Pre-trained Maldi Transformers improve MALDI-TOF MS-based prediction

Gaetan De Waele[1,*], **Gerben Menschaert**[1], **Peter Vandamme**[2] **and Willem Waegeman**[1]

[1]Department of Data Analysis and Mathematical Modelling, Ghent University, [2]Laboratory of Microbiology, Ghent University, [*]Correspondence to <gaetan.dewaele@ugent.be>.

Abstract

For the last decade, matrix-assisted laser desportion/ionization time-of-flight mass spectrometry (MALDI-TOF MS) has been the reference method for species identification in clinical microbiology. Hampered by a historical lack of open data, machine learning research towards models specifically adapted to MALDI-TOF MS remains in its infancy. Given the growing complexity of available datasets (such as large-scale antimicrobial resistance prediction), a need for models that (1) are specifically designed for MALDI-TOF MS data, and (2) have high representational capacity, presents itself.

Here, we introduce Maldi Transformer, an adaptation of the state-of-the-art transformer architecture to the MALDI-TOF mass spectral domain. We propose the first self-supervised pre-training technique adapted to mass spectra. The technique is based on shuffling peaks across spectra, and pre-training the transformer as a peak discriminator. Extensive benchmarks confirm the efficacy of this novel design. The final result is a model exhibiting state-of-the-art (or competitive) performance on downstream prediction tasks. In addition, we show that Maldi Transformer's identification of noisy spectra may be leveraged towards higher predictive performance.

All code supporting this study is distributed on PyPI and is packaged under: https://github.com/gdewael/maldi-nn

Keywords MALDI-TOF MS · Neural networks · Transformers

1. Introduction

Matrix-assisted laser desportion/ionization time-of-flight mass spectrometry (MALDI-TOF MS) is a proteomic technique commonly used to identify microbial species (Dauwalder et al., 2023). First introduced to clinical microbiology at the end of 2010, its routine use is characterized by low-cost, speed, and reliability (Weis et al., 2020a). MALDI-TOF MS generates spectra containing peaks signifying mostly ribosomal proteins (Seng et al., 2009). As such, the spectra can serve as fingerprints indicative of species identity (Bizzini et al., 2011).

For bacterial species identification, clinical diagnostic labs will typically use the solutions provided by MALDI-TOF MS manufacturers. These solutions are built on large, proprietary, in-house databases (Van Belkum et al., 2012). The models used in such solutions presumably rely on querying certain marker peaks to a large database (Florio et al., 2018). This strategy works reasonably well for identification of most species, but some strains remain problematic to identify this way (Cao et al., 2018; Vrioni et al., 2018). In addition, the peak-matching approach does not suit more-difficult prediction tasks such as strain typing (Hettick et al., 2006), antimicrobial resistance prediction (Weis et al., 2022), and virulence factor detection (Gagnaire et al., 2012). In these cases, researchers turn to machine learning in order to possibly mine more-intricate patterns from the spectra.

Historically, machine learning for MALDI-TOF MS has been hampered by a lack of large open data. Because of this, the nascent field has not often progressed beyond off-the-shelf learning techniques (Weis et al., 2020a). Only a handful of examples exist of more advanced machine learning methods specifically adapted to a MALDI-TOF-based task (Vervier et al., 2015; Weis et al., 2020b; Mortier et al., 2021; De Waele et al., 2023). As such, the design of machine learning algorithms specifically for MALDI-TOF mass spectra has not been studied yet in sufficient detail.

Typically, MALDI-TOF MS-based machine learning methods either discretize the (m/z)-axis, or preselect a number of peak locations as features based on training set characteristics. Discretizing the (m/z)-axis results in fixed-length spectral representations, where every feature consists of a bin. If bins are cho-

sen too large, resolution is lost and multiple peaks may be lumped together. On the other hand, small bins result in a higher-dimensional representation where most bins contain no peaks, unnecessarily adding model complexity (Weis et al., 2020b; Mortier et al., 2021). Other works select a fixed set of (m/z) locations based on training set characteristics (Tran et al., 2021). Features for every spectrum are then derived by encoding the peak height (or binary peak presence) for every selected m/z location. The disadvantage here is that spectra may contain important peaks not included in the fixed set of selected m/z locations. Hence, the model is unable to cope with patterns differing too much from general training set characteristics (e.g. rare peaks).

The information in MALDI-TOF mass spectra is defined by their peaks (many associated with ribosomal subunits) (Ryzhov and Fenselau, 2001). As such, we argue that, ideally, a machine learning model for MALDI-TOF mass spectra operates directly on sets of peaks as inputs. This, however, constitutes a non-trivial machine learning setup. A kernel-based technique that processes spectra in this way was previously proposed by Weis et al. (2020b). However, their method is difficult to scale to larger sample sizes. Consequently, in this work, we propose a deep learning-based solution operating on sets of peaks: a transformer for MALDI-TOF data. Transformers elegantly operate on input sets via their permutation-invariant self-attention operations (Vaswani et al., 2017). Self-attention can be interpreted as message passing on a complete directed graph (i.e. peaks are nodes, and all nodes are connected to each other).

Following its rise to dominance in natural language processing, transformers are increasingly adopted towards biological data modalities (Clauwaert and Waegeman, 2020; Jumper et al., 2021; Avsec et al., 2021; Elnaggar et al., 2021). Most closely related to the MALDI-TOF domain, transformers have also been adapted to operate on tandem mass spectra for *de novo* peptide sequencing (Yilmaz et al., 2022). This ever-increasing adoption of transformers across subfields of machine learning speaks to their generality. Because they can be viewed as operating on a complete digraph, they place no inductive bias on the learned patterns between input tokens. This property lends transformers supreme representational capabilities, but is also the reason why they are typically described as data-hungry. Consequently, transformers are usually mentioned in the same breath as self-supervised learning (SSL) (Liu et al., 2021). SSL is the paradigm within deep learning wherein a supervised learning task is designed for data that has not explicitly been labeled (Balestriero et al., 2023). This task is (usually) not a useful prediction problem in itself, but rather serves to pre-train a large model. In doing so, greater downstream performance on tasks of interest can be obtained. SSL is, therefore, most useful in scenarios where labeled data is limited, such as the MALDI-TOF MS domain (Weis et al., 2020a).

In the present study, we introduce Maldi Transformer, a deep learning architecture for processing MALDI-TOF mass spectra. The inputs to Maldi Transformer consist of spectra as sets of peaks. To fully take advantage of the transformer architecture's representational capacity, we propose a novel self-supervised pre-training strategy. The strategy relies on discriminating real peaks from noisy ones introduced to the spectrum via shuffling peaks across samples. We pre-train Maldi Transformer on the large open DRIAMS database (Weis et al., 2020a). Maldi Transformer obtains strong performance on downstream benchmarks, demonstrating the power of the approach.

2. Methods

2.1. Data

To train and test models, we use three large (> 10 000 spectra) MALDI-TOF datasets, two of which are available to the public domain.

First, the recently published DRIAMS database (Weis et al., 2022), is used to both pre-train Maldi Transformer and to fine-tune it on antimicrobial resistance (AMR) prediction (binary classification using a dual-branch recommender system). DRIAMS contains a total of 250 070 spectra, originating from four hospitals in Switzerland. For AMR prediction, the same data splits are used as in earlier work (De Waele et al., 2023). Briefly, DRIAMS-A spectra from before 2018 are split to the training fraction, whereas DRIAMS-A spectra measured during 2018 are evenly split between validation and test set. For pre-training, the same splits are used, but all spectra from DRIAMS-B, -C, and -D are additionally added to the training set. Apart from AMR measurements, DRIAMS contains species labels for many spectra. These labels are derived from the species identification pipelines included with the MALDI-TOF MS machines. After processing (see Appendix A), the pre-training DRIAMS dataset spans 469 species.

The second dataset consists of the public Robert Koch-Institute (RKI) database (Lasch et al., 2023). The final used dataset is a private historical database containing more than 2400 taxonomic reference strains,

Table 1 | Data sources used, along with their use and sizes (in number of spectra).

