Predicting molecular activity on nuclear receptors by multi-task neural networks

Cecile Valsecchi[1], Magda Collarile[1], Francesca Grisoni[2], Roberto Todeschini[1], Davide Ballabio[1]*, Viviana Consonni[1]

[1] Milano Chemometrics and QSAR Research Group, University of Milano Bicocca, P.za della Scienza 1-20126, Milano, Italy

[2] ETH Zurich, Department of Chemistry and Applied Biosciences, Vladimir-Prelog-Weg 4, 8049 Zurich, Switzerland

KEYWORDS

Deep learning, QSAR, multi-task, nuclear receptors, classification, genetic algorithms

ABSTRACT

The interest in multi-task and deep learning strategies has been increasing in the last few years, in application to large and complex dataset for quantitative structure-activity relationship (QSAR) analysis. Multi-task approaches allow the simultaneous prediction of molecular properties that are related, through information sharing, while deep learning strategies increase the potential of capturing nonlinear relationships. In this work we compare the binary classification capability of multi-task deep and shallow neural

networks to single-task strategies used as benchmark (i.e., as k-Nearest Neighbours, N-Nearest Neighbours, Random Forest and Naïve Bayes), as well as multi-task supervised Self-Organizing-Maps.

Comparison was carried out with an extended QSAR dataset containing annotations of molecular binding, agonism and antagonism activity on 11 nuclear receptors, for a total of 14,963 molecules, divided into training and test sets and labelled for their bioactivity on at least one of 30 binary tasks. Additional 304 chemicals were used as external evaluation set to further validate models.

Although no approach systematically overperformed the others, task-specific differences were found, suggesting the benefit of multi-task learning for tasks that are less represented. On average, some of the single-task approaches and multi-task deep learning strategies had similar performances. However, the latter can have advantages, such as a simpler management of predictions and applicability domain assessment for future samples. On the other hand, the parameter tuning required by neural networks are generally time expensive suggesting that the modelling strategy should be evaluated case by case.

1 INTRODUCTION

Multi-task modelling implies the simultaneous learning of several related responses, or tasks[1]. These are expected to share information during this parallel tuning, so that what is modelled for a specific task can help others to be modelled better. Approaches based on deep neural networks are often associated to this type of modelling, where deep learning refers to machine learning strategies based on neural networks with multiple layers of nonlinear processing.[2] The main advantage of deep learning strategies is their potential to capture nonlinear information from big, noisy and complex datasets, thanks

to which they have become the benchmark techniques in different applications, such as image and speech recognition[3–5]. Deep learning methods distinguish from the so-called "shallow" networks, composed of only one processing layer[2].

Quantitative Structure Activity Relationship (QSAR) exploits statistical and mathematical strategies to quantitatively relate a biological or physicochemical property to the molecular structure, numerically encoded within the so-called molecular descriptors[6]. Several recent QSAR studies have shown that deep learning approaches often outperform traditional machine learning approaches both in regression and classification. In particular, deep neural networks have proved to be a valuable tool in drug design and virtual screening[7–13].

The interest in simultaneously modelling more than one biological properties (referred to 'tasks'), has been increasing in the QSAR field[14–16]. However, the issue deriving from predicting multiple responses implies the use of advanced algorithms[17]. The standard approach in machine learning is to learn one task at a time (i.e., single-task modelling), while multi-task learning assumes that training a unique model simultaneously on multiple related tasks allows to process together all the available information, which can help to learn also very difficult tasks[18]. In particular, multi-task deep neural networks allow to obtain information not only from the multiple hidden layers, but also from a shared internal representation deriving from the multiple related tasks[1,15].

Despite the increasing use of deep neural networks in several scientific fields, their superiority to classical models in QSAR is still under debate. While some scientists recommend simpler models for specific applications (e.g., estrogen receptor binding and acute toxicity prediction)[19–21], other studies have reported a statistically significant improvement in performance compared to classical approaches (even if minor in absolute terms)[22,23]. Multi-task neural networks have shown in different QSAR studies to outperform single-task models[14,22,24,25]. The doubtless advantage of the multi-task

approach, which calculates only one model for several tasks, is that it is cheaper in terms of computational requirements than the traditional single-task QSAR modelling, which implies the calculation of as many models as the tasks. In addition, in multi-task modelling, under-represented tasks can benefit from implicit data augmentation and, thus, gain higher performance[1].

In this work, we evaluated advantages and limitations of multi-task neural networks (both based on deep and "shallow" neural networks) in comparison with benchmark single-task approaches, such as Random Forest (RF)[26], k-Nearest Neighbours (kNN)[27], N-Nearest Neighbours (N3)[28] and Nàive Bayes (NB)[29]. Additionally, for the comparison, we introduce a simple multi-task strategy based on a supervised Self-Organizing Map, named XYF[30], which was adapted to handle the multi-task problem. Comparison was performed in order to verify whether the predictive performance on binary classification can justify the use of complex multi-task approaches based on deep and shallow neural networks.

The comparison was carried out with an extended dataset comprising 14,963 molecules labelled for their bioactivity on at least one of 30 binary tasks representing agonism, antagonism or binding (in the form "active"/"inactive") towards 11 nuclear receptors[31].

Molecules were randomly divided in training and test sets. We optimized each model separately in cross-validation through protocols based on grid search, while genetic algorithms were used to tune multi-task neural networks parameters. All the approaches were finally evaluated on the test molecules and with an additional evaluation set of 304 unseen chemicals, considering both classification measures on each task and on the whole set of chemicals.

2 MATERIALS AND METHODS

2.1 Data

2.1.1. NURA dataset

In this study we used a recently published and publicly available dataset (NURA – NUclear Receptor Activity dataset)[31], containing information on nuclear receptor modulation by small molecules. Nuclear receptors (NRs) are a superfamily of transcription factors that control several physiological functions in the human body, such as cell growth, development, homeostasis and metabolism[32,33]. Chemicals can bind to nuclear receptors by activating (agonist) or inhibiting (antagonist) the natural biological response. A binder can, thus, be (i) an agonist, i.e. it can activate the receptor by inducing a physiological response similar to that induced by the naturally occurring physiological ligand, or (ii) an antagonist, by binding to the receptor without activating it and preventing or blocking the action of the natural ligand. While most compounds bind in the same pocket of the natural ligand (orthosteric modulators), others modulate the receptor's action in a non-competitive manner, by binding at a different site than the endogenous ligand (allosteric modulators). These effects can be measured by experimental assay as the half maximal effect (EC_{50}) or inhibition (IC_{50}) concentration of the tested chemical. The NURA dataset contains annotations for binding, agonism and antagonism activity for 15,206 molecules and 11 selected NRs (androgen receptor [AR], estrogen receptor alpha [ERα], estrogen receptor beta [ERβ], progesterone receptor [PR], glucocorticoid receptor [GR], peroxisome proliferator-activated receptor alpha [PPARα], peroxisome proliferator-activated receptor delta [PPARδ], peroxisome proliferator-activated receptor gamma [PPARγ], pregnane X receptor [PXR], retinoid X receptor [RXR], and farnesoid X receptor [FXR]). For each receptor and each activity type (agonism, binding, antagonism), a molecule can have one of the following annotations: (i) "active" if the annotated EC_{50} or IC_{50} is lower than 10 μM; (ii) "weakly active", for EC_{50} or IC_{50} values between 10 μM and 100 μM, (iii) "inactive", for EC_{50} or IC_{50} > 100 μM, (iv) "inconclusive",

highlighting the lack of a final bioactivity assessment, (v) “missing”, in case no bioactivity was reported.

2.1.2. Data curing and molecular descriptors

In this work, each bioactivity type for a given receptor (i.e. binding, agonism or antagonism) was considered as a task (e.g. binding activity for androgen receptor), obtaining a total of 33 tasks. Only active and inactive annotations were considered and tasks containing such annotations for less than 200 molecules were discarded (*i.e.*, antagonism for PPARα, PXR and RXR). The considered dataset is therefore composed of a total of 14,963 chemicals annotated (as active or inactive) for at least one of the selected 30 tasks (Table 1).

Molecules were randomly split into training set (11,970 molecules, 80%) and test set (2,993 molecules, 20%), preserving the proportion between the two classes (actives/inactives) for each task (stratified splitting). The number of molecules for each task and the activity distributions among the tasks are shown in Table 1. For each molecule, we computed extended connectivity fingerprints (ECFPs)[34] as input variables. ECFPs are binary vectors of predefined length which encode the presence/absence of atom-centred substructures through a hashing algorithm. ECFPs were computed using the software Dragon 7[35] with the following options: 1024 as the fingerprint length, two bits to encode each substructure, a fragment radius comprised between 0 and two bonds and the Dragon defaults (Count fragments = True, Atom Options: [Atom type, Aromaticity, Connectivity total, Charge, Bond order])[34].

ECFPs and labels of experimental activity for the 30 tasks for the training and test molecules can be downloaded at the Milano and Chemometrics QSAR Research Group website[36].

2.1.3. External evaluation set

To further evaluate the model predictivity, we collected an additional set of chemicals, hereinafter referred to as evaluation set. Chemicals were retrieved from the latest available release of ChEMBL database (26, released on 3rd March 2020), so that: (i) they were not included in the training or test set, (ii) they had an experimental annotation on at least one of the tasks of interest. The retrieved molecules were curated following the same pipeline of the training set and test chemicals and they were labelled for their bioactivity as in the NURA dataset[31]. Since 97.8% of the chemicals were active, the inactivity data (10 molecules) were considered as not numerous enough to have a reliable estimate of classification accuracy and were thus excluded. The evaluation set was composed of 304 molecules with 435 'active' labels for 21 tasks, as reported in the last column of Table 1. The full list of chemicals included in the evaluation set is available as supplementary material of this manuscript (Table S1).

