ACS OMEGA

Article

Cite This: ACS Omega 2018, 3, 12033−12040

http://pubs.acs.org/journal/acsodf

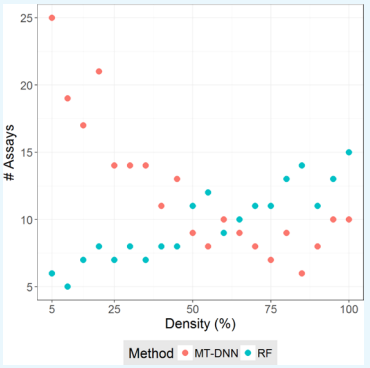
Prediction of Compound Profiling Matrices, Part II: Relative Performance of Multitask Deep Learning and Random Forest Classification on the Basis of Varying Amounts of Training Data

Raquel Rodríguez-Pérez[†,‡] and Jürgen Bajorath*[,†]

[†]Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Endenicher Allee 19c, D-53115 Bonn, Germany

[‡]Department of Medicinal Chemistry, Boehringer Ingelheim Pharma GmbH & Co. KG, Birkendorfer Str. 65, 88397 Biberach/Riß, Germany

ABSTRACT: Currently, there is a high level of interest in deep learning and multitask learning in many scientific fields including the life sciences and chemistry. Herein, we investigate the performance of multitask deep neural networks (MT-DNNs) compared to random forest (RF) classification, a standard method in machine learning, in predicting compound profiling experiments. Predictions were carried out on a large profiling matrix extracted from biological screening data. For model building, submatrices with varying data density of 5−100% were generated to investigate the influence of data sparseness on prediction performance. MT-DNN models were directly compared to RF models, and control calculations were also carried out using single-task DNNs (ST-DNNs). On the basis of compound recall, the performance of ST-DNN was consistently lower than that of the other methods. Compared to RF, MT-DNN models only yielded better prediction performance for individual assays in the profiling matrix when training data were very sparse. However, when the matrix density increased to at least 25−45%, per-assay RF models met or partly exceeded the prediction performance of MT-DNN models. When the average performances of RF and MT-DNN over the grid of all targets were compared, MT-DNN was slightly superior to RF, which was a likely consequence of multitask learning. Overall, there was no consistent advantage of MT-DNN over standard RF classification in predicting the results of compound profiling assays under varying conditions. In the presence of very sparse training data, prediction performance was limited. Under these challenging conditions, MT-DNN was the preferred approach. When more training data became available and prediction performance increased, RF performance was not inferior to MT-DNN.



1. INTRODUCTION

Recently, there has been increasing interest in machine learning (ML) and, especially, deep learning (DL) in many areas of science including pharmaceutical research.[1−3] In ML, one can distinguish between single-task (ST) and multitask (MT) learning. MT learning is based on the idea that the predictive performance of a given task can be improved by using the data available for related tasks.[4] In the context of compound activity prediction, which is a core task in computational medicinal chemistry, this principle implies that some structural features and/or molecular properties should be common to active compounds, regardless of their targets. This "basis set" of activity-relevant features would then be complemented by others to yield target-specific biological activities. Hence, bioactivity data from various assays might be considered to predict activities in a given assay on the basis of shared activity determinants, a key assumption underlying MT learning. By contrast, in ST learning, one trains models on the basis of compounds that were active or inactive in an individual assay in order to predict the potential activities of test compounds.

For MT learning, deep neural network (DNN) architectures (MT-DNNs) have become very popular,[2,3] raising expectations that they might yield further improved predictive performance compared to standard ST−ML approaches.[2,3,5] A frequent reasoning is that MT-DNNs make explicit use of more—and more diverse—training data than ST−ML approaches, which further expands the knowledge base for predictions. For example, Ramsundar et al. compared the performance of MT-DNN with different architectures, ST-DNN, and random forest (RF) predictions on four data sets (Kaggle, Factors, Kinase, and UV). Their results suggested that MT models offered improvements over RF calculations for correlated tasks.[3] However, the effect of training matrix density was not explored. Xu et al. compared the performance of ST-DNNs and MT-DNNs for different quantitative structure−activity relationship prediction tasks.[5] Their results indicated that the prediction performance and relative performance of ST-DNNs and MT-DNNs varied greatly across data sets

Received: July 17, 2018
Accepted: September 12, 2018
Published: September 27, 2018

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

containing either on-target potency values or off-target absorption, distribution, metabolism, and excretion properties. Furthermore, Xu et al. concluded that MT-DNN only outperformed ST-DNN when test compounds showed structural similarity and activity that correlated with training set instances from other tasks.[5] Recently, attempts have also been made to predict experimental compound profiling matrices.[6] Such matrices are obtained by screening a compound collection in different assays against closely related or diverse targets and yield activity profiles of test compounds. Importantly, the composition of such matrices is highly unbalanced because the majority of compounds are usually inactive across assays (otherwise, specific biological activities would not exist). In the first investigation,[6] ST and MT models were derived for individual assays in matrices to predict active compounds. Under conditions of experimental data imbalance, prediction performance using different ML approaches was overall reasonable and DNNs did not further increase the performance over RF or support vector machine (SVM) classifiers.[6]

