LUND
UNIVERSITY

Classification of sequence tags from tandem mass spectrometry spectra using machine learning models

Júlia Ortís Sunyer

BINP51, 45 credits, Bioinformatics

Department of Biology

Supervisor: Lars Malmström

Co-supervisor: Carlos Alberto Gueto Tettay

lars.malmstrom@med.lu.se

Infection Medicine Proteomics, BMC D13, Lund University

Machine learning

Classification of sequence tags from tandem mass spectrometry spectra using machine learning models

Júlia Ortís Sunyer

Infection Medicine Proteomics, BMC D13, Lund University

Supervisor: Dr. Lars Malmström

Co-supervisor: Dr. Carlos Alberto Gueto Tettay

Abstract

Motivation: Proteomics is the large-scale study of all the proteins found in a cell, tissue or organism. In the last few years, and thanks to the development of mass spectrometry and bioinformatics, proteomics has led the research in several fields, ranging from medicine to agriculture. In order to reconstruct the amino acid sequence *de novo* protein sequencing can be used. It uses the protein's molecular weight, its mass spectrometry spectrum, and bioinformatics' tools to reconstruct the sequence without the use of a database. This avoids problems such as the limited amount of data found in the databases. Nonetheless, more research needs to be carried out to optimize the tools and data extraction, specially to deal with the ambiguous spectra of long peptides. In this project, several machine learning algorithms were created using TensorFlow and Keras. The aim was for at least one of the models to correctly identify sequence tags extracted from tandem mass spectrometry spectra from fake tags.

Results: Seven machine learning models were successfully built to classify sequence tags from tandem mass spectrometry spectra. Upon evaluation of the models, two of them delt with the data better, according to several statistical parameters (confusion matrix outcomes, accuracy, precision, recall and area under the curve) and managed to classify the true tags of each spectrum largely correctly.

Contact: ju7605or-s@student.lu.se

Supplementary information: Additional data found in the Supplementary Information appendix. Scripts and documentation (README.md) sent separately.

Background

Introduction

Proteomics is the large-scale study of proteomes (Aslam *et al.*, 2017). Proteomes was a term coined in the nineties to describe the entire set of proteins produced in an organism, cell, or tissue (Wasinger *et al.,* 1995; Wilkins, 1997). Even though proteomics is a fairly new discipline, it has been leading biological research in all the fields, ranging from plants to medicine (Jorrín-Novo *et al.,* 2015; Uemura & Kondo, 2015). This is in part due to the development of bioinformatics and mass spectrometry in the last few years (Gauthier *et al.*, 2019; Shackleton, 2010).

Mass spectrometry (MS) is an analytical tool suitable for measuring the mass-to-charge ratio (m/z) of peptides and calculate their exact molecular weight (Covey *et al.,* 1988). This allows for the identification of unknown compounds and to determine the structure and chemical properties of peptides (Domon & Aebersold, 2006). The most fundamental component in MS is the mass analyzer, which takes ionized masses and separates them based on m/z. This generates information-rich ion mass spectra from the peptide fragments (Dass, 2007).

Two mass analyzers can be coupled together using an additional reaction step in tandem mass spectrometry (MS/MS or MS^2). The first spectrometer (MS_1) separates the ionized molecules by their m/z. The second spectrometer (MS_2) takes the fragmented ions, separates them by their m/z and detects them. The fragmentation step, which occurs between MS_1 and MS_2, allows for the identification and separation of ions with extremely similar m/z (Domon & Aebersold, 2006; Figure 1A). There are several approaches to MS/MS, such as collision-induced dissociation (CID), ion-molecule reaction, and photodissociation (Sleno & Volmer, 2004).

The fragmented ions follow a specific nomenclature. For peptides, fragments containing the N-terminus are labeled a, b, c, while the fragments that contain the C-terminus are labeled x, y, z. In both cases, which letter is used depends on the site of the cleavage while the numbers indicate the amount of amino acid residues found in the fragmented ion (Roepstorff & Fohlman, 1984; Figure 1B).

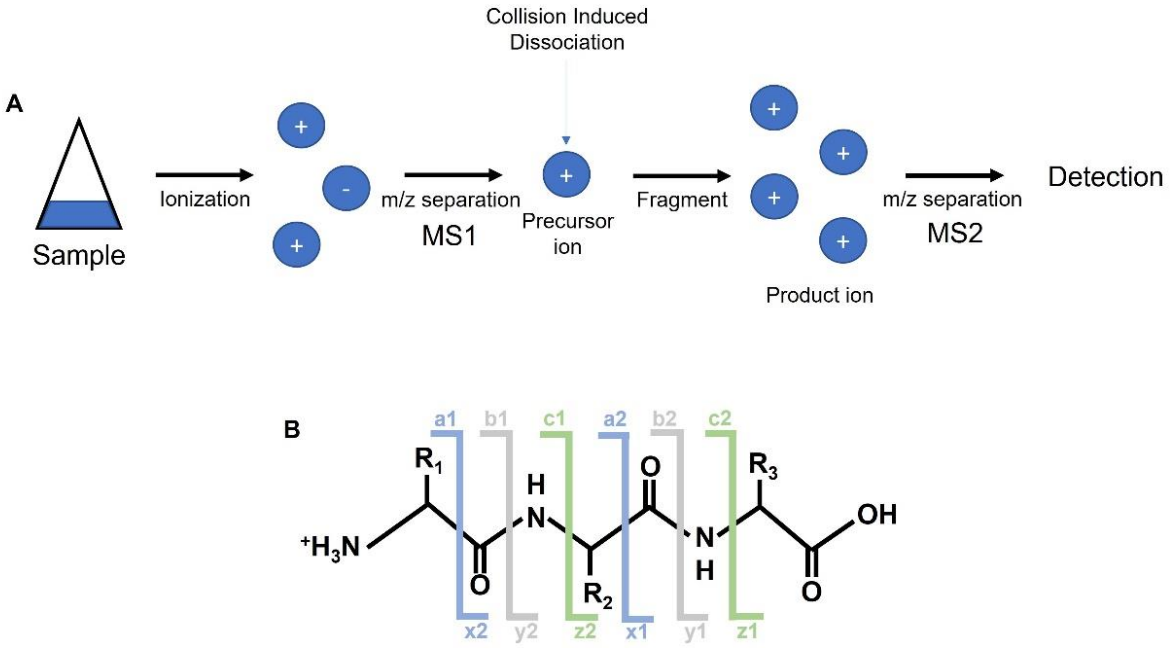

Figure 1. A) Schematic view of MS/MS workflow. In order to generate a mixture of ions the initial sample is ionized. Then, precursor ions of a specific m/z are selected by MS_1 and fragmented to generate a product of ions (MS_2) for detection. **B)** Schematic view of peptide fragmentation. The different fragmented ions are depicted using different colored lines and named according to peptide fragment ion notation. At the top of the image, we see the N-terminus ions while at the bottom, there are the C-terminus ions.

