Cocrystal Prediction by Artificial Neural Networks

Jan-Joris Devogelaer, Hugo Meekes, Paul Tinnemans, Elias Vlieg, Rene de Gelder

Submitted date: 09/06/2020 • Posted date: 11/06/2020
Licence: CC BY-NC-ND 4.0

Citation information: Devogelaer, Jan-Joris; Meekes, Hugo; Tinnemans, Paul; Vlieg, Elias; de Gelder, Rene (2020): Cocrystal Prediction by Artificial Neural Networks. ChemRxiv. Preprint.
https://doi.org/10.26434/chemrxiv.12452849.v1

A significant amount of attention has been given to the design and synthesis of cocrystals by both industry and academia because of its potential to change a molecule’s physicochemical properties. This paper reports on the application of a data-driven cocrystal prediction method, based on two types of artificial neural network models and cocrystal data present in the Cambridge Structural Database. The models accept pairs of coformers and predict whether a cocrystal is likely to form.

File list (2)

Cocrystal prediction by artificial neural networks.pdf (2.94 MiB) view on ChemRxiv download file
Cocrystal prediction by artificial neural networks - Supplem... (2.94 MiB) view on ChemRxiv download file
Cocrystal prediction by artificial neural networks

Jan-Joris Devogelaer, Hugo Meekes, Paul Tinnemans, Elias Vlieg, and René de Gelder*

Radboud University, Institute for Molecules and Materials, Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands
E-mail: r.degelder@science.ru.nl

Abstract

A significant amount of attention has been given to the design and synthesis of cocrystals by both industry and academia because of its potential to change a molecule’s physicochemical properties. Yet, difficulties arise when searching for adequate combinations of molecules (or coformers) to form cocrystals, hampering the efficient exploration of the target’s solid-state landscape. This paper reports on the application of a data-driven cocrystal prediction method, based on two types of artificial neural network models and cocrystal data present in the Cambridge Structural Database. The models accept pairs of coformers and predict whether a cocrystal is likely to form. By combining the output of multiple models of both types, our approach shows to have excellent performance on the proposed cocrystal training and validation sets, and has an estimated accuracy of 80% for molecules for which previous cocrystallization data is unavailable.
1 Introduction

Molecular solids appear in many different ways, and the solid-state landscape of a molecule may cover various crystalline forms, ranging from polymorphs and hydrates to more complex multicomponent crystals [1,2]. The latter have been promoted as excellent tools to modify the physico-chemical characteristics of a target compound, such as the (aqueous) solubility, bioavailability, density, and melting point [3–8]. Multicomponent crystals therefore play a pivotal role in the effective formulation of pharmaceuticals and will continue to be part of the solid form screening process in drug development.

The design of multicomponent crystals is non-trivial and new forms are often identified via trial and error. Unlike salts, where proton transfer leads to strong ionic/coulombic interactions, solvates and cocrystals are assembled through weaker, non-covalent interactions (e.g. hydrogen bonding, π-π interactions,...). Such intermolecular interactions between functional groups are often used to rationalize the possibility of aggregation [9], but with no guarantee that the postulated interactions will emerge.

Whereas polymorphs, salts and solvates are commonly screened using automated high throughput systems [7,10–12], the experimental screening of cocrystals remains labour-intensive and time-consuming. In order to shorten this process, a variety of computational tools [13–24] have been developed to aid in the discovery of adequate combinations of the constituents or coformers. Although these computer-aided methods have succeeded in enhancing cocrystal screening protocols, some of the shortcomings include their bias towards small or structurally related datasets, oversimplified assumptions regarding the mechanisms of interaction, and, in some cases, their computational costs.

Recently, we introduced a holistic approach to study cocrystallization using network science and link prediction [25, 26]. Analysis of a network of coformers extracted from the Cambridge Structural database (CSD) [27] shows that, rather than being a random assembly of coformers, it represents a rational source of cocrystal information that can form a basis for prediction. Therefore, it would be very appealing to develop a method that utilizes all this cocrystal information, and is able to predict cocrystals for coformers lacking any experimental data on cocrystal formation. Such a tool would for instance enable the evaluation of the cocrystal formation propensity for in silico determined drug candidates (prior to their actual synthesis), or aid in the (co-)crystallization of molecules that are amorphous in their pure form.
Artificial neural networks, and in particular deep learning [28], have emerged as promising tools for data-driven prediction. Given an adequate molecular representation, artificial neural networks can be used to, for example, predict physico-chemical properties (e.g. solubility) or classify molecules, hereby assigning the input to a certain class (e.g. toxic or non-toxic).

Driven by the recent advances in artificial neural networks and the promising source of cocrystal information present in the CSD, we introduce a new approach to predict cocrystal formation using neural networks. Two neural network model types are introduced that each accept a pair of coformers as input and classify the combination as a possible cocrystal or not. By optimizing the configuration of each model type, we obtain several equally performing models, which are stored in model ensembles to make a combined prediction. [29] We will demonstrate the excellent performance of the model ensembles by repredicting all available binary cocrystal data in the CSD via cross-validation. In a case study involving carbamazepine, we further validate the approach by analyzing the predictions for known and experimentally tested combinations [24]. Finally, we compiled predictions for ketoprofen, a compound unknown to form cocrystals, and present its first drug-drug cocrystal with carbamazepine.