Dataset	Used for	Train-Val-Test Size
DRIAMS	Pre-training	207 172 - 21 440 - 21 443[a,b]
DRIAMS-A	Fine-tuning on AMR prediction	28 331 - 4 994 - 4 999[c]
RKI	Fine-tuning on species identification	8 442 - 1 350 - 1 263
LM-UGent	Fine-tuning on species identification	88 267 - 8 710 - 8 700

[a] Contains all spectra in DRIAMS with at least 200 detected peaks.
[b] Of which 97 783, 14 055, and 14 183 have species labels in train, val and test splits, respectively.
[c] Numbers reflect spectra. In total, 409 395, 76 431, and 76 133 AMR labels across 65 drugs are associated with those splits.

cultured and analyzed by the Laboratory of Microbiology at Ghent University. In this manuscript, both datasets will be abbreviated as RKI and LM-UGent, respectively. The RKI dataset contains MALDI-TOF mass spectra from highly pathogenic bacteria, covering similar species as in DRIAMS. The LM-UGent dataset, on the other hand, includes a broader taxonomic range. Both datasets are used for fine-tuning on species identification (multi-class classification). As in Mortier et al. (2021), in order to create a challenging training-validation-test split, spectra are split in such a way that there is no overlap in terms of strains. As a consequence, the models are tested whether they can identify unseen strains of (seen) species. The following rules are used in data splitting: all spectra for a species are assigned to the training set if that species only has one strain in the dataset. If the species has more than one strain, strains are split such that 80% of strains of that species are assigned to the training set, and the other 20% evenly split between validation and test set (with a floor value of at least one strain being assigned to either validation or test set). The total number of species in the RKI training set spans 270, of which 106 and 108 are present in the validation and test set, respectively. For the LM-UGent dataset, these numbers are 1088, 200, and 202, for the training, validation and test set, respectively. For more details on the LM-UGent dataset, the reader is referred to Mortier et al. (2021). Table 1 lists a summary of the sizes of all used data.

All MALDI-TOF mass spectra are preprocessed using standard practices (Gibb and Strimmer, 2012). Following Weis et al. (2022), all spectra undergo the following steps: (1) square-root transformation of the intensities, (2) smoothing using a Savitzky-Golay filter with half-window size of 10, (3) baseline correction using 20 iterations of the SNIP algorithm, (4) trimming to the 2000-20000 Da range, (5) intensity calibration so that the total intensity sums to 1. For Maldi Transformer, peaks are then detected on the preprocessed spectrum using the persistence transformation, introduced by Weis et al. (2020b). While the original publication proposes this algorithm to nullify other preprocessing steps, we find that prior preprocessing steps help peak detection (see Appendix B). As in Weis et al. (2020b), we select the highest 200 peaks for every spectrum as inputs[1]. Maldi Transformer is compared against baseline methods, which require a fixed-length input. For these models, instead of running peak detection as a last preprocessing step, spectra are instead binned to a 6000-dimensional vector by summing together intensities in intervals of 3 Da.

2.2. Maldi Transformer

Model To introduce Maldi Transformer, let us denote a spectrum as a set of m peaks $\mathcal{S} = \{((m/z)_i, I_i)\}_{i=1}^{m}$, with each peak characterized by its (m/z) value and (preprocessed) intensity I. An annotated dataset $\mathcal{D}$ then consists of a set of n samples $\mathcal{D} = \{(\mathcal{S}_i, y_i^{\text{spec}})\}_{i=1}^{n}$, with y^{spec} the species label.

Maldi Transformer requires a single input representation $\boldsymbol{x} \in \mathbb{R}^d$ for each peak $((m/z)_i, I_i)$. To achieve this, intensities I are linearly transformed to a d-dimensional space. Similarly, (m/z) values are embedded to sinusoidal positional encodings (Vaswani et al., 2017). For any (m/z) value, the positional encoding (PE) at feature index j is

$$PE_{((m/z),j)} = \begin{cases} sin\left(\frac{(m/z)}{10 \cdot 10000^{j/d}}\right), & \text{if } j \text{ is even} \\ cos\left(\frac{(m/z)}{10 \cdot 10000^{(j-1)/d}}\right), & \text{if } j \text{ is odd} \end{cases}$$

Note that this formulation is identical to the one described in Vaswani et al. (2017), apart from the factor 10 division of the (m/z) positions. This division is performed to bring the numerical range of (m/z) values (2000 Da - 20000 Da) closer to the numerical range of positional indices for which this equation was originally designed[2].

The sinusoidal (m/z) embedding and linear intensity I embedding are summed to a single input representation per peak $\boldsymbol{x}_{i \in \{1,\ldots,m\}}$. A trainable [CLS] vector is prepended to the input for spectrum-level

[1] Note that Maldi Transformer can, in principle, deal with variable number of peaks per spectrum.

[2] Language transformers are often trained with maximum sequence lengths between 512 and 2048. (Beltagy et al., 2020)

prediction, following common practice (Devlin et al., 2018; Dosovitskiy et al., 2020). The final input to the encoder-only transformer is, hence, $\boldsymbol{X} \in \mathbb{R}^{201 \times d}$, with 201 the number of peaks plus the [CLS] token, and d the hidden dimensionality of the model. The encoder-only transformer processes $\boldsymbol{X}$ to a spectrum-level output embedding $\boldsymbol{p}_{[\texttt{CLS}]}$ and output peak-level embeddings $\boldsymbol{p}_{i \in \{1,\ldots,m\}}$. The design of the transformer encoder blocks follows current state-of-the-art practices (Appendix E Figure 6) (Narang et al., 2021). An overview of the model is visualized in Figure 1.

Pre-training task design To boost the performance of Maldi Transformer on supervised tasks with limited labeled data, a novel self-supervised pre-training strategy is designed. We propose to pre-train Maldi Transformers as peak discriminators. That is, in a batch of spectra, some peaks are randomly sampled to use for training. Following Devlin et al. (2018), we sample 15% of the peaks. Half of those sampled peaks are shuffled among all spectra, while the others are kept as part of their original spectrum. Using the shuffled peaks as negative "noise" peaks in a spectrum, a discriminative model is trained to distinguish the noise peaks from the sampled original ones using the cross-entropy loss.

More formally, for a spectrum $\mathcal{S}$, let us denote its version with shuffled peaks as $\mathcal{S}^{\text{shuff}}$. Further, let $\boldsymbol{i}_{\text{train}} = \{i_1, \ldots, i_s\}$ denote the indices of its sampled peaks. Maldi Transformer processes $\mathcal{S}^{\text{shuff}}$ to output peak-level embeddings $\boldsymbol{p}_{i \in \{1,\ldots,m\}}$, which are then processed to predictions $\hat{y}^{\text{peak}}_{i \in \{1,\ldots,m\}} = \sigma\left(\boldsymbol{w}_p^{\top} \boldsymbol{p}_{i \in \{1,\ldots,m\}}\right)$, with $\boldsymbol{w}_p$ a learnable weight vector. The peak discrimination loss is, then, given by

$$\mathcal{L}_{\texttt{peaks}} = \mathbb{E}\Bigg(\sum_{\boldsymbol{i}_{\text{train}}} - \mathbb{I}\left(\mathcal{S}_{ij} = \mathcal{S}^{\text{shuff}}_{ij}\right) \log\left(\hat{y}^{\text{peak}}_{ij}\right) - \mathbb{I}\left(\mathcal{S}_{ij} \neq \mathcal{S}^{\text{shuff}}_{ij}\right) \log\left(1 - \hat{y}^{\text{peak}}_{ij}\right) \Bigg)$$

with $\mathbb{I}$ the indicator function.

Complementary to the peak discrimination strategy, the spectrum embedding $\boldsymbol{p}_{[\texttt{CLS}]}$ is sent to a multi-class linear output head to predict the microbial species identity y^{spec} of the original spectrum[3]. The final pre-training loss function is the sum of peak discrimination binary cross-entropy and species identification multi-class cross-entropy: $\mathcal{L} = \mathcal{L}_{\texttt{peaks}} + \lambda\, \mathcal{L}_{\texttt{spec}}$, with $\lambda \sim \text{Bern}(0.01)$. The species identification loss $\mathcal{L}_{\texttt{spec}}$ is only randomly applied in 1% of the training steps. This is performed because, empirically, overfitting of the species classification task is observed when applied at every training step (see Appendix E Figure 7). A visual representation of the entire pre-training approach is shown in Figure 1. A more-thorough description of the whole pre-training strategy can be found in Appendix C Algorithm 1.

Table 2 | Pre-training configurations for different Maldi Transformer sizes. Species clf λ refers to the probability that the species identification loss is applied per training step.

	Model			
Hyperparameter	**S**	**M**	**L**	**XL**
# params	1.65M	3.27M	6.92M	14.84M
# layers	4	6	8	10
hidden dim d	160	184	232	304
# heads	8	8	8	8
Learning rate	5e-4	5e-4	5e-4	3e-4
Pre-training steps	500 000	500 000	500 000	400 000
Species clf λ	Bern(0.01)	Bern(0.01)	Bern(0.01)	Bern(0.005)

Our novel peak discrimination pre-training strategy is proposed due to conceptual difficulties with porting established pre-training approaches to the MALDI-TOF MS domain, a point further elaborated on in Appendix C. We benchmark this novel pre-training strategy against alternatives in §3.2.