MS/MS spectra can be used in protein sequencing, which consists of determining the amino acid sequence of a peptide (Medzihradszky & Chalkley, 2013). Protein sequencing can be performed by database search, by *de novo* sequencing, or by hybrid methods (Kim & Pevzner, 2014; Medzihradszky & Chalkley, 2013; Taylor & Johnson, 1997). In database search, the mass spectra data of the peptide is run through a directory to find a match with a known peptide sequence (Kim & Pevzner, 2014). This can be done by using peptide sequence tags, which are short amino acid sequences derived from the peptide (Mann & Wilm, 1994). Even though database search is a useful technique, it has a limited number of sequence tags. These are dependent on the data on each database and can lead to bottlenecks in the increase of knowledge (Bork & Koonin, 1998). *De novo* peptide sequencing avoids this issue by reconstructing the amino acid sequence of a peptide using the sequence's tags, an MS/MS spectrum, and the peptide's mass (Medzihradszky & Chalkley, 2013). Thus, *de novo* peptide sequencing from MS/MS spectra is crucial in the description of new protein sequences (Tran *et al*., 2017). Consequently, in the last few years the field has been heavily studied and numerous tools have been proposed, such as DeepNovo, PEAKS, or PepNovo (Frank & Pevzner, 2005; Ma et al., 2003; Tran *et al*., 2017). Nonetheless, several issues still exist with these methods, such as the noise and ambiguity of long peptides in MS/MS spectra (Steen & Mann, 2004). In order to deal with these issues and get a higher optimization, deep learning was introduced in the field (Tran *et al*., 2017). Deep learning is very interesting to use in *de novo* protein sequencing data because it does not need base layers dependent on existing data. Moreover, it can learn from multiple levels of representation of the data, which is useful with MS/MS spectra (Tran *et al*., 2017). Even though it has been seen that amino acid extraction from MS/MS spectra was possible using machine learning models, it was not seen if those sequence tags could be successfully classified using machine learning models. This would be helpful in fields like antibody sequencing, in which *de novo* sequencing is a must due to the lack of comprehensive databases and the different *de novo* methods can lead to different sequence candidates for each spectrum.

Aim

For this reason, the aim of this project was to see if a machine learning model could successfully classify sequence tags from MS/MS spectra. The idea behind this was to build a binary classifier that could correctly classify real tags found in the spectrum from fake tags. This was thought as a subsequent step from the amino acid extraction from the spectra. The data was obtained from a variety of species, to give the model a broader training spectrum. Upon successful training and evaluation of the first architecture, several other models were built to see if the structure of the models affected their classification performance. In this way, the rate of real tags correctly classified by each model is crucial in deciding which model worked best.

Methods

The data used in this project was obtained using higher-energy collisional dissociation (HCD), a type of CID. The MS/MS spectrometer outputs the data in a .raw file that is converted to a mascot generic format (.mgf) file that contains the spectrum and its relevant information. The spectra were annotated using five different search engines. The data came from eleven species: *Escherichia coli*, *Enterococcus faecalis*, *G Streptococcal*, *Equus caballus* (horse), *Homo sapiens* (human), *Mus musculus* (mouse), *Pseudomonas aeruginosa*, *Staphylococcus aureus*, *Streptococcus neumoniae*, *Streptococcus pyogenes* and *Saccharomyces cerevisiae* (yeast). Each species dataset has 200,000 data points, with 50% of true tags and 50% of fake tags. While the real tags were confirmed to be in the spectrum, the fake tags were created using the real tags and random amino acids not found in the real tags. In this way, one, two or all three amino acids in the real tag were changed to a random amino acid to create the fake tag.

In order to create the train and test sets from the data, a data preprocessing workflow was used. Eleven sets of train and test data were created in a leave one out manner, in a similar way as in Tran et al. (2018; Figure 2). Each of these sets contained the vector of differences of all the tags, both real and fake, of each spectrum.

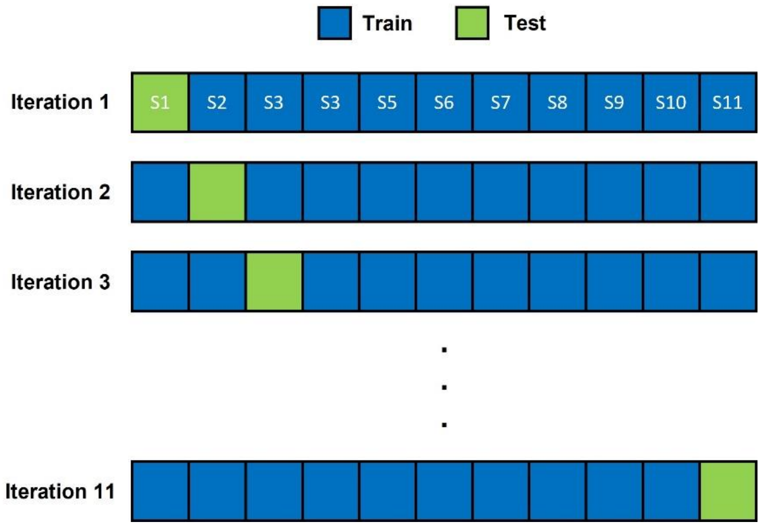

Figure 2. Schematic view of the structure of the leave one out train and test sets in each of the iterations. Depicted in blue are the datasets used to create the training set while in green we can observe the test set. The letter S plus the number inside the squares in the first iteration are to show that each of the squares represents a dataset of one of the eleven species.

As mentioned before, the data used as input for the model needed to be in the form of a vector of differences. The logic behind choosing this method as input for the models was that all the input data is standardized and treated the same way. Moreover, it uses well established data such as the molecular weight of each amino acid and allows for a higher degree of efficiency than other methods.