2 Results and Discussion

2.1 Overview of model design and selection

Two neural network model types are proposed to classify pairs of coformers as possible cocrystals (Figures 1a and 1b), differing in their required molecular representation as input and the initial pre-processing step (blue modules in Figure 1). A wide variety of molecular representations exist, and in an effort to encode both the functionalities and size/shape of the coformers, we opted for circular (or extended-connectivity) fingerprint vectors [30] and molecular graphs [31] as input formats (Figure 2).

Conceptually, both model types pre-process each coformer in an equal manner before combining them together to make the final prediction. Whereas the initial pre-processing step of the molecular fingerprint-based model type (FP; Figure 1a) consists of traditional hidden layers, the molecular graph-based model type uses graph convolutional layers [31–35] (GCN; Figure 1b). Due to their wide applicability and strong performance [36–46], graph convolutional neural networks have gained much popularity in recent years, as they can learn specific molecular fragments and variations thereof decisive for the prediction of a property [31], rather than relying
Figure 1: Model types and ensembles for the prediction of cocrystal formation. (a) FP: Finger- 
print vector-based model type. (b) GCN: Graph convolution-based model type. (c) Model 
ensembles return the average of the predictions of their constituents.

The transformation of a pair of coformers to a prediction of cocrystal formation is learned 
from the cocrystal data available during model training. By adjusting the neural network’s 
internal parameters, the loss parameter, related to the misclassification error on the available 
training data, is minimized. For the present research, 8050 binary cocrystals were extracted from 
the CSD and their constituents were converted to molecular graphs and fingerprint vectors. For 
the successful application of deep learning, however, a data set of invalid coformer combinations 
is also needed. A common issue with databases is their recording of successful cases only, 
not taking failed experiments into account. Therefore, it is impossible to directly extract an 
evidence-based list of invalid coformer combinations from the CSD, nor is screening literature 
for invalid combinations a feasible and unbiased option. To cope with this issue, we generated an 
equally large invalid cocrystal set using our link-prediction method [26], assigning a statistical 
likelihood to the existence of a cocrystal for two coformers based on network science. By 
restricting the sampled invalid cocrystals to highly unlikely combinations of coformers that 
each have at least 5 cocrystals in the CSD, a balanced invalid cocrystal set is found, showing a
Figure 2: Featurization of aspirin (acetylsalicylic acid) as a circular fingerprint vector and molecular graph. (top right) Circular or extended-connectivity fingerprints (ECFP\(_r^2\), where \(r\) is the radius) store substructures around each atom up to a certain radius \(r\) in a binary vector. The presence or absence of these substructures are encoded as 1 or 0, respectively. (bottom right) Molecular graphs store the connectivities between the molecule’s atoms in an \(N_{\text{atoms}} \times N_{\text{atoms}}\) adjacency matrix, and their features (such as atom type, hybridization etc.; see Supplementary Table 1) in an \(N_{\text{atoms}} \times N_{\text{features}}\) feature matrix. Explicit zeros are omitted for clarity.

substantial overlap with coformers present in the valid cocrystal set.

The two model types introduced above are subject to a vast number of adjustable configurational parameters, such as the number of layers, layer sizes, activation function of the layers etc. (see Supplementary Table 2). As each parameter greatly influences the performance of the model types on the data set, it is of paramount importance to tune the model configurations in order to achieve the optimal predictive performance. However, because the space imposed by these configurational parameters is extremely large, it is impossible to manually tune the model configuration and we therefore resorted to using Bayesian Optimization \[47\]. By iteratively assessing the performance of possible configurations on validation sets that are set aside (using the loss parameter), an optimizer constructs a surrogate model, which is used to seek the most optimal configurations for each model type. Fifty such iterations were performed, producing a ranked list of possible model configurations and their associated performance on data set aside for each type.

Although being architecturally different, the performance of the five best-performing FP models and GCN models differs only slightly, and each model has at least an accuracy value of 96% on its respective validation set (Supplementary Table 3 and 4). Moreover, as small differences in the validation sets set aside would lead to small changes in performance metrics and hence a different ranking, it is imprudent to select only a single model per type as final predictor. To solve this issue, we decided to group five models with the lowest losses of each
type in separate model ensembles (FP ensemble and GCN ensemble; Figure 1c). These composite models return the average of their individual constituents’ predictions and are likely to improve the overall robustness of the predictor by cancelling out erroneous mispredictions of single models. Furthermore, in an effort to include a larger and more differentiated amount of molecular information for the final prediction, we also combined the results of both model ensembles in a ten-membered model ensemble (FP + GCN ensemble; Figure 1c).

Details on the model implementation, dataset generation, model selection and model ensembling procedures are described in the Supplementary Information.

2.2 In silico validation of the approach

An evaluation of how well the three model ensembles (FP, GCN, and FP + GCN ensembles) can repredict cocrystals from the data set is obtained with cross-validation. In such a test, the data set is divided in ten equally large random parts, and each part (serving as a validation set) is, in turn, repredicted by the model ensembles that were trained on the remaining nine parts. This test therefore serves as an internal validation check and first step towards model validation.