General reasons for varying MT-DNN performance might include, for example, the high complexity of MT-DNN hyperparameter optimization and lack of transparence and/or the nature of training data that is available.[7,8] For example, Rodríguez-Pérez et al. have shown that activity prediction on the basis of ST-SVM classification and ranking became more accurate and stable with increasing numbers of available training instances and that a lower-bound threshold for active training examples was required.[8] In addition, a recent study by de la Vega de León et al. investigated the effects of missing data on the performance of MT methods.[9] In particular, the authors explored the performance of MT-DNN and Macau (Bayesian factorization) methods at different percentages of missing data. A minimum number of training instances was required to generate effective models, but the predictive ability saturated when increasing amounts of data were added.[9] Furthermore, Reker et al. have shown that only small subsets of ligand−target interaction matrices were required for ML modeling to reach upper limits of predictive performance.[10] In this case, RF models were built for predicting interacting versus noninteracting ligand−protein pairs from concatenated molecular and protein descriptors.[10]

Taken together, the studies discussed above have revealed a significant influence of training set size on the quality of both ST− and MT−ML models. However, the influence of training data sparseness on comparative ST− and MT−ML predictions remains to be investigated. Our current study was designed to address this issue by further extending previous work on the modeling of compound profiling matrices,[6] which is a prediction task of high relevance for biological screening and medicinal chemistry. Herein, a large compound profiling matrix combining different screening assays was used to derive submatrices of systematically increasing density for the training of RF, MT-DNN, and ST-DNN models that were then used to predict the activity profile of test compounds. Thereby, the relative performance of predictions using methods of different computational complexity on training matrices of stepwise increasing data density was investigated, thus directly addressing the issue of training data sparseness for comparative prediction of profiling results. The study design and results of our investigation are presented in the following.

2. RESULTS AND DISCUSSION

2.1. Study Design. *2.1.1. Focusing on Profiling Matrices.* Compound profiling matrices from biological screening represent challenging test cases for ML because of the experimental assay variance and, more importantly, inherent data imbalance. This is the case because most screening compounds are inactive in given assays, which typically yield on the order of ∼0.1−1% active compounds (hits).[11] Previously, we have investigated a variety of ML approaches for predicting the experimental results of assays forming complete or nearly complete matrices using the largest possible amount of training data on a per-assay basis.[6] In a complete (100% dense) matrix, all cells are filled with experimental observations. Matrices of decreasing density have increasing amounts of missing data points ("empty" cells). Here, we change the analysis scheme and attempt assay predictions by systematically deriving submatrices of varying density for training, thereby directly assessing the influence of data sparseness on the model quality.

2.1.2. Matrices of Varying Density. From a large profiling matrix comprising more than 140 000 compounds tested in 53 assays (with 0.8% actives), different series of matrices with stepwise increasing data density were extracted, covering the range of 5−100% density, with increments of 5% per step. Further details are provided in the Materials and Methods section. Hence, 20 matrices with varying density levels were obtained. Figure 1 shows the distribution of the number of active compounds per assay for five exemplary matrices with different densities of 5, 25, 50, 75, and 100%, respectively. The figure illustrates that increasing data density correlated with increasing numbers of active compounds available for training.

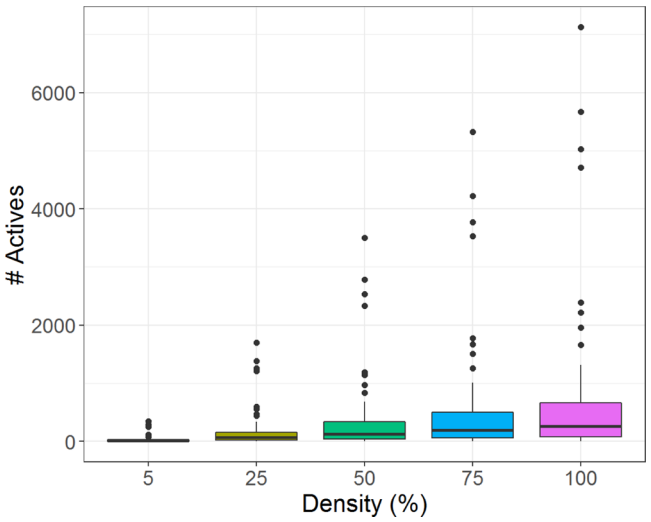


Figure 1. Active compounds per assay. Distributions of the number (#) of active compounds per assay are reported in boxplots for five different matrix density levels. Black points represent outliers.

2.1.3. Training and Predictions. For each of the 20 matrices with increasing density, ML models were derived at each density level. The resulting models were then used to predict active compounds. For ST predictions, an individual model was built for each assay (target) to predict active compounds on a per-assay basis. Individual predictions were then combined. For MT predictions, multioutput models were derived for all assays at each density level to predict the

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

complete activity profile of a compound. The resulting ST and MT models were used to predict a constant test set comprising 25% of the original profiling matrix that was excluded from training.

2.1.4. Selected Methods. As an ST−ML approach, RF was selected. This choice was motivated by the results of our previous ST matrix predictions where RF was the overall best approach, achieving slightly better performance than SVMs and ST-DNNs.[6] As an MT−ML method, MT-DNN was chosen, which represents the currently most complex MT approach. Thus, RF and MT-DNN essentially delineate opposite ends of the ML spectrum ranging from methods of low to high computational complexity and an increasing "black box" character. As a control, ST-DNN models were also generated and evaluated.

In the following, the results of our systematic activity predictions using RF and MT-DNN models trained at different density levels are presented and compared. The results were averaged over three independent trials.