To obtain the vector of differences of each tag, we first need to create a matrix of differences (D) using the m/z from the spectrum of each peptide (z_1, z_2, z_3, z_4, z_5, z_6, …, z_n) (Figure 3A). As an example, a hypothetic peptide ABCDEF is used in this explanation. Once D is obtained, the vector of masses M (m_B, m_C, m_D, m_{BC}, m_{CD}, m_{BCD}) of each possible tag in the peptide, in this case BCD, is used to verify the tag in D. The values in M correspond to each amino acid in BCD and their combinations: (B, C, D, BC, CD, BCD). To verify the tag in D, the minimum absolute value of D - m_k needs to be obtained (Figure 3B). The result of this operation will give the vector c_k, which contains (c_1, c_2, c_3, c_4, c_5, c_6). As before, those values correspond to each amino acid in BCD and their combinations: (B, C, D, BC, CD, BCD). C_k values (c_1, c_2, c_3, c_4, c_5, c_6) are then used as input for the various machine learning models used (Figure 3C and 3D). After training the model, some predictions are made using the test set. These predictions are compared with the real data to determine which model deals better with the data using various statistical tools.

Seven machine learning models were built and tested: ModelA0, ModelA1, ModelA2, ModelA3, ModelA4, ModelA5 and ModelA6 (Figure 3D). They were all constructed using TensorFlow and Keras. Most of the models use a self-built variable signal layer that normalizes the data using the exponential function e^{-kc}. Nonetheless, ModelA0 uses a fixed signal layer instead. Both fixed and variable signal layers use the same function to normalize the data. Nonetheless, the fixed signal layer uses a fixed k for all the inputs while in the variable layer the k value changes depending on the input value c.

All the models have at least a dense layer with rectified linear unit (ReLU) activation that has different dimensionalities of the output space depending on the model; a dropout layer, which always has a dropout rate of 0.2; and an output layer, which is a dense layer with sigmoid activation and a dimensionality of the output of 1. This last layer is crucial for correctly building a model that binary classifies the data. To avoid under and over-fitting, all the models had an early stopper that stopped the training and saved the model if the binary validation accuracy stopped improving.

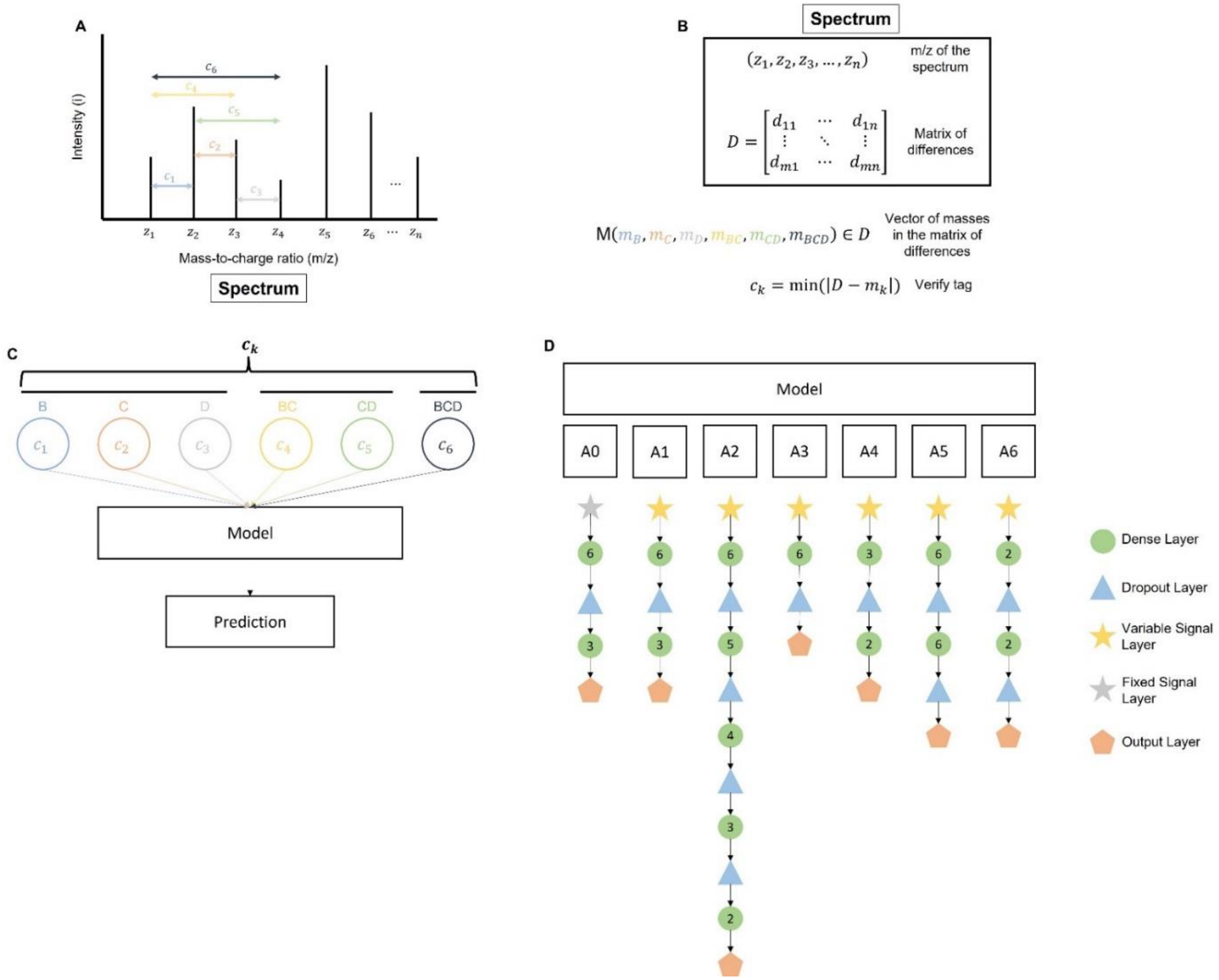

Figure 3. A) Schematic representation of a plot of m/z versus intensity of a MS/MS spectrum of a hypothetic sequence ABCDEF. The various spectrum peaks are represented as (z_1, z_2, z_3, z_4, z_5, z_6, …, z_n). The distance between some of the peaks is represented as (c_1, c_2, c_3, c_4, c_5, c_6). **B)** In order to obtain the values of (c_1, c_2, c_3, c_4, c_5, c_6), depicted as c_k, we first need to use the m/z of the spectrum (z_1, z_2, z_3, z_4, z_5, z_6, …, z_n) to create a matrix of differences, D. We also need to use the amino acids found in the tag we want to find in the spectrum, in this case BCD, to create a vector of masses M (m_B, m_C, m_D, m_{BC}, m_{CD}, m_{BCD}). If this tag belongs to the sequence ABCDEF, M will be found in D. To verify the tag, we find the minimum absolute value of subtracting D minus m_k. This will result in a vector c_k that contains six values, (c_1, c_2, c_3, c_4, c_5, c_6), one for each amino acid and their combinations found in the tag. Real tags should have c_k values very close to zero, because their masses will be very close to some of the values found in D. **C)** C_k values (c_1, c_2, c_3, c_4, c_5, c_6) are then used as input for the machine learning model that will train itself to properly identify real tags found in the spectra. **D)** Schematic representation of the structure of the seven models: A0, A1, A2, A3, A4, A5 and A6. The green circle represents a dense layer, the blue triangle the dropout layer, the yellow star the fixed signal layer, the grey star the fixed signal layer and the orange pentagon the output layer. The number found in the dense layer shape depicts the dimensionality of the output space of the layer. The dropout rate of the dropout layer was 0.2 for all the models. The dimensionality of the output space of the output layer was 1 for all the models.