2.2 Multi-task learning

2.2.1 Feedforward neural networks

Multi-task networks usually are constituted by fully connected neural network layers trained on joint tasks, where the output is shared among all learning tasks and then fed into individual classifiers[14,16]. When some dependence relationships exist among the tasks, the model should learn a joint representation of these tasks[37]. As in single-task feedforward neural networks, the input vectors are mapped to the output vectors with repeated compositions of simpler modules called layers, which are constituted by neurons. When each neuron of a layer is connected to all the neurons of the following layer, the network is called dense or fully connected from input to output layer. The layers between input and output are called hidden layers. Each connection represents a weight, while each node represents a learning function *f* that, in the feedforward phase, processes the information of the previous layer to be fed into the subsequent layer. In

the backpropagation phase, each weight is adjusted according to the loss function and the optimization algorithm[18].

Different types of learning or activation functions exist in literature; the most known functions are sigmoid (σ), REctified Linear Unit (ReLu), hyperbolic tangent (tanh) and leaky ReLu[38–40]. To iteratively adjust the weights, a loss function is computed considering the experimental and the predicted response. Neural network tuning implies setting a learning rate that determines the update of the weights in each iteration with respect to the gradient of the loss function; this parameter can be fixed or changed during the learning (e.g. exponential decay). Furthermore, several strategies called regularization techniques can improve the network's generalizing ability and reduce overfitting. This is the case of dropout and weight decay (L1 or L2 regularization).

In this work, we used the cross-entropy as loss function[41]; it can handle multiple outputs also in the case of some missing data[42]. We considered both 'shallow' (i.e., only one hidden layer) and deep architectures, with hidden layers varying from 2 to 3, and neurons per layer varying between 5 and 100. No "deeper" networks were used, as preliminary results have shown that increased network complexity (from four to ten layers) did not improve the classification performance. We used multi-task networks with and without a bypass net, which is an independent additional layer that 'bypass' shared layers to directly connect inputs with outputs leading to more robust results, as schematized in Figure 1A. The output layer consists of as many nodes as tasks (i.e., 30). The threshold of assignment for the output nodes were optimized on the basis of ROC curves[43] and set to 0.05 for almost all nodes, that is, if the output of the neural network ensemble node is equal or lower than 0.05 the compound is predicted inactive, otherwise active. Only for PPARα binding and agonism nodes the threshold was set to 0.95, being these tasks extremely unbalanced towards active class (Table 1). In Figure S2 an example of ROC curve is shown.

We initialized the network weights randomly according to a truncated normal function and set the number of epochs to 10. To minimize the effect of the random initialization of weights and reduce the potential variance of classification results, we considered the median of the output of an ensemble of five independently trained neural networks[44,45]. To optimize the architecture, i.e. number of layers and nodes per layer, the learning rate, the effect of regularization and the activation function, we applied a genetic algorithm strategy, as further explained in the following optimization section.

2.2.2 Modified XYF networks

Shallow and deep networks were compared to a simpler multi-task neural network, which is the XY-Fused (XYF) network[30]. This is a supervised Self-Organizing Map[46], which we adapted in this work to perform multi-task classification, *i.e.,* simultaneously provide a prediction for more than one task.

A multi-task XYF network is composed of two layers (Kohonen and output); each neuron has a number of Kohonen weights corresponding to the number of input variables (1024 in the case of the ECFPs used in this study) and a number of output weights equal to the number of tasks to be modelled (30 in this case study). A schematic representation of the XYF network architecture is shown in Figure 1B.

Given the *i*-th training molecule, during the learning phase the winner is that neuron associated to the minimum fused distance (d_f) from the *i*-th molecule; d_f is basically a weighted average of the Kohonen ($d_{Kohonen}$) and output (d_{output}) distances:

$$d_f(i,k) = \alpha \cdot d_{Kohonen}(i,k) + (1-\alpha) \cdot d_{output}(i,k) \qquad [1]$$

where d_f is the fused distance measure between the *i*-th molecule and the *k*-th neuron, $d_{Kohonen}$ is the Euclidean distance between the feature vector of the *i*-th molecule (ECFPs) and the weights of the *k*-th neuron in the Kohonen layer, d_{output} is the Euclidean distance between the task vector of the *i*-th molecule and the weights of the *k*-th neuron in the

output layer, α is the tuning parameter with values in the range between 0 and 1. In the traditional XYF network, *α* varies as the number of epochs changes (the epoch being the number of iterations required to process the whole dataset). Herein, we decided to set *α* independent of the number of epochs and its optimal value was determined by cross-validation.

Once the winning neuron has been selected, the weights of the winning neuron and its neighbours are updated by means of the well-known rule of SOM's training[47] up to a certain neighbourhood; the procedure is repeated for every molecule of the dataset and the dataset is processed several times (i.e., epochs). Finally, classification of molecules with respect to each single-task can be carried out on the basis of the output weights of the winning neuron by comparison with a specific assignment threshold, which was optimized on the basis of a ROC approach[43]. A 3-fold cross-validation grid search protocol was used to contemporaneously set the optimal size of the XYF network (toroidal square map with 15 x 15 neurons) and the tuning parameter α (0.9).

2.3 Single-task benchmark models

We compared multi-task models to four classical single-task machine learning approaches:

1. k-Nearest Neighbours (kNN)[27,48] is a local classification method that assigns a target molecule to the most represented class among the *k* most structurally similar chemicals of the training set. In this work, the *k* most similar chemicals (i.e., the neighbours) to a molecule to predict were identified using the Jaccard-Tanimoto [49] coefficient. For each task, we optimized the number of neighbours *k* ($k \in [1,10]$) using a 5-fold cross-validation protocol.

2. N-Nearest Neighbours (N3)[28] is an alternative local classification method, which takes into account all the training molecules to classify a target chemical, instead of considering only *k* neighbours; the contribution of training chemicals to class assignment becomes exponentially less important as the similarity to the target decreases and is modulated by the parameter γ. In this work, the parameter γ was optimized in the range 0.25 to 2.5 (step equal to 0.25) by a 5-fold cross-validation protocol for each task.

3. Naïve Bayes (NB)[29] is a conditional probability approach based on Bayesian posterior probabilities. These are derived from the maximum likelihood probabilities[50], which are calculated for each bit of the fingerprint vector according to the frequency of active/inactive compounds for the current bit. The target molecule is then assigned to the class associated with the highest posterior probability.

4. Random Forest[26] is an extension of decision tree algorithms, which uses an ensemble of trees. Each classification tree is calibrated on a subset of molecules of the training set, selected by the bootstrapping technique (i.e., random sampling with replacement), with the aim to minimize the node's impurity. The class assignment occurs according to the majority vote on the trees of the forest. In this work, the number of trees (1, 10, 100), the class weights (from 0.01 to 1, step equal to 0.01), the split criterion (Gini's diversity index, twoing rule, or cross entropy), the split prediction algorithm, the prior probability for each class (weighted or uniform) and the pruning criterion (impurity or error-based) were optimized for each task through a 5-fold cross-validation grid search approach.

2.4 Defining the model applicability domain

To ensure the reliability of the predictions, we determined an applicability domain (AD), which can be defined as the region of the chemical space where predictions are obtained

by model interpolation and thus are associated with higher confidence[51]. The general principle to define the model applicability domain is that the reliable predictions are limited to the chemicals that are structurally similar to the ones used to build that model. In this study, we applied a previously published approach[51], which is based on a set of local thresholds corresponding to the training data points and defining the width of their neighbourhood. For each *i*-th training molecule, the associated threshold t_i was calculated as the average Jaccard-Tanimoto distance on ECFPs to the first k_i neighbours, the number k_i being variable and depending on the sample density in the chemical space.

If a test molecule exceeds the threshold of all the training molecules, then it is considered as outside the AD and its prediction is considered as unreliable. On the contrary, if the molecule falls inside the neighbourhood of at least one training molecule, it will be considered inside the domain of applicability and associated with a reliable prediction. Therefore, given the training set TR, for each test molecule *j*, the AD decision rule is:

$$j \in \text{AD} \quad \text{iff} \quad \exists i \in \text{TR} : D_{ij} \leq t_i$$

where D_{ij} is the binary Jaccard-Tanimoto distance between the *j*-th test and the *i*-th training molecule.

2.5 Classification performance of single- and multi-task models

The model performance on each *t*-th task was quantified using sensitivity (Sn_t), specificity (Sp_t) and Non Error Rate (NER_t), defined as follows[52]:

$$Sn_t = \frac{TP_t}{TP_t + FN_t} \cdot 100 \quad Sp_t = \frac{TN_t}{TN_t + FP_t} \cdot 100 \quad NER_t = \frac{Sn_t + Sp_t}{2} \qquad [2]$$

where TP_t, TN_t, FP_t and FN_t are the number of true positive, true negative, false positive and false negative molecules for the *t*-th task. To compare the overall performance of models, "global" sensitivity, specificity and Non Error Rate measures (Sn_T, Sp_T, NER_T) were computed as follows:

$$Sn_T = \frac{\sum_{t=1}^{T} TP_t}{\sum_{t=1}^{T} TP_t + \sum_{t=1}^{T} FN_t} \cdot 100 \quad Sp_T = \frac{\sum_{t=1}^{T} TN_t}{\sum_{t=1}^{T} TN_t + \sum_{t=1}^{T} FP_t} \cdot 100 \quad NER_T = \frac{Sn_T + Sp_T}{2} \quad [3]$$

where *t* runs over each task, and *T* is the total number of tasks (30 in our case). Sn_T and Sp_T represent the percentage of active and inactive molecules correctly predicted over all tasks, respectively.

2.6 Optimization of the model parameters

For the classification approaches that require the selection of a limited number of parameters (i.e., XYF networks, kNN, N3 and Random Forest), we carried out a grid search optimization by training models for each of all of the possible combinations of the selected parameters. Then, we selected the parameter combination that led to the best classification performance in cross-validation (i.e., the maximum NER_t and NER_T for single and multi-task approaches, respectively).