All three ensembles demonstrate exceptionally high accuracy values ($\geq 97\%$ averaged over ten validation sets; Supplementary Figure 2), revealing the large potential of deep learning for multicomponent crystal prediction. These findings support the notion that rules for cocrystal formation are encoded in the internal parameters of the neural network models, which in fact define relevant combinations of atomic and molecular fragments. In practice, the reported accuracy suggests that on average more than 19 out of 20 cocrystals from a random subset are classified correctly based on the information of the remaining cocrystals in the data set. As differences in performance for the three model ensembles are small and within one another’s standard deviations, the FP + GCN ensemble was chosen as final predictor for overall robustness.

The high accuracy values obtained with cross-validation are only obtained when significant overlap between the coformers in the training and validation sets is present. To mimic the situation in which the target compound is completely absent in the cocrystal data available from the CSD, we manually removed all valid and invalid cocrystals for carbamazepine (Figure 1c) from our data set and trained the FP + GCN ensemble on the remaining data. By comparing the model ensemble’s output to the experimental outcome for this substantial set of cocrystals of carbamazepine together with a set of experimentally tested combinations by Roca-Paixão et
Figure 3: Model output for the coformers in combination with carbamazepine. (checkmark): Experimental proof available. (cross): Not found by Roca-Paixão et al. [24]. ($\phi$): Invalid combination from the data set. True positives (TP): 38. True negatives (TN): 2. False positives (FP): 10. False negatives (FN): 0. The color around the coformer number is related to the error between true and predicted value (continuous spectrum from green (error = 0) to red (error = 1)). (*): Cocrystal determined in this work.

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | HO | CO | OH | 1.00 | ✓  |
| 2 | HO | CO | OH | 1.00 | ✓  |
| 3 | HO | CO | OH | 1.00 | ✓  |
| 4 |     |     |     | 0.94 | ✓  |
| 5 |     |     |     | 0.98 | ✓  |
| 6 | HO | CO | OH | 1.00 | ✓  |
| 7 | HO | CO | OH | 1.00 | ✓  |
| 8 | HO | CO | OH | 1.00 | ✓  |
| 9 |     |     |     | 0.82 | ✓  |
| 10 |     |     |     | 0.95 | ✓  |
| 11 | HO |     |     | 1.00 | ✓  |
| 12 | HO |     |     | 1.00 | ✓  |
| 13 |     |     |     | 0.90 | ✓  |
| 14 |     |     |     | 0.86 | ✓  |
| 15 |     |     |     | 0.99 | ✓  |
| 16 | HO |     |     | 1.00 | ✓  |
| 17 |     |     |     | 1.00 | ✓  |
| 18 |     |     |     | 0.78 | ✓  |
| 19 |     |     |     | 0.85 | ✓  |
| 20 |     |     |     | 1.00 | ✓  |
| 21 |     |     |     | 1.00 | ✓  |
| 22 |     |     |     | 1.00 | ✓  |
| 23 |     |     |     | 1.00 | ✓  |
| 24 |     |     |     | 1.00 | ✓  |
| 25 |     |     |     | 0.99 | ✓  |

(a)

al. [24], we then are able to evaluate the performance of our approach for its intended purpose. Because the ensemble consists of ten members, its output is no longer binary (i.e 0 or 1) but continuous and can be interpreted as the percentage of models voting for a positive outcome. For the purpose of prediction, values larger than 0.5 are classified as valid and vice versa. Comparison between the true and predicted labels for the structures in Figure 3 leads to an accuracy of 80 % and a precision value of 79%, again confirming that relevant patterns for cocrystallization can be learned from cocrystal data in the CSD, and that these patterns are transferable to unseen cases.

Cocrystals of carbamazepine that were previously verified, were all correctly repredicted with mostly scores of 1.00 and at least 0.78. Invalid cocrystals ($\phi$) that were present in the data set
Figure 3: (continued) Model output for the coformers in combination with carbamazepine. (checkmark): Experimental proof available. (cross): Not found by Roca-Paixão et al. [24]. (ø): Invalid combination from the data set. True positives (TP): 38. True negatives (TN): 2. False positives (FP): 10. False negatives (FN): 0. The color around the coformer number is related to the error between true and predicted value (continuous spectrum from green (error = 0) to red (error = 1)). (*): Cocrystal determined in this work.
are given substantially smaller values, with the emergence of two false positives (coformers 44 and 45). We also compiled the predictions for an additional 67 invalid combinations for carbamazepine (see Supplementary Figure 3) and found that approximately 78% of invalid cocrystals scored lower than 0.5. An important result is therefore that the ensemble can discriminate potential coformer couples from combinations that are unable to interact. Yet, some combinations for carbamazepine that were not experimentally found by Roca-Paixão et al. [24] (crosses) are still given large values. This may indicate that the experimental conditions were possibly not optimal to yield cocrystalline material for these combinations. We therefore extended the experimental search for these cocrystals and have synthesized the cocrystal of carbamazepine with ketoprofen (coformer 42, vide infra). Moreover, cocrystals for the combination of carbamazepine with ibuprofen (coformer 41) have also been reported. [48] In addition to a positive prediction, the identification of a real cocrystal thus also requires the search for and fine-tuning of the experimental conditions for its actual synthesis.

2.3 Application to an active pharmaceutical ingredient (API)

The intended purpose of our approach is to predict cocrystals of molecules for which experimental data is unavailable. Ketoprofen (see Figure 1c) is such a case, as it has no cocrystals in the CSD (and hence our dataset), making it a challenging and perfect candidate to test our approach.