To evaluate the models, a testing set was used. There were a total of eleven testing sets per model, one for each of the left-out species in the training sets. This made up for a total of 2,200,000 data tested per model. As mentioned before, 50% of this data consisted of true tags while the other 50% were fake tags. In order to properly evaluate the models, the confusion matrix of each of them was obtained. The confusion matrix represents the counts from predicted and actual values and has four possible outcomes: true negatives (TN), which show the amount of negative data classified correctly; true positives (TP), which are the positive values classified accurately; false positives (FP), which show values classified as positive when are negative; and false negatives (FN), which depicts actual positive values classified as negative (Gupta *et al*., 2022). Moreover, four statistical representations derived from the confusion matrix were obtained: the accuracy, precision, recall and area under the curve.

The accuracy of the data checks the proportion of correct predictions (TN and TP) in all the data classified (TP, TN, FP, and FN). The accuracy portrays how accurate a model was in predicting the data, or in other words, how rigorous a model is at classifying the data correctly. The precision of the model checks the proportion of true positives (TP), among all the data classified as positive (TP and FP). This is useful because we want to be certain that data classified positively is truly positive to better assess the models. The recall checks the retrieved items (TP) among all the relevant items (TP and TN). It is also known as the sensitivity of the model. The area under the curve shows the model's ability to distinguish between classes. The closer the area under the curve is to one, the better the model correctly differentiates between positive and negative data (Gupta *et al*., 2022).

Results and discussion

As previously mentioned, the purpose of this project was to successfully build a machine learning model that classified sequence tags from MS/MS spectra. Not only this was achieved, but six additional models were built to see if their architecture influenced their classification performance. In order to evaluate the models fairly, the confusion matrix as well as several statistical methods derived from it were used.

Two different plots of the confusion matrix outcomes were built (Figure 4). Figure 4A plots the percentage of the outcomes of the confusion matrix out of the total data. The TP and FN account for 50% of the data, while the TN and FP account for the other 50%. The reason for this is that 50% of the tags are true and the other 50% are fake in the test data so, no matter how they are classified, they should still be there. Even though we aim for the lowest number of FP and FN outcomes, a basal number of FN is expected. The reason for this is that some peptides, specially if they are long, do not have the complete series of ions in their spectra, which is already a known issue in *de novo* protein sequencing (Yang *et al*., 2019). This leads to incomplete or ambiguous spectra that can lead to the erroneous classification of true tags. A small percentage of FN outcomes can be observed in each of the models. Nonetheless, in model A4, 5% of the total outcomes are FN. Not all the FN in that model can be explained by the ambiguity of the spectra, specially when compared to the other models, which have around three times less FN. This suggests that model A4 has issues classifying the tags. The ambiguity of some of the spectra can also explain some of the FP, particularly in the models that have a small percentage of them. Nonetheless, the models A2, A4 and A6 have 5%, 10% and 5% of FP, respectively. This suggests that errors in the classification of the data by those models might lead to the increase of FP, specially when compared to the other models, that have a percentage of FP of around 1.5%. These results imply that models A2, A4 and A6 are not as trustworthy as the other models, as at least 5% of the positively classified tags are in fact, fake.

These results are complemented by the categorical plot of the confusion matrix outcomes seen in figure 4B. This graph depicts the scores in which the data is plotted as one of the outcomes. For example, in model A4 some of the data falsely classified as positive has a prediction score of around 0.9. This means that some fake tags are classified as true with a certainty of 0.9 out of 1. This implies that to be sure that a tag classified as true is really true, the score has to be extremely high so as to avoid the chance of a FP. On the other hand, models A3 and A5 have some FP with a score of around 0.6. This is quite optimal as the certainty that a tag with a prediction score of 0.9 is true and classified as such is higher.

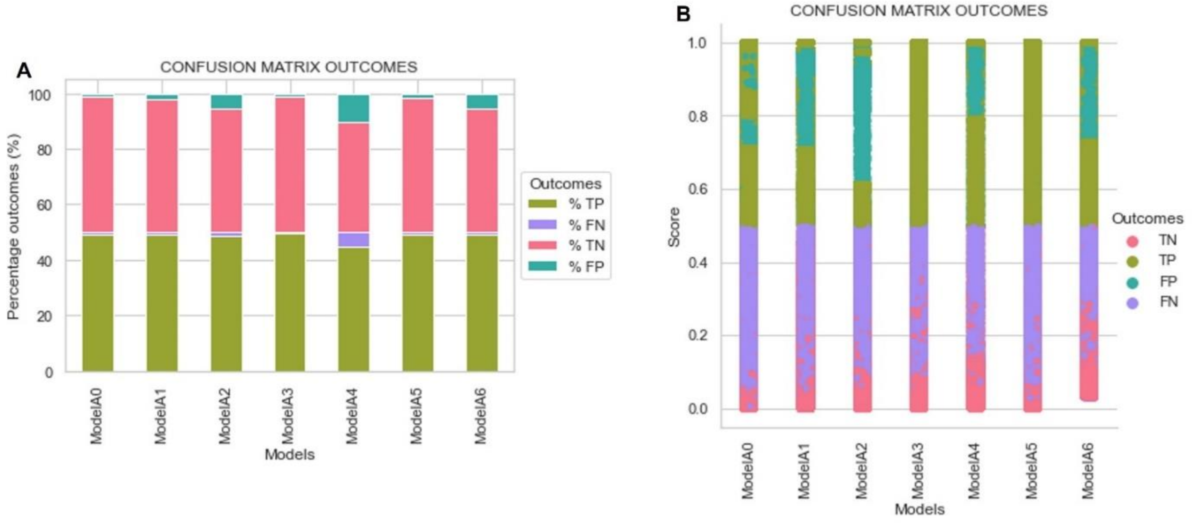

Figure 4. A) Stacked bar plot of the confusion matrix outcomes' percentages. The outcomes are labelled in different colors: true positives (TP) are green, the false positives (FP) blue, the false negatives (FN) purple, and the true negatives (TN) red. **B)** Categorical plot of the confusion matrix outcomes. In the X-axis are the models and in the Y-axis the score. The outcomes are labelled in different colors: true positives (TP) are green, the false positives (FP) blue, the false negatives (FN) purple, and the true negatives (TN) red.

