A Survey of Multi-Task Learning Methods in Chemoinformatics

Sergey Sosnin,[1] Mariia Vashurina,[2] Michael Withnall,[2] Pavel Karpov,[2] Maxim Fedorov[1,3] and Igor V. Tetko[2,4]

[1]Skolkovo Institute of Science and Technology, Skolkovo Innovation Center, Moscow 143026, Russia

[2]Helmholtz Zentrum München - German Research Center for Environmental Health (GmbH), Institute of Structural Biology, Ingolstädter Landstraße 1, D-85764 Neuherberg, Germany

[3]University of Strathclyde, Department of Physics, John Anderson Building, 107 Rottenrow East, G40NG, Glasgow, United Kingdom

[4]BIGCHEM GmbH, Ingolstädter Landstraße 1, b. 60w, D-85764 Neuherberg, Germany

* Address for correspondence
Dr. Igor V. Tetko,
Institute of Structural Biology, Helmholtz Zentrum München - German Research Center for Environmental Health (GmbH), Institute of Structural Biology, Ingolstädter Landstraße 1, D-85764 Neuherberg, Germany
i.tetko@helmholtz-muenchen.de
Tel.: +49-89-3187-3575

Keywords: Multi-task learning, transfer learning, neural networks

Abstract

Despite the increasing volume of available data, the proportion of experimentally measured data remains small compared to the virtual chemical space of possible chemical structures. Therefore, there is a strong interest in simultaneously predicting different ADMET and biological properties of molecules, which are frequently strongly correlated with one another. Such joint data analyses can increase the accuracy of models by exploiting their common representation and identifying common features between individual properties. In this work we review the recent developments in multi-learning approaches as well as cover the freely available tools and packages that can be used to perform such studies.

Introduction

Nowadays, the volume of data that can be generated and processed when modelling tasks has increased dramatically [1]. Machine Learning (ML) techniques, notably Deep Neural Networks (DNNs)[2] are becoming indispensable as a tool for managing and using these vast amounts of generated and measured data effectively. However, data measurement is still a difficult and time-consuming task, and there is a strong interest in how to make the best use of all available data. Biological data, such as ADMETox properties, are highly interrelated. For example, the lipophilicity of compounds is, in one way or another, very important for the majority of these properties. Thus learning several ADMETox properties simultaneously can result in better models. Moreover, some types of data produced with different methods can have different experimental accuracy and/or refer to related but not identical properties. For example, kinetic water solubility is the concentration of a compound in solution at the time when an induced precipitate first appears. This type of solubility can be easily automatized for use in High Throughput Screening (HTS) settings and is actively used in industry due to this. The more biologically relevant solubility is thermodynamic solubility, which is the concentration of a compound in a saturated solution when excess solid is present, and solution and solid are at equilibrium.[3] The co-modelling of both types of solubility simultaneously could potentially develop better models for each of them. This can be achieved with the help of multi-task learning [4], which can be applied to an arbitrary combination of regression and classification tasks (so called heterogeneous multi-tasks).

These multi-learning approaches belong to so-called transfer learning [5], a technique where knowledge gained in one or several (source) tasks is used to improve the target task. The transfer learning approaches differ with respect to whether the source and/or target tasks have labelled data. Thus, they can be classified as semi-supervised or “self-taught” learning (no labelled data in the source domain), transductive learning (labelled data are only in the source domain), unsupervised transfer learning (no labelled data are available) [5] as well as methods which use labelled data for both source and target tasks, which include multi-learning approaches.

The ability to infer relevant knowledge is very important for intelligence. For example, humans, who can draw on vast amounts of previously-learned information, can be trained on a new task with a relatively tiny number of examples. In contrast, traditional machine learning algorithms, which usually learn from scratch, and require large example sets to do so. Therefore, there is active development and interest in machine learning to design new methods having the same speed and accuracy as humans. Early examples of such types of learning have been successfully reported since the mid-1990s, e.g. the use of neural network weights trained with one task as a starting point for new ones to increase the development speed and the accuracy of models [4]. A Library model of Associative Neural Networks[6] is another example, which applied on-the fly correction of predictions for new data by using the errors of the nearest neighbours of the target sample.[7] Transfer of information was also done by developing models for individual properties, and then using those model predictions as additional descriptors for the target property, known as the feature net approach [8]. In the case that the target and source properties are very similar or identical (e.g., measured for different species or at different conditions), one can encode different targets by using additional descriptors (e.g., conditions of experiments) and model all properties simultaneously. Figure 1 schematically illustrates single task as well as several multitask modelling approaches using an example of neural networks. Some of these approaches, such as the feature net, use sequentially-ordered learning.

In our review we will cover new developments in the field, which have appeared during the recent years. Also, we will mainly focus on the methods where the analysed properties are simultaneously modelled within a single model, which corresponds to Figure 1b.

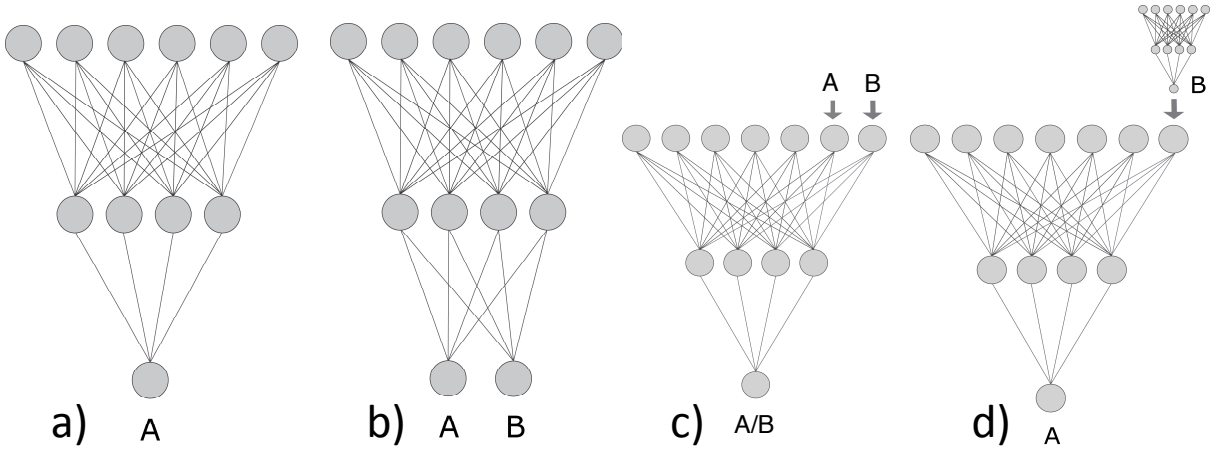

Figure 1. a) Single Task learning; b) Multi-task learning; c) Multi-task learning by property encoding as descriptors; d) Feature net. Adapted with permission from ref. [8]. Copyright (2009) American Chemical Society.