Model configurations We train Maldi Transformer in four different sizes: S, M, L, and XL. Model sizes are chosen so that the total weight numbers roughly correspond to the ones in De Waele et al. (2023). Table 2 lists the size of all models, along with some hyperparameter settings. The Adam optimizer is used to pre-train all models in BFloat16 mixed precision (Kingma and Ba, 2014). Gradients are clipped to a norm of 1. A batch size of 1024 is applied for all models. A linear learning rate warm-up is applied over the first 2500 steps, after which the learning rate remains constant (Table 2). During training, a dropout of 0.2 is applied in the GLU feedforward and over the attention matrix.

2.3. Downstream tasks

As mentioned in §2.1, Maldi Transformer's performance is validated on three downstream supervised

[3] While this is not a purely self-supervised training objective, we justify its use with two reasons. First, in language, pre-training with a sentence-level [CLS] task boosts downstream performance (Devlin et al., 2018). Second, due to the integrated nature of MS manufacturers' species identification pipelines, bacterial species labels are relatively easy to come by. As such, no manual labeling is necessary to obtain these labels. Additionally, it has to be noted that, as not all spectra in DRIAMS carry a species label, the species classification loss is only calculated for labeled spectra.

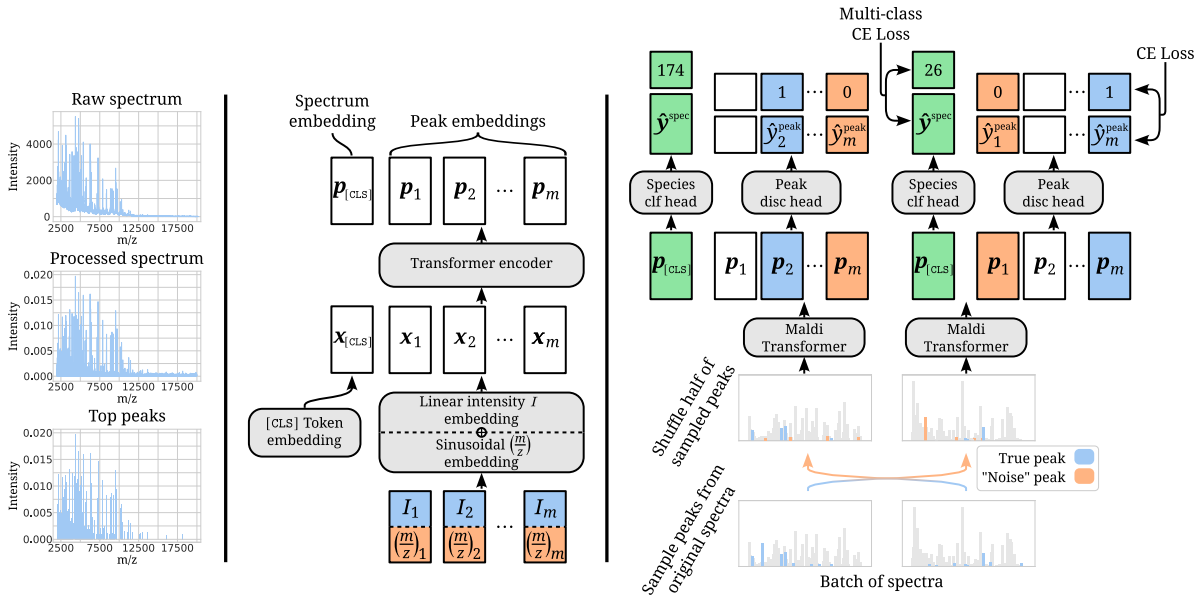

Figure 1 | **Left:** Representation of MALDI-TOF mass spectra. A raw spectrum is preprocessed, after which a topological persistence transformation is performed to detect peaks, resulting in a sparse representation of the spectrum (Weis et al., 2020b). **Middle:** Maldi Transformer. A peak is characterized by an intensity I and (m/z) value. Both are embedded to higher dimensional space by a linear layer and sinusoidal embedding, respectively, and are then summed. A [CLS] token is prepended and the resulting vectors $\boldsymbol{x}$ are sent through multiple Transformer encoder layers. The resulting output vectors $\boldsymbol{p}_{i\in\{1,\ldots,m\}}$ can be used as peak embeddings. Additionally, $\boldsymbol{p}_{[\mathtt{CLS}]}$ can be used as a summary embedding of the whole spectrum. **Right:** Proposed peak discrimination pre-training strategy. In a mini-batch of spectra, 15% of peaks are randomly sampled, half of which are shuffled among all spectra in the batch. The resulting spectra are encoded with Maldi Transformer. The resulting peak embeddings $\boldsymbol{p}_{i\in\{1,\ldots,m\}}$ are sent through a linear output head trying to distinguish original peaks (blue) from shuffled "noise" peaks (orange). Spectrum embeddings $\boldsymbol{p}_{[\mathtt{CLS}]}$ are sent through a separate linear head to predict species identity (green).

tasks. For all three tasks, the pre-trained model is plugged in at initialization and all weights are fine-tuned (i.e. no weight freezing). A task-specific linear output head $\boldsymbol{W}_{out}$ projecting the spectrum embedding $\boldsymbol{p}_{[\mathtt{CLS}]}$ to the desired output space is trained from scratch. For AMR prediction, Maldi Transformer is used as a spectrum embedder in a dual-branch neural network recommender system. To compare its performance against previous results, recommenders are trained with the four best-scoring drug embedders in De Waele et al. (2023). To benchmark Maldi Transformer in terms of species identification, the RKI and LM-UGent datasets are used. Species identification is compared to MLP baselines, Logistic Regression, Random Forest, and k-nearest neighbors (k-NN) models. Details on the exact training setups for each downstream tasks are found in Appendix D.

The LM-UGent dataset covers a broader species diversity than the clinical species found in the pre-training set. As such, this dataset can be considered out-of-distribution for the pre-trained Maldi Transformer. For this reason, a domain adaptation step on the pre-trained Maldi Transformer is performed before supervised fine-tuning. The domain adaptation step consists of pre-training Maldi Transformer for 20 000 additional steps in the same fashion, but now using the LM-UGent dataset, instead of DRIAMS[4].

3. Results

3.1. Maldi Transformer improves performance on downstream tasks

Pre-training curves for Maldi Transformer are shown in Appendix E Figure 8. After pre-training, Maldi Transformer is fine-tuned w.r.t. a downstream task.

[4]By performing domain adaptation, in contrast to the other supervised tasks, the task-specific output head $\boldsymbol{W}_{out}$ does not need to be initialized from scratch anymore, but can be copied from the pre-trained model, as it already has the right dimensions.

Maldi Transformer's downstream performance is compared to preprocessed and binned baselines. For AMR prediction, it is compared to MLPs of similar sizes. For species identification, it is additionally compared to non-neural network baselines.

Figure 2 shows the experimental results for all downstream tasks. For AMR prediction, models are evaluated in terms of micro ROC-AUC and instance-wise ROC-AUC[5]. For species identification, they are evaluated in terms of species- and genus-level accuracy. In general, Maldi Transformer obtains superior performance in comparison to other tested methods. For AMR prediction, Maldi Transformer consistently outperforms all MLP models in terms of micro ROC-AUC. In terms of the instance-wise ROC-AUC, Maldi Transformer is sometimes outperformed by MLP models, but the best-scoring model overall is still one using Maldi Transformer (i.e. the large model paired with a Morgan fingerprint drug embedder).

For species identification on the RKI dataset, Maldi Transformer does not always provide a clear advantage. On species-level, it is consistently outperformed by both Logistic Regression and similar-sized MLPs. For genus-level accuracy, however, Maldi Transformers surpass their MLP counterparts. While a k-NN model provides competitive performance, the best overall model in terms of genus-level accuracy is the large Maldi Transformer.

On the larger and more-difficult LM-UGent dataset, results are again more convincing. The best-performing species-level model is the medium-sized Maldi Transformer, convincingly beating other models with 84% accuracy. The same results are obtained on genus-level accuracy, where Maldi Transformer consistently beats other models.

We hypothesize that, on small MALDI-TOF datasets with relatively-simple prediction tasks (e.g. the RKI dataset), simple models are sufficient, but Maldi Transformer remains competitive. For comparatively-complex tasks, such as AMR prediction with dual-branch neural networks, and species identification across >1000 species, Maldi Transformer consistently delivers state-of-the-art performance.

[5]An explanation of the instance-wise ROC-AUC is given in De Waele et al. (2023).