In order to further support these findings, other statistical methods derived from the confusion matrix were obtained, such as the accuracy, precision, recall and the area under the curve. These methods are useful in further understanding the data, as they give a clearer picture of how the models performed. The raw table with the results of each of these statistics per model and test set can be seen in the Supplementary Information, Table S1. Figure 5 plots the results of Table S1.

Model A4 is less accurate and precise than the other models, which have an accuracy and precision closer to 1 (Figure 5A and 5C). This can be explained by the lower accuracy and precision of the sets of *Streptococcus pyogenes*, *Escherichia coli* and *Staphylococcus aureus* seen in figure 5B and 5D. If we focus on the accuracy, we can observe that for model A4, those sets have an accuracy of 0.5, compared to the accuracy of above 0.9 of the other test sets. This can also be seen, in a lesser measure, in the accuracy of models A1, A2 and A6. This information leads us to the conclusion that some models perform better with certain datasets, while others perform equally well regardless of the left out set. This might be explained by the fact that there can be differences in the protein sequences of different organisms and that some of the machine learning architectures might deal better with those differences without taking a hit on the performance. The variation in protein sequences between organisms have been reported in several studies, specially between eukaryotic and prokaryotic organisms (Bogatyreva *et al.*, 2006; Shemesh *et al.*, 2010). Very similar results for the same models with worse accuracy can be seen for the precision (Figure 5D). When model A1, A2, A4 and A6 are tested with some of the sets, the precision is worse. The clearest example of this is in model A4, where the set for *Staphylococcus aureus* has a precision of 0.

The recall and area under the curve of each of the models were also obtained (Supplementary Information, Figure S1). The recall of all the models is quite good, being close to 1 for most of the models and test sets. Nonetheless, model A4 has a notable outlier, with a recall of 0, for the *Staphylococcus aureus* set. This further shows that model A4 is not only not the most accurate or precise, but also the least sensitive. Regarding the area under the curve, a similar thing as in the accuracy and precision can be observed. Models A2, A4 and A6 have several sets with an area under the curve of 0.5. This means that for those sets, the models are not good at differentiating between positive and negative data.

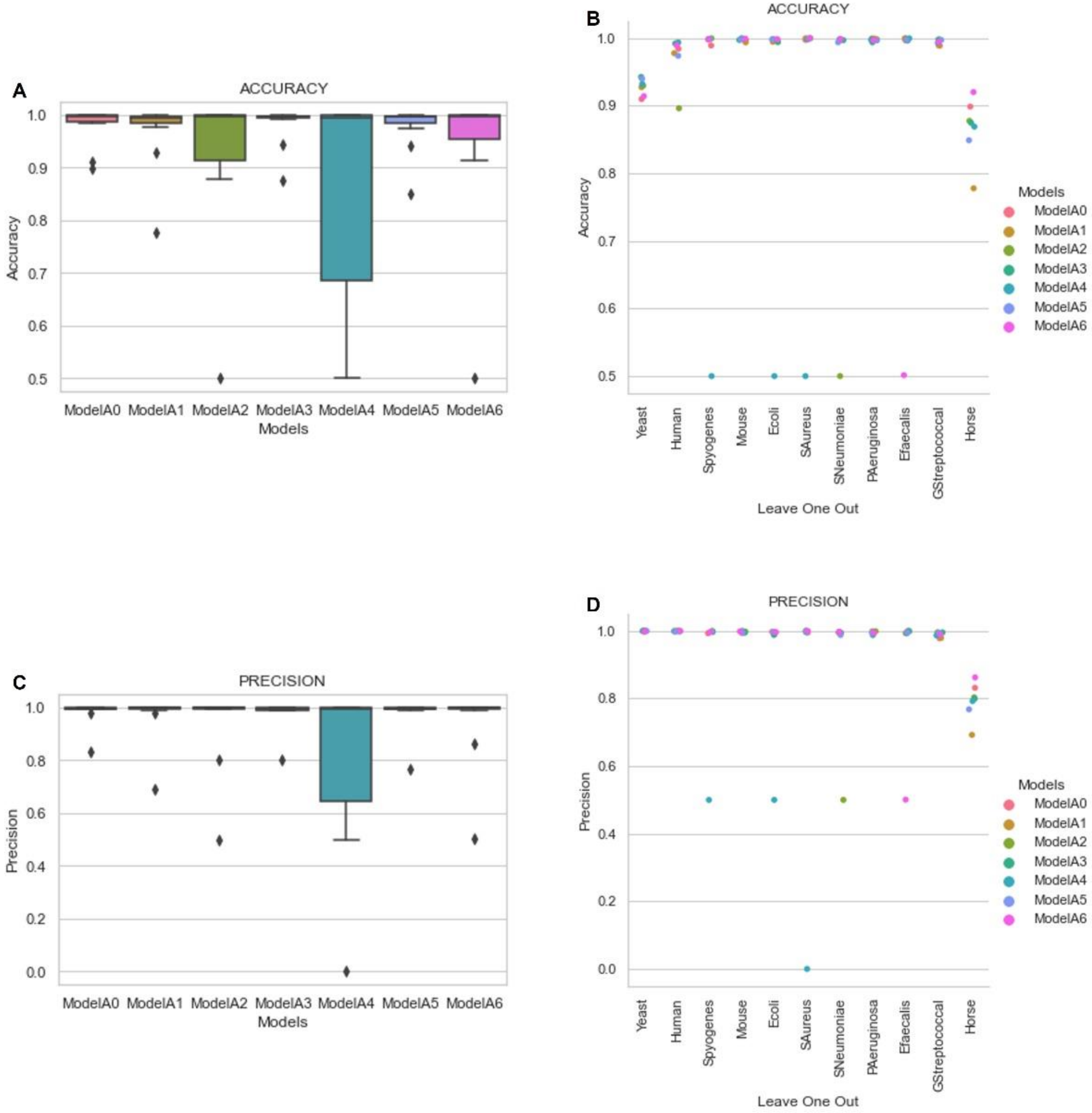

Figure 5. A) Boxplot of the accuracy of the models. On the X-axis are the models and on the Y-Axis the accuracy. **B)** Categorical plot of the accuracy of the models in each test set. The models are depicted in different colors: red for A0, ochre for A1, green for A2, darker green for A3, blue for A4, purple for A5 and pink for A6. **C)** Boxplot of the precision of the models. On the X-axis are the models and on the Y-Axis the precision. **D)** Categorical plot of the precision of the models in each test set. The models are depicted in different colors: red for A0, ochre for A1, green for A2, darker green for A3, blue for A4, purple for A5 and pink for A6.

