acquired and predicted retention times had to be calibrated. The ProteomeTools library[44], SWATH Library and DIA HF data sets were normalized to the indexed retention time (iRT) peptides[45,46]. DeepLC itself supports linear calibration that is similar to iRT calibration[45], but users can supply their own high-confidence identification. This calibration procedure is further explained in the online DeepLC documentation.

The data sets marked 'custom workflow' in Supplementary Table 2 were processed as follows. Raw MS files were downloaded from PRIDE Archive[47] and converted to MGF format with the ThermoRawFileParser[48]. These were then searched using the MS-GF+ search engine[49] with a concatenated target–decoy sequence database containing the respective species' UniProtKB proteome and the common Repository of Adventitious Proteins (https://www.thegpm.org/crap/). Carbamidomethylation of cysteine was set as a fixed modification, oxidation of methionine and acetylation of protein N-termini were set as variable modifications. The full MS-GF+ configuration files for each data set are available on Zenodo. The MS-GF+ search results were postprocessed with Percolator[50] to a FDR of 0.01. Retention times were parsed from the MGF files for all confidently identified peptides. Within each liquid chromatography–mass spectrometry (LC–MS) run, the median retention time for each peptidoform (peptide modifications combination) was calculated. All median retention times were then linearly calibrated across all LC–MS runs for each data set, using the shared peptidoforms as anchor points. Finally, the median calibrated retention time was calculated for each peptidoform across all runs for each data set. These median calibrated retention times were then used to train, validate and test DeepLC. The full custom workflow, including this calibration step, is available in a Snakemake workflow[51].

The data sets marked Custom workflow ProteomeTools in Supplementary Table 2 were processed as follows. MaxQuant[52] identification files were filtered on posterior error probabilities <0.01 and scores >90. The retention times were calibrated with the peptides in Supplementary Table 3. Within a run, the median retention time per peptidoform was used for further analysis. Then, across runs the median retention time per peptidoform was taken for the final retention time.

The data sets marked custom workflow ProteomeTools PTM[30] in Supplementary Table 2 were processed in the same way as the data sets custom workflow ProteomeTools, with the only exception that the retention times were calibrated with the peptides in Supplementary Table 4. A few modifications from the original publication were either grouped or ignored. The modifications 'hydroxyproline' and 'hydroxyisobutyrylation' were grouped under 'oxidation'. The modifications 'monomethylation' and 'dimethylation' on both arginine and lysine were grouped. The modifications 'glutarylation' and 'glyglycylation' were excluded due to the same naming scheme that did not allow for discriminating between them. Finally, 'biotinylation' was excluded due to its uniquely large size.

Each data set was randomly split into a test set (10%), validation set (5%) and training set (85%). The complete set of peptides for all data sets, and which split these were part of, are listed in Supplementary Table 5. The validation set is used for model selection only while all performance results presented here were computed from the test set. Prediction performance is measured using three commonly used metrics: MAE, Pearson correlation and $\Delta t_{95\%}$. The last describes the error for a retention time window that contains 95% of the peptides in the error distribution. To make the MAE and $\Delta t_{95\%}$ comparable between experiments, we divided them by the retention time of the difference between the first and last detected peptide in the respective data set. These metrics are further referred to as relative MAE and relative $\Delta t_{95\%}$.

Training procedure. All models trained are initialized with random weights from a normal distribution ($\mu=0.0$ and $\sigma=1.0$). Two NVIDIA Geforce RTX 2080 TIs graphic cards are used for training. The training consisted of 100 epochs with early stopping on a validation set. Most data sets triggered early stopping around 30 epochs, while larger data sets (>75,000 peptidoforms) triggered early stopping at around 50 epochs.

Open search. Results from an open search by Open-pFind[26] are used to evaluate the ability of DeepLC to identify suspect identifications. Open-pFind allows for open searches by combining an MS^2 tag search and a two-stage (modification restrictive and open) search that is optimized with a Percolator equivalent. Even though Open-pFind uses this sophisticated search strategy open searches are still particularly prone to falsely identify modified peptides. This high false identification rate is due to the larger search space and resulting identification ambiguity[46]. The search was performed on a data set on 17 adult and seven fetal tissues by Kim et al.[32], obtained from the PRIDE repository[53] with identifier PXD000561. The search was run with Open-pFind v.3.1.5. Search parameters were set to: 20 ppm precursor mass error tolerance, peptide length limit from seven to 30 amino acids, modifications were limited to mass deltas from −150 to 500 Da, a maximum of two miscleavages were allowed, and oxidation of M and carbamidomethylation of C were set as variable modifications.