The scoring spectrum of the FP + GCN ensemble for ketoprofen was evaluated for predictions with the 75 most popular coformers as found in the CSD. To gain insight in the correlation between the various coformer types among these 75 molecules and their prediction values, we clustered the coformers based on their number of common cocrystallization partners. By drawing cocrystals in the CSD as a physical network [25], with coformers as its nodes and their cocrystals as its edges, the similarity between two coformers can be expressed as the number of shared partners (or common neighbors) divided by the combined number of cocrystallization partners. Coformers that are highly similar are then merged in clusters using Ward’s hierarchical grouping method [49]. The result of the clustering is presented as a dendrogram (Figure 4, left), showing the distances at which coformers are merged, where higher distances correspond to more dissimilar coformers (or clusters thereof). The actual coformers in these clusters and their respective predicted values are shown in Figure 5. Additional details are available in the Supplementary Information.
Figure 4: (left) Dendrogram of the 75 most popular coformers in the CSD clustered by Ward’s hierarchical method. (right) Predicted values for cocrystal formation with ketoprofen by the 10-membered ensemble model.

Figure 5: Overview of the molecular structures from Figure 4 and their corresponding prediction values. The structures are ordered from left to right based on their appearance in the dendrogram, and predictions values larger than 0.8 are emphasized.
Members of the green, pink and blue clusters are classified as cocrystals with a high likelihood of actual formation, and relatively strong supramolecular interactions can indeed be imagined with ketoprofen. Smaller prediction values are found for the cyan cluster (on average 0.75). While the majority of the models of the ensemble recognize the $\pi$-$\pi$-interaction points on both the cluster’s molecules and ketoprofen, predictions with a smaller confidence are returned. This is likely due to an underpresentation of cocrystals bonded solely through $\pi$-$\pi$-interactions in the CSD, resulting in less exposure of such cases during training. The smallest likelihoods of cocrystal formation are given to the aliphatic dicarboxylic acids in the yellow cluster. Although being omnipresent in other cocrystals, the ensemble recognizes that pairing the structural features of ketoprofen with two carboxylic acid functionalities is not likely to result in a cocrystal, as this is unusual in the training data. Our approach can thus identify plausible coformer combinations, and can additionally make suggestions regarding shape (e.g. arene substitution patterns in the red cluster) and size (decreasing scores in the yellow cluster).

An eye-catching prediction is the drug-drug cocrystal of ketoprofen with carbamazepine ($P_{cc,ensemble} = 0.983$, Figure 5). This combination was given only a modest likelihood of formation by COSMO-RS and was not found by liquid-assisted grinding in methanol. [24] We also tried to synthesize this cocrystal, and by grinding an equimolar mixture of racemic ketoprofen with carbamazepine together with a few drops of acetonitrile, a new phase was found (Supplementary Figure 4). Crystals obtained from slow evaporation in methanol were analyzed with single-crystal X-ray diffraction and proved to be binary cocrystals (Supplementary Figure 5 and Supplementary Table 6). Not only is this the first cocrystal structure of ketoprofen, but also one of the rare cases in which two drugs are found in the same crystal [50].

3 Conclusions

This article introduced a new approach for the prediction of binary cocrystal formation using an ensemble of artificial neural networks. By combining the available binary cocrystal data in the CSD with a large set of invalid combinations of coformers, it becomes possible to train the neural networks for the prediction of cocrystal formation. The approach uses the molecular structures of two coformers, and outputs a likelihood for cocrystal formation based on patterns extracted from the data set. A similar methodology can be used for the prediction of solvates, and it can be extended to applications involving two or more atomic or molecular species (e.g.
metal-organic frameworks).

*In silico* validation of the approach demonstrated its excellent performance (accuracy ≥ 97%), and accuracies around 80% are to be expected for cases where one of the molecules is not found in cocrystals in the CSD. Ketoprofen is such a case, and an analysis of its predictions highlighted the correlation between structural features and model output.

The approach is applicable to virtually any molecule (even prior to actual synthesis), and is therefore envisaged to be an attractive tool for drug design and optimization in the pharmaceutical industry. Predictions can be made as soon as the molecular structure of for instance an active substance is proposed or identified, making it useful in the most early stages of the drug pipeline.

4 Acknowledgements

The authors thank the DeepChem development team for their support and implementations for a variety of neural network layers in Python.

This research received funding as part of the CORE ITN Project by the European Union’s Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Grant Agreement No. 722456 CORE ITN.

5 Competing interests

The authors declare no competing interests.

6 Supplementary information

- Details on the generation of the data set, featurization of the coformers, implementation, selection and validation of the models, additional negative combinations for carbamazepine, clustering of the coformers and experimental procedures (PDF);

References

[1] Aitipamula, S. *et al.* Polymorphs, Salts, and Cocrystals: What’s in a Name? *Crystal Growth \\& Design* **12**, 2147–2152 (2012).

[2] Grothe, E., Meekes, H., Vlieg, E., ter Horst, J. H. & de Gelder, R. Solvates, Salts, and Cocrystals: A Proposal for a Feasible Classification System. *Crystal Growth \\& Design* **16**, 3237–3243 (2016).
[3] Vippagunta, S. R., Brittain, H. G. & Grant, D. J. W. Crystalline solids. *Advanced Drug Delivery Reviews* 48, 3–26 (2001).

[4] Serajuddin, A. T. M. Salt formation to improve drug solubility. *Advanced Drug Delivery Reviews* 59, 603–616 (2007).