2.2. Influence of Matrix Density on Prediction Performance. We first investigated how training sample sizes influenced the predictive ability of ST models based upon data from only one assay or MT models based upon data from all assays. Therefore, a pairwise comparison of ST or MT models at different density levels was carried out using the area under the receiver operating characteristic (ROC) curve (AUC) as a figure of merit. For a given assay, the AUC difference at two density levels was required to exceed 2% to classify one prediction to be superior to another. The training matrix yielding the best (worst) performing model was considered to be of *superior* (*inferior*) density. Figure 2a,b reports the results for RF and MT-DNN, respectively. The number of assays for which a model trained with a given matrix density provides better results compared to another matrix density is reported. In addition to the pairwise comparison shown in the heatmap, the panel on the right reports a cross-density comparison for the same method. For both methods, models trained at higher density levels produced better predictions on a per-assay basis than the models trained at lower density levels, as clearly revealed by the heatmap representations. Thus, consistent with earlier observations, increasing numbers of positive training instances resulted in increasing prediction performance, here for both ST and MT models. The separation between predictions with models trained at higher or lower density was even more extensive for RF than MT-DNN, as also indicated by the distribution of superior assay counts in Figure 2. Hence, RF models were overall more affected by missing data than MT-DNN models.

2.3. Method Comparison. Next, the performance of RF and MT-DNN was compared at different density levels.

2.3.1. Relative Performance for Individual Tasks. Prediction performance was first compared on a per-assay basis using AUC and Matthew's correlation coefficient (MCC). A model was considered superior if it achieved at least 2% better performance than its counterpart. This criterion was used as a disjunctive requirement for the AUC and MCC measures. Then, the number of individual assays in which a method was superior to another was separately calculated for both figures of merit. Figure 3 reports the average number of assays for three independent trials. Figure 3a shows the mean number of assays with larger AUC values for a given method at varying density levels. MT-DNN was clearly superior to RF when very sparse matrices were used for training. However, at increasing density

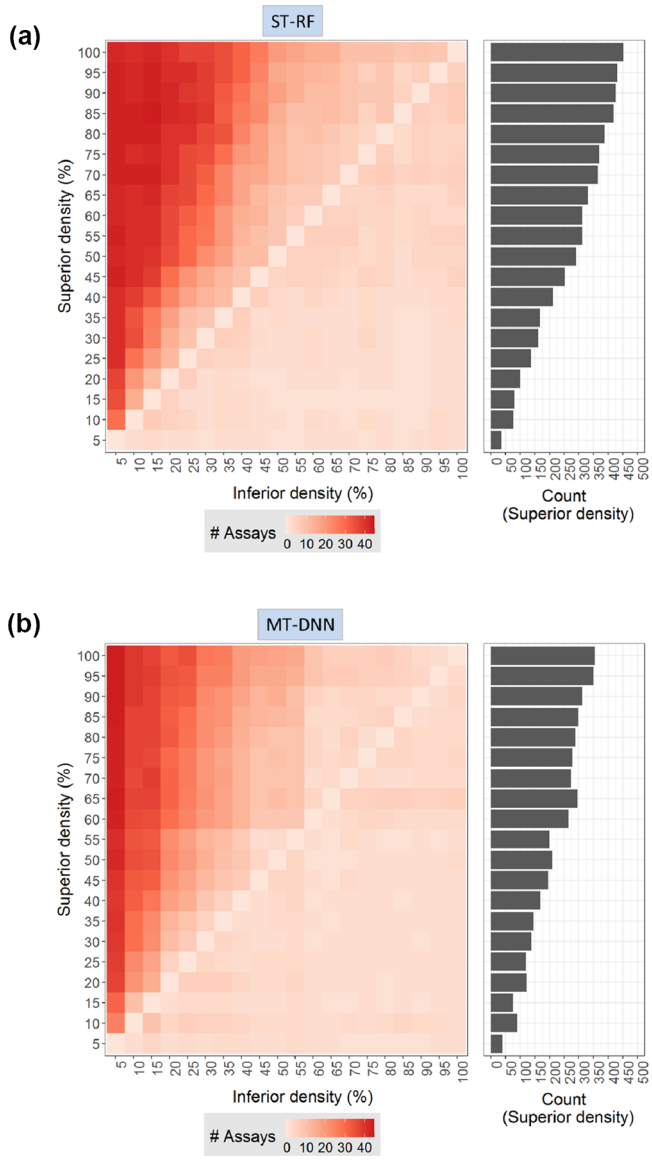


Figure 2. Prediction performance at different matrix densities. Heatmaps record the average number of assays for which larger AUC values were obtained at a given (superior) matrix density (*y*-axis) compared to another (inferior) density (*x*-axis). On the right, bar graphs report the number of assays (count) at a given density level for which better prediction performance was achieved than at any other density for the same method. (a) RF and (b) MT-DNN.

levels, performance differences became smaller, and at a density level of 50% or greater, the performance of RF began to meet and then slightly exceed the performance of MT-DNN. Figure 3b reports the corresponding comparison on the basis of MCC calculations. In this case, MT-DNN models produced better predictions at low density levels of up to 25%. At further increasing density, however, RF models were clearly superior to MT-DNN. Thus, on the basis of the AUC and MCC performance measures, similar trends were observed on a per-assay basis, with MT-DNN models yielding better prediction performance for training on very sparse matrices and RF models having better prediction performance at increasing density levels, especially when evaluated on the basis of MCC calculations. At high density levels, that is, in the presence of large amounts of training data, RF models were