Observed and predicted retention time error distributions that are substantially different from that of carbamidomethyl are marked with a '*'-symbol. This difference is calculated by subtracting 20 equidistant percentiles from 5 to 100% of

both distributions and calculating the summed absolute difference. The 25% error distributions with the largest distance are marked as substantially different.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Data from the following projects were used to train and evaluate DeepLC: HeLa hf[42], ProteomeTools[44], SWATH Library[41], Plasma lumos 1h[54], DIA HF[43], HeLa lumos 2h[54], Pancreas[55], Xbridge[56], ATLANTIS SILICA[56], LUNA SILICA[56], LUNA hydrophilic interaction chromatography[56], strong ion exchange[56], Yeast 2h[57], HeLa lumos 1h[54], Yeast 1h[57], *Arabidopsis*[58], Yeast DeepRT[59], ProteomeTools PTM[30], Plasma lumos 2h[54] and HeLa DeepRT[60]. The files of each data set and open search results are available on Zenodo at https://zenodo.org/record/4542884. The raw files the open search was performed on are available at PRIDE repository under the identifier PXD000561 (ref. [32]).

Code availability

The following Python (v.3.6) libraries were used in DeepLC: Pandas (v.0.25.1)[61], TensorFlow (v.1.14.0)[62], Pyteomics (v.4.1.2)[63], SciPy (v.1.4.0)[64], matplotlib (v.3.1.3)[65], seaborn (v.0.10.0)[66] and Numpy (v.1.17.3)[67]. Other software used for DeepLC are: ThermoRawFileParser[48] (v.1.2.0), FileZilla (v.3.48.1), MS-GF+ (ref. [49]) (v.2019.08.26), Percolator[68] (v3.4) and open-pFind[26] (v.3.1.5). Code used to prepare the data sets, calibrate retention times, generate DeepLC models, make predictions and to reproduce the figures is available on Zenodo at https://zenodo.org/record/4542884.

The DeepLC tool including a GUI (Supplementary Fig. 17) is available for download from the following repositories and package indexes:

- GUI: https://github.com/compomics/DeepLC/releases/latest
- Python package: https://pypi.org/project/deeplc/
- Bioconda package: https://bioconda.github.io/recipes/deeplc/README.html
- Biocontainers docker image: https://quay.io/repository/biocontainers/deeplc
- Streamlit webserver: https://iomics.ugent.be/deeplc/
- Source code: https://github.com/compomics/DeepLC.