3.2. Maldi Transformer pre-training ablation

To further validate the efficacy of the proposed Maldi Transformer, its performance is compared against two alternative realizations of transformers on MALDI-TOF MS data (i.e. using two alternative pre-training strategies). The first one takes inspiration from the masked language model (MLM) BERT (Devlin et al., 2018). Here, intensity values of peaks are randomly masked out, and a transformer is pre-trained to predict the original intensities back using the mean squared error loss. In the second, a discriminative model much like our final proposed strategy is trained. The difference in this model is that negative peaks are sampled from some estimated distribution of peaks, instead of generating negative peaks by shuffling. Both alternative strategies are described in greater detail in Appendix C.

In Figure 3.2A, it is observed that both alternative pre-training techniques are outperformed by our final proposed strategy. Naively porting the MLM strategy to peak intensity regression underdelivers by a wide margin. We hypothesize that the intensity of a peak is of lesser importance compared to whether that peak is present or not, hence making a regression model learn superfluous information. A more-biologically relevant training objective would be to make the model learn over peak co-occurrence. A negative peak sampling strategy paired with a binary discrimination objective would achieve such a task. It can be seen from Figure 3.2A that this task, however close in concept to the final Maldi Transformer, still consistently underperforms. A possible explanation could be that the estimation of negative peaks relies on some estimated overall distribution of peaks. Any inaccuracies in this distribution result in "unrealistic" negative peaks, which are easier to recognize by the model. Thus, instead of truly reasoning over peak co-occurrence, the model takes a shortcut and learns the inaccuracies of the underlying peak sampler. Our final peak shuffling strategy does not have this drawback, and is, therefore, ideally suited for the MALDI-TOF MS domain, also resulting in superior performances.

Figure 3.2B shows ablation results when leaving different parts out of the final pre-training strategy. Maldi Transformer is compared to models where one of the two loss components are left out: either the peak discrimination loss ($\mathcal{L}_{\texttt{peaks}}$), or the species identification loss ($\mathcal{L}_{\texttt{spec}}$). It is also compared to a transformer model without pre-training whatsoever. It can be seen that the latter strategy (i.e. training a separate

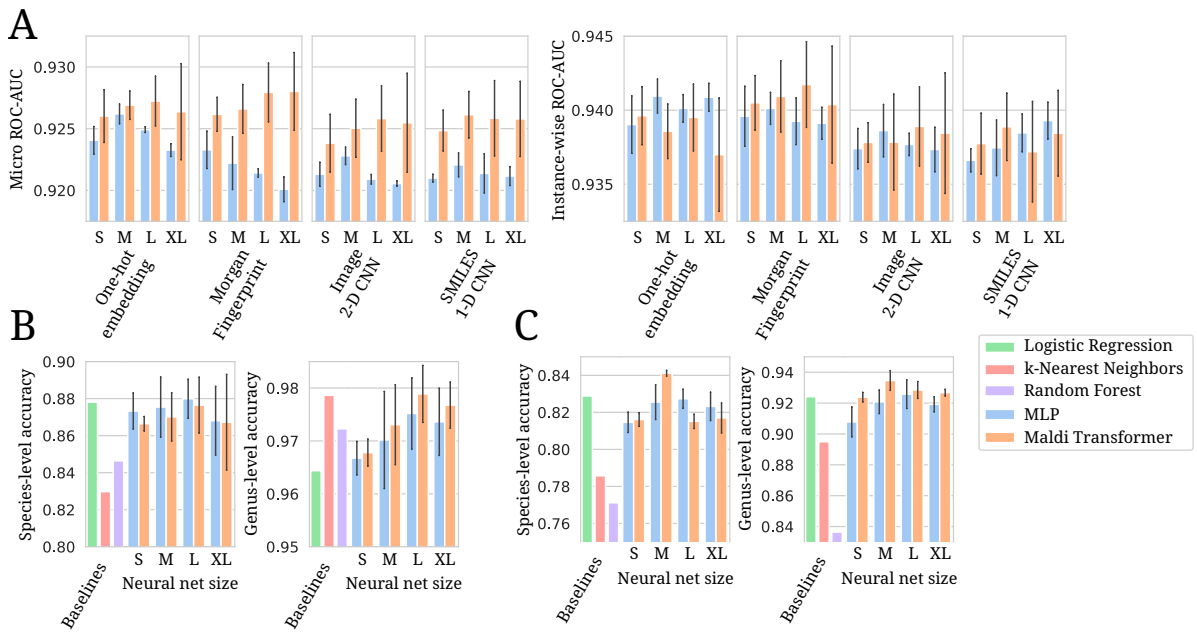

Figure 2 | Barplots of all main experimental results. Error-bars indicate standard deviation over three independent runs. Neural network model sizes range from S to XL. Details per model size for Maldi Transformer and MLP models are listed in Table 2 and Appendix D Table 3, respectively. **A:** AMR prediction results on DRIAMS. Performances are shown for models using four different drug embedders in a dual-branch recommender system. **B:** Species identification results on the RKI dataset. **C:** Species identification results on the LM-UGent dataset.

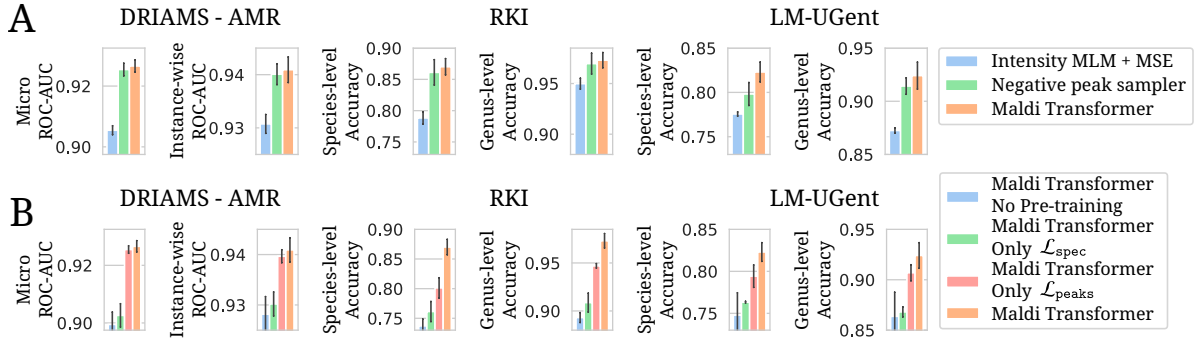

Figure 3 | Barplots of all ablation results. Only results for Medium-sized transformers are shown. The final model as discussed in §2.2 is labeled as Maldi Transformer. Error-bars indicate standard deviation over three independent runs. For DRIAMS AMR prediction, only results using a Morgan Fingerprint drug embedder are shown. For the LM-UGent dataset, results are shown without the domain adaptation step, as this step was also not performed for ablation models. See Appendix E Figure 9 for the effects of this step. **A:** Testing different ways of pre-training a transformer on MALDI-TOF MS data. **B:** Leaving out different parts of the final pre-training strategy.

transformer from scratch for every supervised task), delivers subpar performance. In addition, only pre-training on species identification does not help much. The biggest gain is made from the peak discrimination task. Yet, overall, the effects are additive, showing that each part contributes to the final performance.

3.3. Maldi Transformer model analysis

Due to its pre-training design, Maldi Transformer outputs the probability of a peak belonging to its spectrum for every peak. More formally, let us denote the output probability of a peak i belonging to its spectrum $\mathcal{S}$ as $\Pr(y_i^{\text{peak}} = 1|\mathcal{S})$. This quantity can be

obtained through the pre-trained Maldi Transformer: $\Pr(y_i^{\text{peak}} = 1|\mathcal{S}) = \hat{y}_p^{\text{peak}} = \sigma(\boldsymbol{w}_p^\intercal \cdot \boldsymbol{p}_p)$. During pre-training, as peaks are shuffled between spectra, one expects to encounter both positives and negative true labels. In this section, we examine how peak output probabilities $\Pr(y_i^{\text{peak}} = 1|\mathcal{S})$ may be used during inference (i.e. without shuffling). The interpretation of probabilities requires a model to be calibrated. This property is further examined in Appendix E Figure 10. Results in the following paragraphs are derived from the DRIAMS pre-training test set.