[5] Aakeröy, C. B., Forbes, S. & Desper, J. Using Cocrystals To Systematically Modulate Aqueous Solubility and Melting Behavior of an Anticancer Drug. *Journal of the American Chemical Society* 131, 17048–17049 (2009).

[6] Kalepu, S. & Nekkanti, V. Insoluble drug delivery strategies: review of recent advances and business prospects. *Acta Pharmaceutica Sinica B* 5, 442–453 (2015).

[7] Zvoníek, V. et al. First Crystal Structures of Pharmaceutical Ibrutinib: Systematic Solvate Screening and Characterization. *Crystal Growth & Design* 17, 3116–3127 (2017).

[8] Dai, X.-L., Chen, J.-M. & Lu, T.-B. Pharmaceutical cocrystallization: an effective approach to modulate the physicochemical properties of solid-state drugs. *CrystEngComm* 20, 5292–5316 (2018).

[9] Aakeröy, C. B. & Salmon, D. J. Building co-crystals with molecular sense and supramolecular sensibility. *CrystEngComm* 7, 439–448 (2005).

[10] Allesø, M. et al. Solvent Diversity in Polymorph Screening. *Journal of Pharmaceutical Sciences* 97, 2145–2159 (2008).

[11] Kumar, L., Amin, A. & Bansal, A. K. An overview of automated systems relevant in pharmaceutical salt screening. *Drug Discovery Today* 12, 1046–1053 (2007).

[12] Casares, A. F. et al. An evaluation of salt screening methodologies. *Journal of Pharmacy and Pharmacology* 67, 812–822 (2015).

[13] Galek, P. T. A., Fábián, L., Motherwell, W. D. S., Allen, F. H. & Feeder, N. Knowledge-based model of hydrogen-bonding propensity in organic crystals. *Acta Crystallographica Section B: Structural Science* 63, 768–782 (2007).

[14] CruzCabeza, A. J., Day, G. M. & Jones, W. Towards Prediction of Stoichiometry in Crystalline Multicomponent Complexes. *Chemistry A European Journal* 14, 8830–8836 (2008).

[15] Issa, N., Karamertzanis, P. G., Welch, G. W. A. & Price, S. L. Can the Formation of Pharmaceutical Cocrystals Be Computationally Predicted? I. Comparison of Lattice Energies. *Crystal Growth & Design* 9, 442–453 (2009).

[16] Karamertzanis, P. G. et al. Can the Formation of Pharmaceutical Cocrystals Be Computationally Predicted? 2. Crystal Structure Prediction. *Journal of Chemical Theory and Computation* 5, 1432–1448 (2009).

[17] Fábián, L. Cambridge Structural Database Analysis of Molecular Complementarity in Cocrystals. *Crystal Growth & Design* 9, 1436–1443 (2009).

[18] Delori, A., Galek, P. T. A., Pidcock, E., Patni, M. & Jones, W. Knowledge-based hydrogen bond prediction and the synthesis of salts and cocrystals of the anti-malarial drug pyrimethamine with various drug and GRAS molecules. *CrystEngComm* 15, 2916–2928 (2013).
[19] Grecu, T., Hunter, C. A., Gardiner, E. J. & McCabe, J. F. Validation of a Computational Cocrystal Prediction Tool: Comparison of Virtual and Experimental Cocrystal Screening Results. *Crystal Growth & Design* **14**, 165–171 (2014).

[20] Wicker, J. G. P. *et al*. Will they co-crystallize? *CrystEngComm* **19**, 5336–5340 (2017).

[21] Loschen, C. & Klamt, A. Cocrystal Ternary Phase Diagrams from Density Functional Theory and Solvation Thermodynamics. *Crystal Growth & Design* **18**, 5600–5608 (2018).

[22] Taylor, C. R. & Day, G. M. Evaluating the Energetic Driving Force for Cocrystal Formation. *Crystal Growth & Design* **18**, 892–904 (2018).

[23] Barua, H., Gunnam, A., Yadav, B., Nangia, A. & Shastri, N. R. An ab initio molecular dynamics method for cocrystal prediction: validation of the approach. *CrystEngComm* **21**, 7233–7248 (2019).

[24] Roca-Paixão, L., Correia, N. T. & Affouard, F. Affinity prediction computations and mechanosynthesis of carbamazepine based cocrystals. *CrystEngComm* **21**, 6991–7001 (2019).

[25] Devogelaer, J. J., Meekes, H., Vlieg, E. & de Gelder, R. Cocrystals in the Cambridge Structural Database: a network approach. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials* **75**, 371–383 (2019).

[26] Devogelaer, J.-J. *et al*. Cocrystal design by network-based link prediction. *CrystEngComm* **21**, 6875–6885 (2019).

[27] Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. The Cambridge Structural Database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials* **72**, 171–179 (2016).

[28] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

[29] Dietterich, T. G. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, 1–15 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2000).

[30] Rogers, D. & Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **50**, 742–754 (2010).

[31] Duvenaud, D. K. *et al*. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems 28*, 2224–2232 (Curran Associates, Inc., Red Hook, USA, 2015).

[32] Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. http://arxiv.org/abs/1609.02907 (2016).

[33] Kearnes, S.,McCloskey, K.,Berndl, M.,Pande, V. & Riley, P. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* **30**, 595–608 (2016).