References

37. Fukushima, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Netw.* **1**, 119–130 (1988).
38. Ranzato, M., Huang, F., Boureau, Y. B. & LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proc. 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA* (IEEE, 2007).
39. Parker, J. M. R., Guo, D. & Hodges, R. S. New hydrophilicity scale derived from high-performance liquid chromatography peptide retention data: Correlation of predicted surface residues with antigenicity and X-ray-derived accessible sites. *Biochemistry* **25**, 5425–5432 (1986).
40. Nair, V. & Hinton, G. E. *Rectified Linear Units Improve Restricted Boltzmann Machines* https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf (Univ. Toronto, 2010).
41. Rosenberger, G. et al. A repository of assays to quantify 10,000 human proteins by SWATH-MS. *Sci. Data* **1**, 140031 (2014).
42. Kelstrup, C. D. et al. Performance evaluation of the Q exactive HF-X for shotgun proteomics. *J. Proteome Res.* **17**, 727–738 (2018).
43. Bruderer, R. et al. Optimization of experimental parameters in data-independent mass spectrometry significantly increases depth and reproducibility of results. *Mol. Cell. Proteom.* **16**, 2296–2309 (2017).
44. Zolg, D. P. et al. Building ProteomeTools based on a complete synthetic human proteome. *Nat. Methods* **14**, 259–262 (2017).
45. Escher, C. et al. Using iRT, a normalized retention time for more targeted measurement of peptides. *Proteomics* **12**, 1111–1121 (2012).
46. Zolg, D. P. et al. PROCAL: A set of 40 peptide standards for retention time indexing, column performance monitoring, and collision energy calibration. *Proteomics* **17**, 1700263 (2017).
47. Martens, L. et al. PRIDE: the proteomics identifications database. *Proteomics* **5**, 3537–3545 (2005).
48. Hulstaert, N. et al. ThermoRawFileParser: modular, scalable, and cross-platform RAW file conversion. *J. Proteome Res.* **19**, 537–542 (2020).
49. Kim, S. & Pevzner, P. A. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nat. Commun.* **5**, 5277 (2014).
50. Käll, L., Canterbury, J. D., Weston, J., Noble, W. S. & MacCoss, M. J. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods* **4**, 923–925 (2007).
51. Köster, J. & Rahmann, S. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics* **28**, 2520–2522 (2012).
52. Cox, J. & Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat. Biotechnol.* **26**, 1367–1372 (2008).
53. Vizcaíno, J. A. et al. ProteomeXchange provides globally coordinated proteomics data submission and dissemination. *Nat. Biotechnol.* **32**, 223–226 (2014).
54. Li, W. et al. Assessing the relationship between mass window width and retention time scheduling on protein coverage for data-independent acquisition. *J. Am. Soc. Mass. Spectrom.* **30**, 1396–1405 (2019).
55. Wang, D. et al. A deep proteome and transcriptome abundance atlas of 29 healthy human tissues. *Mol. Syst. Biol.* **15**, e8503 (2019).
56. Gussakovsky, D., Neustaeter, H., Spicer, V. & Krokhin, O. V. Sequence-specific model for peptide retention time prediction in strong cation exchange chromatography. *Anal. Chem.* **89**, 11795–11802 (2017).
57. Jarnuczak, A. F. et al. Analysis of intrinsic peptide detectability via integrated label-free and SRM-based absolute quantitative proteomics. *J. Proteome Res.* **15**, 2945–2959 (2016).
58. Mucha, S. et al. The formation of a camalexin biosynthetic metabolon. *Plant Cell* **31**, 2697–2710 (2019).
59. Nagaraj, N. et al. System-wide perturbation analysis with nearly complete coverage of the yeast proteome by single-shot ultra HPLC runs on a bench top orbitrap. *Mol. Cell. Proteomics* **11**, M111.013722 (2012).
60. Sharma, K. et al. Ultradeep human phosphoproteome reveals a distinct regulatory nature of Tyr and Ser/Thr-based signaling. *Cell Rep.* **8**, 1583–1594 (2014).
61. McKinney, W. pandas: a foundational Python library for data analysis and statistics. *Python High Perform. Sci. Comput.* 1–9, https://www.dlr.de/sc/en/Portaldata/15/Resources/dokumente/pyhpc2011/submissions/pyhpc2011_submission_9.pdf (2011).
62. Abadi, M. et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. Preprint at *arXiv.org* www.tensorflow.org
63. Levitsky, L. I., Klein, J. A., Ivanov, M. V. & Gorshkov, M. V. Pyteomics 4.0: five years of development of a python proteomics framework. *J. Proteome Res.* **18**, 709–714 (2019).
64. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
65. Hunter, J. D. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* **9**, 90–95 (2007).
66. Waskom, M. L. Seaborn: statistical data visualization. *J. Open Source Softw.* **6**, 3021 (2021).
67. Oliphant, T. E. *A Guide to NumPy* Vol. 1 (Trelgol Publishing, 2006).
68. The, M., MacCoss, M. J., Noble, W. S. & Käll, L. Fast and accurate protein false discovery rates on large-scale proteomics data sets with percolator 3.0. *J. Am. Soc. Mass. Spectrom.* **27**, 1719–1727 (2016).

Acknowledgements

R.B. acknowledges funding from the Marie Sklodowska-Curie EU Framework for Research and Innovation Horizon 2020 MASSTRPLAN (grant no. 675132) and Vlaams Agentschap Innoveren en Ondernemen under project number HBC.2020.2205. R.G. acknowledges funding from the Research Foundation Flanders (FWO) (grant no. 1S50918N). S.D. and L.M. acknowledge funding from the European Union's Horizon 2020 Programme (grant nos. H2020-INFRAIA-2018-1 and 823839). N.H. and L.M. acknowledge funding from the Research Foundation Flanders (FWO) (grant nos. G042518N and G028821N). L.M. acknowledges funding from Ghent University Concerted Research Action (grant no. BOF21-GOA-033).

Author contributions

R.B., R.G. and S.D. conceived the study. R.B., R.G., L.M. and S.D. designed the experiments, analyzed the results and wrote the paper. R.G. made the tool available in Python package repositories. N.H. and R.B. built the graphical user interface.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01301-5.

Correspondence and requests for materials should be addressed to Lennart Martens.