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

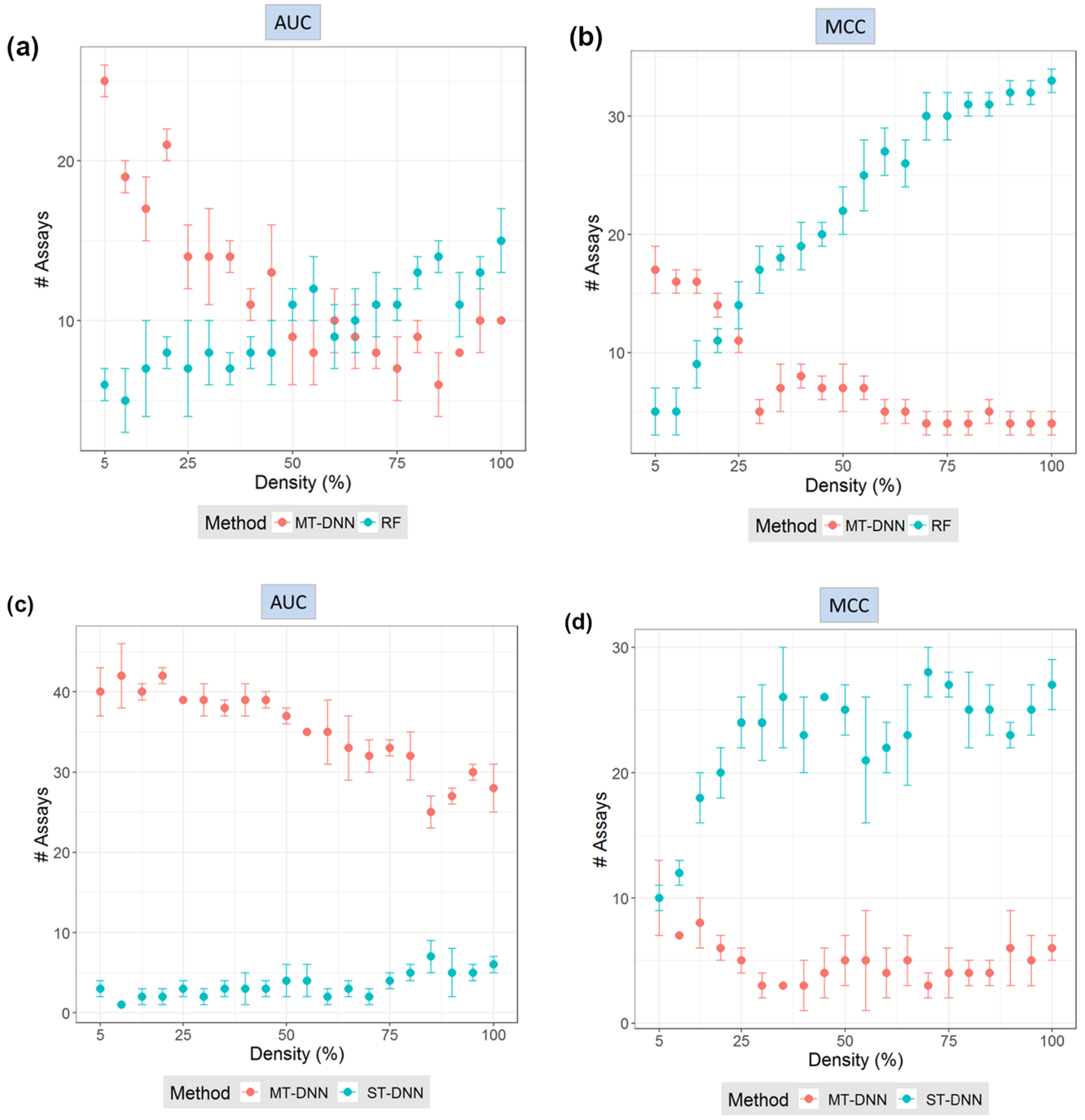


Figure 3. Per-assay comparison of prediction performance using different methods. For different trials covering all matrix density levels, the mean (dot) and standard deviation (error bar) of the number of assays are given for which one method achieved higher prediction performance than the other on the basis of different measures. RF, MT-DNN, and ST-DNN models were compared. (a) MT-DNN vs RF on the basis of AUC, (b) MT-DNN vs RF; MCC, (c) MT-DNN vs ST-DNN; AUC, and (d) MT-DNN vs ST-DNN; MCC.

superior on a per-assay basis to the much more complex MT-DNN models.

To provide additional control calculations, ST-DNN models were also generated. Figure 3c compares ST- and MT-DNN models on the basis of AUC values. MT-DNN models outperformed ST-DNN models in most assays at varying density levels. ST-DNN models only yielded better AUC values in a few cases. At decreasing matrix density, performance differences between MT- and ST-DNN increased, and MT-DNN was progressively superior. Figure 3d shows the results of MCC calculations. Here, ST-DNN models yielded larger MCC values for more assays than MT-DNN models. However, for very sparse training matrices, the relative performances of both methods became comparable.