As briefly mentioned previously, a possible explanation for these results can be found in the models' architecture (Figure 3D). If, firstly, we focus on the models A2, A4 and A6, which performed worse according to the statistical methods, they have distinctive architectures compared to the ones with better performance. Model A2 is the model with the greatest number of dense layers, which seems to be a detriment to the efficiency of the model. Regarding the other two models, A4 and A6, they both have two dense layers, like the best performing models. The difference is that while the best performing models have an initial dense layer with a dimensionality of the output of 6, those two models have an initial dense layer with a lower dimensionality of the output, which is either 3 or 2, respectively. Thus, it seems like the first dense layer is crucial in the performance of the model and that the bigger the dimensionality of the output of the first dense layer, the better.

Regarding the other models, they all seem to classify the data better according to both the confusion matrix and its derived statistics. Model A0 and A1 have a similar architecture: two dense layers with a dimensionality of the output of 6 and 3. The only difference between them lays in the signal layer: A0 has a fixed signal layer while A1 has a variable signal layer. Surprisingly, this does make a slight difference in the true tag classification, especially when the horse dataset is used as a test set. While on the other sets both models perform similarly, on this set model A1 has a worse accuracy and precision. The other two models, model A3 and A5, seem to classify the data with a similar accuracy and precision as model A0. The difference here lays in the certainty that a tag classified as positive is truly positive (Figure 4B). Model A0 has some FP with a prediction score of around 0.9, which make it less trustworthy than the other two models. In regard to the architecture of model A3 and A5, they differ from model A0 on similar things. Model A3 has a variable signal layer followed by one dense layer with a dimensionality of the output of six, while model A5 has a variable signal layer followed by two dense layers with dimensionalities of the output of six. As previously mentioned, all models have the same output layer. Therefore, it seems as if a variable signal layer followed by a maximum of two dense layers with a dimensionality of the output of six works better for the classification of tags from MS/MS spectra.

Conclusion

These results show that it is possible to build a machine learning model that correctly classifies sequence tags from MS/MS spectra. In addition, seven different machine learning architectures were successfully built and tested. Out of the seven models, some architectures proved to have issues classifying the sequence tags (models A2, A4 and A6), while others were notoriously better at the classification (models A3 and A5). Moreover, this project opens the door for a further integration of these models to create a new scoring system of sequence candidates produced by various *de novo* protein sequencing techniques.

Acknowledgements

The author would like to thank the Infection Medicine Proteomics group at Lund's University BMC for the opportunity to work on this project. In particular Dr. Carlos Alberto Gueto Tettay and Dr. Lars Malmström have provided great feedback and guidance on the project.

References

Aslam, B., Basit, M., Nisar, M. A., Khurshid, M. & Rasool, M. H. (2017). Proteomics: Technologies and Their Applications, *Journal of Chromatographic Science*, 55(2):182-196, https://doi.org/10.1093/chromsci/bmw167

Bogatyreva, N. S., Finkelstein, A. V., & Galzitskaya, O. V. (2006). Trend of amino acid composition of proteins of different taxa. *Journal of Bioinformatics and Computational Biology*, *04*(02), 597–608. https://doi.org/10.1142/s0219720006002016

Bork, P., & Koonin, E. V. (1998). Predicting functions from protein sequences—where are the bottlenecks? *Nature Genetics*, *18*(4), 313–318. https://doi.org/10.1038/ng0498-313

Covey, T. R., Bonner, R. F., Shushan, B. I., Henion, J., & Boyd, R. K. (1988). The determination of protein, oligonucleotide and peptide molecular weights by ion-Spray Mass Spectrometry. *Rapid Communications in Mass Spectrometry*, *2*(11), 249–256. https://doi.org/10.1002/rcm.1290021111

Dass, C. (2007). *Fundamentals of Contemporary Mass Spectrometry*. Wiley-Interscience.

Domon, B., & Aebersold, R. (2006). Mass spectrometry and protein analysis. *Science*, *312*(5771), 212–217. https://doi.org/10.1126/science.1124619

Frank, A., & Pevzner, P. (2005). Pepnovo: de novo peptide sequencing via Probabilistic Network modeling. *Analytical Chemistry*, *77*(4), 964–973. https://doi.org/10.1021/ac048788h

Gauthier, J., Vincent, A. T., Charette, S. J. & Derome, N. (2019). A brief history of bioinformatics, Briefings in Bioinformatics, 20(6):1981–1996, https://doi.org/10.1093/bib/bby063

Gupta, D., Kose, U., Khanna, A., & Balas, V. E. (2022). *Deep learning for medical applications with Unique Data*. Academic Press.

Jorrín-Novo, J.V., Pascual, J., Sánchez-Lucas, R., Romero-Rodríguez, M.C., Rodríguez-Ortega, M.J., Lenz, C. & Valledor, L. (2015). Fourteen years of plant proteomics reflected in *Proteomics*: Moving from model species and 2DE-based approaches to orphan species and gel-free platforms. *Proteomics*, 15: 1089-1112. https://doi.org/10.1002/pmic.201400349

Kim, S., & Pevzner, P. A. (2014). MS-gf+ makes progress towards a universal database search tool for proteomics. *Nature Communications*, *5*(1). https://doi.org/10.1038/ncomms6277

Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., & Lajoie, G. (2003). Peaks: Powerful software for Peptide Novo Sequencing by Tandem Mass Spectrometry. *Rapid Communications in Mass Spectrometry*, *17*(20), 2337–2342. https://doi.org/10.1002/rcm.1196

Mann, M., & Wilm, M. (1994). Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical Chemistry*, *66*(24), 4390–4399. https://doi.org/10.1021/ac00096a002