Since multi-task feedforward neural networks require the tuning of nine parameters, we used a genetic algorithm (GA) strategy[53] as a more computationally-efficient approach. GAs are an evolutionary optimization technique inspired by the principles of genetics and natural selection, which can be used for variable selection and optimization of parameters, particularly for highly convoluted response surfaces[54,55].

We adapted the GA approach to tune the neural network architecture and parameters and find an optimal solution. Each chromosome represented one possible set of network

architecture/training parameters, each chromosome being constituted by 13 binary genes that encoded the values of the following parameters to be optimised:

- three genes were used to codify eight possible network architectures: three shallow with one hidden layer each of 10, 100 or 1000 neurons and five deep with two hidden layers, i.e. (10,10), (10, 5), (1000, 100), (100, 100) or three hidden layers, i.e. (1000, 100, 10);
- two genes encoded four learning rate values (0.0005, 0.001, 0.01, 0.025);
- two genes encoded four different activation functions (ReLu, leaky ReLu, Sigmoid and Tanh) applied to all the neurons except for those of the output layer, which were always sigmoid nodes;
- one gene was used to encode the different types of optimization algorithm (Adam or gradient descend);
- five genes were used for the binary parameters (presence/absence of a weight decay of 0.01; L1/L2 regularization; presence/absence of a dropout of 0.5; presence/absence of a bypass net of 100 neurons; presence/absence of the exponential decay).

A population of 20 randomly initialized chromosomes was created. This population was then evolved for eight generations with the following steps:

1. Evaluate the 20 chromosomes in the population; each chromosome corresponds to a specific neural network which is evaluated on the basis of its classification performance (NER_T) with a three-fold cross validation protocol;
2. Rank the chromosomes from the best to the worst according to NER_T in cross-validation (the higher, the better);
3. Select two chromosomes (i.e., parents) through the roulette wheel algorithm[56], from which a new chromosome (i.e., child) with a mutation probability of 0.1 is generated;

4. Repeat step no. 3 other four times to generate a total of five children;
5. Calculate the neural networks with the specific parameters encoded by the children chromosomes and evaluate their classification performance (NER_T in cross-validation);
6. Resort all the chromosomes (i.e. parents and children) according to their NER_T;
7. For the best 20 chromosomes repeat from step no. 3 to step no. 7 for the number of generations.

After five generations, we caused an "invasion" into the current population, replacing the worst 10 chromosomes by new random chromosomes[57]. At the end of the eighth generation, the top ranked chromosome was selected.

As proposed elsewhere[58], the GA evolution process was independently repeated 20 times, to obtain 20 top-ranked chromosomes which resulted to be unique. Finally, the population of the 20 top-ranked chromosomes was further evolved for 250 generations. During this evolution phase, all the generated chromosomes were retained, providing a final population of 1515 chromosomes.

2.7 Software

Fingerprint calculation was performed using Dragon 7 software[35] with default settings (with 2 bits per fragment and an atom-centred radius from 0 to 2 bonds (Count fragments = True, Atom Options: [Atom type, Aromaticity, Connectivity total, Charge, Bond order]). Single-task models and XYF network were calculated in MATLAB[59] by means of in-house scripts. Published and freely-accessible MATLAB code for PCA[60], N3[28], and NB, KNN and RF[61] was used, as available on Milano Chemometrics website[62,63]. Multi-task neural networks were built by means of the "Deepchem 2.1.1" package[64] in Python v3.6[65] and the 'RobustMultitaskModel' function with a Tensorflow 1.14.0 backend. Wilcoxon signed-ranked test was performed in Python v3.6[65] using the SciPy library[66].

3 RESULTS

3.1 GA-based network tuning

Genetic Algorithms (GA) led to 1515 different combinations of parameters (number of layers and number of neurons per layer, learning rate, optimization algorithm, activation function, regularization and bypass layer) for the feedforward neural networks. This set contains all the solutions found by the GA approach. The obtained population of chromosomes was used to evaluate the influence of the parameters on the classification performance of the feedforward neural networks. To this end, the 1515 chromosomes were ranked on the basis of their fitness function (i.e., NER_T in 3-fold cross-validation) and divided into 10 intervals based on the deciles of NER_T. The relative frequency of each parameter in each decile was calculated (Figure 2).

88.5% of the chromosomes included in the highest NER_T decile (D10) have learning rate equal to 0.001, while 5.8% have learning rate of 0.01 and 0.0005, and none of them have learning rate equal to 0.025. Additionally to the observed optimal learning rate (0.001), other settings have a frequent occurrence among the best chromosomes, such as (1) Adam optimisation (frequency larger than 90% in the best five deciles and equal to 100% in D8, D9 and D10) and (2) no exponential decay (always absent in the models belonging to the best 6 deciles).

To get additional insights into the relationship between parameters and classification performance, we carried out a Principal Component Analysis (PCA)[67] on the relative frequencies depicted in Figure 2. We normalized the values, dividing each relative frequency by the maximum relative frequency of each parameter. Then, we performed a PCA on the transposed matrix, using the 10 deciles as the rows and the 28 parameter values as the columns. The first two principal components (Figure 3) capture 57% of the data variance, thus providing a good overview on the relationship between network

parameters and model performance. In fact, the first component (PC1) captures the variation of NER_T among the deciles and, thus, higher PC1 scores correspond to better average classification performance.

PC1 confirms the previous considerations drawn from the numerical analysis of the relative frequencies of Figure 2: learning rate equal to 0.001 (LR01), Adam optimization algorithm (OptA) and no exponential decay (ED0) are related to the highest deciles (D8, D9 and D10), that is, these settings frequently appear in the best models. On the opposite, exponential decay (ED1) and gradient descent optimization (OptGD), with the lowest loadings on the first component, are mainly related to the worst deciles (D1 and D2, where NER_T decreases down to 50%). As far as what is captured by the PCA, weight decay (WD) and dropout (Drp) seem to not affect NER_T, having PC1 loadings between -0.1 and 0.1. The type of activation function results to have a moderate influence on the classification performances, with ReLu (AFr) being the one mostly related to high NER_T values.

In order to select the best performing parameter combination, we then selected the settings associated to the highest PC1 loadings for each parameter, that is, a multi-task neural network (FFNL1) constituted by one hidden layer of 100 neurons (Arch1) with a ReLu activation function (AFr), a bypass net (Bp1), Adam optimization algorithm (OptA), learning rate of 0.001 (LR01) and no regularization (no weight decay, WD0, no dropout, Drp0, and no exponential decay, ED0). This solution is associated with a NER_T in cross-validation equal to 90.6%.

FFNL1 can be considered as a 'shallow' neural network, as it has only one processing layer. To compare the performance of shallow and deep multi-task neural networks, we also considered the best architecture selected by GA having three layers (Arch2, the deepest one). We set the three most relevant parameters to their optimal values as determined by the PCA (LR01, OptA, ED0), and searched for the best combination of

the remaining parameters. The best obtained result (NER_T in cross-validation equal to 90.4%) corresponds to three hidden layers (1000,100,10), leaky ReLu as the activation function (AFl), no bypass net (Bp0), Adam optimization algorithm (OptA), learning rate of 0.001 (LR01), no dropout (Drp0), weight decay type L2 of 0.001 (WD1) and no exponential decay (ED0). This model will be hereinafter referred to as model FFNL3.

3.2 Model comparison on the test set

The optimized single and multi-task models (Table S2) were used to predict the bioactivity of the test set compounds. Only test set compounds within the AD (2970 out of 2993, corresponding to 99.2% of the total) were predicted.

3.2.1 Comparison on individual tasks

Table 2 collects the classification performance of all the models on the test set for each task (expressed as NER_t). The last row of Table 2 collects the average NER_t achieved by each modelling method. All methods could, on average, correctly classify the majority of the test chemicals: the average NER_t is higher than 85% for all methods, with the lowest average NER_t equal to 85.7% (NB) and 86.7% (XYF), and the highest equal to 90.4% (N3), 90.1% (FFNL1) and 89.2% (FFNL3). When comparing the NER_t achieved on the training set (supplementary material, Table S3) and test set (Table 2), the average difference is equal to 9% and 10% for FFNL1 and FFNL3, respectively. Considering these slight differences between fitting and prediction performance, potential presence of overfitting can be excluded.

To provide a graphical representation of the model performance, we carried out a Principal Component Analysis (PCA) by considering the seven classification approaches as the samples (rows) and their NER_t for the 30 tasks as the variables. To aid in the comparison, we added two theoretical benchmarks, consisting of the maximum ('B', best)

and minimum ('W', worst) NER_t values achieved on each task, respectively (Figure 4a). The obtained first component (PC1) is related to the overall predictive capability of the classification methods, since the artificial points 'B' and 'W' have the lowest and highest PC1 scores, respectively.

Multi-task deep (FFNL3) and shallow (FFNL1) neural networks, together with N3, kNN and Random Forest (RF) appear clustered and close to the best point ('B'), indicating their tendency to provide good overall classification. XYF has an intermediate score on PC1, indicating a moderate classification quality. Naive Bayes (NB) shows the worst average performance, as its PC1 score is the highest. These results resemble the average performance shown in the last row of Table 2.