[34] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning* 70, 1263–1272 (JMLR.org, 2017).

[35] Defferrard, M., Bresson, X. & Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. http://arxiv.org/abs/1606.09375 (2017).

[36] Wu, Z. *et al*. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* **9**, 513–530 (2018).
[37] Xie, T. & Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Physical Review Letters* **120**, 145301 (2018).

[38] Feinberg, E. N. *et al.* PotentialNet for Molecular Property Prediction. *ACS Central Science* **4**, 1520–1530 (2018).

[39] Mater, A. C. & Coote, M. L. Deep Learning in Chemistry. *Journal of Chemical Information and Modeling* **59**, 2545–2559 (2019).

[40] Coley, C. W. *et al.* A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical Science* **10**, 370–377 (2019).

[41] Roszak, R., Beker, W., Molga, K. & Grzybowski, B. A. Rapid and Accurate Prediction of pKa Values of CH Acids Using Graph Convolutional Neural Networks. *Journal of the American Chemical Society* **141**, 17142–17149 (2019).

[42] Ishida, S., Terayama, K., Kojima, R., Takasu, K. & Okuno, Y. Prediction and Interpretable Visualization of Retrosynthetic Reactions Using Graph Convolutional Networks. *Journal of Chemical Information and Modeling* **59**, 5026–5033 (2019).

[43] Korolev, V., Mitrofanov, A., Korotcov, A. & Tkachenko, V. Graph Convolutional Neural Networks as General-Purpose Property Predictors: The Universality and Limits of Applicability. *Journal of Chemical Information and Modeling* (2019).

[44] Mayr, A. *et al.* Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science* **9**, 5441–5451 (2018).

[45] Yang, K. *et al.* Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling* **59**, 3370–3388 (2019).

[46] Stokes, J. M. *et al.* A Deep Learning Approach to Antibiotic Discovery. *Cell* **180**, 688–702 (2020).

[47] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE* **104**, 148–175 (2016).

[48] Abd Rahim, S., Rosli, N. A. & Mohd Khalid, S. S. Screening of Carbamazepine-Ibuprofen Co-Crystal Formation Using Non-Stoichiometric and Stoichiometric Methods. *Advanced Materials Research* **1113**, 417–421 (2015).

[49] Ward, J. H. J. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* **58**, 236–244 (1963).

[50] Sekhon, B. S. Drug-drug co-crystals. *DARU Journal of Pharmaceutical Sciences* **20**, 45 (2012).
Supplementary Information for the manuscript:
Cocrystal prediction by artificial neural networks

Jan-Joris Devogelaer, Hugo Meekes, Paul Tinnemans, Elias Vlieg, and René de Gelder*

Radboud University, Institute for Molecules and Materials, Heyendaalseweg 135, 6525 AJ Nijmegen,
The Netherlands
E-mail: r.degelder@science.ru.nl

Contents

1 Data extraction from the CSD S-2
2 Coformer featurization S-3
3 Model implementation and selection S-4
  3.1 Implementation details . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . S-4
  3.2 Selection of the model configurations . . . . . . . . . . . . . . . . . . . . . . . . S-5
4 Model validation S-8
  4.1 Cross-validation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . S-8
  4.2 Additional invalid coformer pairs for carbamazepine . . . . . . . . . . . . . . . S-8
5 Coformer clustering S-12
6 Experimental procedures for cocrystal synthesis and characterization S-12
1 Data extraction from the CSD

The procedures to collect cocrystal data from the CSD are the same as those presented in Refs. 1 and 2. Entries containing two distinct chemical residues that were organic, not ionic or polymeric, error-free and had their three-dimensional coordinates determined (including disorder) were searched in the CSD (v5.40). The individual constituents of the entries were then found by converting their structure data files (SDF) to canonical SMILES strings with OpenBabel [3]. These molecular representations are unique, take aromaticity and chirality into account, and are easily converted into circular fingerprint vectors and molecular graphs. Cocrystals were filtered by comparing each constituent to a predefined list of common solvents and gasses (lists are made available in Ref. 2). In the case of racemic cocrystals (i.e. a cocrystal containing an achiral coformer and a pair of enantiomers of a chiral coformer), a single enantiomer is retained to prevent potential overfitting and unfair reprediction in later stages of model development.

The data set was further restricted to cocrystals with correctly determined explicit valencies of both coformers, as this was a necessary requirement for further processing to fingerprints and molecular graphs. Furthermore, only coformers with up to 60 heavy atoms (approx. 75 % of coformers) were included, easing the conversion to the proposed data formats and focusing the set on cocrystals with directed interactions between relatively small molecules. These procedures resulted in a dataset of 8050 cocrystals, formed by 5334 unique coformers.

Prior to filtering the coformers on their explicit valencies and size, the cocrystals were converted to a physical network and stored in an adjacency matrix $A$, for which the row and column indices correspond to the coformers. Combinations of coformers for which a cocrystal is known in the CSD are labeled in $A$ as 1, and 0 if undetermined. Such a matrix may be used to predict missing cocrystals with link-prediction algorithms [2], assigning large score values to combinations likely to interact based on network principles. Conversely, combinations that are highly unlikely are given smaller score values, which is exploited here to generate an invalid cocrystal set.

