Perception of Chemical Bonds via Machine Learning

Christoph Loschen*

-

E-mail: loschen@gmail.com

Abstract

An approach based on machine-learning is presented that is able to identify chemical bond types such as single, double, triple and aromatic bonds based on spatial atomic coordinates only, as provided for example from quantum chemical calculations or from crystallographic data. The basic idea behind this work is to exploit the various chemical knowledge already assembled in molecular data files in form of connection tables and bond blocks. Rules for novel chemistry or particular functional groups can be learned automatically by training on structure data (.sd/.sdf) files with the respective bond information. Provided that the underlying database is sufficiently large and diverse, the approach is able to identify chemical bond orders in molecules with an accuracy comparable to classical bond-perception tools using hard-coded rules and cheminformatic algorithms. The workflow is implemented in Python using the open source packages RDKit, scikit-learn and pandas (https://github.com/CHLoschen/mamba).

Introduction

The chemical bond is undoubtfully a rather useful concept, as it allows to rationalize and to work in practice with the manifolds of molecular structures available from chemical space.

Consequently, many work flows in computational chemistry and cheminformatics need information on chemical bond types or bond orders for further processing. Prominent examples are mechanical force-fields, molecular 2D or 3D viewers[1,2] and database identifiers.[3] However, chemical bond types are not physcial observables and raw chemical data as provided by quantum chemical structure optimization or by crystallography only obtain spatial atomic coordinates, atomic numbers and total charge as basic information. Therefore, a pre-processing step has to be applied which is termed bond (order) perception. The aim of the bond perception is to identify chemical bond types (i.e. single, double, triple, aromatic etc.) between atoms which define the molecule. Usually quite elaborated algorithms are necessary to deal with all kind of possible occuring special cases and many different bond perception routines have been published. [4,5,5–17] In basically all those approaches, particular rules are used and implemented in order to identify particular bond types. Wang et al. presented an algorithm to determine atom types that are pre-defined in a description table, and an algorithm of assigning bond types based on atomic connectivity.[8] Bruno et al. developed a rule based method that mainly target crystal structures but is also applicable to organometallic compounds.[12] Vanommeslaeghe have published a routine for bond perception and atom typing for the CHARMM General Force Field (CGenFF).[14] Several descriptors of bonds and atoms are used such as valence, bond order, and ring membership. This information is then used to assign CHARMM atom types based on a programmable decision tree. Zhang et al. developed a rule based algorithm for bond perception based on several hard coded guidelines such as so-called hard, length and conjugation rules.[13] An open-source bond-perception has been implemented within the openbabel project.[18] It first defines the bond connectivity based on the atoms covalent radii and then bond orders are determined based on bond angles and geometry similar to on an algorithm proposed by Sayle.[19]

Machine learning features from distance matrices such as bag of bonds and distance histograms have been used in the context of the prediction of quantum-mechanically computed molecular properties.[20,21] In general, so-called machine learning methods are mostly used in

the context of quantitative structure activity or property relationships.[22]

Recently an perception approach using machine learning, in particular support vector machines has been published by Kadukova et al., which is using different cheminformatic descriptors in order to define bonds and atom types.[23] They first distribute atoms and bonds into classification types by some hard-coded rules and then use specifically trained support vector machine classifiers to predict hydridization states and bond orders. The partitioning into classification types affords some prior cheminformatics computations such as establishing a molecular graph and a ring finding algorithm. A typical disadvantage of support vector machine algorithms is that they do not scale well with large sample sizes and training becomes very costly on large data sets.

Those hard-coded rule-based bond percpetion methods have in common often that they are only valid for certain kind of compounds or functional groups or specific structural sources such as the protein daba bank or quantum chemical calculations.

Here, a bond perception algorithm is reported that is using a simple set of features created from the molecular distance matrix, whereas predictions are generated via decision tree based approaches such as a Random Forest and Gradient Boosting which are known for their accuracy and efficiency on large datasets.[24] As input to the learning algorithm a set of structural data files (SDF) can be provided containing bond orders and 3D atomic coordinates.