The second component (PC2) explains the different behaviour of kNN and RF from the other best performing methods (N3, FFNL1, FFNL3), which mainly depends on the low NER_t on the six tasks with the lowest negative loadings on PC2 (RXR agonism, ERα antagonism, PXR agonism, PXR binding, FXR antagonism and PPARγ antagonism). These tasks have a remarkably low number of active chemicals (lower than 6%, Figure 4b) and kNN and RF provide a suboptimal performance, as can be seen from their low sensitivities (Sn_t equal to 72% for ERα antagonism and lower than 58% on the other five tasks, Figure 4c and Table S4). On the contrary, the multi-task feedforward models (FFNL1 and FFNL3) provided the highest NER_t in five out of six cases (RXR agonism, ERα antagonism, PXR agonism, PXR binding and FXR antagonism). In particular, the sensitivity values achieved by FFNL1 and FFNL3, together with N3, on the PXR binding (82.5%, 75.0% and 76.2%, respectively, Table S4) are remarkably higher than those of the majority of other approaches (XYF, kNN and RF have sensitivity lower than 58% for binding). These tasks share a remarkable number of active chemicals with other tasks (75%, 75%, 23% and 23% for ERα antagonism, FXR antagonism, PXR binding and agonism, respectively)[31], potentially suggesting the benefit of multi-task models, where

simultaneous learning can help less represented tasks to be better modelled by exploiting available data from the other tasks.

Looking at individual tasks, there is no approach that always clearly outperforms the others; all of the methods converge to similar performance on easy-to-model tasks. This is the case, for instance, of binding and agonism on PPARδ, for which all classification approaches reach NER_t higher than 95%. The same consideration holds for the discrimination of active or inactive chemicals; for example, all approaches correctly classify more than 91% of active chemicals for FXR binding and more than 94% for FXR agonism (Table S4).

On the contrary tasks associated with lower NER_t values are characterised by higher variation in the results, depending on the adopted modelling approach. For example, N3 and NB achieved considerably higher sensitivity values (86.4% and 72.7% respectively, Figure 4c and Table S4) and higher overall classification performances for PPARγ antagonism (NER_t equal to 77.1% and 74.6%, respectively) than the other methods (NER_t equal or lower than 68.8%).

Deep multi-task neural networks (FFNL3) have an unsatisfactory performance on PPARα agonism (NER_t = 49%). This is the only task, together with PPARα binding, with more than 98% of active molecules, explaining the poor performance of all models when classifying inactive chemicals for PPARα (Sp_t equal to 66.7% for XYF and lower than 50% for all other methods, Figure 4d and Table S4). For all other tasks, the methods tend to better classify inactive chemicals, with specificity (Figure 4d) comparable to or higher than sensitivity (Figure 4c), as expected due to the generally higher number of the inactive compounds.

Finally, for at least three tasks (PPARδ, PPARγ and FXR antagonism), N3 shows an increased sensitivity at the expense of specificity, which slightly decreases compared to other models. This is apparent for example when modelling the PPARδ antagonism, for

which only N3 provides an acceptable sensitivity value (71.4%). The tendency of N3 to favour the less-numerous classes has been already observed,[61] due to the algorithm normalization over the number of the utilized neighbours belonging to a given class for computing the prediction [28].

3.2.2 Model comparison by Wilcoxon signed-rank test

Since the average performance on the tasks of all the models is similar (with a difference of about 5% between the highest and lowest average NER_t, see the last row of Table 2), we verified the statistical significance of the observed differences by a Wilcoxon signed-rank test taking into account all possible pairs of models and separately considering Sn_t, Sp_t and NER_t. The test returned a decision (and an associated p-value) for the null hypothesis that the median rank difference between model performance on all the tasks is zero, that is, for each pair of models and for each classification measure, we tested whether no significant differences (p-value > 0.05) existed. The Wilcoxon signed-rank test results, expressed as p-values, are shown in Figure 5. Considering the models which clustered in the PCA score plot (Figure 4a), we can conclude that no statistically significant differences were detected between N3 and FFNL1, FFNL1 and FFNL3, N3 and FFNL3. When comparing FFNL1 and FFNL3 to kNN, no statistically significant difference on NER_t values was detected, but significant differences in specificity and sensitivity results were found, thus indicating a different behaviour of these methods when predicting active or inactive compounds. Finally, both deep and shallow networks demonstrated to have significantly better classification performance than NB in terms of NER_t, sensitivity and specificity.

3.2.3 Comparison on global performance

Multi-task and single-task approaches were also compared on the basis of the global classification performance on the test set, as quantified by Sn_T, Sp_T and NER_T (Table 3). Unlike task-specific indices, global measures give an insight into the overall model classification capability, regardless of the performance on each individual task. These metrics, in fact, represent the percentage of correctly predicted active (Sn_T) and inactive (Sp_T) chemicals over the whole dataset. Global performances are thus intrinsically biased towards the task cardinality (i.e., number of molecules with annotation in a given task). The higher the number of molecules annotated as active or inactive for a given task, the higher the task influence on the computed overall metrics.

FFNL1 and kNN provided the highest classification capability with NER_T equal to 95.3%; however, FFNL3, RF and N3 achieved very similar performance (95.2%, 95.2%, 94.2%,). All the approaches were able to discriminate active and inactive molecules well (high sensitivity and specificity). Inactive chemicals were better predicted than the active ones, as seen from the specificity values that are generally higher than the sensitivity ones, with the only exception for N3, NB, FFNL1 and FFNL3. For these methods, similar ability to classify active and inactive molecules was observed. kNN and RF reached the highest Sp_T (around 99%), indicating an optimal ability to correctly predict inactive compounds. On the contrary, FFNL1 and FFNL3 achieved the highest sensitivity (Sn_T around 95%), which indicates a good ability to classify active compounds.

3.3 Performance on the evaluation set

Both single-task and multi-task approaches were further tested on the evaluation set molecules. All models were re-fitted on the whole set of training and test molecules (with the previously optimized parameters) and used to predict the 304 active molecules of the external evaluation set. Only four chemicals out of 304 resulted out of the applicability domain and were excluded from the evaluation. Since the evaluation set contains only

active molecules for 21 tasks, with several tasks represented by few molecules (9 tasks with less than 10 molecules, Table 1), only the global sensitivity Sn_T was considered as measure for performance comparison. When looking at the predictions on these compounds (Table 3), RF, XYF and kNN underperformed the other models, having Sn_T ranging from 68.9% (kNN) to 74.9% (RF). FFNL1 has the highest sensitivity (Sn_T = 93.0%). Shallow (FFNL1) multi-task neural networks, together with Naive Bayes (NB), has comparable Sn_T values on the test and evaluation sets (Table 3), with a difference lower than 3%. On the contrary, several discrepancies can be noticed especially for similarity-based approaches, whose Sn_T decreased from 91.7% to 68.9% for kNN, from 94.3% to 81.7% for N3 and from 89.0% to 71.5% for XYF. This might be due to the considered AD approach, which is based on the structural similarity of a target molecule to all the dataset chemicals and does not account only for the chemicals with annotated response for an individual task. This AD method was chosen to have a unique approach regardless of the tasks and modelling algorithms, in order to enhance comparability between single and multi-task models. In our opinion, this issue could mainly affect similarity-based approaches and might explain the observed discrepancies in the sensitivities on the test and evaluation set.

To visualize the differences between the misclassifications of each model, we performed a non-classical multi-dimensional scaling (MDS)[68] on the chemicals of the evaluation set. MDS was calculated on the distance matrix collecting the Jaccard-Tanimoto distances between all the possible pairs of 300 molecules, numerically described as ECFPs fingerprints (Figure 6). Figure 6 shows the obtained two-dimensional MDS, where each chemical is coloured according to the fraction of correct predictions among all annotated tasks. White points indicate molecules of the evaluation set associated only to correct predictions, while black points represent chemicals with all wrong predictions over the

available tasks. For an analogous plot performed on the test set see the supplementary material (Figure S1).

Figure 6 shows that mispredictions are clustered in the chemical space, thus indicating that each modelling method can fail on specific chemical families (darker regions). For example, Imidazo [4,5-c] pyridine derivatives, which were mainly collected from the same scientific study on PPARγ,[69] are grouped in cluster A (Figure 6). The majority of these chemicals are wrongly predicted by N3, kNN, XYF and FFNL3, while FFNL1 and NB provide much better results. Similar considerations can be drawn for the clusters B and D, for which only a few models (e.g. NB and N3) give satisfactory results. NB, despite its general good performance on the whole chemical space, is the only model providing inaccurate predictions for molecules in cluster C. However, these limitations could be mitigated by averaging predictions of models with the application of consensus strategies[70]. On the contrary, groups of chemicals correctly predicted by all models are visible. For example, the molecules of cluster E (Figure 6), which includes compounds annotated for up to six tasks,[71] were correctly predicted by all models. This highlights a certain convergence of the structure-activity relationships captured by all the analysed models when ECFPs are used to describe the molecules.

4 CONCLUSIONS

We compared the classification performance of multi-task deep and shallow neural networks with that of classical multi-task XYF networks and single-task benchmark classification approaches in a QSAR perspective.

The comparison was carried out on 14,963 chemicals, annotated with agonism, antagonism and binding activity for 11 nuclear receptors (i.e., 30 tasks), which were divided in training and test sets. Moreover, an additional evaluation set including 304

chemicals was collected and used to further evaluate the predictive capability of the models. All models were initially optimized and, in particular, we carried out tuning of multi-task neural networks by means of an ad-hoc approach based on genetic algorithms and frequency-based selection. This analysis highlighted that the type of optimisation algorithm, the learning rate and its exponential decay are the network parameters that most affected the overall classification performance.

All approaches achieved a good classification performance on both test and evaluation chemicals. For the considered data, when comparing classical single-task and advanced multi-task networks, results were mainly comparable in terms of average predictive performance, expressed through multiple classification measures, despite some task-dependent exceptions. Deep and shallow feedforward neural networks achieved on average the highest classification performance, which, however, was often only slightly better than those of the other methods and not always significantly better.