Maldi Transformer denoises spectra In Figure 4A, a randomly selected *Staphylococcus epidermidis*[6] spectrum is visualized, each peak colored according to its output probability $\Pr(y_i^{\text{peak}} = 1|\mathcal{S})$. Most peaks are (correctly) assigned a high probability of belonging to their spectrum. The peaks with a low probability could be mistaken by the model, or could represent "true" noise in the spectrum. Such noise may still be present in the final spectrum due to, for example (1) noisy readouts from the spectrometer, or (2) shortcomings in preprocessing. In order to validate this hypothesis, it makes sense to look at patterns across multiple spectra. In Figure 4E, all *S. epidermidis* spectra are visualized together, each peak as a single dot. Black dots represent peaks that are predicted to be noise with high probability ($\Pr(y_i^{\text{peak}} = 1|\mathcal{S}) \leq 0.05$). In the magnified parts of the plot, it can be seen that blue dots cluster together, meaning that *S. epidermidis* spectra often have peaks in the same places. Black dots mainly fall outside or on the edges of those clusters, signifying that those peaks are rightly picked up by the model as noise. Consequently, Maldi Transformers can serve a broader purpose as spectrum denoisers, in addition to their supervised learning capabilities.

Noisy peaks are indicative of lower downstream performance In order to better grasp the effect of noisy peaks on spectra, peak predictions are examined across all spectra in the DRIAMS test set. Figure 4B shows the empirical cumulative distribution of peak output probabilities. Only approximately 5% of all peaks are assigned a probability of belonging to their respective spectrum $\Pr(y_i^{\text{peak}} = 1|\mathcal{S})$ of lower than 50%. Conversely, half of the peaks are predicted with a probability of 98.5% or greater. This shows that while noisy peaks do exist, they typically make up a small percentage of the overall input spectrum.

[6] *Staphylococcus epidermidis* is the most occurring species in the DRIAMS pre-training test set.

Noisy peaks may not uniformly occur across the dataset. Some spectra may have more noisy peaks than others. A good spectrum quality statistic can, hence, be the fraction of peaks that are confidently predicted as "true", e.g. with a probability greater than 95%. Figure 4C shows the distribution of spectra in function of this statistic. A slight left tail in the distribution signifies that some spectra have a large amount of noisy peaks. Aproximately 10% of the spectra have more than half of their peaks predicted with a probability smaller than 95%. Figure 4D shows that this statistic is also indicative of predictive performance. Spectra with more confidently "belonging" peaks have a higher species-level accuracy (on DRIAMS test set labels). Species-level accuracy ramps up from 90% (or lower) for spectra with a lot of noisy peaks, up to nearly-perfect accuracy for spectra with (almost) all "predicted true" peaks[7]. The fraction of "predicted true" peaks also correlates with prediction certainty (see Appendix E Figure 11). These results showcase Maldi Transformer's ability to not only improve performance, but also provide further insights into the data.

4. Discussion

As with many subfields of machine learning, size, quality, and diversity of data constitute a huge bottleneck. In this study, an apparent ceiling in representational capacity has been obtained given the available public data. This is evidenced by the fact that the XL variant of Maldi Transformer typically does not give an advantage in downstream tasks over the M or L model. Its size (~15M weights), however, is still small by self-supervised transformer standards. We hypothesize that the representational capacity of larger transformers is simply not necessary for the relatively-simple datasets that are available in the MALDI-TOF MS domain. Because of this, we argue that more efforts to collect and publish data may be the most important factor for MALDI-TOF-based machine learning research to continue to flourish. As spectral data is routinely generated within hundreds, if not thousands, of hospitals, we envision collaboration efforts with healthcare (as in Weis et al. (2022)) to play a crucial role in this regard.

Deep learning models have been hailed as feature extractors. For this class of models, it is typically ar-

[7] Note that DRIAMS species labels are produced from the MS manufacturers' software and models. It can be expected that it is relatively easy for a model to reproduce the labels (predictions) from another model. Hence, the nearly-perfect species-level accuracy on DRIAMS is not unexpected.

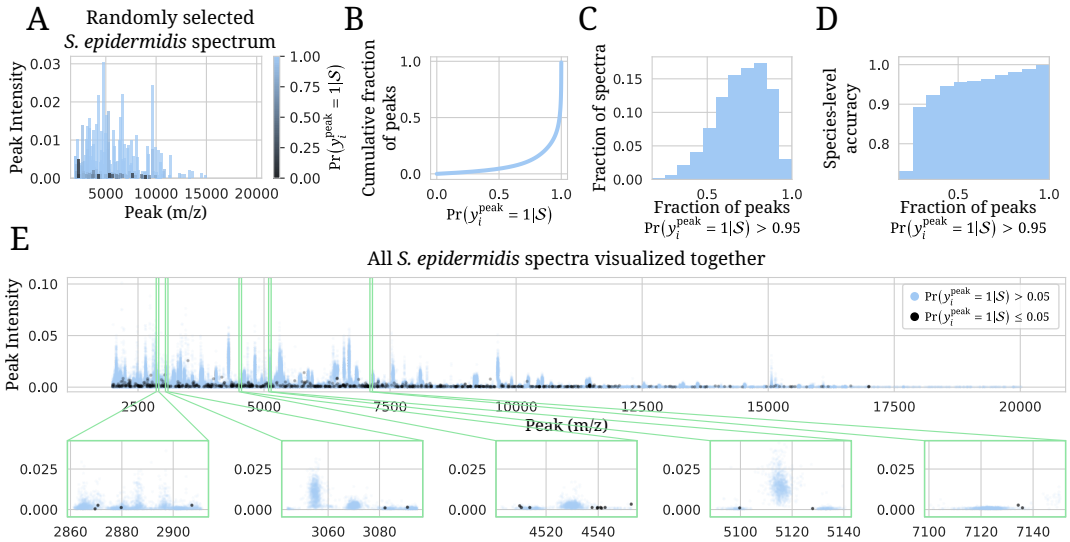

Figure 4 | Maldi Transformer analysis on DRIAMS pre-training test data. Note that no shuffling of peaks across spectra is performed to generate these plots. Each spectrum contains their original detected peaks. **A:** A randomly selected spectrum, each peak colored by the model's output probability of that peak belonging to that spectrum or not $\Pr\left(y_i^{\text{peak}} = 1|\mathcal{S}\right)$. **B:** Empirical cumulative distribution of those output probabilities over all spectra in the DRIAMS test set. **C:** Histogram of fraction of peaks confidently (> 0.95) predicted as "true" peaks per spectrum. **D:** Species-level accuracy in function of fraction of peaks confidently (> 0.95) predicted as "true" peaks per spectrum. **E:** All *S. epidermidis* spectra visualized together, every peak as a separate dot. Peaks confidently predicted as "noise" (≤ 0.05) are shown in black. Zoomed in subplots show that "noise" peaks originate outside clusters of usual peak locations.

gued that it matters less how features are presented to the model, as they can compose the relevant features themselves in their hidden representations. We would caution against this perspective, and state that how inputs are presented to a deep learning model could have far-reaching impacts on what the model can learn. Taking language as an example, character-level language models traditionally underperform the same models trained on (sub)word-level. In this work, MALDI-TOF mass spectra are presented to the model by their peaks. Because of this, any preprocessing steps and algorithms to determine peaks play crucial roles in the final performance of the model. For this reason, experimentation with peak detection algorithms is a promising future research direction.

While this work is not the first to fit a transformer model on mass spectral data (Yilmaz et al., 2022), it is the first to propose a self-supervised learning strategy adapted to this data modality. It is expected that the combination of shuffling peaks to obtain negative examples, paired with a peak discriminator, could be useful in other mass spectral domains — namely, the types of mass spectra where the (co-)occurrence of peaks constitute the most-critical biological signal. For this reason, we hope that our ablations (§3.2) and discussion in Appendix C serve as useful guidelines for adapting this work to other mass spectral data modalities, generated with MS instruments using alternative ionisation and separation methods.

As the pre-trained model can be interpreted as learning peak co-occurences, this property is examined in §3.3. There, it is shown that Maldi Transformer can be used to detect noisy peaks. The predicted absence of noisy peaks is found to correlate with higher predictive performance. These insights go beyond those offered by off-the-shelf machine learning techniques. Paired with its state-of-the-art (or competitive) performance results, it can be concluded that Maldi Transformer enhances what biological practitioners get out of their MALDI-TOF MS data.

Acknowledgements

This work was supported by Research Foundation - Flanders (FWO) [PhD Fellowship fundamental research grant 1153024N to G.D.W.]. W.W. also received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" Programme

References

Olivier Dauwalder, Tiphaine Cecchini, Jean Philippe Rasigade, and François Vandenesch. Matrix assisted laser desorption ionisation/time of flight (maldi/tof) mass spectrometry is not done revolutionizing clinical microbiology diagnostic. *Clinical Microbiology and Infection*, 29(2):127–129, 2023.

Caroline V Weis, Catherine R Jutzeler, and Karsten Borgwardt. Machine learning for microbial identification and antimicrobial susceptibility testing on maldi-tof mass spectra: a systematic review. *Clinical Microbiology and Infection*, 26(10):1310–1317, 2020a.

Piseth Seng, Michel Drancourt, Frédérique Gouriet, Bernard La Scola, Pierre-Edouard Fournier, Jean Marc Rolain, and Didier Raoult. Ongoing revolution in bacteriology: routine identification of bacteria by matrix-assisted laser desorption ionization time-of-flight mass spectrometry. *Clinical infectious diseases*, 49(4):543–551, 2009.

Alain Bizzini, Katia Jaton, Daniel Romo, Jacques Bille, Guy Prod'hom, and Gilbert Greub. Matrix-assisted laser desorption ionization–time of flight mass spectrometry as an alternative to 16s rrna gene sequencing for identification of difficult-to-identify bacterial strains. *Journal of clinical microbiology*, 49(2):693–696, 2011.

Alex Van Belkum, Martin Welker, Marcel Erhard, and Sonia Chatellier. Biomedical mass spectrometry in today's and tomorrow's clinical microbiology laboratories. *Journal of clinical microbiology*, 50(5): 1513–1517, 2012.

Walter Florio, Arianna Tavanti, Simona Barnini, Emilia Ghelardi, and Antonella Lupetti. Recent advances and ongoing challenges in the diagnosis of microbial infections by maldi-tof mass spectrometry. *Frontiers in microbiology*, 9:1097, 2018.

Yan Cao, Lei Wang, Ping Ma, Wenting Fan, Bing Gu, and Shaoqing Ju. Accuracy of matrix-assisted laser desorption ionization–time of flight mass spectrometry for identification of mycobacteria: a systematic review and meta-analysis. *Scientific reports*, 8(1):1–9, 2018.

Georgia Vrioni, Constantinos Tsiamis, George Oikonomidis, Kalliopi Theodoridou, Violeta Kapsimali, and Athanasios Tsakris. Maldi-tof mass spectrometry technology for detecting biomarkers of antimicrobial resistance: current achievements and future perspectives. *Annals of translational medicine*, 6 (12), 2018.

Justin M Hettick, Michael L Kashon, James E Slaven, Yan Ma, Janet P Simpson, Paul D Siegel, Gerald N Mazurek, and David N Weissman. Discrimination of intact mycobacteria at the strain level: a combined maldi-tof ms and biostatistical analysis. *Proteomics*, 6(24):6416–6425, 2006.

Caroline Weis, Aline Cuénod, Bastian Rieck, Olivier Dubuis, Susanne Graf, Claudia Lang, Michael Oberle, Maximilian Brackmann, Kirstine K Søgaard, Michael Osthoff, et al. Direct antimicrobial resistance prediction from clinical maldi-tof mass spectra using machine learning. *Nature Medicine*, 28(1):164–174, 2022.

Julie Gagnaire, Olivier Dauwalder, Sandrine Boisset, David Khau, Anne-Marie Freydiere, Florence Ader, Michèle Bes, Gerard Lina, Anne Tristan, Marie-Elisabeth Reverdy, et al. Detection of staphylococcus aureus delta-toxin production by whole-cell maldi-tof mass spectrometry. *PloS one*, 7(7):e40660, 2012.

Gaetan De Waele, Gerben Menschaert, and Willem Waegeman. An antimicrobial drug recommender system using maldi-tof ms and dual-branch neural networks. *bioRxiv*, pages 2023–09, 2023.

Thomas Mortier, Anneleen D Wieme, Peter Vandamme, and Willem Waegeman. Bacterial species identification using maldi-tof mass spectrometry and machine learning techniques: A large-scale benchmarking study. *Computational and Structural Biotechnology Journal*, 19:6157–6168, 2021.

Caroline Weis, Max Horn, Bastian Rieck, Aline Cuénod, Adrian Egli, and Karsten Borgwardt. Topological and kernel-based microbial phenotype prediction from maldi-tof mass spectra. *Bioinformatics*, 36(Supplement_1):i30–i38, 2020b.

Kévin Vervier, Pierre Mahé, Jean-Baptiste Veyrieras, and Jean-Philippe Vert. Benchmark of structured machine learning methods for microbial identification from mass-spectrometry data. *arXiv preprint arXiv:1506.07251*, 2015.

Nam K Tran, Taylor Howard, Ryan Walsh, John Pepper, Julia Loegering, Brett Phinney, Michelle R Salemi, and Hooman H Rashidi. Novel application of automated machine learning with maldi-tof-ms for rapid high-throughput screening of covid-19: A proof of concept. *Scientific reports*, 11(1):8219, 2021.

Victor Ryzhov and Catherine Fenselau. Characteriza-

tion of the protein subset desorbed by maldi from whole bacterial cells. *Analytical chemistry*, 73(4): 746–750, 2001.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

Jim Clauwaert and Willem Waegeman. Novel transformer networks for improved sequence labeling in genomics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(1):97–106, 2020.

Jörg Franke, Frederic Runge, and Frank Hutter. Probabilistic transformer: Modelling ambiguities and distributions for rna folding and molecule design. *Advances in Neural Information Processing Systems*, 35:26856–26873, 2022.

Melih Yilmaz, William Fondrie, Wout Bittremieux, Sewoong Oh, and William S Noble. De novo mass spectrometry peptide sequencing with a transformer model. In *International Conference on Machine Learning*, pages 25514–25522. PMLR, 2022.

Peter Lasch, Maren Stämmler, and Andy Schneider. Version 4 (20230306) of the maldi-tof mass spectrometry database for identification and classification of highly pathogenic microorganisms from the robert koch-institute (rki), 2023. URL https://doi.org/10.5281/zenodo.7702375.

Sebastian Gibb and Korbinian Strimmer. Maldiquant: a versatile r package for the analysis of mass spectrometry data. *Bioinformatics*, 28(17):2270–2271, 2012.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Sharan Narang, Hyung Won Chung, Yi Tay, William Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, et al. Do transformer modifications transfer across implementations and applications? *arXiv preprint arXiv:2102.11972*, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

A. DRIAMS species processing

DRIAMS contains species labels for many spectra. These species labels are derived from the species identification pipelines included with the MALDI-TOF MS machines. As such, some preprocessing steps are taken to make the labels as presentable as possible to an ML model.

Firstly, labels in DRIAMS containing the string `"not reliable identification"` are integrally deleted. Second, many spectra are labeled as `"MIX!species"`, indicating that the spectrum potentially contains an impure mixture of species. As such, species labels for these spectra are also not used. Additionally, species labels that occur fewer than five times in the training set are removed. Finally, species labels occurring only in the validation or test set, but not in the training set, are similarly removed. After processing, DRIAMS contains 469 different species labels. Note that the previous steps involve removing of labels, not spectra themselves. Corresponding spectra are still kept in the dataset, albeit without a species label.

B. On preprocessing and persistence transformation

While Weis et al. (2020b) argue that persistence transformation nullifies the need for the parameter-heavy chain of preprocessing steps, we advocate for the opposite. To support this stance, a simple exploratory visualization is made.

In Figure 5, all *Bacillus anthracis*[8] spectra in the RKI training set are put through two preprocessing chains. The first only does (1) trimming to the 2000-20000 Da range, (2) intensity calibration so that the total intensity sums to 1, and (3) persistence transformation, and keeps the 200 highest peaks. The second preprocessing chain performs all those steps preceded by variance stabilization, smoothing, and background removal, as in §2.1. Then, across all preprocessed spectra, the occurrence of detected peaks in bins of 3 Da is counted and plotted in Figure 5. The resulting profile can be considered a summary profile of all detected peaks in *Bacillus anthracis* spectra. The detected peaks with extra preprocessing result in a cleaner profile, with peaks more concentrated in regions that clearly correspond to biologically-informative signal (especially notable when inspecting the left tail of the spectra). In contrast, without prior preprocessing, detected peaks are seemingly more-randomly distributed, hinting that noise in the spectrum negatively affects peak detection.

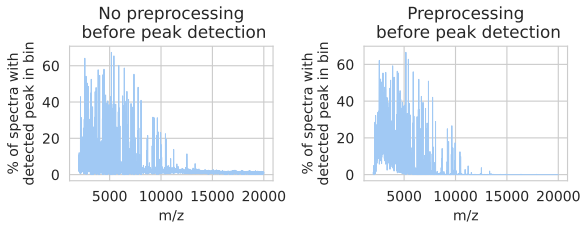

Figure 5 | Visualization of summary profiles of *Bacillus anthracis* RKI training spectra obtained when doing peak detection with (R) or without (L) preprocessing first.