For coformers in the valid cocrystal set (5334 coformers), having more than five determined cocrystals in the CSD, all possible combinations were evaluated with the bipartite resource allocation index (detailed explanation in Ref. 2), storing those for which the score $\equiv 0$, corresponding to invalid coformer combinations. 8050 random samples were taken from this set, forming the invalid cocrystal set.
2 Coformer featurization

When applying deep learning to molecular data, the input (in this case a pair of coformers) should be provided in the form of a chemical representation. A wide variety of such representations exist [4], including SMILES strings [5], molecular descriptor vectors [6], key-based [7] and circular [8] (i.e. extended-connectivity) fingerprint vectors, and molecular graphs [9]. Extended-connectivity (or radial; Morgan-type) fingerprints were directly generated from canonical SMILES strings of the coformers using a DeepChem [10] wrapper-function for RDKit [11] (DeepChem v2.3.0 with GPU-enabled support, installed for Python v3.5.6). The radius and length of the fingerprints were not a priori set to fixed values but assumed to be configurational parameters of the model, meaning that the featurization to fingerprints is different per FP-model.

Similarly, the canonical SMILES of the coformers were transformed to molecular graphs with DeepChem. Molecules are characterized by an $N_{\text{atoms}} \times N_{\text{atoms}}$ adjacency matrix, containing the connectivities between the atoms, and an $N_{\text{atoms}} \times N_{\text{features}}$ feature matrix (Figure 2) describing the features of each atom. As features are mostly categorical of nature (see Supplementary Table 1), the feature vector is one-hot encoded to allow for their further processing with machine learning techniques. Therefore, each feature is transformed into an array with a length equal to its number of choices. Each option is given its own bit in the array, which is set to 1 when present. For example, if one desires to encode the atom type and has three options (e.g. carbon, oxygen or nitrogen), then carbon corresponds to $[1, 0, 0]$, oxygen to $[0, 1, 0]$ and nitrogen to $[0, 0, 1]$. Finally, the one-hot encoded bit vectors of all properties are joined together, forming the atomic feature vector (of length 78).

Although these two representations are technically two-dimensional in the sense that no atomic coordinates are taken into account, the inclusion of for example the hybridization state of the atoms and optionally their chirality results in the subtle presence of three-dimensional information. Therefore, these molecular representations seem to be very suitable for the purpose of cocrystal prediction.
Supplementary Table 1: Atom features that are encoded in the feature matrix. The length corresponds to the number of bits occupied by the specific one-hot encoded feature in the final feature vector (of length 78). Only a single bit is required for boolean properties (e.g. aromatic).

| Feature                  | Options                                                                 | Length |
|--------------------------|-------------------------------------------------------------------------|--------|
| Element type             | C, N, O, S, F, Si, P, Cl, Br, Mg, Na, Ca, Fe, As, Al, I, B, V, K, Tl, Yb, Sb, Sn, Ag, Pd, Co, Se, Ti, Zn, H, Li, Ge, Cu, Au, Ni, Cd, In, Mn, Zr, Cr, Pt, Hg, Pb, Unknown | 44     |
| Degree                   | 0, 1, 2, ..., 9, 10                                                     | 11     |
| Implicit valence         | 0, 1, 2, ..., 5, 6                                                      | 7      |
| Formal charge            | 0 or 1                                                                 | 1      |
| Number of radical electrons | 0 or 1                                                                    | 1      |
| Hybridization            | sp, sp$^2$, sp$^3$, sp$^3$d or sp$^3$d$^2$                             | 5      |
| Aromatic                 | True or False                                                           | 1      |
| Number of hydrogens      | 0, 1, 2, 3, 4                                                           | 5      |
| Chirality                | R, S                                                                    | 2      |
| Chirality possible       | True or False                                                           | 1      |

$\Sigma = 78$

3 Model implementation and selection

3.1 Implementation details

Both model types (Figure 2a and 2b) were implemented in Python (v3.5.6) with Keras [12] and DeepChem [10], and are divided into 4 modules (Supplementary Table 2). Each model first preprocesses both coformers in a shared manner (modifying each coformer in the same way) and afterwards merges them into a learnable, united cocrystal vector. The latter is then further processed through a sequence of hidden layers and used for the final prediction. In fact, each hidden layer consists of a sequence of a fully-connected (or dense) layer, a batch normalization layer [13] and a dropout layer [14]. Several of such layer sequences may be present within the module.

Fully-connected (or dense) layers receive as input a weighted linear combination of the output from all nodes in the previous layer. After subtraction of a bias term, the input is passed though a non-linear activation function and a single outcome is transmitted to the next layer, repeating the same computation. During the training phase, the model is initialized with random weights and bias terms, and is subsequently exposed to batches of labeled (or known) training data, for which predictions are generated. The error on these predictions contributes to a loss function, which is simultaneously minimized by adjusting (or learning) the model’s weights while cycling over the available training data (i.e. supervised learning). The model weights are adjusted in the direction opposite to the loss gradients with the backpropagation algorithm.
For the preprocessing module of the molecular graph-based model type, the fully-connected layer is swapped for a graph convolutional layer (open-source implementations of Altac-Tran et al. [16]). The molecular graphs are passed through a series of learnable convolution layers, updating their node features with those of their local chemical environment with each convolutional pass. This creates both a tunable and hierarchical representation of the molecule [17], which, after transformation into a one-dimensional array, is combined into a cocrystal vector and used as input for the abovementioned neural networks containing only hidden layers. In unreported results, dropout for such layers did not appear to affect the training outcome and was therefore omitted. Also, after a batch normalization layer, each node of the graph is (max) pooled, updating its features with the maximum activation across itself and its neighbors. At the end of the preprocessing, the molecular graph of each coformer is condensed to a one-dimensional array by passing it through a graph gathering layer, after which both coformers are merged.