Based on the results of this study, no method clearly outperforming all the others was found. The single-task approaches considered in this work have the advantage of avoiding the optimisation of several parameters, thus being less computationally demanding than feedforward neural networks. Thus, we recommend using traditional single-task QSAR approaches when only a few molecular properties have to be predicted. However, when many tasks have to be modelled contemporaneously (like the 30 tasks modelled in this study), multi-task approaches might offer several advantages, such as (1) the possibility of leveraging information on related tasks and (2) modelling less represented tasks. Ideally, based on these considerations, multi-task models could represent a solution to identify selective compounds on desired and less-represented biological targets. In addition, having a unique model can facilitate several desirable aspects of machine learning in chemistry, such as (1) the applicability domain definition,

(b) the development of a 'joint' model interpretation for the problem under analysis, therefore enabling a better mechanistic understanding.

Finally, our study highlighted the difference in structure-activity relationships captured by each approach, underscoring potential benefits of consensus modelling integrating single- and multi-task approaches to increase classification performance.

Figure captions

Figure 1. Schematic representation of the considered multi-task models. (A) Multi-task feedforward neural network with bypass layer; input vector is mapped to output layer with repeated compositions of hidden layers, a bypass layer connects directly the input with each task-specific sigmoid neuron in the output layer. (B) Multi-task XYF network; for each input vector composed of ECFP and tasks, the neuron with the minimum distance is selected (winning neuron) and its weights and those of the neighbouring neurons are updated in a radial manner.

Figure 2. Relative frequency (%) of network parameters in the chromosomes of the final GA population; for each NER_T–based decile (D1,..., D10), the relative frequency of architecture and training parameters is reported and coloured according to a red (minimum) - green (maximum) scale. The first two rows report mean and minimum-maximum values of NER_T (%) for each decile. *The relative frequencies of weight decay type (L1, L2) were calculated considering only the chromosomes with weight decay equal to 0.01 (WD1).

Figure 3. PCA biplot of relative frequencies of network parameters. Scores (the deciles: D1, D2,..,D10) are coloured according to the average NER_T. The loading labels (networks parameters) are reported in Figure 2. FFNL1 and FFNL3 stand for the two multi-task feedforward neural networks whose selected parameter combinations are highlighted in red and blue, respectively.

Figure 4. Analysis of the classification performance on the individual tasks. (a) Score plot of PCA on the task-specific classification performances, expressed as NER_t, of all the considered models; B and W represent the theoretical best and worst performance, respectively; multi-task and single-task models are represented by green and red circles, respectively. (b) PCA loading plot; each circle represents a task and its size is proportional to the percentage of active chemicals. Radar plots of (c) sensitivity (Sn_t) and (d) specificity (Sp_t) achieved on test molecules for each task.

Figure 5. P-values of the paired Wilcoxon signed-rank test performed on each pair of classification approaches and for each performance measure (Sn_t, Sp_t and NER_t). P-values greater than 0.05, which support the evidence that the null hypothesis is true (i.e., no statistically significant median difference between model performance on all the tasks) are highlighted in bold. The couples of models with p-values always greater than 0.05 are highlighted with a grey background.

Figure 6. Multidimensional scaling calculated over the fingerprints (ECFPs) of the evaluation set (stress = 0.30). Each circle represents a molecule and its brightness is proportional to the fraction of correct predictions among all annotated tasks. Single-task and multi-task models are shown in the first and second row, respectively. The last score plot shows the number of tasks with known activities per molecule in terms of the circle's colour. Representative scaffolds of the chemical structures belonging to cluster A, B, C, D and E are reported below, where R_n indicates every possible residue.

Table 1. Dataset description: number of molecules and class distributions among the tasks for the training and test sets. The last column reports the number of active molecules in the evaluation set.

Task information			Training set		Test set		External evaluation set
Receptor	**Activity**	**Task Label**	**No. mol.**	**No. actives (%)**	**No. mol.**	**No. actives (%)**	**No. actives**
Androgen	binding	AR bind	5221	1113 (21.3%)	1328	306 (23%)	5
	agonism	AR ago	4861	409 (8.4%)	1230	104 (8.5%)	5
	antagonism	AR ant	4566	615 (13.5%)	1152	161 (14%)	22
Estrogen (α)	binding	ERα bind	4927	1031 (20.9%)	1221	256 (21%)	32
	agonism	ERα ago	4420	370 (8.4%)	1116	106 (9.5%)	7
	antagonism	ERα ant	4405	275 (6.2%)	1117	87 (7.8%)	39
Estrogen (β)	binding	ERβ bind	5370	927 (17.3%)	1343	232 (17.3%)	33
	agonism	ERβ ago	4814	226 (4.7%)	1216	60 (4.9%)	24
	antagonism	ERβ ant	4291	179 (4.2%)	1066	45 (4.2%)	17
Farnesoid X	binding	FXR bind	4627	432 (9.3%)	1195	118 (9.9%)	-
	agonism	FXR ago	4551	293 (6.4%)	1170	79 (6.8%)	40
	antagonism	FXR ant	3939	96 (2.4%)	1014	28 (2.8%)	31
Glucocorticoid	binding	GR bind	5644	1461 (25.9%)	1399	354 (25.3%)	-
	agonism	GR ago	4879	586 (12%)	1242	151 (12.2%)	2
	antagonism	GR ant	4173	518 (12.4%)	1061	139 (13.1%)	44
Peroxisome	binding	PPARα bind	991	979 (98.8%)	244	241 (98.8%)	-

proliferator-activated (α)	agonism	PPARα ago	808	796 (98.5%)	204	202 (99%)	-
Peroxisome proliferator-activated (γ)	binding	PPARγ bind	5719	1363 (23.8%)	1438	336 (23.4%)	-
	agonism	PPARγ ago	5276	1094 (20.7%)	1299	258 (19.9%)	6
	antagonism	PPARγ ant	4261	66 (1.5%)	1076	22 (2%)	4
Peroxisome proliferator-activated (δ)	binding	PPARδ bind	5165	578 (11.2%)	1307	152 (11.6%)	17
	agonism	PPARδ ago	5005	485 (9.7%)	1274	131 (10.3%)	1
	antagonism	PPARδ ant	4463	21 (0.5%)	1126	7 (0.6%)	1
Progesterone	binding	PR bind	5029	1008 (20%)	1262	243 (19.3%)	89
	agonism	PR ago	4799	291 (6.1%)	1220	58 (4.8%)	4
	antagonism	PR ant	4099	586 (14.3%)	1042	155 (14.9%)	-
Pregnane X	binding	PXR bind	3264	191 (5.9%)	835	42 (5%)	-
	agonism	PXR ago	3260	187 (5.7%)	834	41 (4.9%)	12
Retinoid X	binding	RXR bind	4352	703 (16.2%)	1078	158 (14.7%)	-
	agonism	RXR ago	3738	104 (2.8%)	941	26 (2.8%)	-

Table 2. Classification performance (NER_t) on the test set for each task. The last row collects the average NER_t of each modelling method. For each task, the best NER_t is highlighted in bold while the worst NER_t is underlined.

task	FFNL1	FFNL3	XYF	NB	N3	kNN	RF
AR bind	97.4	96.3	92.8	89.6	97.7	97.8	**97.9**
AR ago	**95.3**	93.2	90.5	89.8	93.0	93.5	94.4
AR ant	92.3	92.0	86.6	84.2	**92.6**	92.5	91.7
ERα bind	95.3	**95.9**	93.3	86.8	95.5	95.0	95.1
ERα ago	87.8	87.7	84.0	80.9	**87.9**	86.1	85.7
ERα ant	88.3	**89.8**	83.2	85.9	87.4	85.0	84.3
ERβ bind	97.6	97.7	97.0	87.0	97.8	**98.0**	97.0
ERβ ago	94.2	93.3	94.0	85.6	**95.4**	89.8	91.3
ERβ ant	**89.9**	89.5	88.3	84.7	89.6	88.1	88.5
FXR bind	96.2	95.4	94.9	91.9	**97.0**	95.2	95.4
FXR ago	97.4	98.1	96.0	94.3	**99.5**	97.1	97.3
FXR ant	**87.0**	84.4	78.8	82.9	85.6	78.3	76.7
GR bind	96.6	95.9	95.7	91.0	97.9	**98.1**	97.9
GR ago	97.0	96.6	96.5	93.9	98.1	97.9	**98.4**
GR ant	92.4	93.2	91.3	88.2	93.5	**94.0**	92.8
PPARα bind	65.8	65.8	**71.1**	64.2	65.8	66.5	66.0
PPARα ago	74.0	49.0	64.1	72.8	74.0	**74.8**	74.3

PPARδ bind	99.3	**99.4**	96.5	96.0	98.7	99.0	99.2
PPARδ ago	99.0	**99.5**	95.5	96.1	99.1	99.0	98.7
PPARδ ant	74.9	75.0	68.2	75.0	75.9	**78.6**	71.4
PPARγ bind	95.2	94.2	92.0	90.8	95.6	**96.0**	95.5
PPARγ ago	95.1	95.3	93.4	92.0	96.9	**97.1**	95.9
PPARγ ant	62.5	68.8	62.6	74.6	**77.1**	56.1	56.5
PR bind	98.4	98.1	95.6	90.6	98.9	**99.0**	98.6
PR ago	**98.9**	98.0	97.3	93.9	98.3	98.7	98.8
PR ant	**94.6**	94.4	86.8	87.8	94.0	93.3	91.5
PXR bind	**85.9**	82.9	72.5	75.9	77.5	61.7	71.4
PXR ago	**85.8**	83.8	70.7	75.6	78.3	60.7	73.3
RXR bind	94.0	94.4	95.3	91.1	95.2	95.2	**95.5**
RXR ago	74.6	**78.9**	76.1	76.8	77.9	76.8	74.8
average	90.1	89.2	86.7	85.7	**90.4**	88.0	88.2

Table 3. Global classification measures expressed as NER_T, Sn_T, *and* Sp_T (as defined in equation 3) achieved on the test set and global sensitivity Sn_T on the external evaluation set.