Multi-task Learning (MTL) is a technique which aims improve ML efficacy by simultaneously co-modelling multiple properties within a single model. A lot of developments in this field were done in in 1990s by Rich Caruana[4], who investigated how to improve related task performance by leveraging domain-specific information, and inductively transferring it between the tasks. In comparison to the other transfer learning approaches, which use labelled data for both source and target tasks, the aim of MTL is to improve the performance of all tasks with no task prioritised.

MTL trains tasks in parallel, sharing their representation internally. As a result, the training data from the extra tasks serve as an inductive bias, acting in effect as constraints for the others, improving general accuracy and the speed of learning. Caruana noted mechanisms by which MTL may show improvement over Single Task Learning (STL) to be a) amplification of statistical data; b) attention focusing (finding a better signal in noisy data); c) eavesdropping (learning “hints” from simpler tasks); d) representation bias and feature selection and e) regularisation (less overfitting) [4].

As MTL implies sharing information between all tasks, it is possible to define three main types of MTL based on the type of data sharing: feature, instance and parameter-based[9]. Feature-based MTL models learn a common feature representation among all the tasks by assuming that such a representation can increase the performance of the algorithm vs. single-tasks. Parameter-based approaches explore the similarity between target properties and include task clustering, learning of task relationships, as well as multilevel hierarchical approaches. Instance-based MTL identifies individual data within a task, which can be effectively used in other tasks for information sharing. [10] However, we did not find applications for the latter in chemoinformatics and thus will not cover them in our review. Let us consider some examples of the other two MTL approaches and their combination.

Feature based approaches

Neural networks are the primary platform for multi-learning. Rich Caruana was one of the first to develop multi-task learning using backpropagated neural networks. He found out that four separate neural networks performing only one task can be reduced to one network with multiple outputs that performs the tasks simultaneously. As a result, he created a multi-task neural network able to perform parallel learning. One should also mention the earlier work of Suddarth and Kergosien [11], who used an additional layer to inject rule hints and to guide the neural network as to what should be learned.

The network forms a set of features on the hidden layer(s), which can fit several tasks simultaneously. Moreover, the activation patterns of neurons in neural networks with several hidden layers contribute to the formation of features, which are known to be important for the analysed type of properties, e.g. toxicophores for the prediction of toxicological end-points.[12]

One of the first successful applications of MTL in chemoinformatics was done by Varnek et al [8], who demonstrated that learning several tissue/air partitioning coefficients by using Associative Neural Networks provided models with statistically-significantly higher accuracy compared to the respective single task models. The neural network models analysed by Varnek et al were examples of so-called “shallow” neural networks since they included only one hidden layer. The appearance of new training algorithms and in particular GPU-accelerated computing has brought about the rise of Deep Neural Networks,[2] which incorporate multiple hidden layers with much larger numbers of neurons. This greater flexibility of DNN networks allows them to learn more complex relationships and patterns in the data.

Regarding multi-learning one can distinguish two primary architectures with respect to the sharing of parameters: hard and soft. “Hard” parameter sharing is similar to that of shallow neural networks and implies the sharing of hidden layers between all tasks, except some task-specific output layers. “Soft” parameter sharing gives each task its own model with its own parameters, where these model parameters have a regularized distance to facilitate the sharing of learning.[13] Soft parameter sharing has not yet received sufficient attention in chemoinformatics and will be briefly outlined in the section “Simultaneous Feature and Task similarity learning”.

J. Ma et al [14] performed several experiments on STL and MTL neural networks. They found out that in some cases multi-task learning deep neural networks (MTL DNN) are better than single task learning deep neural networks (STL DNNs). The authors suggested that better performance of MTL DNN is based mainly on the size of data sets: MTL DNNs are useful for small and mixed (small and large) datasets and STL DNNs are good for large data sets.

Multi-task learning provided the best model according to the ROC AUC (Receiver Operator Characteristic Area Under Curve) metric for the Tox21 challenge.[12] The authors showed that such networks learned on their hidden layers chemical features resembling toxicophores identified by human experts. The networks used these features to classify active and inactive (toxic and nontoxic) compounds. It is also of note that the second best approach was based on “shallow” STL associative neural networks.[15]

In another comprehensive study the authors compared the performance of MTL and STL approaches in predicting the toxicity of chemical compounds from the Registry of Toxic Effects of Chemical Substances (RTECT) database totalling 29 toxicity end-points and more than 120k measurements.[16] MTL significantly outperformed STL thus showing the utility of this approach to model complex *in vivo* endpoints.

Xu et al [17] investigated why an MTL DNN can outperform separate STL DNNs and under what scenarios the multi-task approach is advantageous. The result of this study lead to two main findings regarding the efficacy of multi-task deep neural networks:

- Similar molecules modelling correlated properties will boost the predictive performance of the DNN, and likewise uncorrelated properties will degrade performance.
- Structurally dissimilar molecules have no influence on the predictive performance of the MTL DNN, regardless of whether or not tasks are correlated.

Their conclusions are important for the identification of strategies for designing datasets for MTL learning.

MTL can be used to simultaneously learn both regression and classification in one model, as was demonstrated by Xu et al [18] for the prediction of acute oral toxicity. The authors used convolutional neural networks and reported that their model provided higher accuracy compared to conventional methods.