2.3.2. Global Prediction Performance. Figure 4a shows the mean AUC, balanced accuracy (BA), and MCC values over all assays at varying density levels. Values of different performance measures are reported in Table 1. Different from the results obtained for individual assays, on average, predictions were slightly superior for MT-DNN compared to RF when assessed on the basis of AUC and clearly superior on the basis of BA calculations. However, on the basis of MCC calculations, the

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

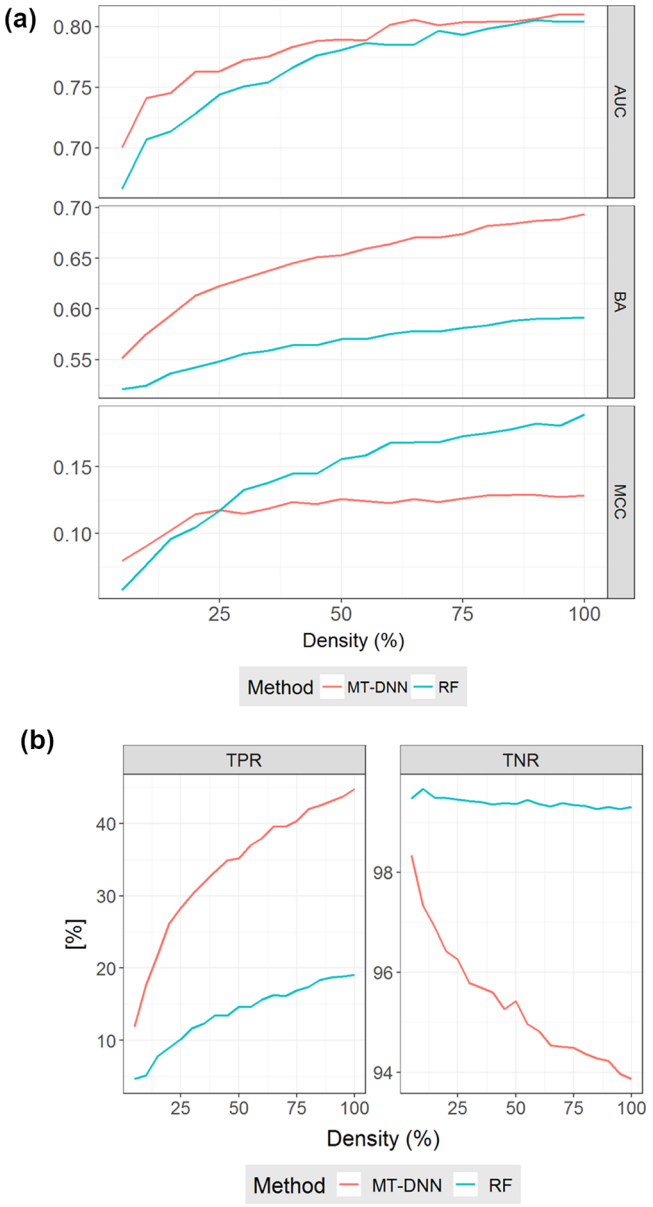


Figure 4. Global prediction performance using different methods. For different trials covering all matrix density levels, the mean prediction performance over all assays is compared for MT-DNN and RF using different measures. (a) AUC (top), BA (middle), and MCC (bottom), (b) TPR (right), and TNR (left).

global prediction was only slightly better for MT-DNN models at very low density levels of up to 25%. Then, the prediction performance of RF models gradually exceeded the performance of MT-DNN models, consistent with the results in Figure 3b. Hence, Figure 4a shows that different performance measures produced different results. As a consensus, we would conclude that average results over all assays were slightly better for MT-DNN than RF.

To better understand apparent differences resulting from the application of alternative performance measures, confusion matrices were generated at different density levels using mean values. Rates derived from raw counts of true positives (TPs), false positives (FPs), false negatives (FNs), and true negatives (TNs) were calculated. Figure 4b shows the TP rate (TPR) and TN rate (TNR), which are defined as follows: TPR = TP/(TP + FN) and TNR = TN/(TN + FP). Therefore, TPR and TNR are related to FN rates (FNR) and FP rates (FPR), respectively. TPR and FNR displayed the same tendency for RF and MT-DNN. At increasing density, TPR increased and FNR decreased. However, for MT-DNN, FPR increased and TNR decreased at increasing density levels, whereas they remained essentially constant for RF across all levels. Thus, MT-DNN predicted more FPs than RF at increasing density. We note that the constantly used test set contained a mean of 35 523 inactive and only 305 active compounds per target, given the inherent data imbalance. Consequently, figures of merit that use absolute values such as MCC are strongly affected by the different magnitudes of the numbers of active and inactive compounds. Conversely, other measures relying on proportions only yield small differences, which correspond, however, to large differences in the absolute number of errors.

On the basis of MCC calculations, MT-DNN model performance was clearly inferior to RF, except at lower density levels, when the number of FPs and TNs decreased and increased, respectively. On the other hand, the model performance assessed by BA taking only the TPR and TNR into account was superior for MT-DNN, given that the TPR was consistently higher for MT-DNN and differences in TNR were comparably small. These aspects must be taken into consideration when judging relative prediction performance on imbalanced data sets using alternative figures of merit.

ST-DNN was also included in the global comparison as a control. On the basis of AUC values, ST-DNN performed consistently worse than the other two methods. In addition, ST-DNN models produced BA values falling in between those of RF and MT-DNN and MCC values that were overall comparable to RF.

The consensus view emerging from the results comparing MT-DNN and RF shown in Figures 3 and 4 was that MT-DNN was only superior to RF when models were trained on the basis of very sparse matrices. When examining the relative prediction performance (Figure 3), MT-DNN models only

Table 1. Evaluation of Predictions Applying Different Performance Measures[a]

	AUC		BA		MCC		TPR		TNR	
matrix density (%)	MT-DNN	RF	MT-DNN	RF	MT-DNN	RF	MT-DNN	RF	MT-DNN	RF
5	0.700	0.666	0.551	0.521	0.080	0.058	11.9	4.6	98.3	99.5
25	0.763	0.744	0.623	0.548	0.117	0.117	28.3	10.2	96.3	99.5
50	0.790	0.781	0.653	0.570	0.126	0.156	35.2	14.7	95.4	99.4
75	0.803	0.793	0.674	0.581	0.126	0.173	40.3	16.9	94.5	99.3
100	0.810	0.804	0.693	0.591	0.128	0.190	44.8	19.1	93.9	99.3

[a]Reported are mean AUC, BA, and MCC values for global predictions using MT-DNN and RF models trained at varying matrix density levels. In addition, mean TPR and TNR values are given.