Model	Test set			Evaluation set
	NER_T	Sp_T	Sn_T	Sn_T
FFNL1	95.3	95.3	95.4	93.0
FFNL3	95.2	95.4	95.1	82.6
XYF	90.8	92.6	89.0	71.5
NB	89.7	90.3	89.1	89.1
N3	94.2	94.2	94.3	81.7
kNN	95.3	98.9	91.7	68.9
RF	95.2	99.2	91.2	74.9

REFERENCES

1. Caruana R. Multitask Learning. *Mach Learn*. 1997;**28**(1):41-75.
2. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;**521**(7553):436-444.
3. Cherkasov A, Muratov EN, Fourches D, et al. QSAR modeling: where have you been? Where are you going to? *J Med Chem*. 2014;**57**(12):4977-5010.
4. Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process Mag*. 2012;**29**(6):82-97.
5. Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis*. 2015;**115**(3):211-252.
6. Todeschini R, Consonni V. *Molecular Descriptors for Chemoinformatics: Volume I*. John Wiley & Sons; 2009.
7. Unterthiner T, Mayr A, Unter Klambauer G¨, et al. Deep Learning as an Opportunity in Virtual Screening. *Advances in Neural Information Processing Systems*. 2014.
8. Imrie F, Bradley AR, Van Der Schaar M, Deane CM. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J Chem Inf Model*. 2018;**58**:2319-2330.
9. Korotcov A, Tkachenko V, Russo DP, Ekins S. Comparison of Deep Learning with Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets. *Mol Pharm*. 2017;**14**(12):4462-4475.
10. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *J Chem Inf Model*. 2015;**55**(2):263-274.
11. Lenselink EB, Ten Dijke N, Bongers B, et al. Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. *J Cheminform*. 2017;**9**(1):45.
12. Gini G, Zanoli F, Gamba A, Raitano G. Could deep learning in neural networks improve the QSAR models? *SAR QSAR Environ Res*. 2019.
13. Koutsoukas A, Monaghan KJ, Li X, Huan J. Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *J Cheminform*. 2017;**9**(1). doi:10.1186/s13321-017-0226-y.
14. Ramsundar B, Kearnes S, Riley P, et al. Massively Multitask Networks for Drug Discovery. *arXiv*. 2015. https://tripod.
15. Sosnin S, Vashurina M, Withnall M, Karpov P, Fedorov M, Tetko IV. A Survey of Multi-Task Learning Methods in Chemoinformatics. *Mol Inform*. November

2018:minf.201800108.

16. Dahl GE, Jaitly N, Salakhutdinov R. Multi-task Neural Networks for QSAR Predictions. *arXiv*. June 2014. http://arxiv.org/abs/1406.1231.

17. Wold S, Sjöström M, Eriksson L. PLS-regression: a basic tool of chemometrics. *Chemometrics Intellig Lab Syst*. 2001;**58**(2):109-130.

18. Wenzel J, Matter H, Schmidt F. Predictive Multitask Deep Neural Network Models for ADME-Tox Properties: Learning from Large Data Sets. *J Chem Inf Model*. 2019;**59**(3):1253-1268.

19. Russo DP, Zorn KM, Clark AM, Zhu H, Ekins S. Comparing Multiple Machine Learning Algorithms and Metrics for Estrogen Receptor Binding Prediction. *Mol Pharm*. 2018;**15**(10):4361-4370.

20. Liu R, Wang H, Glover KP, Feasel MG, Wallqvist A. Dissecting Machine-Learning Prediction of Molecular Activity: Is an Applicability Domain Needed for Quantitative Structure--Activity Relationship Models Based on Deep Neural Networks? *J Chem Inf Model*. 2018;**59**(1):117-126.

21. Liu R, Madore M, Glover KP, Feasel MG, Wallqvist A. Assessing Deep and Shallow Learning Methods for Quantitative Prediction of Acute Chemical Toxicity. *Toxicol Sci*. 2018;**164**(2):512-526.

22. Mayr A, Klambauer G, Unterthiner T, et al. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem Sci*. 2018;**9**(24):5441-5451.

23. Rodríguez-Pérez R, Bajorath J. Multitask Machine Learning for Classifying Highly and Weakly Potent Kinase Inhibitors. *ACS Omega*. 2019;**4**(2):4367-4375.

24. Ramsundar B, Liu B, Wu Z, et al. Is Multitask Deep Learning Practical for Pharma? *J Chem Inf Model*. 2017;**57**:10.

25. Ciallella HL, Russo DP, Aleksunes LM, Grimm FA, Zhu H. Predictive modeling of estrogen receptor agonism, antagonism, and binding activities using machine- and deep-learning approaches. *Lab Invest*. August 2020. doi:10.1038/s41374-020-00477-2.

26. Breiman L, Last M, Rice J. Random Forests: Finding Quasars. *Statistical Challenges in Astronomy*. 2003:243-254.

27. Wilkinson GN, Eckert SR, Hancock TW, Mayo O. Nearest Neighbour (Nn) Analysis of Field Experiments. *J R Stat Soc Series B Stat Methodol*. 1983;**45**(2):151-178.

28. Todeschini R, Ballabio D, Cassotti M, Consonni V. N3 and BNN: Two New Similarity Based Classification Methods in Comparison with Other Classifiers. *J Chem Inf Model*. 2015;**55**(11):2365-2374.

29. Townsend JA, Glen RC, Mussa HY. Note on naive Bayes based on binary descriptors in cheminformatics. *J Chem Inf Model*. 2012;**52**(10):2494-2500.

30. Melssen W, Wehrens R, Buydens L. Supervised Kohonen networks for classification problems. *Chemometrics Intellig Lab Syst*. 2006;**83**(2):99-113.

31. Valsecchi C, Grisoni F, Motta S, Bonati L, Ballabio D. NURA: a curated dataset of nuclear receptor modulators. *Toxicol. Appl. Pharmacol.* 2020;**407**:115244.

32. Francis GA, Fayard E, Picard F, Auwerx J. Nuclear Receptors and the Control of Metabolism. *Annu Rev Physiol*. 2003;**65**(1):261-311.

33. Huang P, Chandra V, Rastinejad F. Structural overview of the nuclear receptor superfamily: insights into physiology and therapeutics. *Annu Rev Physiol*. 2010;**72**:247-272.

34. Rogers D, Hahn M. Extended-connectivity fingerprints. *J Chem Inf Model*. 2010;**50**(5):742-754.

35. KodeSrl. *Dragon (software for Molecular Descriptor Calculation)*.; 2017.

36. Multitask QSAR data. https://michem.unimib.it/download/data/multitask-qsar-data/. Accessed August 25, 2020.

37. Ruder S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv [csLG]*. June 2017. http://arxiv.org/abs/1706.05098.

38. Nair V, Hinton GE. Rectified linear units improve Restricted Boltzmann machines. In: *Proc. Int. Conf. Mach. Learn.* ICML; 2010.

39. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: *Proc. Int. Conf. Mach. Learn.* Vol 30. ; 2013:3.

40. Agostinelli F, Hoffman M, Sadowski P, Baldi P. Learning Activation Functions to Improve Deep Neural Networks. *arXiv [csNE]*. December 2014. http://arxiv.org/abs/1412.6830.

41. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016.

42. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016.

43. Fawcett T. An introduction to ROC analysis. *Pattern Recognition Letters*. 2006;**27**(8):861-874.

44. Naftaly U, Intrator N, Horn D. Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems*. 1997;**8**(3):283-296.

45. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc.; 2012:1097-1105.

46. Murtagh F, Hernández-Pajares M. The Kohonen self-organizing map method: an assessment. *J Classification*. 1995.

47. Novic M. Kohonen and counterpropagation neural networks applied for mapping and interpretation of IR spectra. *Methods Mol Biol*. 2008;**458**:45-60.

48. Keller JM, Gray MR, Givens JA. A fuzzy K-nearest neighbor algorithm. *IEEE Trans Syst Man Cybern*. 1985;**SMC-15**(4):580-585.

49. Tanimoto TT. IBM internal report 1957. *An elementary mathematical theory of classification and prediction*. 1957.

50. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Science & Business Media; 2009.

51. Sahigara F, Ballabio D, Todeschini R, Consonni V. Defining a novel k-nearest neighbours approach to assess the applicability domain of a QSAR model for reliable predictions. *J Cheminform*. 2013;**5**(1):27.

52. Ballabio D, Grisoni F, Todeschini R. Multivariate comparison of classification performance measures. *Chemometrics Intellig Lab Syst*. 2018;**174**:33-44.

53. Leardi R, Lupiáñez González A. Genetic algorithms applied to feature selection in PLS regression: How and when to use them. *Chemometrics Intellig Lab Syst*. 1998;**41**(2):195-207.

54. Leardi R. Genetic algorithms in chemometrics and chemistry: a review. *J Chemom*. 2001;**15**(7):559-569.

55. Ballabio D, Vasighi M, Consonni V, Kompany-Zareh M. Genetic Algorithms for architecture optimisation of Counter-Propagation Artificial Neural Networks. *Chemometrics Intellig Lab Syst*. 2011;**105**(1):56-64.

56. Blickle T, Thiele L. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evol Comput*. 1996;**4**(4):361-394.

57. Drezner Z, Hamacher HW. *Facility Location: Applications and Theory*. Springer Science & Business Media; 2001.

58. Leardi R, Boggia R, Terrile M. Genetic algorithms as a strategy for feature selection. *J Chemom*. 1992;**6**(5):267-281.

59. The MathWorks, Inc. , Natick, Massachusetts, United States. *MATLAB*.; 2018. www.mathworks.com.

60. Ballabio D. A MATLAB toolbox for Principal Component Analysis and unsupervised exploration of data structure. *Chemometrics Intellig Lab Syst*. 2015.