Results and discussion

The essential idea of the present approach is to computationally learn from well-defined molecular structure files (such as .sd/.sdf files) containing already bond order/type information. This assembled knowledge can then be used to predict bond types from raw atomic coordinates as obtained from quantum chemical and crystallographic data. The set of molecular structure files used for learning may be constructed by different bond perception algo-

rithms and should be as diverse as possible. In order to be usable by a machine learnig algorithm either for training or prediction the molecular data needs to be cast in a suitable digestable format. Figure 1 illustrates the concept that is used for the construction of the feature matrix. First, the 3D coordinates of the n atoms are used to build up the $n \times n$ euclidean distance matrix of the molecule. This guarantees invariance of the input under translation and rotation of the coordinate system. For all distances d below a certain threshold a feature vector is constructed.

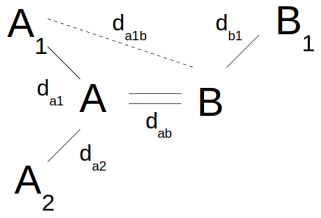

Figure 1: Illustration of how the distance information from the neighborhood of bonded atoms A and B is used to build the feature matrix. Each distance below a certain threshold serves as a feature vector.

The feature (row) vector contains the atomic number of atom A and B and the corresponding distance. A an B are ordered according to their atomic number. Then, for the first atom A its m closest neighbors and their respective distance are attached to feature vector - the same for the second original atom B. Missing entries for atoms with a coordination number less than the maximum allowed valence (e.g. 4 or 6) get an arbitrary value for the atomic number (e.g. 999). Here, the coordination number is defined as the number of atoms that are within the given cut off radius. Furthermore, the distances of all neighbors of A to atom B and vice versa are used as descriptors (d_{a1b} in Figure 1). The addition of this last set of features ensures, that enough structural information is provided which allows for example to learn implicitly the bond angles (e.g. from the law of cosines) sourrounding the target bond.

Finally, a target column vector consisting out of the bond types for each row is constructed. Allowed bond types are encoded as integers and correspond to the common con-

vention: 0 for no bond, 1 for single bonds, 2 for double bonds, 3 four triple bonds and 4 for aromatic bonds. An example exerpt of the feature matrix is shown in table 1.

Table 1: An example of the feature matrix as used for training and prediction. Each row represents a possible bond. In this example the maximum coordination number is 4. Empty cells in rows of low-coordinated atoms with a valence less than 4 (i.e. no further nearby atom below the distance threshold) get the arbitrary "atomic number" 0 as well as a arbitrary "distance" of 9999.

A	B	d_{ab}	A1	d_{a1}	A2	d_{a2}	A3	d_{a3}	B1	d_{b1}	B2	d_{b2}	B3	d_{b3}	bond
6	6	1.33	1	1.08	1	1.08	1	2.09	1	1.08	7	1.43	1	2.09	2
7	6	2.40	1	1.04	1	1.04	6	1.43	1	1.08	1	1.08	6	1.33	0
6	1	1.08	1	1.08	6	1.33	1	2.09	1	1.87	6	2.09	1	2.41	1
6	1	1.08	1	1.08	6	1.33	1	2.09	1	1.87	6	2.10	1	2.46	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
6	1	2.16	1	1.08	6	1.33	7	1.43	7	1.04	1	1.80	1	2.46	0
7	1	2.64	1	1.04	1	1.04	6	1.43	6	1.08	1	1.87	6	2.10	0
7	1	2.17	1	1.04	1	1.04	6	1.43	6	1.08	6	2.09	1	2.41	0
7	1	1.04	1	1.04	6	1.43	1	2.17	1	1.80	6	2.15	1	2.48	1
7	1	1.04	1	1.04	6	1.43	1	2.17	1	1.80	6	2.16	1	2.46	1

Using raw features gives already good classification results, even for untidy structural data containing a few unreasonable bond lengths being either too long or too short. Nevertheless, some care has to be taken not to train the classification algorithm with too noisy information in order not to deterior its predictive capabilities. Therefore, for the construction of the training feature matrix some simple checks were introduced, for example clearly unreasonable long bonds were removed such as C-C bonds longer than 1.8Å. In such cases the whole molecule is dumped and not used for training at all. As the target vector contains the bond types encoded as integers from 0 to 4 this results in a multiclass classification problem.

The feature matrix and the target vector as constructed from many diverse molecules are then fed into a machine learning algorithm, in this case either bagged[25] or boosted[26] decision trees. Probably other machine learning algorithms would be suitable for this problem as well, however best performances were obtained using decision tree based algorithms. Deep neural networks were tried as well but did not achieve a comparable performance on these data with

a tabular data structure. As a typical bagging algorithm, a Random Forest fits a decision tree many times to different subsets of the data (usually bootstrapped), and finally a vote each is casted from the ensemble of fitted trees in order to predict the final class (here bond type).[24] For Gradient boosting many decision trees are trained sequentially, whereas the loss function is minimized by each model using the gradient descent method.[24] For model building the classifiers as implemented in the Python based scikit-learn library have been used.[27]

In addition to the prediction of a final bond type / class, the voting mechanism also allows for the precition of probabilities for each of the possible classes, which can also be a valuable piece of information, see also Table 2. Random forests also have the advantage that no extensive parameter fitting is necessary, and good results are obtained using a set of standard parameters. Therefore, no extensive cross-validation is afforded. The following scikit-learn parameters were obtained on 5-fold cross-validation loop for the first training set: ($n\_estimators = 250$), i.e. 250 trees considering 50% of the features at each split of the construction of the tree ($max\_features = 0.5$). Gradient boosting in general needs more thorough parameter tuning, here we arrived after cross-validation at the following set of parameters: $n\_estimators = 1000$, $learning\_rate = 0.1$ and $max\_depth = 5$.

Table 2: Exemplary results for bond type prediction. In addition to the atomic numbers for A and B, and the bond type, probabilities p are specified for each bond type (p(-): single bonds, p(=): double bonds, p(#): triple bonds, p(a): aromatic bonds, p(X): no bond.

A	B	d_{ab}	bond	p(-)	p(=)	p(#)	p(a)	p(X)
6	6	1.329	2	0.0145	0.950	0	0.036	0
7	6	2.400	0	0	0	0	0	1
6	1	1.082	1	1	0	0	0	0
6	1	1.082	1	1	0	0	0	0
6	1	2.087	0	0	0	0	0	1
6	1	2.667	0	0	0	0	0	1
7	6	1.427	1	0.701	0.0777	0	0.218	0.002
6	1	2.092	0	0	0	0	0	1
6	1	2.101	0	0	0	0	0	1
6	1	1.082	1	1	0	0	0	0

In order to evaluate the approach several different datasets were generated. First, a sufficient large set of about 28000 smiles[16] was randomly divided in a train and a test set. Then 3-dimensional structures were created using the distance geometry approach as implemented in RDKit.[28] Subsequently, the structures were optimized with the UFF (Universal Force-Field),[29] the MMFF and also refined using the empirical ETKDG approach by Riniker and Landrum,[30] as implemented in the RDKit, yielding 3 strucurally different data sets. Force-field optimization have been stopped after 200 iterations, the idea behind this was besides to speed-up data set generation, to introduce some slight structural noise and hence make the training less prone to overfitting. Altogether about 14000 molecules and 1.6M bonds are used in those 3 trainings and test sets respectively. Another dataset consisting mainly out of drug molecules stemming from different sources as collected in a 3Dqsar study[31] was used. The structures from the datatset gdb1k are mainly optimized by density functional theory and are taken from the github repository of the deepchem project.[32] Their bond types have been obtained originally via Open Babel. Finally, in order to evaluate the approach for cases where typically no information on the position of the hydrogen atoms are available, a subset of the refined pdbbind dataset was used,[33] containing the coordinates of the heavy atoms (i.e. no hydrogen) of about 3500 ligands from pdb structures (Table 3).

Table 3: Different datasets as used for training and testing. Number of molecules and bonds in the training and test datasets are given as n_m and n_b,respectively.

label	source	n_m,train	n_b,train	n_m,test	n_b,test
bradley1	melting point dataset,UFF[16]	14140	1660483	14160	1681224
bradley2	melting point dataset,MMFF[16]	13979	1653215	14005	1673920
bradley3	melting point dataset,ETDKG[16]	14153	1678074	14191	1699041
3dqsar	3Dqsar dataset[31]	990	179311	259	46156
gdb1k	qm optimized dataset[32]	787	41748	213	11506
pdbbind	structures from pdb,no H[33]	3535	279389	817	65921

During the processing of all those datasets a few basic rules were introduced in order to get rid of obviously non-reasonable bond information, such as much too short or very long bond distances. In order to asses the quality of the approach, a comparison with the bond

perception algorithm implemented in the open source code Open Babel[18] was carried out by creating SD files and a subsequent analysis of the predicted bonds. For the comparison, hydrogen have been completely removed in order to avoid difficulties with re-ordering of X-H bonds.

Table 4: Accuracy for Random Forest classifiers (RF) and Gradient boosting classifiers (GB) on different evaluation datasets. The F1-score for RF predictions is computed per predicted sample(bond). The accuracy acc(mol) is computed per molecule. Additionally, the accuracy is given for the same evaluation sets using a 30% fraction of all datasets (except pdbbind) acc(mols,all) for training. Finally, the accuracy using the Open Babel bond perception algorithm (OB) is given for comparison.

	RF			GB		OB
dataset	F1-score	acc(mol)	acc(mol,all)	acc(mol)	acc(mol,all)	acc
bradley1	1.000	0.997	0.993	0.997	0.994	0.941
bradley2	0.999	0.914	0.882	0.905	0.976	0.933
bradley3	1.000	0.982	0.969	0.988	0.883	0.941
3dqsar	0.999	0.907	0.857	0.950	0.915	0.952
gdb1k	0.998	0.953	0.887	0.934	0.901	(0.990)
pdbbind	0.994	0.800	-	0.918	-	-
mean	1.00	0.93	0.92	0.95	0.93	0.94

The f1-score for each class is shown in Table 4, which is just the harmonic mean of precision and recall: For a specific bond type, precision gives the fraction of correctly classified bonds among all the predicted bonds of that type, while recall (also known as sensitivity) is the fraction of correctly classified bonds that have been retrieved from all existing bonds of that type. The high F1-score can be a bit misleading, as in practice a false predicted bond renders the prediction for the whole molecule wrong. Hence, a somewhat more meaningful metric, the prediction accuracy with regard to whole molecules is computed.

Compared to the open babel predictions results are competitive in allmost all cases, given an accuracy of more than 90%. As the predictions using only a single classifier trained on a subset of the first five datasets (acc(mol,all)) shows, the amount of overfitting to a certain Forcefield or data source is low. If hydrogens are omitted (pdbbind) from the input coordinates the approach also achieves a high accuracy even on a comparatively noisy dataset

as the refined pddbind structures. Note, that on the pdbbind data set only mediocre results could be obtained with Open Babel, giving an accuracy of only 0.7, which is probably due to the low quality of the data set concerning bond types. Furthermore, the gdb1k dataset bond types have been originally obtained by Open Babel, explaining the high accuracy score for respective prediction. Both scores have been removed from the overall average.

Although the prediction accuracy per molecule is the more meaningful metric, breaking down the accuracy for specific bonds can give some additional insights. Figure 2 shows the confusion matrix which is obtained after training on a 30% subsets of datasets bradley1-3, gdb1 and 3dqsar. The matrix shows that the partitioning into bonds and non-bonds is pretty flawless, as only 3 out of more than 270000 bonds have not been recognized correctly as bonds. Furthermore, most failures seems to occur for aromatic bonds, that are not recognized as such (about 300 of 270000).

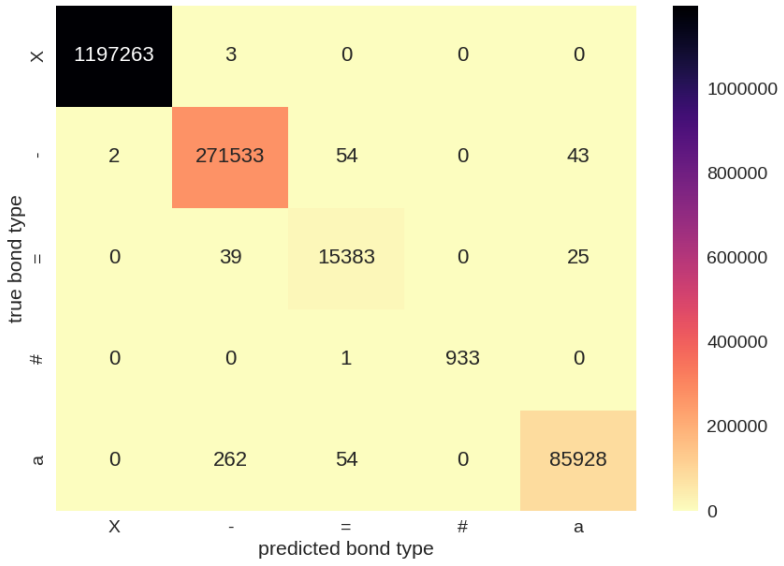

Figure 2: Confusion matrix for predictions on all evaluation sets.

The approach was been further visually evaluated on some difficult cases which have been reported previously.[14] Figure 3 displays predicted structures and bonds from some molecules denoted as *special* cases published along the work of Vanommeslaeghe.[14]

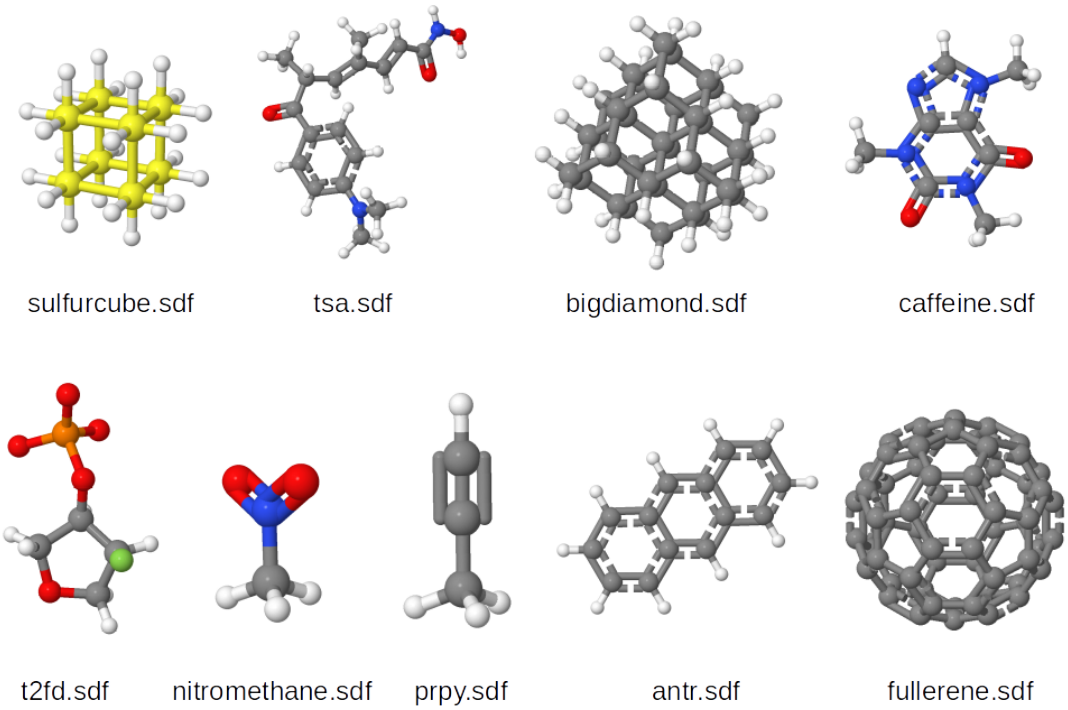

Figure 3: Predicted bonds for molecules from the dataset from the work of Vanommeslaeghe.[14] Aromatic bonds are shown as dashed cylinders in order to differentiate them from usual double bonds. Images were created with jmol using the option "set autobond off".

An interesting case is the outcome of the prediction for fullerene (fullerene.sdf from the special set). The shorter bonds (1.38Å) of the 6-ring members are predicted to be aromatic with a significant double bond character, whereas longer bonds of the 5-rings (1.46Å) are predicted to be single bonds.[34] This corresponds to the common description of the bonding situation in fullerene as being composed of hexatrienes and 5-radialenes.[35]

An yet unsolved problem are non-aromatic delocalized bonds as for example the two identical C-O bonds in the carboxylate-anion. The coordinates of acetate coming from a quantum chemical calculation will have both C-O bonds with identical length in a symmetric environment. Therefore, a standard machine-learning algorithm will assign identical probabilities that are somewhat between a single and a double bond to those two bonds. Symmetry breaking into a $C - O^-$ single and a $C = O$ double bond can not occur. Another example is the nitro-group in nitromethane (Figure 3) which is predicted to have a 5-coordinate nitrogen center, which may chemically not be correct, however is a common alternative description of this group often used in SMILES notation. This problem may be considered as a consequence of shortcomings of chemical bonding theory, which is unable to assign a single unique Lewis structure to some bonding situations. The issue may possibly be solved by stacking two classifiers in series, whereas the second one incorporates the predicted bonds from the first one to break the symmetry constraints. Some promising attempts have been undertaken in this direction however more work is needed to refine this approach, which may be covered in a subsequent study.

The classification accuracy is 93% on the average and as such comparable to other bond perception algorithms. There is probably still some optimization potential of this approach and also there may be some failures for special functional groups, that have been missed so far in the training sets. However, in contrast to classical algorithms where a failure requires a code modification, the machine learning approach needs just some new or improved training examples for an update. Those examples can be provided by anyone anytime as desired for a specific compound class, which can be considered as a significant advantage in terms of

usability compared to classical algorithms.

Conclusion

A machine-learning approach is presented that determines chemical bond types from spatial molecular coordinates. The method adopts prior knowledge from other bond perception algorithms in a rather automatic and general fashion. The approach is also rather flexible in the sense that it can easily be adopted for problematic cases by just updating it with the corresponding structure data files without the need to modify the program code. For example the approach can simply be adopted to the prediction of pdb ligand structures which may have hydrogen atoms missing by omitting hydrogen positions during training or by training directly on such pdb data. Furthermore, accurately predicted bond types should provide all the necessary ingredients for an extension of the approach for the automatic detection of forcefield atom types. Of course, the quality of the predictions largely depends on the quality of the input stream and hence input data should be carefully selected in order to avoid misteaching of the machine-learning algorithm.

Supplemental material

A Python script is provided that can be trained on either SMILES files (.smi) or structure data files (.sdf) and can then be used to convert .xyz or .pdb coordinate files to .sdf files. The script can be found at: https://github.com/CHLoschen/mamba

References

(1) Steffen, C.; Thomas, K.; Huniar, U.; Hellweg, A.; Rubner, O.; Schroer, A. TmoleX - A Graphical User Interface for TURBOMOLE. *31*, 2967–2970.

(2) Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchi-

son, G. R. Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *Journal of Cheminformatics* **2012**, *4*, 17.

(3) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics* **2015**, *7*, 23.

(4) Baber, J. C.; Hodgkin, E. E. Automatic assignment of chemical connectivity to organic molecules in the Cambridge Structural Database. *Journal of Chemical Information and Computer Sciences* **1992**, *32*, 401–406.

(5) Hendlich, M.; Rippmann, F.; Barnickel, G. BALI: Automatic Assignment of Bond and Atom Types for Protein Ligands in the Brookhaven Protein Databank. *Journal of Chemical Information and Computer Sciences* **1997**, *37*, 774–778.

(6) Froeyen, M.; Herdewijn, P. Correct Bond Order Assignment in a Molecular Framework Using Integer Linear Programming with Application to Molecules Where Only Non-Hydrogen Atom Coordinates Are Available. *Journal of Chemical Information and Modeling* **2005**, *45*, 1267–1274, PMID: 16180903.

(7) Labute, P. On the Perception of Molecules from 3D Atomic Coordinates. *Journal of Chemical Information and Modeling* **2005**, *45*, 215–221, PMID: 15807481.

(8) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling* **2006**, *25*, 247 – 260.

(9) Zhao, Y.; Cheng, T.; Wang, R. Automatic Perception of Organic Molecules Based on Essential Structural Information. *Journal of Chemical Information and Modeling* **2007**, *47*, 1379–1385, PMID: 17530839.

(10) Neudert, G.; Klebe, G. fconv: format conversion, manipulation and feature computation of molecular data. *Bioinformatics* **2011**, *27*, 1021–1022.

(11) Dehof, A. K.; Rurainski, A.; Bui, Q. B. A.; Bcker, S.; Lenhof, H.-P.; Hildebrandt, A. Automated bond order assignment as an optimization problem. *Bioinformatics* **2011**, *27*, 619–625.

(12) Bruno, I. J.; Shields, G. P.; Taylor, R. Deducing chemical structure from crystallographically determined atomic coordinates. *Acta Crystallographica Section B* **2011**, *67*, 333–349.

(13) Zhang, Q.; Zhang, W.; Li, Y.; Wang, J.; Zhang, L.; Hou, T. A rule-based algorithm for automatic bond type perception. *Journal of Cheminformatics* **2012**, *4*, 26.

(14) Vanommeslaeghe, K.; MacKerell, A. D. Automation of the CHARMM General Force Field (CGenFF) I: Bond Perception and Atom Typing. *Journal of Chemical Information and Modeling* **2012**, *52*, 3144–3154, PMID: 23146088.

(15) Urbaczek, S.; Kolodzik, A.; Groth, I.; Heuser, S.; Rarey, M. Reading PDB: Perception of Molecules from 3D Atomic Coordinates. *Journal of Chemical Information and Modeling* **2013**, *53*, 76–87, PMID: 23176552.

(16) Bradley, J.-C.; Williams, A.; Lang, A. Jean-Claude Bradley Open Melting Point Dataset. **2014**, [Online; accessed 27-July-2018].

(17) Svetlana, A.; Lonard, J.; Stephane, R. Automatic molecular structure perception for the universal force field. *Journal of Computational Chemistry 37*, 1191–1205.

(18) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *Journal of Cheminformatics* **2011**, *3*, 33.

(19) PDB: Cruft to Content, Perception of Molecular Connectivity from 3D Coordinates. `http://www.daylight.com/meetings/mug01/Sayle/m4xbondage.html`, 2001.

(20) Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **2015**, *6*, 2326–2331, PMID: 26113956.

(21) Faber, F. A.; Hutchison, L.; Huang, B.; Gilmer, J.; Schoenholz, S. S.; Dahl, G. E.; Vinyals, O.; Kearnes, S.; Riley, P. F.; von Lilienfeld, O. A. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *Journal of Chemical Theory and Computation* **2017**, *13*, 5255–5264, PMID: 28926232.

(22) Mitchell, J. B. O. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*, 468–481.

(23) Kadukova, M.; Grudinin, S. Knodle: A Support Vector Machines-Based Automatic Perception of Organic Molecules from 3D Coordinates. *Journal of Chemical Information and Modeling* **2016**, *56*, 1410–1419, PMID: 27405533.

(24) Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer New York Inc.: New York, NY, USA, 2001.

(25) Breiman, L. Random Forests. *Machine Learning* **2001**, *45*, 5–32.

(26) Freund, Y.; Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* **1997**, *55*, 119–139.

(27) Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(28) RDKit: Open-source cheminformatics. `http://www.rdkit.org`, [Online; accessed 27-October-2018].

(29) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. UFF, a full

periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society* **1992**, *114*, 10024–10035.

(30) Riniker, S.; Landrum, G. A. Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation. *Journal of Chemical Information and Modeling* **2015**, *55*, 2562–2574, PMID: 26575315.

(31) Sutherland, J. J.; O'Brien, L. A.; Weaver, D. F. A Comparison of Methods for Modeling Quantitative Structureactivity Relationships. *47*, 5541–5554.

(32) Democratizing Deep-Learning for Drug Discovery, Quantum Chemistry, Materials Science and Biology. `https://github.com/deepchem/deepchem`, 2016.

(33) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind Database: Collection of Binding Affinities for ProteinLigand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry* **2004**, *47*, 2977–2980, PMID: 15163179.

(34) Hedberg, K.; Hedberg, L.; Bethune, D. S.; Brown, C. A.; Dorn, H. C.; Johnson, R. D.; De Vries, M. Bond Lengths in Free Molecules of Buckminsterfullerene, C60, from Gas-Phase Electron Diffraction. *Science* **1991**, *254*, 410–412.

(35) Bühl, M.; Hirsch, A. Spherical Aromaticity of Fullerenes. *Chemical Reviews* **2001**, *101*, 1153–1184, PMID: 11710216.