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

displayed superior performance to RF models at training matrix density levels of up to 25−45%, depending on the performance measures that were applied. By contrast, at increasing matrix density, RF calculations often met or exceeded the prediction performance of MT-DNN at the level of individual assays. Global prediction results (Figure 4) also showed that when enough training data were available, RF models were at least as good as MT-DNN models. Only global BA values were consistently higher for MT-DNN, but for the remaining performance measures (AUC, MCC), MT learning only provided a notable advantage at low matrix density levels.

2.4. Concluding Discussion. In this work, we have systematically explored the effects of using varying amounts of training data on MT-DNN and RF modeling. As a prediction task representing experimental results, a large compound profiling matrix was selected. The analysis was facilitated by generating assay submatrices of varying density for model derivation. The resulting models were then compared on the basis of a consistently used test submatrix of 100% density. There was no significant global correlation between prediction tasks. Differences in the performance of (low-complexity) RF and (high-complexity) MT-DNN models were observed at different density levels.

When trained on very sparse matrices, MT-DNN models yielded better prediction performance than RF models. However, when the density increased to 25−45%, per-assay RF models met or slightly exceeded the prediction performance of MT-DNN models. Thus, compared to a RF, a standard ML classifier, MT-DNN models only provided a learning advantage for individual assays when training data were very limited. However, when predictions were averaged over all assays, MT-DNN was the overall superior approach, albeit by a confined margin, depending on the applied performance measures. This observation reflected the presence of more stable predictions as a likely consequence of MT learning. On the basis of AUC values, ST-DNN was consistently inferior to MT-DNN and RF but produced higher MCC values than MT-DNN for matrices of increasing density. In all instances, performance assessment yielded partly different results, depending on the measures that were used, emphasizing the need to consider alternative performance measures in ML.

Taken together, the results of our analysis show that there was no consistent advantage of MT-DNNs over RF in predicting profiling assay results, as one might have anticipated, given high expectations often associated with MT DL. These findings should balance such expectations, at least for applications of DL in compound screening. However, they are also encouraging from the point of view that reasonable prediction performance was also achieved on a complicated prediction task with a standard ML classifier of much lower complexity than DNN architectures. Clearly, under most challenging conditions of data sparseness, when prediction performance was limited, MT-DNN was the superior approach. When increasing amounts of training data became available, and the model quality generally improved, the performance of MT-DNN and RF was comparable.

Taken together, our findings also suggest that MT-DNN might be preferred over standard classification methods such as RF in special situations, for example, when the main objective is modeling a single task (activity) and only very little training data are available for this task, but extensive data are available for related (correlated) tasks (such as similar activities). In addition, MT-DNN might be an approach of choice when the main objective is improving global prediction performance over multiple screens, and only sparse training matrices are available.

In future work, additional prediction tasks in chemistry and other challenging prediction conditions should be explored to further evaluate potentially significant advantages of DL and MT learning over standard ML approaches.

3. MATERIALS AND METHODS

3.1. Assay Data. A large compound profiling matrix was algorithmically extracted[12] from PubChem confirmatory assays as described previously[6] and provided the basis for our analysis. This matrix consisted of 143 310 compounds tested in 53 assays (covering a diverse range of 53 unique target proteins).[6] In the matrix, activity versus inactivity of compounds in assays was recorded in a binary format (i.e., 1 vs 0). The matrix density of experimental observations was 96.4%. As reported in Table 2, the majority of screened compounds (77%) were consistently inactive in all assays, 13% of the compounds had single-target activity, and 10% had multitarget activity. The resulting global proportion of matrix cells containing activity annotations was 0.8%. As reported previously,[6] the intra- and interassay similarity of active matrix compounds was generally low.

Table 2. Matrix Compounds with Different Activity Status[a]

activity status	number of compounds
consistently inactive	110 272 (77%)
single-target activity	19 054 (13%)
multitarget activity	13 984 (10%)

[a]Reported are the numbers of matrix compounds with different activities across all assays.

3.2. Matrix Modifications. For computational modeling, the matrix was completed (100% density) by conventional zero filling,[13] that is, missing experimental data (3.6%) were compensated for by inactivity annotations. The complete matrix was then randomly divided into training (75%) and test data (25%). The test set submatrix was complete (100% density). By contrast, training sets of varying density were created ranging from 5 to 100% density, with increments of 5%. To these ends, 95% of the compound-assay annotations were randomly removed, and assay data were added back in 5% increments, yielding cumulatively built training sets of stepwise increasing density.

3.3. Machine Learning. Using a consistent molecular representation, two distinct ML approaches of different designs and computational complexity were investigated including (ST-)RF and MT-DNN. As a control, ST-DNN calculations were carried out.