61. Grisoni F, Consonni V, Ballabio D. Machine Learning Consensus To Predict the Binding to the Androgen Receptor within the CoMPARA Project. *J Chem Inf Model*. February 2019:acs.jcim.8b00794.

62. MICHEM. Milano Chemometrics and QSAR Research Group - Department of Earth and Environmental Sciences - University of Milano-Bicocca. Milano Chemometrics and QSAR Research Group. https://michem.unimib.it/. Published 2015. Accessed July 2, 2020.

63. Ballabio D, Consonni V. Classification tools in chemistry. Part 1: linear models. PLS-DA. *Anal Methods*. 2013.

64. Ramsundar B, Eastman P, Walters P, Pande V. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. “O’Reilly Media, Inc.”; 2019.

65. Python Software Foundation. *Python Language Reference,*.; 2016. https://www.python.org/.

66. Virtanen, P, Gommers R, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* 2020, **17** (3):261–272.

67. Jolliffe IT. Principal component analysis. *Technometrics*. 2003;**45**(3):276.

68. Krzanowski W. *Principles of Multivariate Analysis*. OUP Oxford; 2000.

69. Shinozuka T, Tsukada T, Fujii K, et al. Discovery of DS-6930, a potent selective PPARγ modulator. Part II: Lead optimization. *Bioorg Med Chem*. 2018;**26**(18):5099-5117.

70. Valsecchi C, Grisoni F, Consonni V, Ballabio D. Consensus versus Individual QSARs in Classification: Comparison on a Large-Scale Case Study. *J Chem Inf Model*. 2020;**60**(3):1215-1223.

71. Ning W, Hu Z, Tang C, et al. Novel Hybrid Conjugates with Dual Suppression of Estrogenic and Inflammatory Activities Display Significantly Improved Potency against Breast Cancer. *J Med Chem*. 2018;**61**(18):8155-8173.

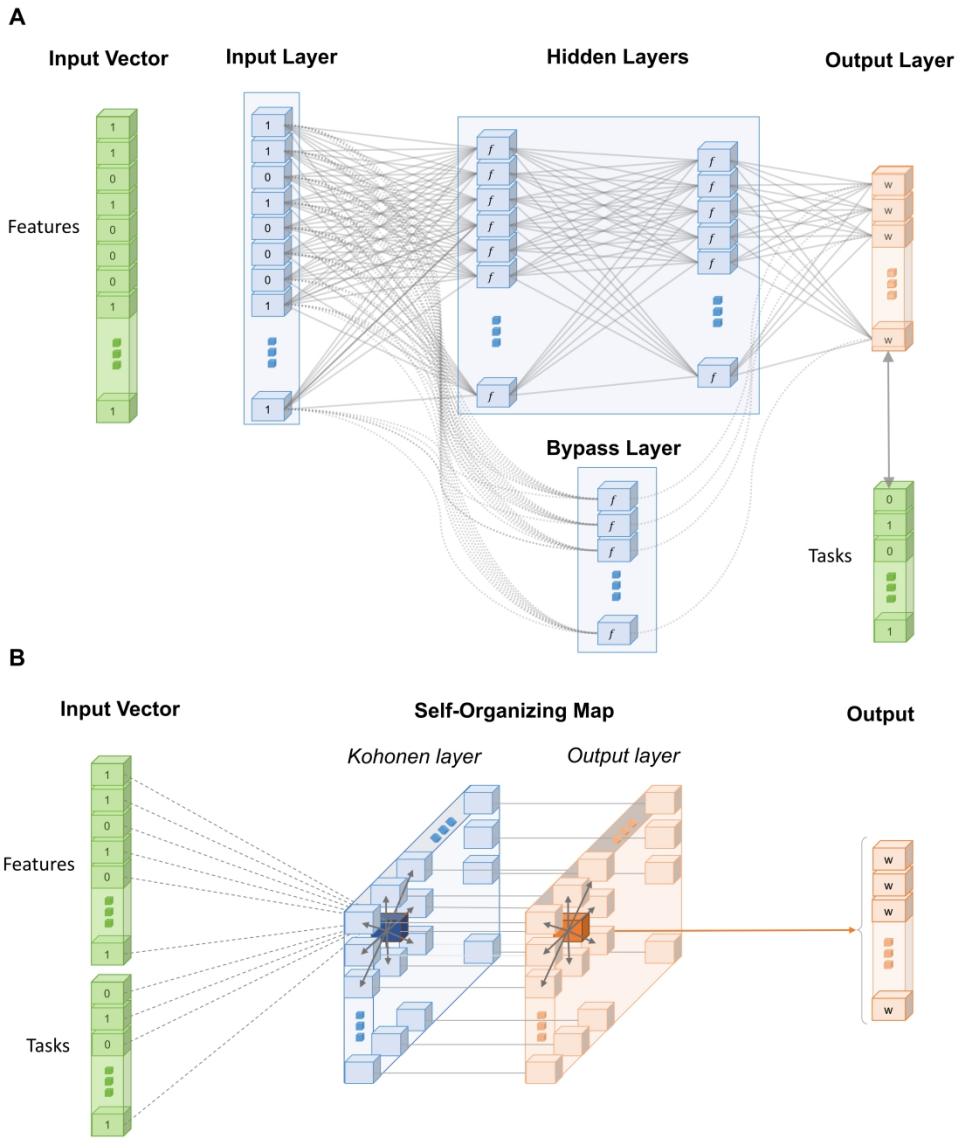


Figure 1. Schematic representation of the considered multi-task models. (A) Multi-task feedforward neural network with bypass layer; input vector is mapped to output layer with repeated compositions of hidden layers, a bypass layer connects directly the input with each task-specific sigmoid neuron in the output layer. (B) Multi-task XYF network; for each input vector composed of ECFP and tasks, the neuron with the minimum distance is selected (winning neuron) and its weights and those of the neighbouring neurons are updated in a radial manner.

		NER_T (%)	Relative frequency for NER_T-based decile intervals (%)									
		mean	72.1	83.6	86.7	88.1	88.9	89.3	89.5	89.6	89.8	90.1
		min-max	(50.0-81.8)	(81.8-85.4)	(85.4-87.5)	(87.5-88.7)	(88.7-89.1)	(89.1-89.4)	(89.4-89.6)	(89.6-89.7)	(89.7-89.9)	(89.9-90.6)
Parameter	**Value**	**Label**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**
Architecture	100	Arch1	9.6	2.0	5.9	15.7	7.8	11.8	5.9	21.6	19.6	17.3
	1000, 100, 10	Arch2	19.2	19.6	11.8	17.6	9.8	3.9	5.9	3.9	7.8	17.3
	1000, 100	Arch3	5.8	7.8	7.8	0.0	5.9	5.9	11.8	9.8	9.8	13.5
	100, 100	Arch4	21.2	17.6	21.6	15.7	21.6	21.6	29.4	13.7	17.6	34.6
	1000	Arch5	7.7	15.7	13.7	21.6	7.8	9.8	15.7	15.7	29.4	7.7
	10	Arch6	5.8	9.8	23.5	11.8	15.7	13.7	9.8	13.7	7.8	3.8
	10, 5	Arch7	15.4	15.7	9.8	9.8	23.5	27.5	11.8	7.8	5.9	3.8
	10, 10	Arch8	15.4	11.8	5.9	7.8	7.8	5.9	9.8	13.7	2.0	1.9
Learning rate	0.01	LR1	11.5	23.5	17.6	39.2	37.3	19.6	13.7	9.8	9.8	5.8
	0.001	LR01	57.7	25.5	35.3	23.5	45.1	60.8	76.5	86.3	90.2	88.5
	0.0005	LR005	19.2	17.6	9.8	5.9	17.6	19.6	9.8	3.9	0.0	5.8
	0.025	LR25	11.5	33.3	37.3	31.4	0.0	0.0	0.0	0.0	0.0	0.0
Weight decay	0	WD0	32.7	39.2	49.0	49.0	45.1	35.3	29.4	39.2	37.3	50.0
	0.01	WD1	67.3	60.8	51.0	51.0	54.9	64.7	70.6	60.8	62.7	50.0
Weight decay type*	L2	L2	48.1	47.1	60.8	70.6	58.8	64.7	58.8	51.0	66.7	67.3
	L1	L1	51.9	52.9	39.2	29.4	41.2	35.3	41.2	49.0	33.3	32.7
Dropout	0	Drp0	55.8	54.9	74.5	72.5	64.7	51.0	56.9	60.8	64.7	88.5
	0.5	Drp1	44.2	45.1	25.5	27.5	35.3	49.0	43.1	39.2	35.3	11.5
Activation function	ReLu	AFr	26.9	41.2	41.2	25.5	31.4	56.9	31.4	47.1	41.2	53.8
	Leaky ReLu	AFl	15.4	9.8	13.7	35.3	11.8	5.9	25.5	5.9	17.6	15.4
	Tanh	AFt	25.0	39.2	31.4	19.6	25.5	29.4	37.3	43.1	39.2	30.8
	Sigmoid	AFs	32.7	9.8	13.7	19.6	31.4	7.8	5.9	3.9	2.0	0.0
Bypass layer	1	Bp1	48.1	51.0	47.1	56.9	47.1	51.0	72.5	70.6	43.1	61.5
	0	Bp0	51.9	49.0	52.9	43.1	52.9	49.0	27.5	29.4	56.9	38.5
Optimization algorithm	Adam	optA	46.2	66.7	49.0	78.4	92.2	100.0	96.1	100.0	100.0	100.0
	Gradient Descent	optGD	53.8	33.3	51.0	21.6	7.8	0.0	3.9	0.0	0.0	0.0
Exponential decay	0	ED0	25.0	49.0	96.1	92.2	100.0	100.0	100.0	100.0	100.0	100.0
	1	ED1	75.0	51.0	3.9	7.8	0.0	0.0	0.0	0.0	0.0	0.0

* Relative frequency calculated considering only chromosomes with WD1

Figure 2. Relative frequency (%) of network parameters in the chromosomes of the final GA population; for each NER_T-based decile (D1,..., D10), the relative frequency of architecture and training parameters is reported and coloured according to a red (minimum) - green (maximum) scale. The first two rows report mean and minimum-maximum values of NER_T(%) for each decile. *The relative frequencies of weight decay type (L1, L2) were calculated considering only the chromosomes with weight decay equal to 0.01 (WD1).