Medzihradszky, K. F., & Chalkley, R. J. (2013). Lessons in de novo peptide sequencing by Tandem Mass Spectrometry. *Mass Spectrometry Reviews*, *34*(1), 43–63. https://doi.org/10.1002/mas.21406

Roepstorff, P., & Fohlman, J. (1984). Proposal for a common nomenclature for sequence ions in mass spectra of peptides. *Biological Mass Spectrometry*, *11*(11), 601–601. https://doi.org/10.1002/bms.1200111109

Shackleton, C. (2010). Clinical steroid mass spectrometry: A 45-year history culminating in HPLC–ms/MS becoming an essential tool for patient diagnosis. *The Journal of Steroid Biochemistry and Molecular Biology*, *121*(3-5), 481–490. https://doi.org/10.1016/j.jsbmb.2010.02.017

Shemesh, R., Novik, A., & Cohen, Y. (2010). Follow the leader: Preference for specific amino acids directly following the initial methionine in proteins of different organisms. *Genomics, Proteomics & Bioinformatics*, *8*(3), 180–189. https://doi.org/10.1016/s1672-0229(10)60020-4

Sleno, L., & Volmer, D. A. (2004). Ion activation methods for tandem mass spectrometry. *Journal of Mass Spectrometry*, *39*(10), 1091–1112. https://doi.org/10.1002/jms.703

Steen, H., & Mann, M. (2004). The ABC's (and XYZ's) of peptide sequencing. *Nature Reviews Molecular Cell Biology*, *5*(9), 699–711. https://doi.org/10.1038/nrm1468

Taylor, J. A., & Johnson, R. S. (1997). Sequence database searches via de novo peptide sequencing by Tandem Mass Spectrometry. *Rapid Communications in Mass Spectrometry*, *11*(9), 1067–1075. https://doi.org/10.1002/(sici)1097-0231(19970615)11:9<1067::aid-rcm953>3.0.co;2-l

Tran, N. H., Zhang, X., Xin, L., Shan, B., & Li, M. (2017). De novo peptide sequencing by Deep Learning. *Proceedings of the National Academy of Sciences*, *114*(31), 8247–8252. https://doi.org/10.1073/pnas.1705691114

Tran, N. H., Qiao, R., Xin, L., Chen, X., Liu, C., Zhang, X., Shan, B., Ghodsi, A., & Li, M. (2018). Deep learning enables de novo peptide sequencing from data-independent-acquisition Mass Spectrometry. *Nature Methods*, *16*(1), 63–66. https://doi.org/10.1038/s41592-018-0260-3

Uemura, N. & Kondo, T. (2015). Current advances in esophageal cancer proteomics. *Biochimica et Biophysica Acta. Proteins and Proteomics*, 1854-6: 687-695, https://doi.org/10.1016/j.bbapap.2014.09.011

Wasinger, V.C., Cordwell, S.J., Cerpa-Poljak, A., Yan, J.X., Gooley, A.A., Wilkins, M.R., Duncan, M.W., Harris, R., Williams, K.L. & Humphery-Smith, I. (1995). Progress with gene-product mapping of the Mollicutes: *Mycoplasma genitalium*. *ELECTROPHORESIS*, 16: 1090-1094. https://doi.org/10.1002/elps.11501601185

Wilkins, M. R. (1997). *Proteome Research: New frontiers in functional genomics*. Springer.

Yang, H., Li, Y.-C., Zhao, M.-Z., Wu, F.-L., Wang, X., Xiao, W.-D., Wang, Y.-H., Zhang, J.-L., Wang, F.-Q., Xu, F., Zeng, W.-F., Overall, C. M., He, S.-M., Chi, H., & Xu, P. (2019). Precision de novo peptide sequencing using mirror proteases of AC-Lysarginase and trypsin for large-scale proteomics. *Molecular & Cellular Proteomics*, *18*(4), 773–785. https://doi.org/10.1074/mcp.tir118.000918

Supplementary Information

Table S1. Raw table of the accuracy, precision, recall and area under the curve (AUC). In this table we can observe the data for each of the models' test sets.

Model	Name	Accuracy	Precision	Recall	AUC
ModelA0	Yeast	0.910	1.000	0.820	0.994
	Human	0.984	0.999	0.969	0.999
	Spyogenes	0.989	0.993	0.985	0.999
	Mouse	0.997	0.994	1.000	1.000
	Ecoli	0.999	0.997	1.000	1.000
	SAureus	0.997	0.995	1.000	1.000
	SNeumoniae	0.997	0.994	1.000	1.000
	PAeruginosa	0.999	0.999	1.000	1.000
	Efaecalis	1.000	0.999	1.000	1.000
	GStreptococcal	0.989	0.978	1.000	1.000
	Horse	0.899	0.831	1.000	0.978
ModelA1	Yeast	0.927	0.999	0.856	0.987
	Human	0.978	1.000	0.956	0.999
	Spyogenes	0.998	0.999	0.998	1.000
	Mouse	0.994	0.998	0.989	1.000
	Ecoli	0.995	0.989	1.000	1.000
	SAureus	1.000	0.999	1.000	1.000
	SNeumoniae	0.998	0.995	1.000	1.000
	PAeruginosa	0.998	0.996	1.000	1.000
	Efaecalis	0.997	0.993	1.000	1.000
	GStreptococcal	0.989	0.979	1.000	0.996
	Horse	0.778	0.692	1.000	1.000
ModelA2	Yeast	0.930	0.999	0.860	0.985
	Human	0.896	1.000	0.793	0.993
	Spyogenes	1.000	0.999	1.000	1.000
	Mouse	0.999	0.998	1.000	0.999
	Ecoli	0.998	0.996	1.000	1.000
	SAureus	1.000	1.000	1.000	1.000
	SNeumoniae	0.500	0.500	1.000	0.500
	PAeruginosa	0.999	0.999	1.000	1.000
	Efaecalis	1.000	0.999	1.000	1.000
	GStreptococcal	0.998	0.996	1.000	1.000
	Horse	0.878	0.803	1.000	0.898
ModelA3	Yeast	0.943	1.000	0.885	0.993
	Human	0.992	0.999	0.985	1.000
	Spyogenes	0.998	0.996	1.000	1.000
	Mouse	0.997	0.995	1.000	1.000
	Ecoli	0.994	0.988	1.000	1.000
	SAureus	0.998	0.996	1.000	1.000
	SNeumoniae	0.997	0.994	1.000	1.000
	PAeruginosa	0.994	0.988	1.000	1.000
	Efaecalis	0.996	0.993	1.000	1.000
	GStreptococcal	0.998	0.995	1.000	1.000
	Horse	0.875	0.799	1.000	0.986
ModelA4	Yeast	0.932	1.000	0.865	0.994
	Human	0.994	0.999	0.988	1.000
	Spyogenes	0.500	0.500	1.000	0.500
	Mouse	1.000	1.000	1.000	1.000
	Ecoli	0.500	0.500	1.000	0.500
	SAureus	0.500	0.000	0.000	0.500
	SNeumoniae	0.999	0.997	1.000	1.000
	PAeruginosa	0.997	0.995	1.000	1.000
	Efaecalis	1.000	0.999	1.000	1.000
	GStreptococcal	0.993	0.987	1.000	1.000
	Horse	0.869	0.792	1.000	0.998
ModelA5	Yeast	0.940	1.000	0.880	0.995
	Human	0.974	0.998	0.950	0.997
	Spyogenes	0.999	0.998	1.000	1.000
	Mouse	0.998	0.995	1.000	1.000
	Ecoli	0.998	0.996	1.000	1.000
	SAureus	0.999	0.999	1.000	1.000
	SNeumoniae	0.994	0.988	1.000	1.000
	PAeruginosa	0.997	0.993	1.000	1.000
	Efaecalis	0.997	0.994	1.000	1.000
	GStreptococcal	0.997	0.995	1.000	1.000
	Horse	0.849	0.768	1.000	0.991
ModelA6	Yeast	0.914	0.998	0.830	0.989
	Human	0.989	1.000	0.979	0.995
	Spyogenes	0.998	0.996	1.000	1.000
	Mouse	0.999	0.998	1.000	1.000
	Ecoli	0.998	0.996	1.000	1.000
	SAureus	1.000	0.999	1.000	1.000
	SNeumoniae	0.998	0.997	1.000	1.000
	PAeruginosa	0.997	0.994	1.000	1.000
	Efaecalis	0.501	0.501	1.000	0.502
	GStreptococcal	0.994	0.989	1.000	0.999
	Horse	0.920	0.862	1.000	1.000