Peer review information *Nature Methods* thanks Lukas Reiter and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Arunima Singh was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

nature research

Corresponding author(s): Prof. Lennart Martens

Last updated by author(s): Aug 11, 2021

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	FileZilla (3.48.1) was used to access the FTP server at PRIDE; for other sources the systems default browser was used.
Data analysis	The following Python (v3.6) libraries were used in DeepLC: Pandas (v0.25.1), TensorFlow (v1.14.0), Pyteomics (v4.1.2), SciPy (v1.4.0), matplotlib (v3.1.3), seaborn (v0.10.0), and Numpy (v1.17.3). Other software used for DeepLC are: ThermoRawFileParser45 (v1.2.0), MS-GF+ (v2019.08.26), Percolator (v3.4),and open-pFind (v3.1.5). Source code is available under Apache 2.0 at github: https://github.com/compomics/DeepLC. The models, predictions and code to reproduce figures are available at zenodo: https://doi.org/10.5281/zenodo.4542884.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Figure 1
SWATH library (https://doi.org/10.1038/sdata.2014.31; PXD000953-954)
HeLa HF (https://doi.org/10.1021/acs.jproteome.7b00602; PXD006932)
DIA HF (https://doi.org/10.1074/mcp.ra117.000314; PXD005573)

Figure 2
HeLa hf (https://doi.org/10.1021/acs.jproteome.7b00602; PXD006932)
ProteomeTools (https://doi.org/10.1038/nmeth.4153; PXD010595)
SWATH library (https://doi.org/10.1038/sdata.2014.31; PXD000953-954)
Plasma lumos 1h (https://doi.org/10.1007/s13361-019-02243-1; PXD013477)
DIA HF (https://doi.org/10.1074/mcp.ra117.000314; PXD005573)
HeLa lumos 2h (https://doi.org/10.1007/s13361-019-02243-1; PXD013477)
Pancreas (https://doi.org/10.15252/msb.20188503; PXD010154)
Xbridge (https://doi.org/10.1021/acs.analchem.7b03436)
ATLANTIS SILICA (https://doi.org/10.1021/acs.analchem.7b03436)
LUNA SILICA (https://doi.org/10.1021/acs.analchem.7b03436)
LUNA HILIC (https://doi.org/10.1021/acs.analchem.7b03436)
SCX (https://doi.org/10.1021/acs.analchem.7b03436)
Yeast 2h (https://doi.org/10.1021/acs.jproteome.6b00048; PXD003472)
HeLa lumos 1h (https://doi.org/10.1007/s13361-019-02243-1; PXD013477)
Yeast 1h (https://doi.org/10.1021/acs.jproteome.6b00048; PXD003472)
Arabidopsis (https://doi.org/10.1105/tpc.19.00403; PXD008812)
Yeast DeepRT (https://doi.org/10.1074/mcp.m111.013722)
ProteomeTools PTM (https://doi.org/10.1074/mcp.tir118.000783; PXD009449)
Plasma lumos 2h (https://doi.org/10.1007/s13361-019-02243-1; PXD013477)
HeLa DeepRT (https://doi.org/10.1016/j.celrep.2014.07.036)

Figure 3
SWATH library (https://doi.org/10.1038/sdata.2014.31; PXD000953-954)
HeLa HF (https://doi.org/10.1021/acs.jproteome.7b00602; PXD006932)
DIA HF (https://doi.org/10.1074/mcp.ra117.000314; PXD005573)

Figure 4
ProteomeTools PTM (https://doi.org/10.1074/mcp.tir118.000783; PXD009449)

Figure 5
DIA HF (https://doi.org/10.1074/mcp.ra117.000314; PXD005573)
HeLa DeepRT (https://doi.org/10.1016/j.celrep.2014.07.036)

Figure 6
Pandey (https://doi.org/10.1038/nature13302; PXD000561)

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	The available data was split into three sets; training (85.5 %), validation (4.5 %), and test (10 %). Data set sizes range from 161193 to 3411 unique peptide sequences.
Data exclusions	Peptides from previously published analysis (DeepRT and Guan et al. were used). As described in the manuscript for self-processed data the median value within and between runs was taken for the same peptidoforms. No other data was excluded or summarized.
Replication	DeepLC was applied and evaluated on different 20 data sets. On all data sets DeepLC shows high accuracy in predicting retention times.
Randomization	Random division into training,validation, and test splits.
Blinding	Since the model learns the parameters it was kept "blind" from the test data in the pipeline. Traditional blinding of investigators is not considered a relevant technique in machine learning.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging