An antimicrobial drug recommender system using MALDI-TOF MS and dual-branch neural networks

Gaetan De Waele[1,*], **Willem Waegeman**[1] **and Gerben Menschaert**[1]
[1]Ghent University, [*]Correspondence to <gaetan.dewaele@ugent.be>.

Abstract

Timely and effective use of antimicrobial drugs can improve patient outcomes, as well as help safeguard against resistance development. Matrix-assisted laser desorption/ionization time-of-flight mass spectrometry (MALDI-TOF MS) is currently routinely used in clinical diagnostics for rapid species identification. Mining additional data from said spectra in the form of antimicrobial resistance (AMR) profiles is, therefore, highly promising. Such AMR profiles could serve as a drop-in solution for drastically improving treatment efficiency, effectiveness, and costs.
This study endeavours to develop the first machine learning models capable of predicting AMR profiles for the whole repertoire of species and drugs encountered in clinical microbiology. The resulting model can be interpreted as a drug recommender system for infectious diseases. We find that our dual-branch method delivers considerably higher performance compared to previous approaches. In addition, experiments show that the models can be efficiently fine-tuned to data from other clinical laboratories. MALDI-TOF-based AMR recommender systems can, hence, greatly extend the value of MALDI-TOF MS for clinical diagnostics.
All code supporting this study is distributed on PyPI and is packaged under: `https://github.com/gdewael/maldi-nn`

Keywords Recommender systems · MALDI-TOF MS · Neural networks · Antimicrobial resistance

1. Introduction

In diagnostic laboratories, matrix-assisted laser desorption/ionization time-of-flight mass spectrometry (MALDI-TOF MS) is routinely used for microbial species identification (Hou et al., 2019). Usually, microbial samples only require an overnight culturing step before being analyzed with mass spectrometry (Van Veen et al., 2010; Cuénod et al., 2021). Consequently, the technology provides a time- and cost-efficient way to accurately identify the pathogen underlying an infection.

Due to the rapid evolution of antibiotic resistant strains, it is increasingly difficult to determine a treatment based on only species identity. It has been estimated that infections caused by antibiotic-resistant bacteria have caused the deaths of 1.27 million people in 2019, making AMR one of the leading causes of death on earth (Murray et al., 2022). Projections have estimated that this annual number could rise to 10 million by 2050 (O'Neill, 2016), highlighting the need for responsible antimicrobial drug use. In light of this, diagnostic laboratories will often perform various tests, such as dilution arrays or disc diffusion tests, to probe which drug will be effective (Khan et al., 2019). Such experiments typically require further culturing and are either costly, labor-intensive, time-intensive, or a mixture of the above (Humphries, 2022).

Given that MALDI-TOF spectra are already routinely used for identification, it is worth investigating to which extent they can contain further information regarding the resistance status of strains (Weis et al., 2020a). Mining this information from the spectra could help inform healthcare workers of candidate drugs. This may nullify the need for phenotypical experiments, or (at least) direct the tests by narrowing down the choices. Furthermore, possessing a detailed resistance profile allows to treat with more specifically-working drugs (instead of broad-spectrum antibiotics) (Weis et al., 2022). Consequently, predicting resistance status from MALDI-TOF spectra could help towards the goals of antibiotic stewardship (Shlaes et al., 1997).

It has been described that some known resistance mechanisms are outside of the m/z range that MALDI-TOF spectrometers can accurately measure (Humphries, 2022). Still, it remains largely unknown to which extent co-evolved traits, such as subtle changes in metabolism caused by the resistance

mechanism, can be detected by MALDI-TOF spectra. A number of studies have shown that some resistant strains can reliably be predicted from MALDI-TOF MS, either by identifying and detecting specific markers (e.g. peaks) or by learning patterns from data (see §2.1). To our knowledge, all of these studies have modeled AMR prediction for specific species-drug combinations. For this reason, they learn very specific markers of resistance, not guaranteed to extrapolate well to other drugs and species. As susceptibility rapidly evolves, it is practically impossible to perform such studies for all clinically-relevant species-drug combinations. As such, the value of aforementioned studies remains of exploratory nature with limited practical value. In addition, their performance remains limited owing to small sample sizes and, likely, the inability of MALDI-TOF spectra to fully discriminate between the characteristics of interest (Bai et al., 2017). The recently published DRIAMS dataset (Weis et al., 2022) contains phenotypic AMR data covering a wide range of species and drugs, allowing to study MALDI-TOF-based AMR prediction on an unprecedented scale.

We posit that the most pertinent challenge healthcare workers face regarding AMR is to choose between all possible drugs given an infection, not whether one specific drug will be effective or not. For this reason, we argue that our models and evaluation metrics should be designed to optimally answer that question. In this study, a single model is proposed that can predict AMR for the whole range of pathogens and drugs encountered in clinical microbiology. The method jointly learns representations for antibiotic drugs and bacterial MALDI-TOF spectra. It can be used to recommend the most-likely drug to work for any drug-spectrum combination, irregardless of the species identity of the underlying pathogen. Consequently, the model is broadly-applicable and practical to use. Our contributions are as follows:

1. We formulate a dual-branch neural network recommender system for the prediction of AMR profiles. The model operates on MALDI-TOF spectra, as well as a representation of the candidate drug.
2. We evaluate multiple state-of-the-art techniques for representing drug identity in the model.
3. We perform evaluations by comparing our method to non-recommender system baselines.
4. We show that the model efficiently transfers to data from diagnostic laboratories it wasn't trained on. Making the model easy to adopt for hospitals lacking the means and/or volume to collect large data.

2. Related Work

2.1. MALDI-TOF-based machine learning

The most canonical task for MALDI-TOF-based methods is species identification. Identification solutions are usually provided by the MS manufacturers and are built on large, proprietary, in-house databases (Van Belkum et al., 2012). It is unclear how these closed-source identification pipelines work, but it is likely that query spectra are directly compared to the in-house database in an approach akin to nearest neighbors (Dauwalder et al., 2023). While this approach works excellently for identification of most species, some strains remain problematic (Cao et al., 2018; Vrioni et al., 2018). Furthermore, by presumably focusing on the presence or absence of specific peaks, a lot of spectral information stands unused (Florio et al., 2018).

For various difficult prediction cases, such as strain typing, researchers often resort to machine learning (Hettick et al., 2006; Wang et al., 2018; De Bruyne et al., 2011). Stifled by a historical lack of large open data, machine learning research on MALDI-TOF data remains in its infancy. Most studies have narrow scopes and simple datasets (e.g. binary classification), only warranting standard preprocessing and off-the-shelf learning techniques (Yu et al., 2022; Zhang et al., 2023; Chung et al., 2023). Only a handful of examples exist of more advanced learning techniques specifically adapted to a MALDI-TOF-based task (Mortier et al., 2021; Weis et al., 2020a; Vervier et al., 2015). For a more thorough overview of MALDI-TOF-based machine learning, readers are referred to the review of Weis et al. (2020b).

2.2. Dual-branch neural networks

The idea of processing and combining two separate streams of information with two neural networks is applied in many fields of machine learning, collectively referred to as deep multi-target prediction (Waegeman et al., 2019; Iliadis et al., 2022).

In collaborative filtering, the goal is to predict the preference of a user to items (He et al., 2017). In its most elementary neural form, both users and items are represented by one-hot encodings, generating a model unable to make salient predictions for new users or items without having seen them during training. To solve this, a body of works exists on trying to communicate user- and item-identity to the model via side-information encoded in features (Zheng et al., 2017).

Dual-branch neural networks are also prevalent in language and vision. Recent advances in (multimodal) contrastive learning of image (and text) representations often rely on two neural encoders to learn a matching score between two views of the same or discordant objects (Radford et al., 2021; Chen et al., 2020). Language retrieval systems typically compare input vectors with a database of key vectors, each derived from a neural network, using approximate nearest neighbor search techniques (Karpukhin et al., 2020). In biology, fields of research employing dual-branch neural networks include (1) drug-target interaction (Lee et al., 2019), (2) single-cell multi-omics analysis (Lance et al., 2022), and (3) transcription factor binding prediction (Yang et al., 2020), among countless others.

Most of these applications can, to varying extents, be interpreted as (collaborative filtering) recommender systems. For example, contrastive language-image models have been used to retrieve the most-semantically similar images to a piece of text (Beaumont, 2022).

3. Methods

3.1. Data

To train models, we use the recently published DRIAMS database, consisting of 765 048 AMR measurements derived from 55 773 spectra across four different hospitals, spanning in total 74 different drugs[1] (Weis et al., 2022). Every drug is characterized by a canonical SMILES string obtained from PubChem (Kim et al., 2023). As in the original DRIAMS publication, AMR measurements are binarized according to the EUCAST norms per drug. Specifically, intermediate or resistant values are assigned a positive label, and susceptible samples a negative one. Furthermore, spectra are identically processed as in the original publication. Briefly, the following steps are performed: (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, and (6) binning the intensities by summing all values in intervals of 3 Da. After preprocessing, every spectrum is represented as a 6000-dimensional vector.

The main experiments concern models that are trained on data from one hospital only (DRIAMS-A, University Hospital Basel). All spectra and measurements derived from the other three hospitals in DRIAMS are left out for transfer learning experiments (see §4.3). Within DRIAMS-A, all spectra from before 2018 are allocated to the training set, and all spectra measured during 2018 are evenly split between validation and test set. This split in time reflects a realistic evaluation scenario, as models trained on historical data need to generalize to new patients possibly infected by newly-evolved strains. The final sizes of all splits are as follows: 409 395 labels across 28 331 spectra for the training set, 76 431 labels across 4 994 spectra for the validation set, and 76 133 labels across 4 999 spectra for the test set.

3.2. Model architecture

We formulate AMR prediction as a multi-target classification problem with side-information for both instances and targets, also referred to as dyadic prediction (Waegeman et al., 2019). In this context, let us denote a sample in the dataset $\mathcal{D}$ by a triplet $(\boldsymbol{s}_i, \boldsymbol{d}_j, y_{ij})$, where y_{ij} denotes the resistance label of a microbial spectrum $\boldsymbol{s}_{i \in \{1,\ldots,n\}}$ w.r.t. a drug $\boldsymbol{d}_{j \in \{1,\ldots,m\}}$. This dataset can be arranged in an incomplete score matrix $\boldsymbol{Y} \in \{0,1\}^{n \times m}$. In what follows, the final architectural set-ups used to present the results are described. For details on hyperparameter tuning, readers are referred to Appendix B.

The model consists of two separate neural network embedders $E_s(\cdot)$ and $E_d(\cdot)$ for processing the spectra and drugs, respectively. The resulting instance and target embeddings $\boldsymbol{x}_i$ and $\boldsymbol{t}_j$ are then combined into a single score by their scaled dot product $\hat{y} = \frac{\boldsymbol{x}_i \cdot \boldsymbol{t}_j}{\sqrt{h}}$ (Rendle et al., 2020). The scaling factor $\sqrt{h}$, with h the dimensionality of both embeddings, is inspired by the formulation of self-attention (Vaswani et al., 2017). It ensures the dot products to be of manageable magnitudes, even for large values of h. This score can be used together with the sigmoid function and the cross-entropy loss to optimize the two-branch neural network to map a spectrum-drug pair to a resistance label (Iliadis et al., 2022). An overview of the model is visualized in Figure 1.

The representations of the instance vectors $\boldsymbol{x}_i$ are

[1] These figures reflect the size of the dataset as downloaded from the original Dryad repository `https://doi.org/10.5061/dryad.bzkh1899q`, and after processing. For example, the number of spectra listed here corresponds to all spectra in DRIAMS for which there exists at least one AMR measurement. The total number of spectra in DRIAMS counts 250 070, but no labels are associated with these extra spectra. Further, the naming of drugs was further preprocessed such that every drug can be linked to a single chemical identifier. For more information on which drugs were merged and how this was performed, see Appendix A

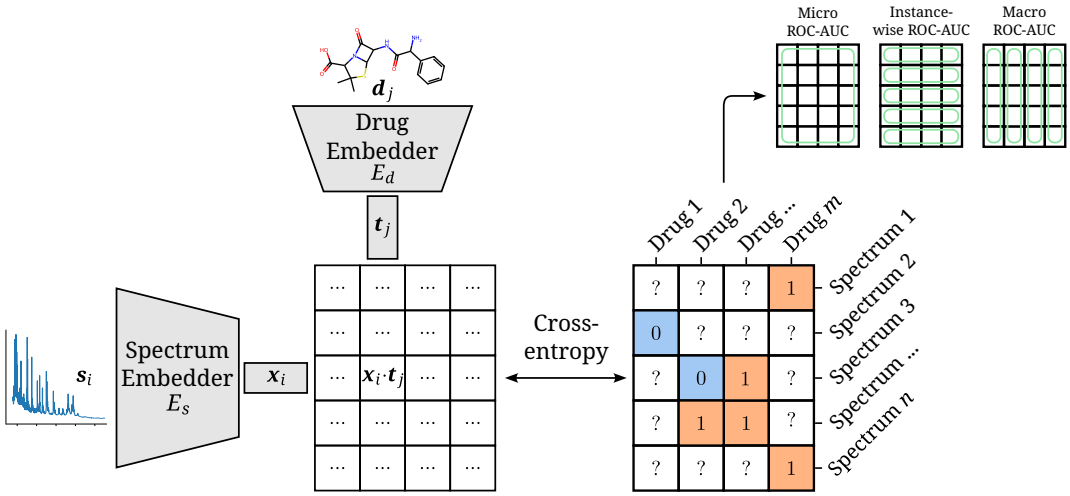

Figure 1 | Architectural overview of the proposed model. AMR labels of spectrum-drug pairs can be represented in an incomplete matrix. A microbial sample that is susceptible to a drug is denoted by a negative label (orange), whereas positive labels (blue) signify an intermediate or resistant combination. Instance (spectrum) and target (drug) embeddings $\boldsymbol{x}_i$ and $\boldsymbol{t}_j$ are obtained from their respective input representations passed through their respective neural network branch. The two resulting embeddings are aggregated to a single score by their (scaled) dot product. The cross-entropy loss optimizes this score to be maximal or minimal for positive or negative combinations of microbial spectra and drugs, respectively. On the upper right-hand side, different metrics are visualized. Whereas micro ROC-AUC takes all prediction-label pairs together, the instance-wise and macro ROC-AUC compute their score per spectrum or drug, respectively, and then average.

Table 1 | All tested model sizes for the (instance) spectrum branch. Hidden sizes represent the evolution of the hidden state dimensionality as it goes through the model, with every hyphen defining one fully connected layer. The listed number of parameters only include those of the instance (spectrum) branch.

Size	# Weights	Hidden sizes
S	1 578 176	6000-256-128-64
M	3 246 784	6000-512-256-128-64
L	6 846 144	6000-1024-512-256-128-64
XL	15 093 440	6000-2048-1024-512-256-128-64

extracted from a neural network $E_s(\cdot)$ operating on the processed and binned MALDI-TOF spectra $\boldsymbol{s}_i$. $E_s(\cdot)$ is parameterized by a multi-layer perceptron (MLP), consisting of a series of fully-connected layers. Between every two such layers, a series of operations consisting of (1) a GeLU activation (Hendrycks and Gimpel, 2016), (2) a dropout rate of 0.2 (Srivastava et al., 2014), and (3) layer normalization (Ba et al., 2016), is applied. We include multiple model sizes in our final results (Table 1). To make comparisons easier, all models output the same number of hidden dimensions that are used in the dot product, $\boldsymbol{x}_i \in \mathbb{R}^{64}$.

Drug identity can be communicated to the model in a number of ways. In this work, we study the following different input representations $\boldsymbol{d}_j$ and embedder $E_d(\cdot)$ combinations:

1. As indices in a one-hot encoding paired with a single linear layer.
2. As Extended Connectivity Fingerprints paired with a single linear layer.
3. As DeepSMILES strings (O'Boyle and Dalke, 2018) paired with a 1D convolutional neural network (CNN).
4. As DeepSMILES strings paired with a gated recurrent unit neural network (GRU).
5. As DeepSMILES strings paired with a transformer neural network.
6. As images paired with a 2D CNN.
7. As rows of a pre-computed string kernel on the SMILES strings (LINGO (Vidal et al., 2005)), paired with a single linear layer.

For all these combinations, the embedder outputs target embeddings $\boldsymbol{t}_j \in \mathbb{R}^{64}$. For more details on the different drug embedders and their hyperparameters (as well as their tuning), see Appendix B. For every combination of spectrum embedder (four sizes: S, M, L, and XL) and drug embedder (seven types), four different learning rates ({1e-4, 5e-4, 1e-3, 5e-3})

are tested. 1e−10 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.

All models are trained with the Adam optimizer (Kingma and Ba, 2014) for a maximum of 50 epochs with a batch size of 128. A linear learning rate warm-up over the first 250 steps is applied, after which the rate is kept constant. As every epoch constitutes one pass over every label and, hence, multiple passes over every individual drug and spectrum, a branch can technically already be overfitting before the end of the first epoch. Because of this, performance on the validation set is checked every tenth of an epoch. 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.

4. Results

The following section will first relay the results of the different dual-branch model configurations. After, a comprehensive comparison is made with baselines consisting of specialist models trained on specific species-drug combinations, as is now common practice in existing works. Finally, the models' capabilities and representations are examined through transfer learning and embeddings.

4.1. Encoding species and drugs effectively

Figure 2 shows the performance of all trained models in terms of their areas under the receiver operating characteristic curve (ROC-AUC). The ROC-AUC measures the probability that any positive (resistant or intermediate) sample is assigned a higher predicted probability of being positive as compared to any negative (susceptible) sample. The micro ROC-AUC computes this probability for any pair of positive and negative samples, whereas the instance-wise ROC-AUC only compares pairs within the same spectrum, respectively[2] (Waegeman and Dembczynski, 2018). In a realistic scenario, an AMR recommender system, such as this one, would be used to evaluate which drugs work for new patients. Instance-wise metrics reflect on the average performance across drugs for a single patient. The instance-wise ROC-AUC, for instance, measures the average quality of the ranking of suggested drugs to a patient.

From Figure 2 it can be seen that, in general, performance differences between model configurations occupy a small margin. However, trends can still be found. One-hot drug embedders typically outperform those using molecular structure information. We offer two hypotheses for this observation: (1) the number of drugs in the training dataset is too small to learn salient representations based on molecular information, and/or (2) wildly different resistance profiles may be obtained from very similar molecular structures. In the latter case, including molecular structure in the model may actually hinder the model, rather than helping it. Among the other drug embedders, the best performance is found in models using Morgan fingerprints. On the spectrum embedder side, it is observed that the small or medium-sized variants typically perform best.

Performance in terms of Macro ROC-AUC and Instance-wise Prec@1 of the negative class can be found in Appendix C Figure 8. The Macro ROC-AUC averages the ROC-AUC for every individual drug. The Prec@1 evaluates how often the top-ranked prediction is correct. The Prec@1 of the negative class, hence, reports the proportion of cases for which the "most-likely susceptible drug" prediction is actually an effective one. In a scenario where the top recommended drug is always administered, it corresponds to the percentage of correctly-suggested treatments. It can be observed that, in terms of Macro ROC-AUC, Morgan fingerprints (and sometimes other drug embedders) outperform one-hot drug embedders. Hence, molecular structures are a useful inductive bias to learn salient representations for drugs not encountered much in the data. For the Instance-wise Prec@1, a similar observation can be made. The full list of performances can be found in Appendix C Table 4.

In Appendix C Figure 9, the performance of the spectrum embedder sizes is compared against a linear baseline. The linear baseline uses the same preprocessed input spectrum representation, but only uses a single linear combination to produce an embedding. For this comparison, only the one-hot and Morgan fingerprint drug embedders are used, as they produce the best-performing models overall. Models

[2] An instance-wise metric can be seen as a "macro"-averaged metric. But, whereas macro-averaged metrics usually compute an average performance per class (here, drug), with instance-wise metrics, the performance is averaged per instance (here, a spectrum).

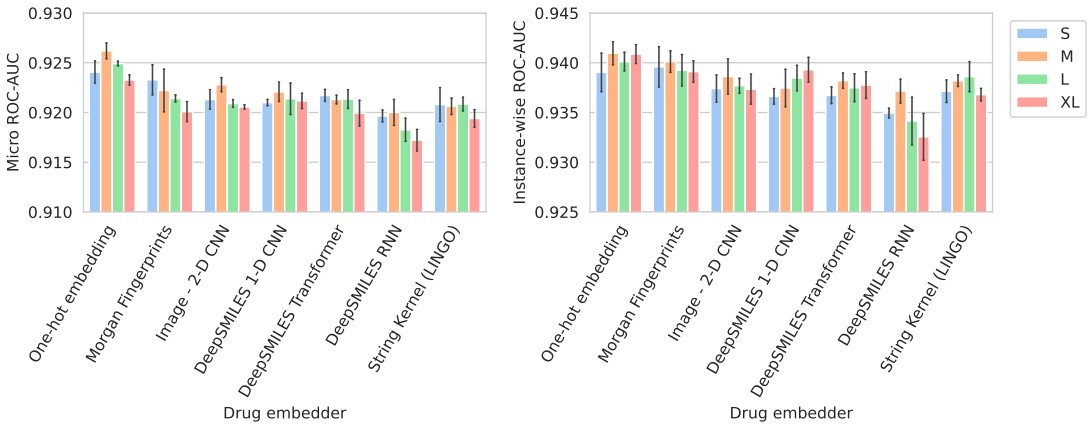

Figure 2 | Barplots showing test performance results for all trained models. Errorbars represent the standard deviation over three random seeds. The x-axis and colors show the different drug and spectrum embedders, respectively.

using non-linear multi-layer spectrum embedders obtain considerably better performance over linear embedders.

4.2. Dual-branch modeling improves AMR prediction

Previous studies have studied AMR prediction in specific species-drug combinations (Weis et al., 2020b). To the best of our knowledge, this study represents the first attempt at modeling MALDI-TOF-based AMR prediction across a wide array of species and drugs using a single model. For this reason, it is useful to compare how the dual-branch setup weighs up against training separate models for separate species and drugs. To test this, binary logistic regression, XGBoost (Chen and Guestrin, 2016) and MLPs are trained for the 200 combinations of species and drugs for which most training labels are present. The tested MLPs come in the same four sizes as the spectrum branches of the dual-branch models. Other than having an output node of size 1 for binary classification, they share all hyperparameters with the tested spectrum branches.

There exist many species-drug combinations for which there are either only positive or only negative labels. As it is impossible to train and evaluate models for these cases, models are trained only for the 200 most-occurring combinations for which both labels are present in the training, validation and test set. We refer to these models as "specialists", as they are optimized to recognize patterns for specific cases of AMR. For details on the training and tuning procedure of all baselines, see Appendix B.2.2.

The performances of specialist models are compared to two dual-branch recommenders (Table 2). Two representative dual-branch models are chosen based on their validation micro ROC-AUC. One model is chosen among those using one-hot drug embedders, and another is chosen among those not using a one-hot drug embedder. Performance is computed on the subset of labels spanning the 200 most-common species-drug combinations. This subset spans 4017 spectra, 35 drugs, and 53503 labels (covering 70.28% of the original test set). Dual-branch recommenders outperform specialist baselines on all but one metric. Logistic regression baselines result in the best average ROC-AUC for individual species-drug combinations. By all other metrics, dual-branch recommenders outshine a collection of specialist models. These results are perhaps not surprising, as one could argue that either type of models is designed to be optimal for the metric they outperform the other in. It's illustrated that, when the question is to choose between drugs for a patient (i.e. instance-wise metrics), a model designed as a recommender will outperform binary classification models trained to predict AMR for specific drugs. On the other hand, when the pathogen is already known, and only one drug is in consideration, a binary classification model will be optimal.

Multi-task learning can explain why dual-branch models outperform a collection of specialists (Waegeman et al., 2019; Caruana, 1997). As species will often share some common biological processes causing AMR, representations learned on one species may strengthen those for another. In doing so, the combination of data from multiple species in one model

Table 2 | Test performance of selected dual-branch models, compared to the performance of a collection of models — each trained on only one species-drug combination — coined "specialists". Performance is computed on the subset of labels spanning the 200 most-common species-drug combinations.

Model	Micro ROC-AUC	Instance-wise ROC-AUC	Macro ROC-AUC	Instance-wise Prec@1 (neg class)	Species-drug Macro ROC-AUC
Dual-branch Recommender (One-hot - M)	**0.8983**	**0.9018**	**0.8180**	**0.9676**	0.6758
Dual-branch Recommender (Morgan Fingerprints - M)	0.8958	0.9016	0.8095	0.9693	0.6679
Specialists (Logistic Regression)	0.8819	0.8811	0.7953	0.9423	**0.7085**
Specialists (XGBoost)	0.8640	0.8474	0.7577	0.9331	0.6430
Specialists (MLP - S)	0.8719	0.8562	0.8020	0.9545	0.7013
Specialists (MLP - M)	0.8678	0.8488	0.8077	0.9545	0.6789
Specialists (MLP - L)	0.8803	0.8664	0.8054	0.9562	0.6699
Specialists (MLP - XL)	0.8830	0.8819	0.8160	0.9621	0.6800

helps to produce more expressive spectrum representations overall. Similarly, many antimicrobial drugs share common structure and mechanisms. As such, combining the data from different drugs into one dataset and model may similarly lead to performance gains.

4.3. Efficient transfer learning to new hospitals

An AMR prediction model trained using data from one hospital may not be suitable for use in other hospitals for several reasons. First, protocols such as sample preparation and culturing media differ from hospital to hospital, resulting in systematic differences in MALDI-TOF spectra (Weis et al., 2022). Second, epidemiology is spatially varied. Drug-resistant clades may be prevalent in one region or country, but absent in another (Humphries, 2022). Finally, the MALDI-TOF instruments themselves may also be specific to the hospital and influence the readout. This influences prediction models, as a hospital-specific effect is reported by the study introducing the DRIAMS dataset (Weis et al., 2022). They find that models typically perform best when trained with data from the same hospital. Here, hospital transferability is studied in the context of transfer learning.

Data from DRIAMS-B, -C and -D, are split into training, validation and test set. The train set for these hospitals consists of 1000 randomly-drawn spectra, simulating a small data scenario where the hospital has not spent considerable efforts in data collection. The remaining spectra for all three hospitals are evenly split among validation and test set.

For all three hospitals, we train models in the same way as previously (see §3.2). A comparison is made between fine-tuning starting from models trained on DRIAMS-A (i.e. models from previous sections) and dual-branch models trained from scratch (Figure 3). Over all three hospitals, models fine-tuned from a DRIAMS-A checkpoint generally outperform models trained from scratch. This trend holds true over different numbers of spectra available in the fine-tuning set. In general, it can be seen that pre-trained models require very little fine-tuning spectra to obtain performances in the same order of magnitude as with DRIAMS-A (§4.1). Curiously, whereas the one-hot drug embedders typically outperform Morgan fingerprints-based models on DRIAMS-A, here, we see an opposite trend. This effect may be explained by the fact that not all hospitals use the same set of drugs. Whereas a Morgan fingerprint-based drug embedder may (to an extent) instantly generalize to new drugs, a one-hot drug embedder needs to learn a representation from scratch for every drug it hasn't seen before. Performance comparisons of the same models in terms of other metrics are shown in Appendix C Figure 10

Lowering the amount of data required is paramount to expedite the uptake of AMR models in clinical diagnostics. The transfer learning qualities of dual-branch models may be ascribed

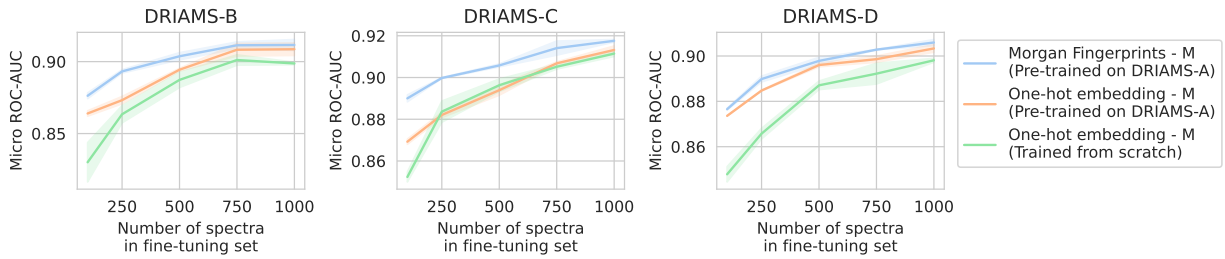

Figure 3 | Transfer learning of DRIAMS-A models to other hospitals. Errorbands show the standard deviation over three runs. Results in terms of other evaluation metrics are shown in Appendix C Figure 10.

to multiple properties. First of all, since different hospitals use much of the same drugs, transferred drug embedders allow for expressively representing drugs out of the box. Secondly, owing to multi-task learning, even with a limited number of spectra, a considerable fine-tuning dataset may be obtained, as all available data is "thrown on one pile".

4.4. MALDI-TOF spectra embeddings

To investigate what the dual-branch models have learned to represent, MALDI-TOF spectra embeddings are examined. For this purpose, the model with the best validation micro ROC-AUC is used, which is a model using a medium-size spectrum embedder and one-hot drug embedder. Here, we visualize the embeddings $\boldsymbol{x}_i \in \mathbb{R}^{64}$ of all test set spectra from the 25 most-occurring pathogens. To visualize in a 2-dimensional space, UMAP is applied (using default parameters apart from `min_dist=0.75`[3]). Figure 4 shows the resulting embeddings, colored by species identity, as well as by their AMR status to a selection of drugs.

The MALDI-TOF embeddings are grouped primarily per species. This shows that, without being instructed to discriminate between species, the model has learned to group spectra of the same species together. Furthermore, species under the same genus are typically grouped close together, illustrating that the model can pick up hierarchical relations in the tree of life from the data. Within species clusters, the AMR status subplots show that samples are often grouped according to their resistance. For example, for *S. epidermidis* and *S. aureus*, multidrug resistant variants clearly form subclusters. In addition, the cluster of *E. coli* spectra shows a clear tail with samples resistant to ciprofloxacin. UMAP embedding plots for other drugs are shown in Appendix C Figure 11.

[3]Increasing this parameter helps reduce UMAP packing points too tightly together, hence, making for a more-legible plot.

5. Discussion

Prior work on AMR prediction has always modeled within the boundaries of one clade and drug(class), using standard machine learning practices. This work differentiates itself from others by constructing one model for the whole range of pathogens and drugs encountered in clinical diagnostics. We propose to model AMR prediction via dual-branch neural networks, producing a novel MALDI-TOF-based AMR recommender system. The proposed models come with improved performance over the approaches taken in previous works. Only when evaluated on a per-pathogen-and-drug basis, it makes more sense to have separate models. It is worth noting, though, that the dual-branch models can also be fine-tuned to similar "specialist" models on a certain of subset of data (spanning, for example, one species and drug). Aside from this point, we argue that, in any case, instance-wise metrics are more appropriate for AMR prediction. They judge the quality of predictions on a patient-per-patient basis, as it would happen in usage of such models in clinical diagnostics.

We postulate that the performance of the proposed models is still limited due to (1) lacking a MALDI-TOF-specific learning architecture, (2) collection of more data, especially on rarely-encountered species and drugs, and (3) inherent technological limitations of MALDI-TOF MS. Whilst the former is the subject of further machine learning research, the latter two can be considered by equipping the model with some notion of uncertainty, epistemic and aleatoric, respectively (Hüllermeier and Waegeman, 2021). In medical decision-making applications, effective uncertainty estimates would be an invaluable tool to aid understanding the models' predictions. A fourth factor to

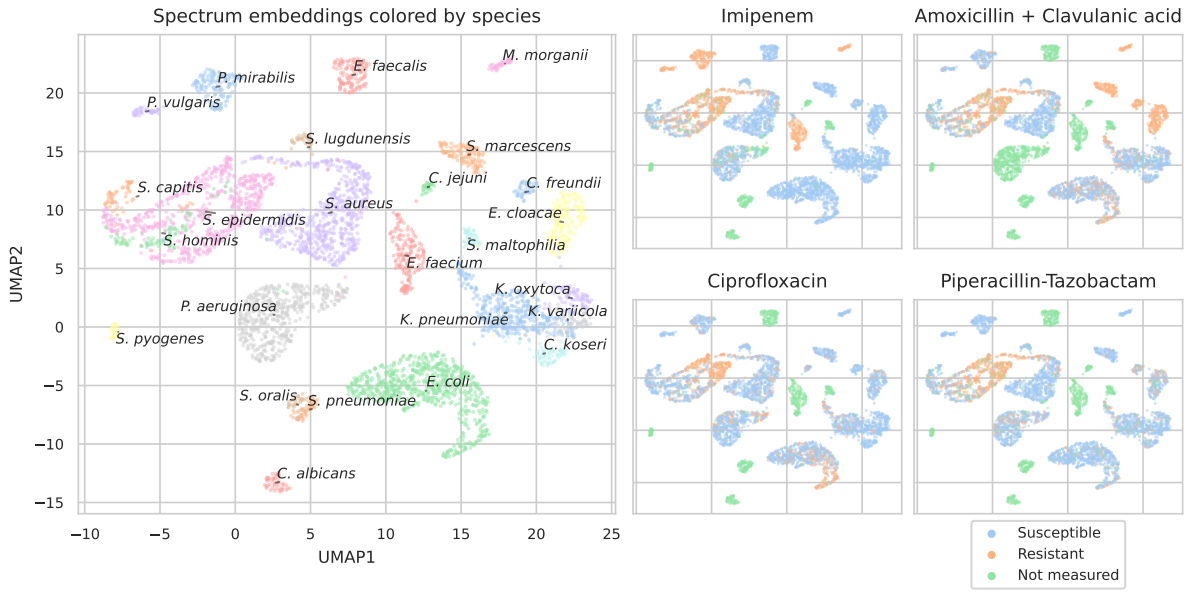

Figure 4 | UMAP scatterplots of test set MALDI-TOF spectra embeddings $\boldsymbol{x}_i$. Only embeddings belonging to the 25 most-occurring species in the test set are shown. The panels on the right show the same embeddings as on the left, but colored according to its AMR status to a certain drug. The four displayed drugs are selected based on a ranking of the product of the number of positive and negative labels $\sum_{i=1}^{n}[y_{ij} = 0] \cdot \sum_{i=1}^{n}[y_{ij} = 1]$. In this way, the drugs that have a lot of observed labels, both positives and negatives, are displayed.

consider is that perfect test set performance may also be unattainable due to labeling errors. This comes as a consequence of (1) error-prone laboratory measurements of MIC values, and (2) the fact that EUCAST norms change over time, resulting in outdated label thresholds for historical data.

As bacterial strains readily adapt resistance to new and frequently-used antibiotics, it is impossible for a single machine learning model to maintain its performance over time. Due to this obvious need for continual data collection and online machine learning, it is clear that ML for AMR prediction will prove most valuable when integrated tightly in the inner workings of healthcare (Lee and Lee, 2020).

It stands to reason that blindly following the recommender system's predictions spells misery. For example, healthcare practitioners should additionally take into account host-specific factors such as patient age, medical history, and concurrent medication. Additionally, as the model is trained on the whole repertoire of antimicrobial drugs, it will have learnt that broad-spectrum antibiotics are typically effective. Hence, it may overrecommend their use. As a consequence, the model's proposed treatment strategies may not be aligned with antibiotic stewardship, instead exacerbating the very issue it is designed to mitigate. To tackle this problem, one could downweigh the prediction probabilities of undesirable drugs, or, alternatively, train a dual-branch model on only more-specifically-working drugs.

In summary, this study serves as the first proof-of-concept for large MALDI-TOF-based antimicrobial drug recommenders. In this context, we highlight the need for appropriate metrics, proposing that instance-wise metrics are most suitable. Extensive experiments on our proposed dual-branch model allow us to assemble some conclusions w.r.t. its use. Firstly, given the present data at hand, medium-sized MLP spectrum embedders (counting 3.2M weights) generally perform best. Second, embedding drugs without structural information (by one-hot encoding) works best for the larger DRIAMS-A dataset. For the smaller datasets used in the transfer learning experiments, the structural inductive bias lent to the model via Morgan fingerprints delivers best results. Our experiments demonstrate that dual-branch recommenders outperform non-recommender baselines on relevant metrics. In the above discussion, some considerations are listed w.r.t. its practical implementation in healthcare. Taken together, this work demonstrates the potential of AMR recommenders to greatly extend the value of MALDI-TOF MS for clinical diagnostics.

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

Tsung-Yun Hou, Chuan Chiang-Ni, and Shih-Hua Teng. Current status of maldi-tof mass spectrometry in clinical microbiology. *Journal of food and drug analysis*, 27(2):404–414, 2019.

SQ Van Veen, ECJ Claas, and Ed J Kuijper. High-throughput identification of bacteria and yeast by matrix-assisted laser desorption ionization-time of flight mass spectrometry in conventional medical microbiology laboratories. *Journal of clinical microbiology*, 48(3):900–907, 2010.

Aline Cuénod, Frédéric Foucault, Valentin Pflüger, and Adrian Egli. Factors associated with maldi-tof mass spectral quality of species identification in clinical routine diagnostics. *Frontiers in Cellular and Infection Microbiology*, 11:646648, 2021.

Christopher JL Murray, Kevin Shunji Ikuta, Fablina Sharara, Lucien Swetschinski, Gisela Robles Aguilar, Authia Gray, Chieh Han, Catherine Bisignano, Puja Rao, Eve Wool, et al. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *The Lancet*, 399(10325):629–655, 2022.

Jim O'Neill. Tackling drug-resistant infections globally: final report and recommendations. 2016.

Zeeshan A Khan, Mohd F Siddiqui, and Seungkyung Park. Current and emerging methods of antibiotic susceptibility testing. *Diagnostics*, 9(2):49, 2019.

Romney M Humphries. Ad hoc antimicrobial susceptibility testing from maldi-tof ms spectra in the clinical microbiology laboratory. *Clinical Chemistry*, 68 (9):1118–1120, 2022.

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, 2020a.

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.

David M Shlaes, Dale N Gerding, Joseph F John, William A Craig, Donald L Bornstein, Robert A Duncan, Mark R Eckman, William E Farrer, William H Greene, Victor Lorian, et al. Society for healthcare epidemiology of america and infectious diseases society of america joint committee on the prevention of antimicrobial resistance guidelines for the prevention of antimicrobial resistance in hospitals. *Infection Control & Hospital Epidemiology*, 18(4):275–291, 1997.

J Bai, ZC Fan, LP Zhang, XY Xu, and ZL Zhang. Classification of methicillin-resistant and methicillin-susceptible staphylococcus aureus using an improved genetic algorithm for feature selection based on mass spectra. In *Proceedings of the 9th International Conference on Bioinformatics and Biomedical Technology*, pages 57–63, 2017.

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.

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.

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.

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.

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.

Hsin-Yao Wang, Tzong-Yi Lee, Yi-Ju Tseng, Tsui-Ping Liu, Kai-Yao Huang, Yung-Ta Chang, Chun-Hsien Chen, and Jang-Jih Lu. A new scheme for strain typing of methicillin-resistant staphylococcus aureus on the basis of matrix-assisted laser desorption ionization time-of-flight mass spectrometry by using machine learning approach. *PloS one*, 13(3): e0194289, 2018.

Katrien De Bruyne, Bram Slabbinck, Willem Waegeman, Paul Vauterin, Bernard De Baets, and Peter Vandamme. Bacterial species identification from maldi-tof mass spectra through data analysis and machine learning. *Systematic and applied microbiology*, 34(1):20–29, 2011.

Jiaxin Yu, Ni Tien, Yu-Ching Liu, Der-Yang Cho, Jia-Wen Chen, Yin-Tai Tsai, Yu-Chen Huang, Huei-Jen Chao, and Chao-Jung Chen. Rapid identification of methicillin-resistant staphylococcus aureus using maldi-tof ms and machine learning from over 20,000 clinical isolates. *Microbiology Spectrum*, 10 (2):e00483–22, 2022.

Yu-Ming Zhang, Mei-Fen Tsao, Ching-Yu Chang, Kuan-Ting Lin, Joseph Jordan Keller, and Hsiu-Chen Lin. Rapid identification of carbapenem-resistant klebsiella pneumoniae based on matrix-assisted laser desorption ionization time-of-flight mass spectrometry and an artificial neural network model. *Journal of Biomedical Science*, 30(1):25, 2023.

Chia-Ru Chung, Hsin-Yao Wang, Chun-Han Yao, Li-Ching Wu, Jang-Jih Lu, Jorng-Tzong Horng, and Tzong-Yi Lee. Data-driven two-stage framework for identification and characterization of different antibiotic-resistant escherichia coli isolates based on mass spectrometry data. *Microbiology Spectrum*, pages e03479–22, 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.

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.

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, 2020b.

Willem Waegeman, Krzysztof Dembczyński, and Eyke Hüllermeier. Multi-target prediction: a unifying view on problems and methods. *Data Mining and Knowledge Discovery*, 33:293–324, 2019.

Dimitrios Iliadis, Bernard De Baets, and Willem Waegeman. Multi-target prediction for dummies using two-branch neural networks. *Machine Learning*, pages 1–34, 2022.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 425–434, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Ingoo Lee, Jongsoo Keum, and Hojung Nam. Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS computational biology*, 15 (6):e1007129, 2019.

Christopher Lance, Malte D Luecken, Daniel B Burkhardt, Robrecht Cannoodt, Pia Rautenstrauch, Anna Christine Laddach, Aidyn Ubingazhibov, Zhi-Jie Cao, Kaiwen Deng, Sumeer Khan, et al. Multimodal single cell data integration challenge: results and lessons learned. *bioRxiv*, pages 2022–04, 2022.

Shu Yang, Xiaoxi Liu, and Raymond T Ng. Proberating: a recommender system to infer binding profiles for nucleic acid-binding proteins. *Bioinformatics*, 36(18):4797–4804, 2020.

Romain Beaumont. Clip retrieval: Easily compute clip embeddings and build a clip retrieval system with them. https://github.com/rom1504/clip-retrieval, 2022.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.

Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 240–248, 2020.

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.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014.

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

Noel O'Boyle and Andrew Dalke. Deepsmiles: an adaptation of smiles for use in machine-learning of chemical structures. 2018.

David Vidal, Michael Thormann, and Miquel Pons. Lingo, an efficient holographic text based method to calculate biophysical properties and intermolecular similarities. *Journal of chemical information and modeling*, 45(2):386–393, 2005.

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

Willem Waegeman and Krzysztof Dembczynski. Multi-target prediction: a unifying view on problems and methods. Tutorial presented at ECML/PKDD 2018, 2018.

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

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

Cecilia S Lee and Aaron Y Lee. Clinical applications of continual learning machine learning. *The Lancet Digital Health*, 2(6):e279–e281, 2020.

Alice Capecchi, Daniel Probst, and Jean-Louis Reymond. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. *Journal of cheminformatics*, 12(1):1–15, 2020.

Greg Landrum. Rdkit documentation. *Release*, 1(1-79): 4, 2013.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

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

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. A comparative study of smiles-based compound similarity functions for drug-target interaction prediction. *BMC bioinformatics*, 17(1):1–11, 2016.

Table 3 | Full list of modifications made to drug names in DRIAMS. Modifications consist of (1) removal of drugs, (2) merging of drugs, and (3) renaming drugs.

Original drug name	Step undertaken
Quinolones	Removed
Aminoglycosides	Removed
Ofloxacin	Merged with Levofloxacin
Benzylpenicillin	Merged with Penicillin
Benzylpenicillin_others	Merged with Penicillin
Benzylpenicillin_with_meningitis	Merged with Penicillin
Benzylpenicillin_with_pneumonia	Merged with Penicillin
Penicillin_with_endokarditis	Merged with Penicillin
Penicillin_without_endokarditis	Merged with Penicillin
Penicillin_without_meningitis	Merged with Penicillin
Penicillin_with_meningitis	Merged with Penicillin
Penicillin_with_pneumonia	Merged with Penicillin
Penicillin_with_other_infections	Merged with Penicillin
Cefuroxime.1	Merged with Cefuroxime
Cotrimoxazol	Merged with Cotrimoxazole
Gentamicin_high_level	Merged with Gentamicin
Cefoxitin_screen	Merged with Cefoxitin
Teicoplanin_GRD	Merged with Teicoplanin
Vancomycin_GRD	Merged with Vancomycin
Rifampicin_1mg-l	Merged with Rifampicin
Meropenem_with_meningitis	Merged with Meropenem
Meropenem_without_meningitis	Merged with Meropenem
Meropenem_with_pneumonia	Merged with Meropenem
Amoxicillin-Clavulanic acid_uncomplicated_HWI	Merged with Amoxicillin-Clavulanic acid
Strepomycin_high_level	Renamed to Streptomycin
Bacitracin	Renamed to Bacitracin A
Ceftarolin	Renamed to Ceftaroline fosamil
Fosfomycin-Trometamol	Renamed to Fosfomycin Tromethamine

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 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.

A. DRIAMS processing

As our models require every target to correspond to one specific drug (for which a SMILES string can be obtained), data provided by Weis et al. (2022) is further cleaned up. First, as "Quinolones", and "Aminoglycosides" constitute classes of drugs rather than single ones, these drugs and their corresponding measurements are removed from the dataset. Second, some drug names in DRIAMS that refer to the same chemical structure are merged to a single drug. As this merging of drugs also combines their labels, care is taken so that no conflicting labels are combined. If, for a single spectrum, labels exist for both of the merging drugs in question, the label is only kept if both measurements are congruent (either both resistant, intermediate or susceptible). Otherwise, the merged label is discarded. Finally, some drugs are renamed such that there is less ambiguity as to exactly which compound is referred to by their name. The full list of modifications to drug names is listed in Table 3.

To present drugs to the model, all names of drugs are converted to SMILES strings. In this work, PubChem's canonical SMILES strings of every compound are used. In PubChem, canonical SMILES are not isomeric, which means that stereochemistry is ignored. As such, two drugs that are stereoisomers are treated as a single drug, this is the case for Ofloxacin and Levofloxacin. Furthermore, many drugs in the dataset refer to the co-administration of two compounds (such as, for example, Ampicillin-Sulbactam or Amoxicillin-Clavulanic acid). These cases are treated as a single drug with a SMILES string consisting of the strings of both constituent compounds separated by a "." character, as is common practice with SMILES strings.

B. Modeling set-up

B.1. Drug embedders

In this paper, seven ways to encode drugs in a model are tested out. In this section, those seven drug embedders are described in detail. All descriptions correspond to the final set-up used to present results, hyperparameter tuning results are presented in Appendix B.2.1.

All drug embedders encode drugs to a vector $\boldsymbol{t}_j \in \mathbb{R}^{64}$. The most simple way to obtain a dense vector of that size for every drug is via a **one-hot embedding**. Every drug gets assigned an index in a vector, and the resulting vectors are embedded to a dense representation via a single linear layer. Encoding drugs in this way is the most straightforward, but no structural information of the underlying active compound is included. No inductive bias is presented to the model that will give structurally-similar drugs comparable embeddings. As such, all this information must be learnt from data. Similarly, such drug embedders can not be generalized out-of-the-box to drugs it hasn't seen in the training data, as there are no indices - and learnt embeddings - for them.

The (local) structure of drugs can be encoded via fingerprints. A molecular fingerprint corresponds to a bit-vector in which every bit corresponds to the presence or absence of a substructure (Capecchi et al., 2020). In this paper, **Morgan fingerprints** with a diameter of 4 and consisting of 512 bits are derived from RDKit (Landrum, 2013). The resulting vector is embedded with a linear layer to get a dense drug representation. Embedding drugs using such structural features overcomes the aforementioned drawbacks with one-hot embeddings.

Similarly, the identity of a drug can be communi-

cated via the textual representation known as SMILES strings (Weininger, 1988). Here, an adaptation of SMILES for machine and deep learning applications is used, called DeepSMILES (O'Boyle and Dalke, 2018). All the different letters in the alphabet are assigned an index in a one-hot vector. Hence, every molecule can be encoded to a matrix $\boldsymbol{S}_j \in \mathbb{R}^{v \times l}$, with v the vocabulary size of the DeepSMILES alphabet and l the string length of the molecule. This representation can be processed to a vector embedding using any neural network type that is appropriate for variable-length sequences.

A **1D CNN** detects and composes local patterns in the DeepSMILES string to a final drug embedding. Every input channel corresponds to a specific letter in the SMILES alphabet. The convolutional network used here consists of a position-wise linear layer to embed the channels to 64 dimensions, four convolutional blocks placed in sequence, followed by a global max-pooling operation across the length axis and a final linear layer to return a vector $\boldsymbol{t}_j \in \mathbb{R}^{64}$. The global max-pooling layer allows the same network to be used for variable-length inputs. Each convolutional block consists of a structure similar to the one found in transformers (Vaswani et al., 2017). A first layer normalization is followed by a (padded) convolutional layer with a kernel size of 5, a first residual connection is wrapped around these two operations. After this, a position-wise feedforward makes up the second half of the convolutional block. The position-wise feedforward consists of a layer normalization, after which a GeLU-based gated linear unit identical to the one introduced by Shazeer (2020) is employed: $z = Dropout_{0.2}((GeLU(\boldsymbol{xW}) \odot \boldsymbol{xV}))\boldsymbol{W}_2$. First, the input is sent to two position-wise linear layers via $\boldsymbol{W}$ and $\boldsymbol{V}$, each of them exploding the hidden dimensions of the input by a factor of four. By sending the result of the first linear layer to a GeLU activation and multiplying element-wise with the result of the second linear layer, a gated linear unit structure is obtained. The output of this gated linear unit is sent to a dropout layer with rate 0.2 and then returned to the original dimension size via a final linear layer $\boldsymbol{W}_2$. Around this second LayerNorm and feedforward structure, a residual connection is again wrapped. All residual blocks have an input and output hidden dimension of 64. Figure 5 shows the structure of this convolutional block. Its design adopts the current state-of-the-art practices in transformers, which are increasingly being used in convolutional networks (Liu et al., 2022).

A **transformer** can be used to learn and compose signals in the DeepSMILES strings that occur sequence-wide, as opposed to the local pattern detection with a 1D CNN. The DeepSMILES strings are similarly embedded to 64 dimensions per character. After this, sinusoidal positional encodings (Vaswani et al., 2017) are added, and a CLS token embedding is prepended to the sequence. Four transformer blocks are employed, each with 64 as hidden dimension. The structure of the blocks are identical as with the 1D CNN (Figure 5), but using scaled dot-product self-attention instead of 1D convolutions. The self-attention operation uses 8 heads. The output at the CLS token is used as a "summary" of the content in the sequence (as opposed to the global max pooling with the CNN). A final linear layer on the output of the CLS token returns the drug embedding $\boldsymbol{t}_j \in \mathbb{R}^{64}$.

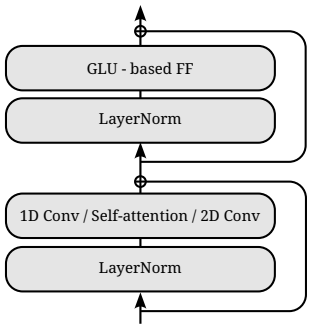

Figure 5 | Structure used for the residual blocks, used in the 1D CNN, 2D CNN, and transformer. In the case of convolutions, the output is zero padded so as to produce the same output dimensions as in the input.

A **recurrent neural network** (RNN) is used to process the DeepSMILES strings sequentially. The RNN used here consists of a bidirectional GRU with 64 hidden dimensions (Cho et al., 2014). The two final hidden states of the GRU are used as "summaries" of the content in the sequence. These two final states are averaged (element-wise) and sent to a final linear layer returning $\boldsymbol{t}_j \in \mathbb{R}^{64}$.

All three aforementioned neural network structures work on variable-length (Deep)SMILES strings. With mini-batches, input drugs are (zero) padded so that everything fits into a tensor $\mathbf{S} \in \mathbb{R}^{b \times v \times l}$, with b the batch size, v the vocabulary size, and l the longest length of a drug in the batch. The three aforementioned neural nets are adapted so that no information can flow from masked tokens to actual drug tokens (through masking after convolutions or in the attention matrices).

As (Deep)SMILES are a 1D representation of a 3D molecular structure, a more-detailed view of the drug may be obtained by permitting an extra dimension into its input representation. Drawings of drugs

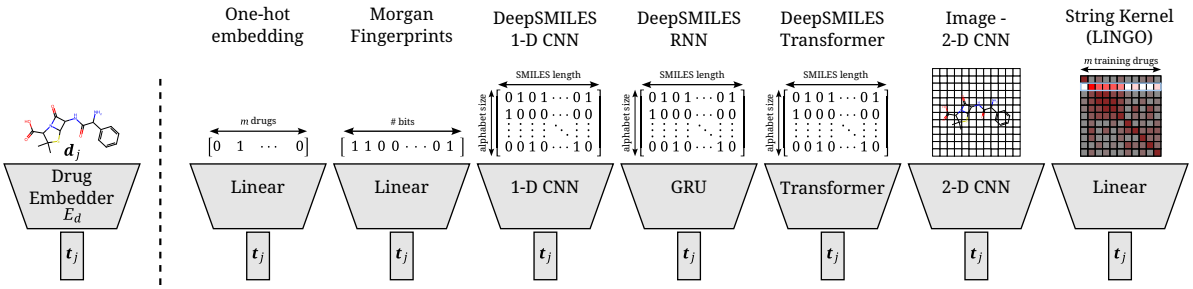

Figure 6 | Overview of all different drug embedders tested in this work. One-hot embeddings are the only technique not incorporating prior knowledge of the structure of the compound. Hence, they are the only technique incapable of directly transferring to new compounds. Morgan fingerprints produce a bit-vector containing information on the presence of certain substructures. DeepSMILES strings are encoded and processed with a 1D CNN, GRU, or transformer. Drawings of molecules are processed with a 2D CNN. A string kernel on SMILES strings produces a numerical vector for every drug (taken as the row in the resulting Gram matrix).

achieve this 2D view of the molecule. Here, 128×128 drawings of drugs are obtained through RDKit. The RGB values are inverted so as to make the parts of the image containing molecule "activated". Also, the RGB values are scaled to the range of 0 to 1 by dividing by 255. A **2D CNN** processes the images to a drug embedding. The CNN consists of an input convolutional layer with kernel size and stride of 2. The input layer takes the three input channels and returns 32 hidden dimensions. After, two convolutional blocks of the same structure as with the transformer and 1D CNN are placed in tandem (Figure 5). The 2D convolutional operation used in the convolutional operation has a kernel size of 5. Hereafter, a global max-pooling operation across the height and width of the image is performed, followed by a final linear layer producing the drug embedding $\boldsymbol{t}_j \in \mathbb{R}^{64}$.

A final way to obtain a numerical representation of drugs tested here is through similarity matrices. A **string kernel** is used to create a Gram matrix of all drugs in the training set. The input representation of a drug is then simply a row in said Gram matrix. This approach is generalizable to unseen drugs at inference time, as obtaining a representation for them involves running the kernel function of the new compound to all training drugs. In this work, the LINGO string kernel (using 4-mers) is used (Vidal et al., 2005), as this kernel performed well in a recent benchmark (Öztürk et al., 2016). Note that here, SMILES strings are used instead of DeepSMILES (as with the 1D CNN, RNN, and transformer). A linear layer produces the final drug embedding $\boldsymbol{t}_j \in \mathbb{R}^{64}$ from a row in the Gram matrix.

A visual overview of all seven drug embedders in given in Figure 6.

B.2. Hyperparameter tuning

B.2.1. Dual-branch models

Due to the complexity of tuning two branches and the size of the dataset, tuning is mostly done in an *ad hoc* fashion, relying on knowledge of current best practices in deep learning. Only some hyperparameters of interest are tuned on the validation set. Here, we present validation model results of those experiments. All results presented here concern models that are trained with a medium-sized spectrum embedder, with hyperparameters otherwise as described in Appendix B.1. All numbers indicate an average over three runs, similarly choosing the best average out of four tested learning rates (1e-4, 5e-4, 1e-3, 5e-3).

Figure 7A shows validation set performances for a grid of different kernel sizes and hidden dimensionalities for the SMILES 1-D CNN. The best-performing hidden dimensionality (64) is copied to the (Deep)SMILES Transformer and GRU without further tuning. In Figure 7B, a similar grid is shown for the Image 2-D CNN, where it is found that a smaller hidden size is favored. Figure 7C shows the performance for using different molecular string representations as input to the 1-D CNN model: SMILES, DeepSMILES (O'Boyle and Dalke, 2018), and SELFIES (Krenn et al., 2020). While all techniques perform competitively, DeepSMILES strings outperform the other two by a small margin. Similarly, DeepSMILES are thus selected as input representations for the Transformer and GRU, without further tuning. Figure 7D shows how sinusoidal positional encodings outperform learned positional encodings (as in Devlin et al. (2018)). It is found that a bidirectional GRU considerably outpeforms a unidirectional one (Figure 7E). Finally, the number of bits in the Morgan fin-

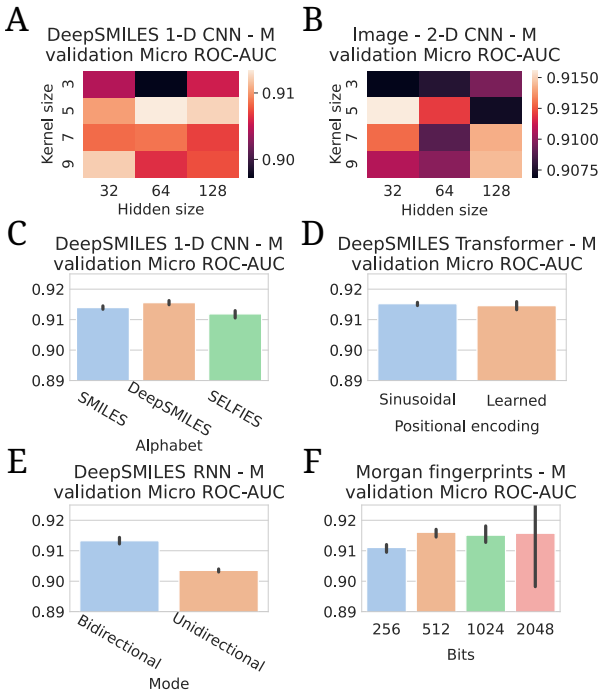

Figure 7 | All hyperparameter tuning experiments. All evaluations are listed in terms of validation Micro ROC-AUCs. All numbers are averages of three model runs, with errorbars showing standard deviations. In every experiment, the highest average is chosen to use in the final models.

gerprint encoding is also tuned (Figure 7F). It is seen that including lower than 512 bits degenerates performance, but including more than 512 introduces instabilities in model training, as the model becomes prone to overfitting the drug branch.

B.2.2. Specialist baselines

All baselines are trained using the same data splits as used with the dual-branch model. In essence: all DRIAMS-A spectra before the year 2018 are in the training set. The remaining spectra from 2018 are evenly divided among validation and test (with which belonging to which corresponding with the splits used for the dual-branch experiments). The same preprocessed 6000-dimensional spectrum representations are used as input.

Logistic regression baselines are trained with the L-BFGS solver for a maximum of 500 training iterations. For every species-drug combination, a grid search is performed on various hyperparameters, selecting the best based on validation ROC-AUC. The hyperparameters that are tuned are the scaling method on the features (either none, or standard scaling), and the L2 regularization strength ($C \in \{10^{-3}, 10^{-2}, ..., 10^{2}, 10^{3}\}$)

For XGBoost, default parameters are used apart from those tuned. For every species-drug combination, a grid is run, testing different numbers of trees (`n_estimators` $\in \{25, 50, 100, 200\}$) and learning rate (`learning_rate` $\in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$).

For the MLP baselines, the same hyperparameters are used as for the spectrum branch. Briefly recapitulated: between every two fully-connected layers, a series of operations consisting of (1) a GeLU activation, (2) a dropout rate of 0.2, and (3) layer normalization, is applied. The sizes of the models are as in Table 1, but then ending in 1 node instead of 64. For every species-drug combination, models are trained using the cross-entropy loss and the Adam optimizer for a maximum of 250 epochs. A batch size of 128 is employed. A linear learning rate warm-up is applied over the first 250 steps. Early stopping based on validation ROC-AUC is applied with a patience of 10 epochs. The model with the best validation ROC-AUC during training is kept as final model. the best model out of four different learning rates (`learning_rate` $\in$ {1e-4, 5e-4, 1e-3, 5e-3}) is chosen based on their validation ROC-AUC.

C. Tables and figures supporting the results section

Table 4 | Full table of test results. The listed averages and standard deviations are calculated over three independent runs of the same model. The best models for every metric per drug embedder are underlined. The overall best model for every metric is in bold face.

Drug embedder	Spectrum embedder	Micro ROC-AUC	Instance-wise ROC-AUC	Macro ROC-AUC	Instance-wise Prec@1 (neg class)
One-hot embedding	S	0.9241 ± 0.0011	0.9390 ± 0.0019	0.8047 ± 0.0216	0.9887 ± 0.0011
	M	**0.9262 ± 0.0008**	**0.9410 ± 0.0012**	0.8090 ± 0.0124	0.9890 ± 0.0025
	L	0.9249 ± 0.0002	0.9401 ± 0.0009	0.7909 ± 0.0199	0.9877 ± 0.0016
	XL	0.9233 ± 0.0005	0.9409 ± 0.0009	0.7812 ± 0.0213	0.9870 ± 0.0007
Morgan Fingerprints	S	0.9233 ± 0.0015	0.9396 ± 0.0020	**0.8136 ± 0.0104**	0.9889 ± 0.0016
	M	0.9222 ± 0.0021	0.9401 ± 0.0011	0.8084 ± 0.0020	**0.9918 ± 0.0013**
	L	0.9214 ± 0.0003	0.9392 ± 0.0016	0.7975 ± 0.0071	0.9900 ± 0.0016
	XL	0.9201 ± 0.0010	0.9391 ± 0.0011	0.7804 ± 0.0072	0.9917 ± 0.0005
DeepSMILES 1-D CNN	S	0.9210 ± 0.0003	0.9366 ± 0.0008	0.7989 ± 0.0093	0.9871 ± 0.0024
	M	0.9221 ± 0.0010	0.9375 ± 0.0019	0.7879 ± 0.0222	0.9879 ± 0.0009
	L	0.9214 ± 0.0016	0.9385 ± 0.0013	0.7919 ± 0.0212	0.9876 ± 0.0020
	XL	0.9212 ± 0.0008	0.9393 ± 0.0012	0.7862 ± 0.0101	0.9904 ± 0.0005
DeepSMILES Transformer	S	0.9217 ± 0.0006	0.9367 ± 0.0008	0.8071 ± 0.0164	0.9876 ± 0.0013
	M	0.9213 ± 0.0004	0.9382 ± 0.0008	0.7862 ± 0.0032	0.9873 ± 0.0009
	L	0.9213 ± 0.0009	0.9375 ± 0.0014	0.7939 ± 0.0104	0.9878 ± 0.0029
	XL	0.9199 ± 0.0013	0.9378 ± 0.0013	0.7812 ± 0.0089	0.9870 ± 0.0029
DeepSMILES RNN	S	0.9196 ± 0.0006	0.9349 ± 0.0005	0.7848 ± 0.0021	0.9822 ± 0.0009
	M	0.9200 ± 0.0013	0.9372 ± 0.0012	0.7716 ± 0.0175	0.9858 ± 0.0029
	L	0.9183 ± 0.0012	0.9341 ± 0.0024	0.7578 ± 0.0066	0.9835 ± 0.0026
	XL	0.9172 ± 0.0011	0.9325 ± 0.0024	0.7680 ± 0.0063	0.9857 ± 0.0040
Image - 2-D CNN	S	0.9213 ± 0.0010	0.9374 ± 0.0014	0.7933 ± 0.0139	0.9878 ± 0.0024
	M	0.9228 ± 0.0007	0.9386 ± 0.0018	0.7792 ± 0.0093	0.9880 ± 0.0009
	L	0.9209 ± 0.0004	0.9377 ± 0.0008	0.7794 ± 0.0048	0.9896 ± 0.0024
	XL	0.9206 ± 0.0002	0.9373 ± 0.0015	0.7690 ± 0.0048	0.9858 ± 0.0011
String Kernel (LINGO)	S	0.9208 ± 0.0017	0.9371 ± 0.0011	0.7780 ± 0.0072	0.9883 ± 0.0016
	M	0.9206 ± 0.0008	0.9382 ± 0.0006	0.7870 ± 0.0086	0.9869 ± 0.0010
	L	0.9208 ± 0.0007	0.9386 ± 0.0015	0.7914 ± 0.0038	0.9880 ± 0.0034
	XL	0.9194 ± 0.0009	0.9368 ± 0.0006	0.7845 ± 0.0100	0.9881 ± 0.0014

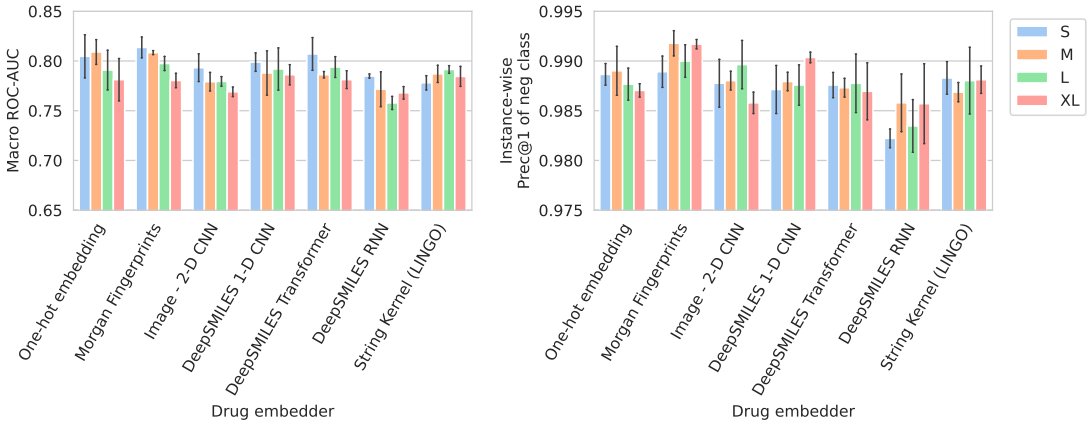

Figure 8 | Barplots showing test performance results for all trained models. Colors represent the different spectrum embedder model sizes. Performance is shown in terms of Macro ROC-AUC (computed per drug and averaged) and in terms of Instance-wise Prec@1 of the negative class. The Prec@1 evaluates how often the top-ranked prediction is correct. The Prec@1 of the negative class, hence, reports the proportion of cases for which the "most-likely susceptible drug" prediction is actually an effective one. In a scenario where the top recommended drug is always administered, it corresponds to the percentage of correctly-suggested treatments. Errorbars represent the standard deviation over three random seeds.

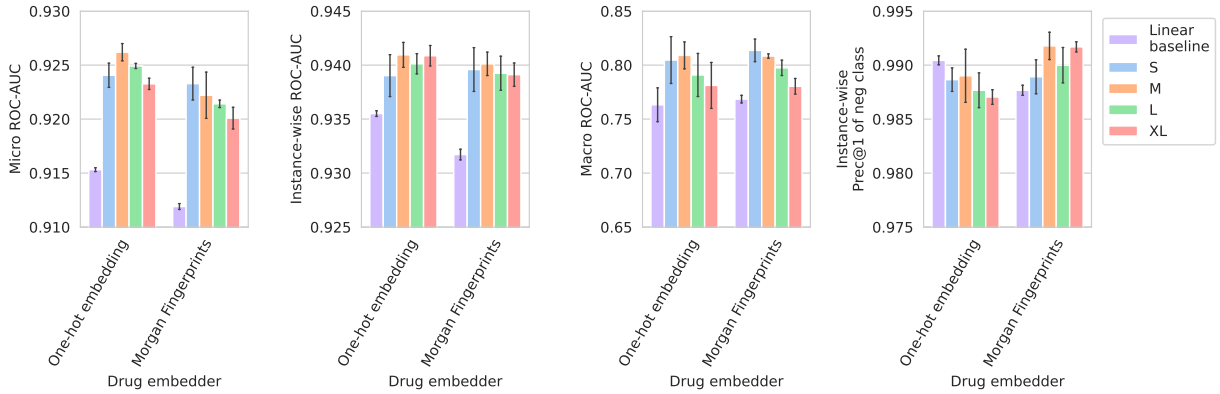

Figure 9 | Performance of models compared against a linear spectrum embedder baseline. The comparison is only shown for the two best-performing drug embedders (in terms of Micro ROC-AUC). Errorbars represent the standard deviation over three random seeds.

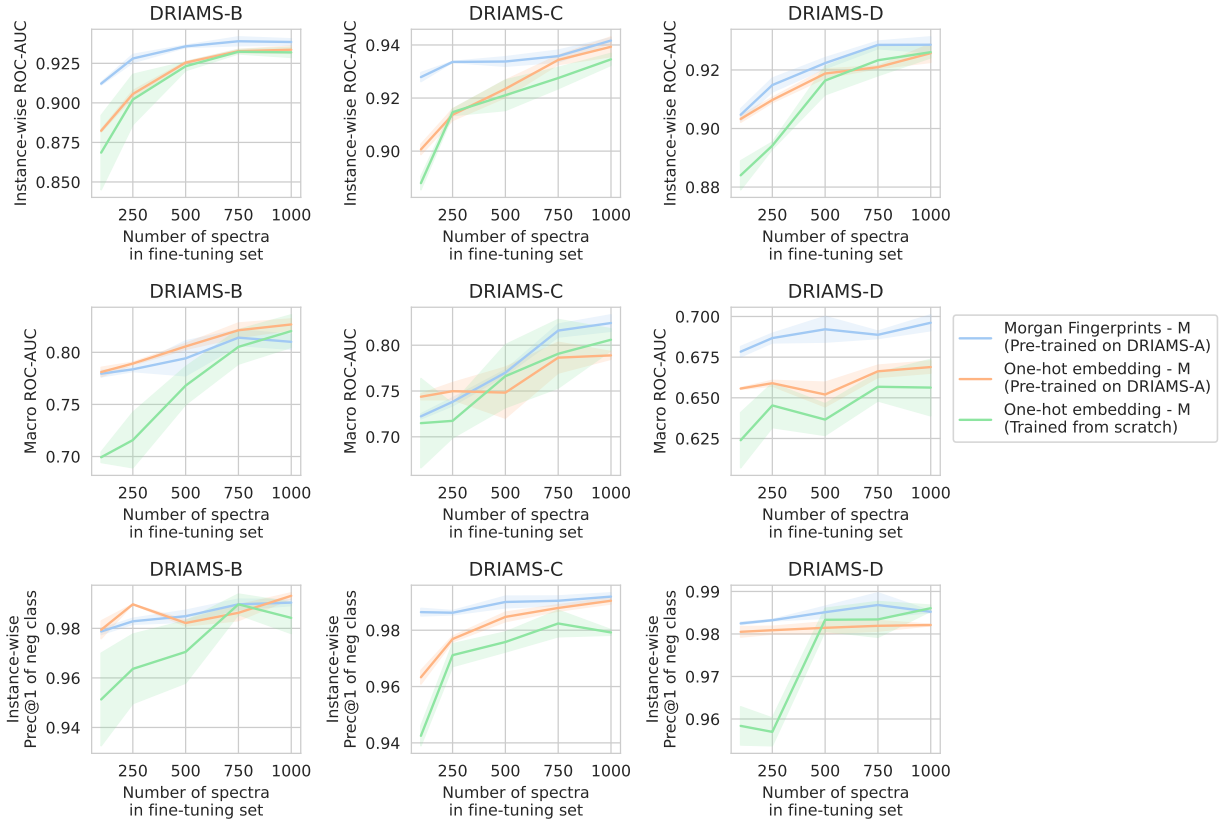

Figure 10 | Transfer learning of DRIAMS-A models to other hospitals. Errorbands show the standard deviation over three runs.

Figure 11 | UMAP scatterplots of test set MALDI-TOF spectra embeddings $\boldsymbol{x}_i$. Only embeddings belonging to the 25 most-occurring species in the test set are shown. Spectra are colored according to its AMR status to a certain drug. The twenty displayed drugs are selected based on a ranking of the product of the number of positive and negative labels $\sum_{i=1}^{n}[y_{ij} = 0] \cdot \sum_{i=1}^{n}[y_{ij} = 1]$. In this way, the drugs that have a lot of observed labels, both positives and negatives, are displayed. The drugs here are ranked 5-24 (the first four are shown in Figure 4).