C. Design of a self-supervised pre-training task

A large part of this study concerns the design of a self-supervised pre-training task for MALDI-TOF MS data. This is motivated by the scarcity of data in this domain. Through self-supervised learning, a greater representational capacity can be obtained, maximally utilizing patterns in the data. The following paragraphs chronicle thoughts and experiments w.r.t. the design of the SSL task.

MALDI-TOF MS spectra can be represented as sets of peaks. As each peak is characterized by an intensity and an (m/z) value, a natural parallel is drawn with language data. Instead of a sequence of words,

[8] *Bacillus anthracis* is chosen because it is the most occurring species in the RKI training dataset.

a sequence of peaks is processed. A crucial difference is that, for language transformers, positional indices for each text "token" are set to the integer range numbering 0 to $n-1$, with n the number of tokens in the input sequence. With MALDI-TOF MS spectra, positional indices (i.e. (m/z) values) are irregularly spaced and real-valued. Additionally, whereas words have a categorical identity in text, MALDI-TOF MS intensity is also real-valued. These two factors have to be taken into consideration when porting self-supervised learning techniques from one modality to the mass spectral domain.

In language transformer pre-training, perhaps the two most-classically quoted techniques are masked language modeling (MLM) (Devlin et al., 2018) and autoregressive modeling (AR) (Radford et al., 2018). The GPT series of models best exemplifies the success of the latter category (Brown et al., 2020). For MALDI-TOF MS, however, it is unclear how to autoregressively model a set of peaks. Autoregressive models imply a certain ordering of data, whereas this perspective is ill-fitting for the set-valued peaks input. For example, how does one train to generate the "next" peak given the previous peaks, if it is unclear how to define what the "next" peak is? For example, does one choose to order peaks by their height, or by their (m/z) value? If the model predicts the second-next peak instead of the next, is that necessarily wrong? If not, how to efficienctly construct the loss to take this into account?

Instead of answering the issues with autoregressively modeling a set-valued input, we may consider the alternative strategy: MLM, introduced by BERT (Devlin et al., 2018). For example, instead of masking out words, intensities can be masked, and a model can be trained to recover the intensities with the use of the mean-square error loss. This brings us to first ablated pre-training technique in §3.2. The training for this strategy is performed identical to the final strategy outlined in §2.2. The only difference being that the peak discrimination loss $\mathcal{L}_{\texttt{peaks}}$ is exchanged for the mean-square error loss on masked peaks. Peaks to train on are similarly selected with 15% probability. Of those 15%, 80% are assigned masked intensities and 20% are left unchanged.

After fitting a regression MLM on intensities with limited success, a logical next step is to design a self-supervised classification task. The following paragraphs describe the exact procedure resulting in the negative peak sampler model in §3.2.

Taking inspiration from contrastive learning (Liu et al., 2021), a per-peak classification task can ask a model whether a peak belongs to the rest of the spectrum or not. Just as in contrastive learning, this strategy requires to sample negative peaks to deliver negative samples. One way to generate negative peaks is to randomly generate them from some estimated distribution of peaks. Here, we estimate the distribution of peaks in two steps. First, all peak locations in the training dataset are collected: $P_{\text{train}} = \left\{ \left(\frac{m}{z}\right)_j \middle| \left(\frac{m}{z}\right)_j \in S_i \right\}_{i=1}^{n}$, and a probability mass function over discrete bins is calculated:

$$\Pr(b_i) = \frac{\left| \{ p_j | p_j \in P_{\text{train}} \wedge p_j \in b_i \} \right|}{\left| \{ p_j | p_j \in P_{\text{train}} \} \right|} \tag{1}$$

with bins chosen by 1 Da intervals: $b_i \in \,]i, i+1]$, for $i \in \{2000, 2001, \ldots, 19\,999\}$. In other words, a probability mass function over 1 Da bins is created by counting how many times peaks fall into each bin in the training data. Next, within each bin, quantiles of the found intensities therein are calculated: $Q_{b_i} = \{q_{0.00}, q_{0.01}, \ldots, q_{0.99}, q_{1.00}\}$. To sample a negative peak, an (m/z) value is first sampled by uniformly sampling a location within a sampled bin: $(m/z) \sim \Pr(b_i) + \text{Unif}[0,1]$. After, an intensity is drawn by uniformly sampling within a uniformly sampled interquantile range: $I \sim \text{Unif}[q_j, q_{j+1} \mid q \in Q_{b_i}]$, with $j \sim \text{Unif}\{0.00, 0.01, \ldots 0.99\}$. The training for this strategy is performed identical to the final strategy outlined in §2.2. The only difference being that, instead of shuffling peaks around to generate negatives, negative peaks are now sampled. In every training step, 15% of the peaks are selected for training, half of which are exchanged for sampled negative ones.

As discussed in §3.2, generating negative peaks is not ideal for forcing the model to reason over peak co-occurrences. The model is allowed to learn any misrepresentation in negatives as a shortcut. A way to circumvent the issues with generating negative peaks, is to use real ones. One way to present real peaks as negatives, is to take them from other spectra. This is where we land on the final self-supervised strategy that we test, and ultimately propose in our main text (see §2.2, Figure 1, and Algorithm 1).

D. Downstream tasks and baselines

For all three downstream tasks, the pre-trained model is plugged in at initialization and all weights are fine-

Algorithm 1: Maldi Transformer peak discrimination pre-training.

Input: $\boldsymbol{X} \in \mathbb{R}^{b \times m \times d}$, $\boldsymbol{y}^{\text{spec}} \in \mathbb{R}^{b}$
b = batch size, m = number of peaks, d = hidden dim
$\boldsymbol{X}$ = encoded batch of spectra, $\boldsymbol{y}^{\text{spec}}$ = species labels
$\boldsymbol{y}^{\text{spec}}$ contains int labels and NaN if unlabeled

Let $\boldsymbol{U} \in \{0,1\}^{b \times m}$, with $U_{ij} \sim \text{Bern}(0.15)$ # Sample 15% of peaks
Let $\boldsymbol{P} \in \{0,1\}^{b \times m}$, with $P_{ij} \sim \text{Bern}(0.50)$ # Sample 50% of peaks
$\boldsymbol{p} = [(i,j) : U_{ij} = 1 \wedge P_{ij} = 1]$ # Positive label indices
$\boldsymbol{n} = [(i,j) : U_{ij} = 1 \wedge P_{ij} = 0]$ # Negative label indices
$\boldsymbol{X}^{\text{shuff}} = \boldsymbol{X}[\texttt{permute}(\boldsymbol{n})]$ # Shuffle neg peaks
$\boldsymbol{p}_{i \in \{1,\ldots,m\}}, \boldsymbol{p}_{[\texttt{CLS}]} = \texttt{MaldiTransformer}(\boldsymbol{X}^{\text{shuff}})$
$\hat{\boldsymbol{Y}}^{\text{peak}} = \sigma(\boldsymbol{w}_p^{\intercal} \cdot \boldsymbol{p}_{i \in \{1,\ldots,m\}})$ # Projection of peak embeddings
$\hat{\boldsymbol{Y}}^{\text{spec}} = \texttt{Softmax}(\boldsymbol{W}_s^{\intercal} \cdot \boldsymbol{p}_{[\texttt{CLS}]})$ # Projection of spec embed
$\mathcal{L}_{\text{peaks}} = \mathbb{E}\Big(\sum_{\substack{(i,j) \\ U_{ij}=1}} -\mathbb{I}\left(X_{ij} = X_{ij}^{\text{shuff}}\right) \log\left(\hat{Y}_{ij}^{\text{peak}}\right) - \mathbb{I}\left(X_{ij} \neq X_{ij}^{\text{shuff}}\right) - \log\left(1 - \hat{Y}_{ij}^{\text{peak}}\right) \Big)$ # Binary CE on peaks
$\mathcal{L}_{\text{spec}} = \mathbb{E}\left(\sum_i^b -\log \hat{Y}_{i,c}^{\text{spec}} \right)$ # with $c = y_i^{\text{spec}}$ the class index
$\mathcal{L} = \mathcal{L}_{\text{peaks}} + \lambda \cdot \mathcal{L}_{\text{spec}}$, with $\lambda \sim \text{Bern}(0.01)$
Output: $\mathcal{L}$ # Final loss

tuned (i.e. no weight freezing). A task-specific linear output head $\boldsymbol{W}_{out}$ projecting the spectrum embedding $\boldsymbol{p}_{[\texttt{CLS}]}$ to the desired output space is trained from scratch. All downstream models are similarly trained with the Adam optimizer with a batch size of 128 and dropout of 0.2. A linear learning rate warm-up over the first 250 steps is applied, after which the rate is kept constant.