3.3.1. Molecular Representation. The folded (1024-bit) version of the extended connectivity fingerprint with bond diameter 4 (ECFP4) was used as a molecular representation.[14] ECFP4 was computed using in-house Python scripts based upon the OEChem Toolkit.[15]

3.3.2. Calculation Protocol. For each matrix density level, RF and MT-DNN models were trained and used to predict the same test data. Three independent trials with different random seeds were carried out for training sets covering all density levels, as detailed above. In each trial, RF and MT-DNN models were built for individual assays using the same cumulative training sets and compared. The use of different

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

random seeds for modeling modified the initialization of MT-DNN and cross-validation partitions of RF models.

Models were only built for assays for which the training matrices and the test matrix consistently contained active compounds. The different training sets included active compounds from all 53 assays, whereas the test set was found to contain active compounds from 47 assays. Thus, RF and MT-DNN models were ultimately built for 47 assays (targets).

Figure 5 shows the distribution of pairwise Pearson correlation coefficients (r) between learning tasks encoded by the matrix. The maximum r value was 0.79 and the mean r value 0.03, which indicated very low global correlation between tasks (while significant correlation between tasks typically supports transfer and MT learning).

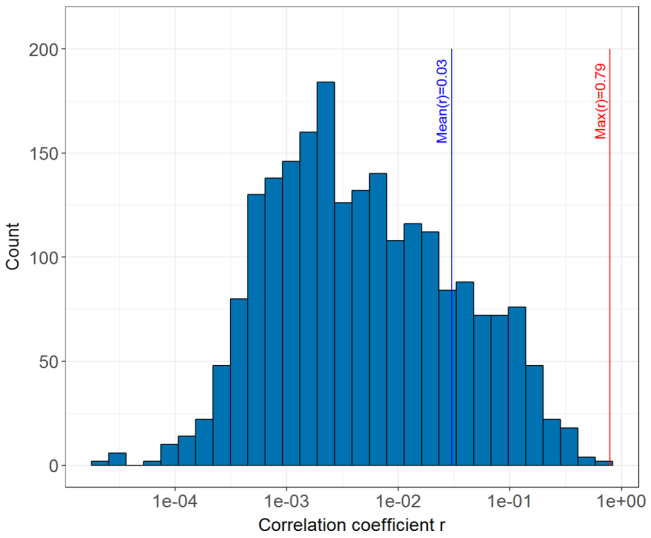


Figure 5. Correlation between learning tasks. The histogram shows the distribution of pairwise Pearson correlation coefficients (r) between all learning tasks (training data) on a logarithmic x-scale.

3.3.3. Random Forest. The RF approach utilizes an ensemble of decision trees that are built with different subsets of samples by bootstrapping.[16] Variance is reduced by training decision trees using different subsets of the training set. Moreover, a random sample of features is considered during node splitting, which avoids the presence of correlated trees because of feature dominance.[16] In this study, the *scikit-learn* implementation of RF was used.[17] The number of trees was set at 100, and two hyperparameters were optimized using twofold cross-validation including the number of randomly selected features available at each bifurcation (*max_features*) and the minimum number of samples required to reach a leaf node (*min_samples_leaf*). Cross-validation optimization was independently carried out on a per-assay basis such that different optimum hyperparameters could be derived for each RF model. Tested values for *max_features* included the total number of features, the square root, and the logarithm to base two of the number of features. In addition, for *min_samples_leaf*, candidate values were 1, 5, and 10. Class weights were set according to the ratio of samples from each training label (i.e., active vs inactive) such that errors in the minority class were preferentially penalized.[7] Default values were used for all remaining hyperparameters.[17]

3.3.4. Multitask Deep Neural Networks. Feed-forward DNNs learn a function that approximates the input values to an output (class) without backward connections or loops within the network architecture.[18,19] DNNs can be used for MT activity predictions by considering multiple nodes in the output layer, yielding MT-DNNs.[19] A DNN is constituted by different layers including an input layer, hidden layers, and an output layer.[20] Each layer contains a number of neurons that assign weights to the values originating from the previous layer, adds them, and passes the sum through an activation function

$$y_k = f\left(\sum_j w_{kj}x_j + b_k\right)$$

Here, y_k is the output and x_j is the input of neuron k, f is the activation function, w_{kj} are the weights connecting neuron k with x_j, and b_k is the so-called bias.[21] Ultimately, the output layer transforms the values of the last hidden layer into the output values (classes). Weights are derived during training by the iterative value modification to obtain the desired output y. Gradient descent is computed using back-propagation to optimize the weights and biases.[20] For weight and bias adjustment, back-propagation required the actual labels (active/inactive) of the training set. For MT-DNN calculations, Keras[22] and TensorFlow[23] Python implementations were used.

For MT-DNNs, many optimization-relevant hyperparameters are available. Because 20 successive density levels and three trials per level were investigated, an exhaustive evaluation of alternative hyperparameter settings was computationally infeasible. Instead, a set of hyperparameters permitting validation loss convergence was chosen for comparison of different density levels, as suggested by previous optimization studies.[6,20] These parameter settings included, first, a pyramidal network architecture with two hidden layers of 2000 and 1000 neurons, respectively. In addition, the rectified linear unit (ReLU) function was chosen as an activation function, except for the output layer, in which the sigmoid function was employed. Furthermore, as an optimization function, stochastic gradient descent (SGD) was used, the batch size was 1024, and the initial learning rate (LR) was set to 0.01 and iteratively decreased when the training loss reached a plateau and remained constant. To avoid overfitting, a fall-out rate of 25% was applied. A total of 800 epochs were computed, and the best resulting model was used for prediction. Class weights were considered. For internal validation, an 80−20% data split was applied. Binary cross-entropy was used as the loss function and the reduction of the LR and the choice of the best model after 800 epochs were based on minimizing this validation loss.