Human cytochrome P450 inhibition for 5 kinases were predicted using a pre-trained autoencoder-based DNN [19]. On the pre-training stage, the first layers were trained to reconstruct the original input layer on the whole database. The authors proved that an autoencoder-based DNN can achieve better quality than other popular methods of machine learning for cytochrome P450 inhibition prediction, and a multi-target DNN approach can significantly outperform single-target DNNs. The flexibility of neural networks makes it possible to use them not only with descriptors derived from chemical structures in the traditional way, but also apply them to directly analyse chemical structures represented as SMILES or chemical graphs. We will review several approaches in the "Implementations of multi-learning approaches" section below.

Multi-task feature learning for sparse data using other methods. The problem of feature-selection has an exact mathematical formulation and an analytical solution for linear methods. For example, Varnek et al[8] compared the performance of neural networks with Partial Least Squares (PLS). PLS could also provide multi-task learning by identifying common internal representations, so called latent variables, for several analysed properties simultaneously. In addition to the PLS method, there are other approaches for identifying sparse features or to perform multi-feature selection as comprehensively analysed in a recent review.[9] These methods can be used directly with linear or kernel methods, or to provide features for training other methods.

One such method is Macau [20]. It is based on Bayesian Probabilistic Matrix Factorisation (BPMF). After BPMF was used to win the Netflix prize for predicting film recommendation, the interest in this method notably increased. One of the problems during multi-learning are missing values; frequently not all measurements are available for all targets. For some other tasks the matrix of responses can be extremely sparse, e.g. only 1.2% of all users-combinations were available for the Netflix competition. Some methods, such as neural networks, can naturally work with missing values by ignoring the error contribution from missing values when calculating the loss for backpropagation. The BPMF allows imputing missing values in the matrix thus enabling the application of standard techniques, such as singular value decomposition and principal component analysis. In contrast to classical algorithms of matrix factorization, Macau is able to handle side relations i.e. fingerprints of chemical compounds or phylogenetic distance between protein targets. Another useful feature of Macau is the ability to work with multi-dimensional data and perform tensor decomposition. The capacity to deal with multi-dimensional biological sparse data was studied by de Vega et al[21], who applied this technique to inhibition activities of 15073 compounds for 346 targets extracted from ChEMBL. The authors showed that Macau provided performance similar to that of neural networks methods but did not require GPU-accelerated computing.

Task learning approaches

Task learning explores task relationships to better learn common parameters of models as overviewed below.

Metric-learning algorithms. k-Nearest Neighbour approaches provide predictions for new samples based on their nearest neighbours. Usually, it uses a Mahalanobis distance, which is defined as:

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)} \qquad (1)$$

where and x_i and x_j are two samples and M is a matrix, which acts as a global linear transformation of the input space. The M matrix is thus an optimizable parameter of the method. The most straightforward way is to use the same metric to model all tasks simultaneously. However, better performance can be expected by using different matrices, which are optimised to each individual class. If tasks are correlated, the matrix M can be decomposed into a common M_0 and individual task-specific M_t parts, as

$$d_t(x_i, x_j) = \sqrt{(x_i - x_j)^T (M_0 + M_t)(x_i - x_j)} \qquad (2)$$

where M_0 and $M_1, \ldots, M_T$ are the global matrix and task-specific additional matrices respectively. The larger the similarity is between the tasks, the larger the determinant of matrix M_0 relative to those of individual tasks M_t. This idea was first applied to multi kNN by Parameswaran et al[22]. Since that time many different algorithms have been developed for metric learning, as overviewed by Yang et al.[23]

Similarity learning. Metric learning, in contrast to feature selection, directly optimises the parameters of the method itself. The main idea is that similar tasks can provide better generalization by using similar parameters. For example, when classifying several related properties one can identify a common separation hyperplane given by a vector w_0, which will be only slightly different for separation hyperplanes w_i for individual properties

$$w_i = w_0 + v_i \qquad (3)$$

where v_i accounts for features specific for property *i*. This separation is thus similar to that used for global and task-specific matrices in eq. (2) where w_0 and v_i correspond to matrices M_0 and M_t respectively. Figure 2 exemplifies the intuition underlying this idea used to develop the Multi-Task Least Square Support Vector Regression (MLS-SVR) approach.[24]

One of the promising current approaches in the field is based on MTL networks with “soft” parameter sharing (see Figure 3). The network facilitates multi-task learning by regularising weights as well as features (which are defined as neural network activation patterns at the last layers) across the networks [25]. The regularisation of weights corresponds to the sharing of model parameters while the regularisation of learning features across the last networks’ layers corresponds to feature regularisation. The algorithm can also be applied if no measurements are available for one of the tasks.

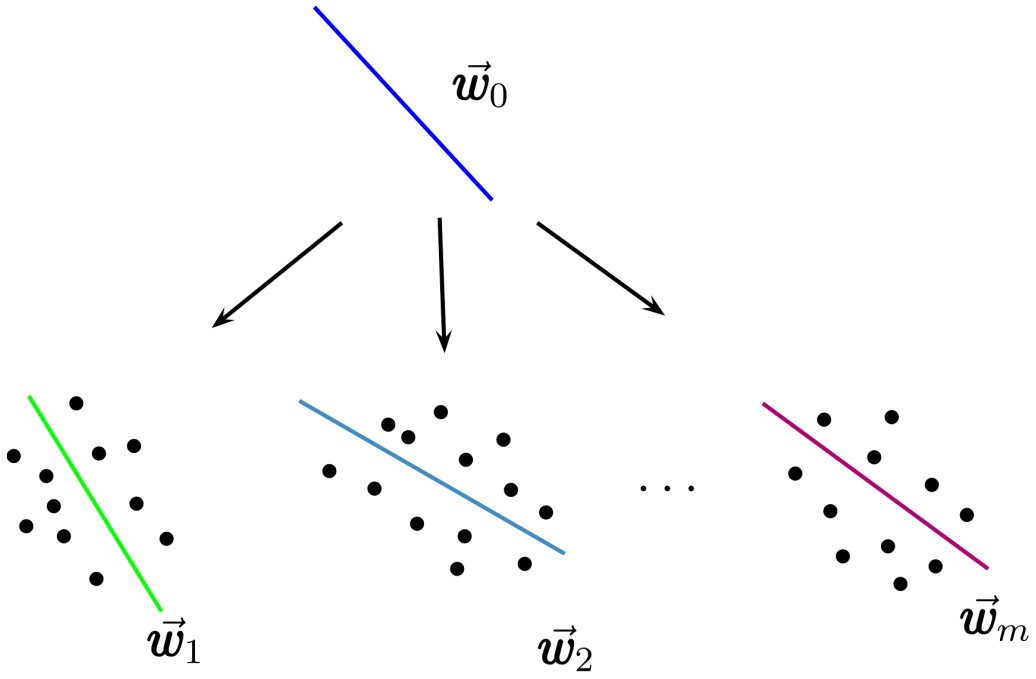

Figure 2. Multi-task learning in Least Square Support Vector Regression (MLS-SVR) identifies a common hyperplane w_0, which carries the information of the commonality and $w_i = w_0 + v_i$, where the vector v_i carries the information of the specialty. (Reprinted from *Pattern Recognition Letters, vol. 34*, Xu, S.; An, X.; Qiao, X.; Zhu, L.; Li, L., Multi-output least-squares support vector regression machines, Copyright (2013), with permission from Elsevier).

The information about task dependency can be used as *a priori* information and an example of multi-task learning with the integration of taxonomy information has been presented by Rosenbaum et [26]. The authors used a dataset of 112 human kinases extracted from ChEMBL. The Graph-regularized multi-task (GRMT) Support Vector Machine Regression and Top-Down Multi-task SVR were used to consider the relationship between these targets during modelling. The authors showed that hierarchical learning provided significantly better results compared to base models, as developed using STL approaches such as STL, or a model which combined all data and ignored the kinases' types.

Similarity learning is also a feature of Generative Topographic Mapping (GTM),[27] which can be used both for visualization and molecular property prediction.[28] GTM constructs a projection from a high-dimensional descriptor space into a latent (usually 2D) space. Probabilities of the latent representations of molecules can be regarded as GTM descriptors and be used to build classification or regression models. Gaspar et al[29] proposed the Stargate GTM method, which projects both descriptors and multi-target activity spaces into corresponding latent spaces and iteratively optimizes the joint probability distribution between the two mappings. The authors compared the method on data extracted from ChEMBL and showed that the Stargate GTM slightly outperformed conventional GTM but had a lower accuracy than Random Forest. It was also stressed that the model can act as a "gate", which both predicts the activity profiles for a compound and finds areas in a descriptor space that are likely to have the desired activity profile. The latest feature is a particular advantage of Stargate GTM.

In machine learning there are a number of other approaches that can explore task similarity, including task clustering or multi-level approaches as reviewed elsewhere [9].

Simultaneous Feature and Task similarity learning

As it was aforementioned, networks with soft parameter sharing can provide a richer variety of network architectures (for review see [13]). Such networks can be used to simultaneously provide feature selection and task similarity learning. Let us show how this method could potentially be used to address domain adaptation. This problem is well known in the chemical industry and has been deeply studied by Sheridan,[30] who demonstrated a loss of prediction accuracy in models for prospective validation of compounds, due to a time shift in chemical diversity. The problem of prospective validation can be easily cast to the multi-learning domain by considering two tasks (prediction of past and new data, for which one may have just a few measurements) as two separate tasks.

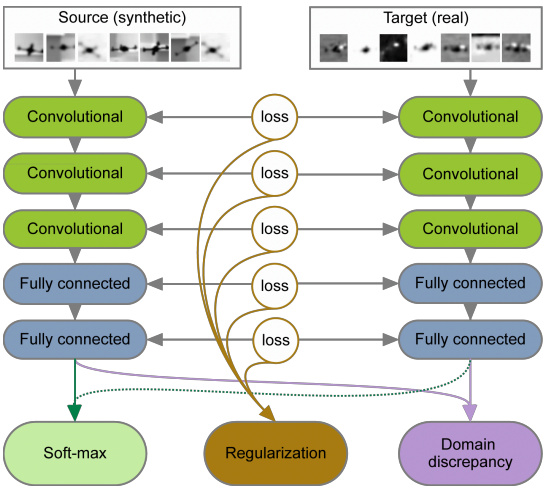

Figure 3. An example of neural network model using "soft parameter" sharing. Two networks are trained in parallel for each individual task. The soft parameter sharing is done by introducing a penalty function, which prevents neural network weights in both models from differing greatly, as well as by regularising neural network features at the last layer. Reprinted from ref. [25] under the Creative Commons license CC-BY 4.0.

Implementations of multi-task learning approaches

Multiple software packages exist and are available in the computer science field, which provide tools for multi-learning. As a rule, many articles are published by the authors together with their source code, which is frequently deposited on online repositories such as GitHub, allowing wide and immediate dissemination of information. The use of these software tools in chemoinformatics is not necessarily straightforward due to the need to make an interface with chemical structures. However, several efforts to port these packages to chemoinformatics are currently on-going. In Table 1 we review several complete packages, which were developed to bring multi-learning approaches to analyse chemical structures.

Table 1. "Chemistry aware" multi-task learning approaches

Package	Examples of supported algorithms	Availability

Chainer Chemistry	NFP, GGNN, RSGCN, WeaveNet, SchNet	https://github.com/pfnet-research/chainer-chemistry
DeepChem	DAG, NNF, MPNN, TEXTCNN, WEAVE, IRV	https://github.com/deepchem/deepchem
OCHEM	The methods from ChemChainer, DEEPCHEM, DNN, MLS-SVM as well as multi-task learning by property encoding as descriptors and feature net.	http://ochem.eu

Chainer Chemistry (ChemChainer) ports several neural network architectures, which were recently introduced to work with graphs, to chemical structures. DeepChem supports the majority of ChemChainer methods as well as providing several other approaches, some of which were originally developed by the authors of the toolbox. DeepChem also provides a port of machine learning methods from the Scikit-learn python package. Since the latter methods support only single-task learning, DeepChem uses an embedded wrapper to calculate models for each task, and provides a combined result of STL models in way similar to that of MTL, thus allowing an easy comparison of STL and MTL models. Thus, the user can apply both types of methods to datasets containing several properties using a similar interface. ChemChainer and DeepChem are based on Python and are built around Chainer and TensorFlow frameworks, respectively. Both packages use the RDkit library,[31] which provides a framework to translate chemical structures to graphs and the required representation for both packages.