For AMR prediction, training of Maldi Transformer-based recommenders is performed identical to the MLP-based baselines in De Waele et al. (2023). Briefly explained, for every combination of spectrum embedder (four sizes: S, M, L, and XL) and drug embedder (four types), four different learning rates ({1e-5, 5e-5, 1e-4, 5e-4}) are tested. For all these different combinations, three models are trained (using different random seeds for model initialization and batching of data). For every spectrum and drug embedder combination, only results from the best learning rate are presented; that is, the learning rate resulting in the best average validation micro ROC-AUC for that combination. The validation set is checked every tenth of an epoch. Models are trained for a maximum of 50 epochs, and their training is halted early when validation micro ROC-AUC hasn't improved for 10 validation set checks. The checkpoint of the best performing model (in terms of validation micro ROC-AUC) is used as the final model. The baselines for the AMR prediction task are the models described in De Waele et al. (2023), which describes the recommender model structure in greater detail.

The pre-training of Maldi Transformer for species identification is performed in a similar way. The differences consist of: (1) $\boldsymbol{W}_{out}$ returning 270, or 1088, for the RKI and LM-UGent dataset, respectively, (2) five different learning rates are tested: ({1e-5, 5e-5, 1e-4, 5e-4, 1e-3}), and (3) validation species-level accuracy is tracked to halt training early (for a maximum of 250 epochs, and it is only checked once per epoch). As species identification is a multi-class classification task, models are then optimized using a softmax operation, combined with the cross-entropy loss. Species identification is compared to MLP baselines, Logistic Regression, Random Forest, and k-nearest neighbors (k-NN) models. All of these baselines are trained on preprocessed and binned spectra (see §2.1). The S, M, L, and XL MLPs are identical in construction to the spectrum embedders in De Waele et al. (2023), but with n-dimensional outputs instead of 64, with n the number of classes (see Table 3). MLP baselines are trained using the same strategy as Maldi Transformers. That is, for all model sizes, five different learning rates ({1e-5, 5e-5, 1e-4, 5e-4, 1e-3}) are trained in triplicates. Model results from the best learning rate (in terms of validation species-level accuracy) are presented. Model training halts before its maximum of 250 epochs if validation species-level accuracy hasn't increased in 10 epochs, and the model with the best validation accuracy is saved.

Table 3 | All tested model sizes for the MLP baseline Hidden sizes represent the evolution of the hidden state dimensionality as it goes through the model, with every hyphen defining one fully connected layer. n represents the number of output nodes. n equals 64, 270, and 1088 for DRIAMS AMR prediction, species identification on RKI, and LM-UGent, respectively.

Size	# Weights	Hidden sizes
S	1.58M	6000-256-128-n
M	3.25M	6000-512-256-128-n
L	6.85M	6000-1024-512-256-128-n
XL	15.09M	6000-2048-1024-512-256-128-n

For non-neural baseline classifiers (Random Forest, Logistic Regression, and k-NN), a grid-search is performed to find optimal hyperparameters. Given the non-stochastic nature of their implementations, only one model is trained after tuning and, hence, only one test performance is reported. The parameter grid for Random Forest consists of {`max_depth` = `[25,50,75,100]`, `min_samples_split` = `[2,5,10]`, `max_features` = `[10,25,50,100]`}. All random forests are trained with 200 trees. For Logistic Regression: {`standardscaling` = `[True,False]`, `L2_norm` = `[1e3,1e2,10,1,0.1,1e-2,1e-3]`}. And for k-NN:

`{standardscaling = [True,False], n_neighbors = [1,2,3,4,5,6,7,9,10,25].`

E. Figures and tables supporting the methods and results sections

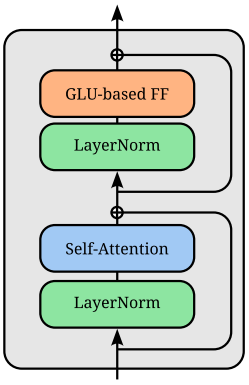

Figure 6 | Utilized transformer encoder block. The network uses pre-LayerNorms and GeLU gated linear units (GLU) in the feedforward (FF) networks (Ba et al., 2016; Shazeer, 2020).

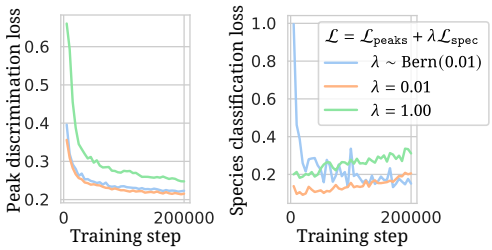

Figure 7 | Pre-training dynamics using different strategies for combining the peak discrimination loss $\mathcal{L}_{\text{peaks}}$ and the species identification loss $\mathcal{L}_{\text{spec}}$. Loss on the validation set is shown. The figure is shown only for the first 200 000 pre-training steps using a medium-sized Maldi Transformer to illustrate. It is observed that if $\mathcal{L}_{\text{spec}}$ is applied at every step (using a weight of 0.01 or 1.00), this component starts overfitting well before the $\mathcal{L}_{\text{peaks}}$ component is converged (training is performed for 500 000 steps total). By not applying $\mathcal{L}_{\text{spec}}$ at every training step, the Adam optimizer momentum is dominated by $\mathcal{L}_{\text{peaks}}$. As $\mathcal{L}_{\text{spec}}$ is the "easier" task and more prone to overfitting, this regime benefits the final model.

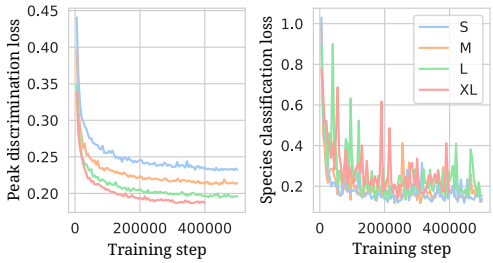

Figure 8 | Pre-training validation loss curves for all Maldi Transformer model sizes.

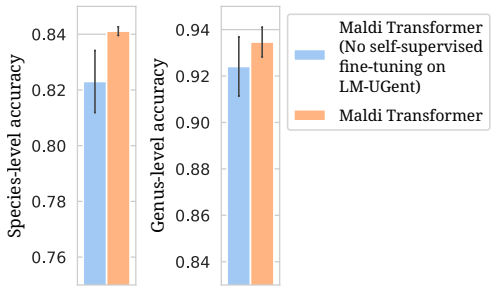

Figure 9 | LM-UGent performance improves when, prior to supervised fine-tuning, the pre-trained model is first fine-tuned using the self-supervised training task. In the main text, we refer to this step as the domain adaptation step.

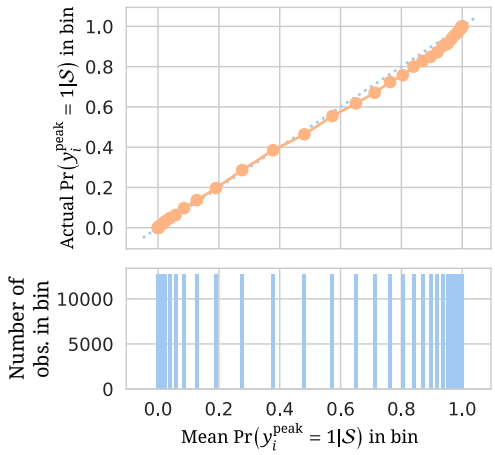

Figure 10 | Calibration curve for pre-training peak discrimination. On the DRIAMS test-set, the usual peak shuffling and peak discrimination is performed (see §2.2). Predictions are then sorted and split into equal frequency bins (on the x-axis). Within those bins, the average predicted value is plotted against the actual fraction of positive true labels in that bin. A calibrated model requires predictions to be interpretable in a frequentist manner, i.e. a sample with an output probability of 80% is expected to be positive 80% of the times. Hence, the aforementioned plot is expected to follow the diagonal. The plot shows that the model is reasonably-well calibrated, with slight overconfidence.

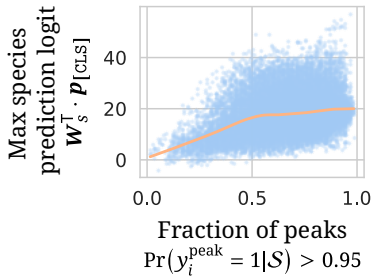

Figure 11 | Max species prediction logit $\boldsymbol{W}_s^{\mathsf{T}} \cdot \boldsymbol{p}_{[\text{CLS}]}$ in function of number of confidently "belonging" peaks for a spectrum for DRIAMS test set spectra. A higher max logit for the species prediction task corresponds to a higher max output probability post-softmax, and, hence, higher confidence. It is displayed that the model is more confident in its prediction for spectra with a higher number of confidently belonging peaks.