3.3.5. Single-Task Deep Neural Networks. As additional control calculations, ST-DNN models were built and evaluated at the same 20 density levels. Hyperparameter values were set according to previous optimization results.[6] The ST-DNN network architecture included two hidden layers with 2000 and 1000 neurons, respectively. ReLU was the activation function, except for the output layer, which used the softmax function. The optimization function was SGD, the batch size was set to 128, and the LR was set to 0.0001. To avoid overfitting, a drop-out rate of 25% was permitted, and L2-regularization was applied. Furthermore, batch normalization was applied to all layers, and a total of 100 epochs were computed.

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040

3.4. Performance Measures. The performance of ML models was evaluated using confusion matrices and three different measures including the area under the ROC curve (AUC),[24] MCC,[25] and BA.[26] AUC evaluates the global ranking of test compounds. MCC and BA are defined below

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

$$\mathrm{BA} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$$

AUTHOR INFORMATION

Corresponding Author

*E-mail: bajorath@bit.uni-bonn.de. Phone: 49-228-7369-100 (J.B.).

ORCID

Jürgen Bajorath: 0000-0002-0557-5714

Author Contributions

The study was carried out, and the manuscript was written with contributions of all authors. All authors have approved the final version of the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The project leading to this report has received funding (for R.R.P.) from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 676434, "Big Data in Chemistry" ("BIGCHEM", http://bigchem.eu). The article reflects only the authors' view and neither the European Commission nor the Research Executive Agency (REA) are responsible for any use that may be made of the information it contains. The authors thank the OpenEye Scientific Software, Inc., for providing a free academic license of the OpenEye toolkit. The authors also thank Nils Weskamp for support and helpful discussions.

REFERENCES

(1) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The Rise of Deep Learning in Drug Discovery. *Drug Discovery Today* **2018**, *23*, 1241−1250.

(2) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.

(3) Ramsundar, B.; Liu, B.; Wu, Z.; Verras, A.; Tudor, M.; Sheridan, R. P.; Pande, V. Is Multitask Deep Learning Practical for Pharma? *J. Chem. Inf. Model.* **2017**, *57*, 2068−2076.

(4) Caruana, R. Multitask Learning. In *Learning to Learn*; Thrun, S., Pratt, L., Eds.; Springer: New York, 1998; pp 95−133.

(5) Xu, Y.; Ma, J.; Liaw, A.; Sheridan, R. P.; Svetnik, V. Demystifying Multitask Deep Neural Networks for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2017**, *57*, 2490−2504.

(6) Rodríguez-Pérez, R.; Miyao, T.; Jasial, S.; Vogt, M.; Bajorath, J. Prediction of Compound Profiling Matrices Using Machine Learning. *ACS Omega* **2018**, *3*, 4713−4723.

(7) Kurczab, R.; Bojarski, A. J. The influence of the negative-positive ratio and screening database size on the performance of machine learning-based virtual screening. *PLoS One* **2017**, *12*, No. e0175410.

(8) Rodríguez-Pérez, R.; Vogt, M.; Bajorath, J. Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds. *J. Chem. Inf. Model.* **2017**, *57*, 710−716.

(9) de la Vega de León, A.; Chen, B.; Gillet, V. J. Effect of Missing Data on Multitask Prediction Methods. *J. Cheminf.* **2018**, *10*, 26.

(10) Reker, D.; Schneider, P.; Schneider, G.; Brown, J. B. Active Learning for Computational Chemogenomics. *Future Med. Chem.* **2017**, *9*, 381−402.

(11) Zhu, T.; Cao, S.; Su, P.; Patel, R.; Shah, D.; Chokshi, H.; Szukala, R.; Johnson, M.; Hevener, K. E. Hit Identification and Optimization in Virtual Screening: Practical Recommendations Based Upon a Critical Literature Analysis. *J. Med. Chem.* **2014**, *56*, 6560−6572.

(12) Vogt, M.; Jasial, S.; Bajorath, J. Extracting Compound Profiling Matrices from Screening Data. *ACS Omega* **2018**, *3*, 4706−4712.

(13) Tanrikulu, Y.; Kondru, R.; Schneider, G.; So, W. V.; Bitter, H.-M. Missing Value Estimation for Compound-Target Activity Data. *Mol. Inf.* **2010**, *29*, 678−684.

(14) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(15) *OEChem TK version 2.0.0*; OpenEye Scientific Software: Santa Fe, NM, 2015.

(16) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.

(17) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(18) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **2016**, *35*, 3−14.

(19) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, 2016.

(20) Nielsen, M. A. *Neural Networks and Deep Learning*; Determination Press, 2015.

(21) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer: New York, 2006.

(22) Chollet, F. Keras, version 2.1.3, 2015 https://github.com/keras-team/keras (accessed Jan. 17, 2018).

(23) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M., Yu, Y.; Zheng, X. TensorFlow: A System for Large-scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*: Savannah, GA, 2016.

(24) Bradley, A. P. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recogn.* **1997**, *30*, 1145−1159.

(25) Matthews, B. W. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochim. Biophys. Acta* **1975**, *405*, 442−451.

(26) Brodersen, K. H.; Ong, C. S.; Stephan, K. E.; Buhmann, J. M. The Balanced Accuracy and Its Posterior Distribution. *20th International Conference on Pattern Recognition*, 2010.

DOI: 10.1021/acsomega.8b01682
ACS Omega 2018, 3, 12033−12040