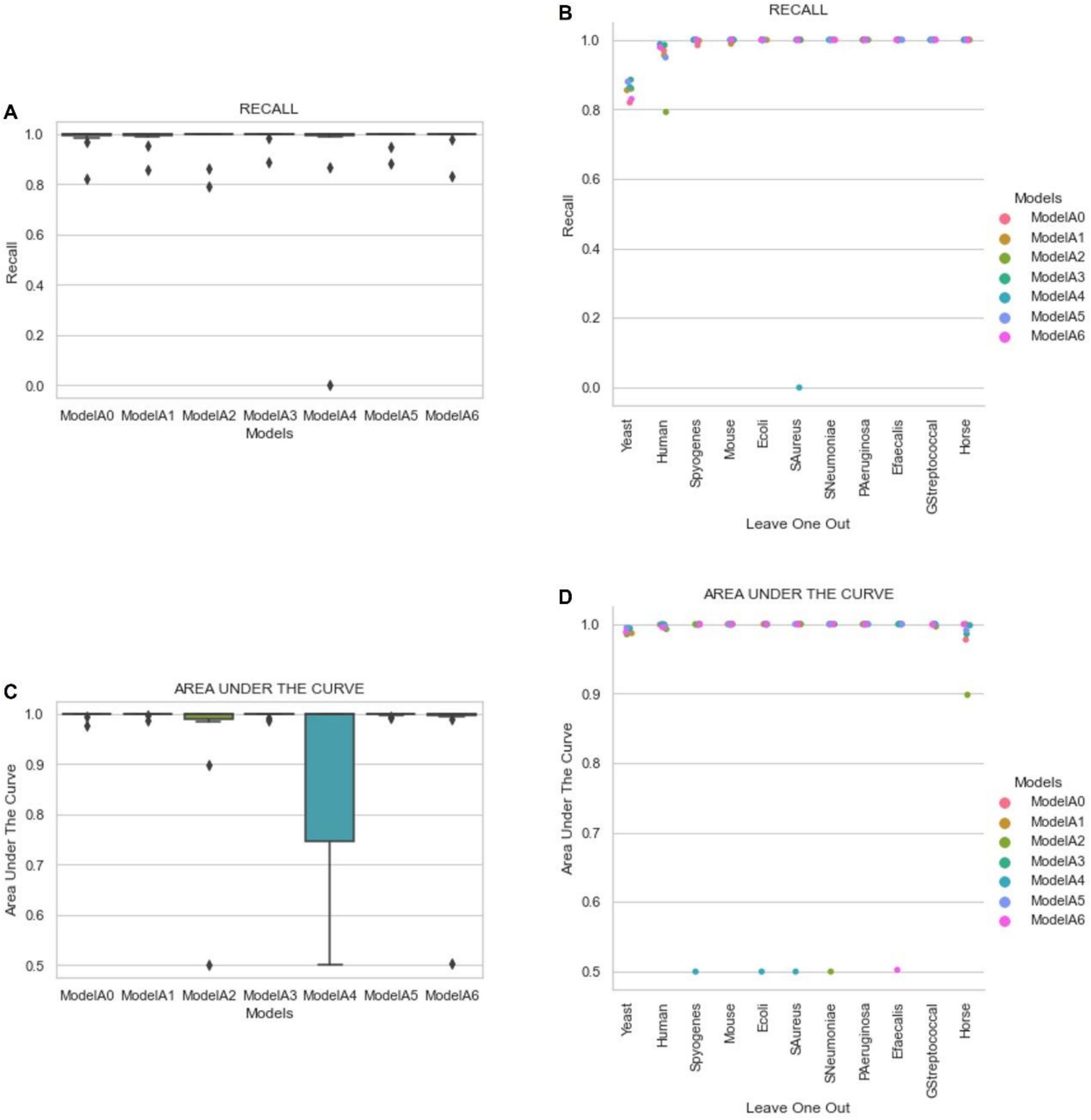

Figure S1. A) Boxplot of the recall of the models. On the X-axis are the models and on the Y-Axis the recall. **B)** Categorical plot of the recall of the models in each test set. The models are depicted in different colors: red for A0, ochre for A1, green for A2, darker green for A3, blue for A4, purple for A5 and pink for A6. **C)** Boxplot of the area under the curve of the models. On the X-axis are the models and on the Y-Axis the area under the curve. **D)** Categorical plot of the area under the curve of the models in each test set. The models are depicted in different colors: red for A0, ochre for A1, green for A2, darker green for A3, blue for A4, purple for A5 and pink for A6.