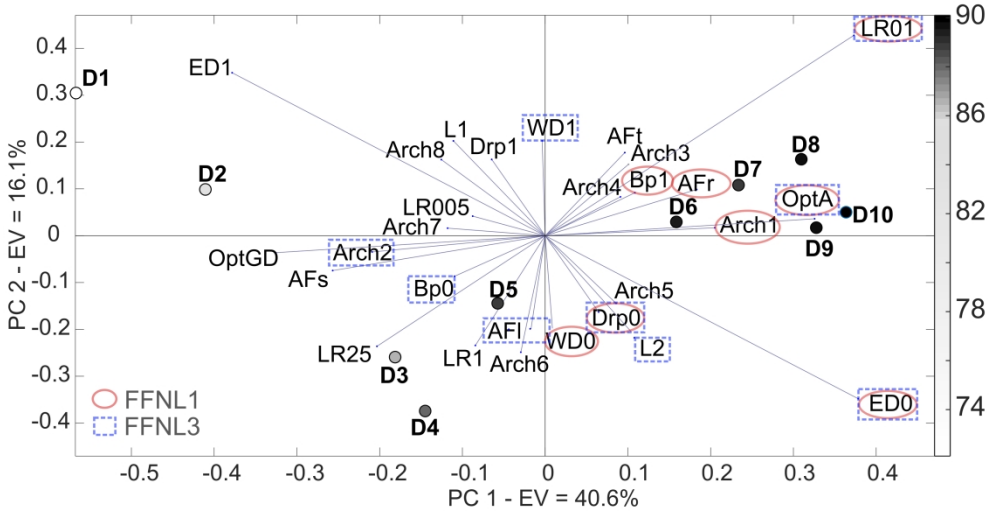
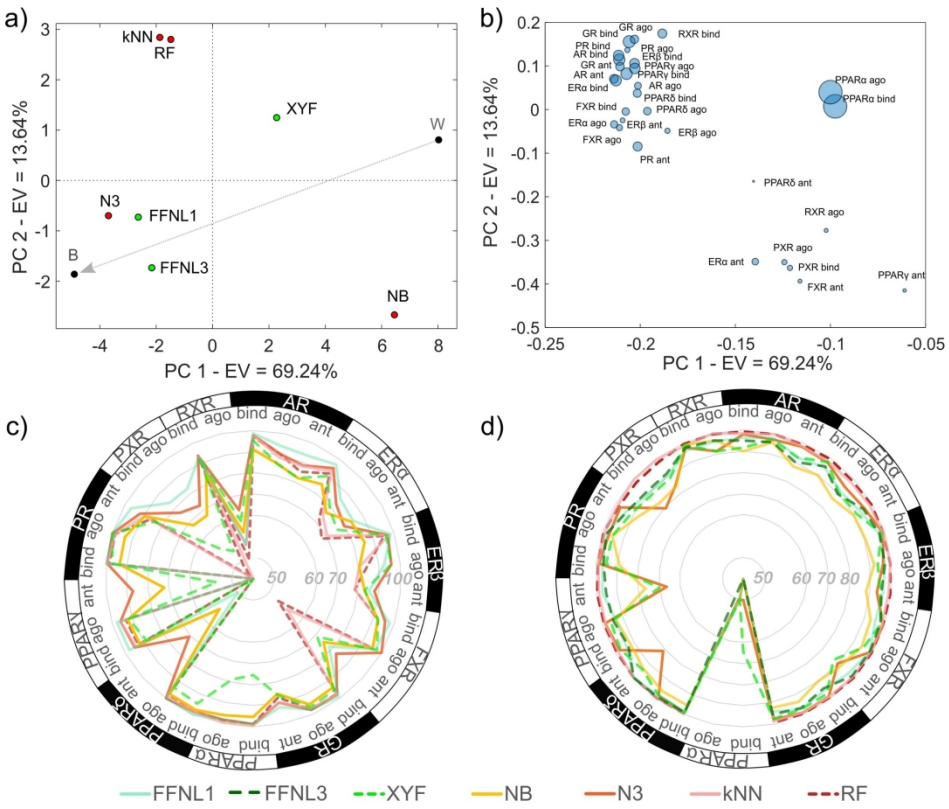


Figure 3. PCA biplot of relative frequencies of network parameters. Scores (the deciles: D1, D2,..,D10) are coloured according to the average NER_T. The loading labels (networks parameters) are reported in Figure 2. FFNL1 and FFNL3 stand for the two multi-task feedforward neural networks whose selected parameter combinations are highlighted in red and blue, respectively.



Analysis of the classification performance on the individual tasks. (a) Score plot of PCA on the task-specific classification performances, expressed as NER_t, of all the considered models; B and W represent the theoretical best and worst performance, respectively; multi-task and single-task models are represented by green and red circles, respectively. (b) PCA loading plot; each circle represents a task and its size is proportional to the percentage of active chemicals. Radar plots of (c) sensitivity (Sn_t) and (d) specificity (Sp_t) achieved on test molecules for each task.

p-value		*FFNL1*					
	Sn_t	**0.391**					
FFNL3	Sp_t	**0.891**	***FFNL3***				
	NER_t	**0.222**					
	Sn_t	0.000	0.000				
XYF	Sp_t	**0.284**	**0.405**	***XYF***			
	NER_t	0.000	0.000				
	Sn_t	0.000	0.000	**0.424**			
NB	Sp_t	0.000	0.000	0.001	***NB***		
	NER_t	0.000	0.000	**0.147**			
	Sn_t	**0.367**	**0.923**	0.000	0.000		
N3	Sp_t	**0.600**	**0.417**	**0.139**	0.003	***N3***	
	NER_t	**0.214**	**0.057**	0.000	0.000		
	Sn_t	0.000	0.000	**0.738**	**0.459**	0.000	
kNN	Sp_t	0.000	0.000	0.000	0.000	0.000	***kNN***
	NER_t	**0.428**	**0.558**	0.039	0.008	**0.082**	
	Sn_t	0.000	0.000	**0.683**	**0.909**	0.000	**0.052**
RF	Sp_t	0.000	0.000	0.000	0.000	0.000	0.016
	NER_t	0.030	**0.098**	0.007	0.001	0.001	**0.229**

P-values of the paired Wilcoxon signed-rank test performed on each pair of classification approaches and for each performance measure (Sn_t, Sp_t and NER_t). P-values greater than 0.05, which support the evidence that the null hypothesis is true (i.e., no statistically significant median difference between model performance on all the tasks) are highlighted in bold. The couples of models with p-values always greater than 0.05 are highlighted with a grey background.

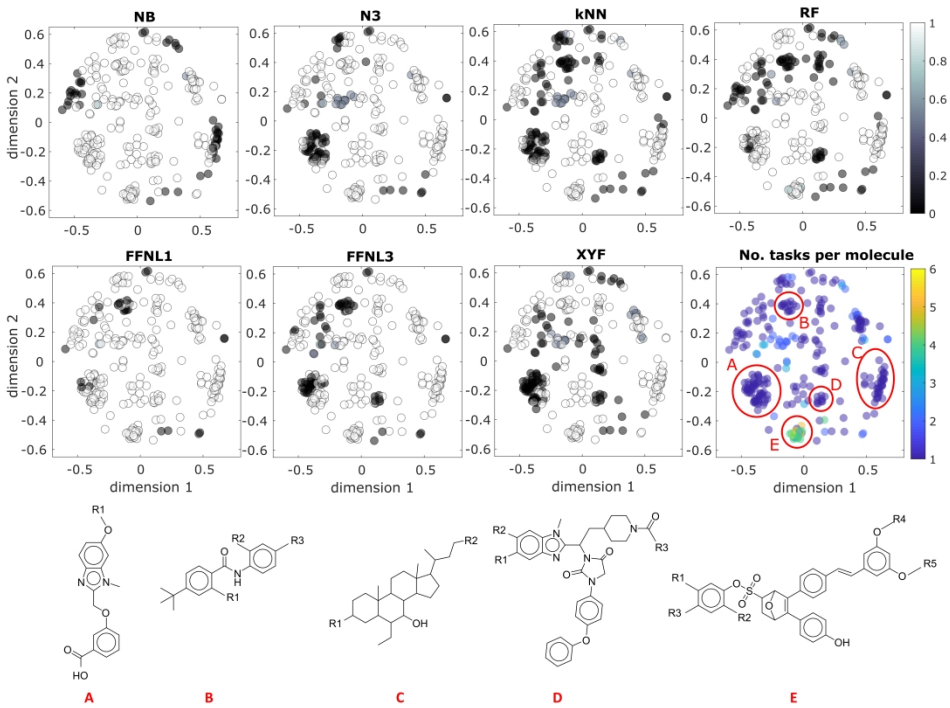


Figure 6. Multidimensional scaling calculated over the fingerprints (ECFPs) of the evaluation set (stress = 0.30). Each circle represents a molecule and its brightness is proportional to the fraction of correct predictions among all annotated tasks. Single-task and multi-task models are shown in the first and second row, respectively. The last score plot shows the number of tasks with known activities per molecule in terms of the circle's colour. Representative scaffolds of the chemical structures belonging to cluster A, B, C, D and E are reported below, where Rn indicates every possible residue.