OCHEM provides [32] a uniform interface to methods from both of these packages as well as several other methods supporting multi-task learning, such as Associative Neural Networks, an implementation of Deep Neural Networks, a GPU implementation of Least Squares Support Vector Machines [33] and several other approaches. An example of simultaneous prediction of tissue/air partitioning coefficients from Varnek et al [8] by different methods is shown in Figure 4.

Below we overview several methods implemented in these packages. The majority of these methods are neural networks that operate on chemical graphs. Thus, these approaches are different from traditional ones that analyse molecules by converting them to a set of descriptors. The first publication about the direct application of neural networks to graphs was proposed as an extension of recurrent neural networks in 2005 [34]. Interestingly, the first models based on chemical graphs were presented in the field of chemoinformatics about eight years earlier by Baskin et al.[35]

Neural Network Fingerprints (NNF). The method shows that the representation of chemical structures as circular fingerprints (e.g. Morgan fingerprints or Extended Connectivity Circular Fingerprints (ECFP)) can be extended with a more advanced method based on neural networks.[36]

Weave network[37] This network was developed as an inspiration of convolutional neural networks. This network recreates atom and pair features on each layer based on the information in the previous layer, which resembles a weaving propagation of information through the network. The multiple layers (“weaves”) can be stacked to produce networks with more complex architectures.

Renormalized Spectral Graph Convolutional Network (RSGCN) [38] This network was developed to learn large graph-structured networks, where the classification information is only available for a

small number of samples but valuable additional information can be derived from the data graph structure of a much large number of unlabeled data points.

A continuous-filter convolutional neural network for modeling quantum interactions (SchNet)[39] was developed to overcome the limitations of using grid-based approaches, which work with discretized signals such as image pixels. The Comparative molecular field analysis (CoMFA)[40] represents another example of a similar grid-based approach coupled with PLS.

Gated Graph Neural Network (GGNN). This network was specifically developed to predict sequences of outputs, allowing better predictions of their relationships.[41] This algorithm was introduced by testing its performance on the bAbI suite tasks where it demonstrated a remarkable performance over existing algorithms. The bAbI tasks were specifically developed to test the reasoning capabilities of artificial intelligence systems, such as Path Finding and Shortest Path Finding, or automatic program verification.

Predicted property: Cblood/Cair(Human)
Training set: tissue/air set

Metrics R2 for Training set Validation: Cross-Validation (52 models)

	ASNN	MTL	DNN	ASNN(2)	STL	DNN(2)
CDK2 (constitutional, topological, geometrical, electronic, ...	0.88 0.83 0.94 0.73 0.86 0.9 0.87 0.86 0.85 0.9 0.71 (0.848)		0.79 0.76 0.8 0.63 0.74 0.8 0.868 0.68 0.79 0.78 0.55 (0.744)	0.88 0.7 0.76 0.4 0.77 0.71 0.88 0.6 0.82 0.85 0.69 (0.733)		0.79 0.58 0.64 0.3 0.4 0.55 0.83 0.49 0.77 0.82 0.2 (0.579)
Dragon6 (blocks: 1-29)	0.73 0.84 0.84 0.78 0.73 0.74 0.85 0.6 0.75 0.79 0.5 (0.741)		0.86 0.74 0.92 0.78 0.87 0.9 0.878 0.76 0.84 0.83 0.61 (0.817)	0.85 0.4 0.6 0.19 0.4 0.4 0.76 0.39 0.73 0.79 0.43 (0.54)		0.83 0.74 0.6 0.25 0.43 0.5 0.81 0.45 0.77 0.79 0.5 (0.606)
ALogPS, OEstate	0.86 0.81 0.93 0.7 0.85 0.83 0.82 0.74 0.86 0.89 0.65 (0.813)		0.87 0.81 0.83 0.7 0.81 0.8 0.85 0.77 0.81 0.85 0.67 (0.797)	0.87 0.51 0.73 0.3 0.56 0.68 0.79 0.41 0.75 0.84 0.5 (0.631)		0.87 0.73 0.79 0.52 0.81 0.73 0.82 0.59 0.79 0.85 0.5 (0.727)
Fragmentor (Length 2 - 4)	0.79 0.56 0.86 0.5 0.7 0.82 0.7 0.66 0.82 0.85 0.65 (0.719)		0.76 0.79 0.79 0.72 0.73 0.76 0.76 0.63 0.75 0.79 0.56 (0.731)	0.81 0.4 0.79 0.2 0.5 0.73 0.74 0.41 0.74 0.8 0.7 (0.62)		0.76 0.4 0.6 0.22 0.4 0.6 0.75 0.3 0.68 0.78 0.5 (0.545)
ChemaxonDescriptors (pH 0 - 14:1)	0.85 0.82 0.92 0.73 0.89 0.87 0.85 0.7 0.81 0.85 0.82 (0.828)		0.79 0.8 0.83 0.64 0.79 0.8 0.79 0.73 0.81 0.81 0.58 (0.761)	0.83 0.67 0.8 0.4 0.74 0.75 0.84 0.6 0.8 0.81 0.7 (0.722)		0.78 0.41 0.64 0.1 0.2 0.74 0.73 0.51 0.71 0.76 0.2 (0.525)
StructuralAlerts	0.76 0.6 0.75 0.4 0.76 0.77 0.77 0.72 0.79 0.88 0.68 (0.716)		0.83 0.75 0.81 0.6 0.77 0.8 0.832 0.78 0.81 0.86 0.64 (0.771)	0.77 0.38 0.7 0.4 0.6 0.69 0.77 0.45 0.69 0.83 0.6 (0.625)		0.82 0.4 0.38 0.03 0.6 0.51 0.71 0.38 0.73 0.8 0.6 (0.542)
SIRMS (LABELING = CHARGE;LOGP;HB;REFRACTIVITY noH (1-4))	0.77 0.72 0.83 0.6 0.745 0.81 0.63 0.7 0.78 0.83 0.5 (0.72)		0.72 0.63 0.6 0.4 0.5 0.5 0.69 0.6 0.65 0.69 0.41 (0.581)	0.79 0.4 0.75 0.09 0.61 0.7 0.59 0.45 0.7 0.78 0.5 (0.578)		0.74 0.32 0.14 0.07 0.1 0.25 0.67 0.33 0.64 0.68 0.3 (0.385)
MOPAC2016 (MOPAC basic)	0.62 0.57 0.5 0.3 0.5 0.5 0.68 0.4 0.61 0.61 0.6 (0.535)		0.61 0.67 0.77 0.45 0.72 0.75 0.69 0.37 0.55 0.51 0.19 (0.571)	0.66 0.51 0.57 0.4 0.5 0.6 0.75 0.5 0.65 0.64 0.6 (0.58)		0.58 0.3 0.1 0.01 0.02 0.1 0.59 0.01 0.35 0.4 0.09 (0.232)
PyDescriptor (PyDescriptor)	0.86 0.72 0.89 0.8 0.8 0.86 0.86 0.65 0.76 0.84 0.55 (0.781)		0.88 0.85 0.89 0.75 0.86 0.87 0.89 0.75 0.86 0.88 0.55 (0.821)	0.89 0.6 0.61 0.4 0.53 0.65 0.81 0.3 0.78 0.84 0.67 (0.644)		0.85 0.68 0.6 0.4 0.5 0.59 0.85 0.49 0.73 0.81 0.57 (0.643)

	DAG	TEXTCNN	DAG(2)	TEXTCNN(2)
DEEPCHEM	0.51 0.62 0.54 0.6 0.3 0.5 0.74 0.33 0.59 0.61 0.2 (0.504)	0.8 0.74 0.8 0.66 0.75 0.79 0.85 0.51 0.77 0.78 0.7 (0.741)	0.71 0.6 0.3 0.34 0.02 0.1 0.8 0.16 0.64 0.67 0 (0.395)	0.72 0.4 0.64 0.2 0.5 0.35 0.78 0.13 0.69 0.74 0.49 (0.513)

	GGNN	NFP	GGNN(2)	NFP(2)
CHEMCHAINER	0.81 0.86 0.71 0.75 0.71 0.66 0.88 0.59 0.79 0.82 0.5 (0.735)	0.76 0.77 0.74 0.55 0.72 0.66 0.8 0.42 0.67 0.72 0.3 (0.646)	0.7 0.4 0.06 0.53 0.1 0.05 0.84 0.09 0.72 0.72 0.2 (0.401)	0.76 0.49 0.22 0.2 0.54 0.11 0.82 0.02 0.55 0.63 0 (0.395)

Figure 4. Example of MTL and STL using the comprehensive-modelling view of the OCHEM platform. The RMSE of models on the left-side columns (MTL) provide a higher squared correlation coefficient, R^2, than models developed for each analysed property regardless of the descriptor set or method used. The models developed using DEEPCHEM and ChemChainer are based on chemical graphs. The values in parentheses indicate the average value or R^2 for each analysis. ASNN - Associative Neural Networks [6]; DNN - Deep Neural Network [16]; DAG - Directed Acyclic Graphs [42]; TEXTCNN - Text

Convolutional Neural Network [43]; NFP – Neural Network Fingerprint [36]; GGCN - Gated Graph Neural Network [41]; STL - S

Message Passing Neural Networks (MPNN)[44] are a generalisation of neural network architectures, which operate on graphs and update their node states using message passing. Examples of such networks are the NNF, GGNN, Weave and RSGCN networks considered above. The developed network was based on the GGNN architecture and had several improvements to decrease the computational cost and increase performance, e.g. optimisation of the final layers of the network (readout function which maps the whole graph to a feature vector), improvement of the scalability of training, etc. This allowed the authors to achieve superior results for 13 targets when co-modelling electronic and energetic properties of molecules.

Directed Acyclic Graphs (DAG)[42] (or DAG Recursive Neural Network) consider molecules as directed graphs by iteratively taking each atom as a central one and defining the directions of all other bonds as outgoing from the central atom. The algorithm uses the atoms and their atomic features to propagate information through the graph to calculate properties. This operation is repeated for all atoms in a molecule and the result is used to train a neural network.

Influence Relevance Voters (IRV)[45] is a variation of a metric-learning algorithm applied to molecular graphs. The motivation of this algorithm was to simulate the ability of humans to learn using just few examples or in a limit with a single example.

Text Convolutional Neural Networks (TEXTCNN)[43] uses neural network vectors trained on billions of words from Google News. These pre-trained vectors serve as "universal" feature extractors that can be used to achieve excellent results for various problems. The method was adapted to work with SMILES by the developers of DEEPCHEM.

The variety of powerful and freely accessible methods will enable their wide use to address various multi-learning tasks.

Open issues

Despite the promising performance of MTL there are several issues, which either have not been properly addressed or remain open. Surprisingly, there is no good understanding as to which tasks are considered similar and could thus profit from multi-learning [13,46-48]. The main outstanding issue being that some tasks help each other and some do not; some compete for network capacity so that training them together actually worsens performance. Chen et al.[47] stressed that, in general, multi-learning neural networks can be rather hard to train because different tasks bring imbalances in the gradient calculations. The authors proposed an adaptive algorithm to estimate the weights of tasks dynamically during the training to improve prediction accuracy. Much remains to be explored in the design of neural network architectures, especially in the area of DNNs. A recent publication by Sturm et al[49] analysing the performance of DNNs on the ExCAPE-DB of 70 million SAR datapoints, demonstrated a large dependency of the performance upon the hyperparameter choices. Optimising such parameters can be a costly operation, so determining general guidelines for estimating initial settings should be a point of future investigation. However, one can also formulate an even broader question*: "Can we derive non-linear dependences between tasks from data and use them to improve multi-task learning?"* Zamir et al. (a best paper award at the CVPR2018 conference)[48] provided a method for automatic creation of taxonomy graphs for tasks. This approach has great prospects in chemoinformatics, e.g., for deriving and using the taxonomy of protein targets, viruses, toxicity endpoints, etc. in a fully data-driven mode.

Summary

The multi-task learning approaches are gaining popularity in various fields of science, including chemoinformatics. Successful use of these methods can result in models with higher prediction accuracies compared to the development of models for each individual property. The conditions when MTL can provide better results over STL are clearly formulated by Xu et al [17]. As concluded by the authors MTL should be used for modelling correlated properties, but will degrade performance for uncorrelated properties. Structurally dissimilar molecules have no influence on the predictive performance of MTL, regardless of whether or not tasks are correlated. While these recommendations were for deep neural networks, they are likely to be valid for other multi-learning approaches too and should be considered before deciding whether an MTL method can be employed. Finally, the development of a single MTL model is much faster and such a model occupies less memory and disk space compared to multiple single task models. This feature becomes important when increasing the number of simultaneously analysed properties. Examples of data sets that could potentially benefit from transfer learning and MTL with regards to QSAR modelling are given by Simoes et al[50] and include a) similar compounds measured under different experimental conditions; b) antimicrobial activities against genetically similar microorganisms; c) compounds with the same mechanism of action in homologous targets and high degrees of similarity in the binding pocket; d) non-specific endpoints such as toxicity. When the endpoint has been measured exactly, but under different conditions or on e.g. different but correlated target organisms, one can also encode conditions as input descriptors. The availability of tools to perform multi-learning is important for the widespread adoption and use of these methods by the scientific community.

Outlook

Both industrial and academic partners share high expectations from “Big Data” in chemistry, which is a new emerging area of research on the borders of several disciplines [1]. Transductive learning in general, as well as multi-learning approaches, will help to fully exploit the potential of such data by contributing models with higher prediction ability and coverage. These approaches will be important within the new federated learning project, a call for which was recently launched by the IMI. The future developments in this area should accommodate different data precision and accuracy from different sources, unbalanced datasets as well as sound calculation of the applicability domain and accuracy of predictions of multi-models, which will be important for the use of these methods. Moreover, MTL can be combined with other types of networks, such as Recurrent Neural Networks (RNNs), to automatically design new chemicals with desired predicted properties [51].

Acknowledgements

The project leading to this article has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 676434, “Big Data in Chemistry”. The article reflects only the authors’ view and neither the European Commission nor the Research Executive Agency are responsible for any use that may be made of the information it contains. This study was partially supported by Russian Science Foundation, grant No 14-43-00024. The authors would like to thank Prof. Victor Lempitsky for his helpful remarks.

Conflict of interests

IVT is CEO of BIGCHEM GmbH, which licenses the OCHEM [32]. The other authors declared that they have no actual or potential conflicts of interests.

References

1. Tetko, I.V.; Engkvist, O.; Koch, U.; Reymond, J.L.; Chen, H. Bigchem: Challenges and opportunities for big data analysis in chemistry. *Mol. Inform.* **2016**, *35*, 615-621.
2. Baskin, I.I.; Winkler, D.; Tetko, I.V. A renaissance of neural networks in drug discovery. *Expert Opin. Drug Discov.* **2016**, *11*, 785-795.
3. Comer, J. In *The relationships between lipophilicity, solubility and pka for ionizable molecules*, PhysChem forum for Physical Chemists by Physical Chemists, UK, 2005; UK.
4. Caruana, R. Multitask learning. In *Learning to learn*, Thrun, S.; Pratt, L., Eds. Springer US: Boston, MA, 1998; pp 95-133.
5. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* **2010**, *22*, 1345-1359.
6. Tetko, I.V. Associative neural network. *Methods Mol. Biol.* **2008**, *458*, 185-202.
7. Tetko, I.V.; Poda, G.I. Application of alogps 2.1 to predict log d distribution coefficient for pfizer proprietary compounds. *J. Med. Chem.* **2004**, *47*, 5601-5604.
8. Varnek, A.; Gaudin, C.; Marcou, G.; Baskin, I.; Pandey, A.K.; Tetko, I.V. Inductive transfer of knowledge: Application of multi-task learning and feature net approaches to model tissue-air partition coefficients. *J. Chem. Inf. Model.* **2009**, *49*, 133-144.
9. Zhang, Y.; Yang, Q. A survey on multi-task learning. *eprint arXiv:1707.08114* **2017**, arXiv:1707.08114.
10. Bickel, S.; Bogojeska, J.; Lengauer, T.; Scheffer, T. Multi-task learning for hiv therapy screening. In *Proceedings of the 25th international conference on Machine learning*, ACM: Helsinki, Finland, 2008; pp 56-63.
11. Suddarth, S.C.; Kergosien, Y.L. In *Rule-injection hints as a means of improving network performance and learning time*, Neural Netw., Berlin, Heidelberg, 1990//, 1990; Almeida, L.B.; Wellekens, C.J., Eds. Springer Berlin Heidelberg: Berlin, Heidelberg, pp 120-129.
12. Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. Deeptox: Toxicity prediction using deep learning. *Frontiers Environ. Sci.* **2016**, *3*, 80.
13. Ruder, S. An overview of multi-task learning in deep neural networks. *eprint arXiv:1706.05098* **2017**, arXiv:1706.05098.
14. 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.
15. Abdelaziz, A.; Spahn-Langguth, H.; Werner-Schramm, K.; Tetko, I.V. Consensus modeling for hts assays using in silico descriptors calculates the best balanced accuracy in tox21 challenge. *Frontiers Environ. Sci.* **2016**, *4*, 2.
16. Sosnin, S.; Karlov, D.; Tetko, I.V.; Fedorov, M.V. A comparative study of prediction of multi-target toxicity for a broad chemical space. *J. Chem. Inf. Model.* **2018**, *in press*.
17. 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.
18. Xu, Y.; Pei, J.; Lai, L. Deep learning based regression and multiclass models for acute oral toxicity prediction with automatic chemical feature extraction. *J. Chem. Inf. Model.* **2017**, *57*, 2672-2685.
19. Li, X.; Xu, Y.; Lai, L.; Pei, J. Prediction of human cytochrome p450 inhibition using a multitask deep autoencoder neural network. *Mol. Pharm.* **2018**.
20. Simm, J.; Arany, A.; Zakeri, P.; Haber, T.; Wegner, J.K.; Chupakhin, V.; Ceulemans, H.; Moreau, Y. Macau: Scalable bayesian multi-relational factorization with side information using mcmc. *ArXiv e-prints* **2015**, 1509.04610.
21. de la Vega de Leon, A.; Chen, B.; Gillet, V.J. Effect of missing data on multitask prediction methods. *J. Cheminform.* **2018**, *10*, 26.

22. Parameswaran, S.; Weinberger, K.Q. Large margin multi-task metric learning. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, Curran Associates Inc.: Vancouver, British Columbia, Canada, 2010; pp 1867-1875.
23. Yang, P.; Huang, K.; Hussain, A. A review on multi-task metric learning. *Big Data Analytics* **2018**, *3*, 3.
24. Xu, S.; An, X.; Qiao, X.; Zhu, L.; Li, L. Multi-output least-squares support vector regression machines. *Pattern Recognition Letters* **2013**, *34*, 1078-1084.
25. Rozantsev, A.; Salzmann, M.; Fua, P. Beyond sharing weights for deep domain adaptation. *eprint arXiv:1603.06432* **2016**, arXiv:1603.06432.
26. Rosenbaum, L.; Dorr, A.; Bauer, M.R.; Boeckler, F.M.; Zell, A. Inferring multi-target qsar models with taxonomy-based multi-task learning. *J. Cheminform.* **2013**, *5*, 33.
27. Bishop, C.M.; Svensén, M.; Williams, C.K.I. Gtm: The generative topographic mapping. *Neural Comput.* **1998**, *10*, 215-234.
28. Gaspar, H.A.; Baskin, I.I.; Marcou, G.; Horvath, D.; Varnek, A. Gtm-based qsar models and their applicability domains. *Mol. Inf.* **2015**, *34*, 348-356.
29. Gaspar, H.A.; Baskin, I.I.; Marcou, G.; Horvath, D.; Varnek, A. Stargate gtm: Bridging descriptor and activity spaces. *J. Chem. Inf. Model.* **2015**, *55*, 2403-2410.
30. Sheridan, R.P. Time-split cross-validation as a method for estimating the goodness of prospective prediction. *J. Chem. Inf. Model.* **2013**, *53*, 783-790.
31. Landrum, G.A. Rdkit, open-source cheminformatics. http://www.rdkit.org
32. Sushko, I.; Novotarskyi, S.; Korner, R.; Pandey, A.K.; Rupp, M.; Teetz, W.; Brandmaier, S.; Abdelaziz, A.; Prokopenko, V.V.; Tanchuk, V.Y.*, et al.* Online chemical modeling environment (ochem): Web platform for data storage, model development and publishing of chemical information. *J. Comput. Aided. Mol. Des.* **2011**, *25*, 533-554.
33. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293-300.
34. Gori, M.; Monfardini, G.; Scarselli, F. In *A new model for learning in graph domains*, Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., 31 July-4 Aug. 2005, 2005; pp 729-734 vol. 722.
35. Baskin, I.I.; Palyulin, V.A.; Zefirov, N.S. A neural device for searching direct correlations between structures and properties of chemical compounds. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 715-721.
36. Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R.P. Convolutional networks on graphs for learning molecular fingerprints. *ArXiv e-prints* **2015**, 1509.09292v09292.
37. Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: Moving beyond fingerprints. *J. Comput. Aided. Mol. Des.* **2016**, *30*, 595-608.
38. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *CoRR* **2016**, *abs/1609.02907*.
39. Schütt, K.T.; Kindermans, P.-J.; Sauceda, H.E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *eprint arXiv:1706.08566* **2017**, arXiv:1706.08566.
40. Cramer, R.D.; Patterson, D.E.; Bunce, J.D. Comparative molecular field analysis (comfa). 1. Effect of shape on binding of steroids to carrier proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959-5967.
41. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated graph sequence neural networks. *eprint arXiv:1511.05493* **2015**, arXiv:1511.05493.
42. Lusci, A.; Pollastri, G.; Baldi, P. Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. *J. Chem. Inf. Model.* **2013**, *53*, 1563-1575.

43. Kim, Y. In *Convolutional neural networks for sentence classification*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014//, 2014; Association for Computational Linguistics: pp 1746-1751.
44. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. *ArXiv e-prints* **2017**, *1704*, arXiv:1704.01212.
45. Altae-Tran, H.; Ramsundar, B.; Pappu, A.S.; Pande, V. Low data drug discovery with one-shot learning. *ACS Cent. Sci.* **2017**.
46. Kokkinos, I. Ubernet: Training a `universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *eprint arXiv:1609.02132* **2016**, arXiv:1609.02132.
47. Chen, Z.; Badrinarayanan, V.; Lee, C.-Y.; Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *eprint arXiv:1711.02257* **2017**, arXiv:1711.02257.
48. Zamir, A.; Sax, A.; Shen, W.; Guibas, L.; Malik, J.; Savarese, S. Taskonomy: Disentangling task transfer learning. *eprint arXiv:1804.08328* **2018**, arXiv:1804.08328.
49. Noé, S.; Jiangming, S.; Yves, V.; Andreas, M.; Günter, K.; Lars-Anders, C.; Ola, E.; Hongming, C. *Application of bioactivity profile based fingerprints for building machine learning models*. 2018.
50. Simoes, R.S.; Maltarollo, V.G.; Oliveira, P.R.; Honorio, K.M. Transfer and multi-task learning in qsar modeling: Advances and challenges. *Front. Pharmacol.* **2018**, *9*, 74.
51. Gomez-Bombarelli, R.; Wei, J.N.; Duvenaud, D.; Hernandez-Lobato, J.M.; Sanchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T.D.; Adams, R.P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268-276.