Both model types process through batches of training data and are optimized with Adam optimizer [18] (learning rate=0.001, $\beta_1=0.9$, $\beta_2=0.999$). The model training was performed on an Intel® Core™ i9-7940X (CPU) and an NVIDIA GeForce© RTX 2080 SUPER™ (GPU). The training of one FP model took approximately one minute and that of a GCN model around ten minutes, resulting in a total training time of a little less than one hour for the ten-membered ensemble.

### 3.2 Selection of the model configurations

The number of configurational parameters for both model types that are adjustable is quite large (Supplementary Table 2). As each parameter is variable and modules can consist of multiple layers, each with independently defined sizes, the space of possible model configurations becomes too large to manually search for an optimum. Therefore, sequential model-based optimization techniques such as Bayesian optimization [19] provide a convenient tool to explore the large parameter space for the best model configurations. For this purpose, we used the Python package Hyperopt [20,21].

The procedure for finding the optimal configurations is identical for both model types. The available cocrystal data set is first randomly split into a training (90%) and validation (10%) set. Next, a model (defined by the combination of a model type and set of config-
Supplementary Table 2: Configurational parameters and their possible values for both model types.

| Model | FP model | GCN model |
|-------|----------|-----------|
| Fingerprint size \((2^x)\) | \(x \in \{7, 8, \ldots, 11, 12\}\) | - |
| Fingerprint radius \((r)\) | \(r \in \{1, 2, 3\}\) | - |
| Batch size \((2^b)\) | \(b \in \{6, 7, 8\}\) | 7 |

Preprocessing module

| Layer type | Dense | Graph Convolution |
|------------|-------|-------------------|
| Layer size \((2^x)\) | \(x \in \{7, 8, 9, 10, 11\}\) | \(x \in \{6, 7, 8\}\) |
| Layer activation function | ReLU, ELU or Tanh | ReLU, ELU or Tanh |
| Layer dropout \((d)\) | \(d \in [0, 0.75]\) | 0 |
| Number of layers | 1,2 or 3 | 1,2 or 3 |
| Graph gathering activation function | - | ReLU, ELU or Tanh |

Merging module

| Vector operation | Add or Concatenate | Add or Concatenate |

Feedforward module

| Layer type | Dense | Dense |
|------------|-------|-------|
| Layer size \((2^x)\) | \(x \in \{7, 8, 9, 10, 11\}\) | \(x \in \{6, 7, 8, 9, 10\}\) |
| Layer activation function | ReLU, ELU or Tanh | ReLU, ELU or Tanh |
| Layer dropout \((d)\) | \(d \in [0, 0.75]\) | \(d \in [0.1, 0.6]\) |
| Number of layers | 1,2 or 3 | 1,2 or 3 |

Predictive module

| Layer type | Dense | Dense |
|------------|-------|-------|
| Layer size | 1 | 2 |
| Layer activation function | Sigmoid | Softmax |
| Loss function | Binary cross entropy | Softmax cross entropy |
Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.

Supplementary Figure 1: Example of training a GCN model. (left) The loss on both the training and validation set quickly decreases with an increasing number of epochs. (right) Evolution of the model’s accuracy.
to the Bayesian optimizer (e.g. Supplementary Figure 1, epoch 36), which repeatedly selects the next configuration to be tested based on previous evaluations by optimizing the Expected Improvement with the Tree-structured Parzen Estimator approach (TPE) [22]. Initially, this procedure is run for three random configurations from the space, which the optimizer uses as a starting point. After 50 iterations, a ranked list of possible model configurations and their associated performance on data set aside is produced for each model type, which was used to select the five best models for each type (Supplementary Table 3 and 4). These models were placed in their individual model ensembles and a combined model ensemble, containing all ten models.

4 Model validation

4.1 Cross-validation

The performance of the FP, GCN, and FP + GCN model ensembles was validated on the available cocrystal data by ten-fold cross-validation. The data set was first randomly divided in ten equal parts or folds. Each fold is used once for validation and nine times for training, and the performance on each of the validation sets is recorded while training on the data from residual nine sets. To prevent overfitting, the FP models and GCN models were trained for 30 and 50 epochs, respectively. The average precision, accuracy and loss values over ten folds were computed for the model ensembles and are shown in Supplementary Figure 2.

4.2 Additional invalid coformer pairs for carbamazepine

Besides the four invalid cocrystal combinations that were already in our data set for carbamazepine, we found an additional 67 eligible couples according to the procedures mentioned in section 1. The models were again trained on all data points except for the carbamazepine combinations, and the prediction values for these invalid pairs are shown in Supplementary Figure 3. Clearly, most combinations (approx. 78%) are scored below the 0.5 threshold, and are therefore assumed to be non-existing. Although our set of invalid cocrystals is to a certain degree artificial and lacks real experimental evidence, its usefulness is thus demonstrated in Supplementary Figure 3, as well as by the high precision values (and therefore small occurrence of false positives) presented in Supplementary Figure 2.
Supplementary Table 3: Configurational parameters (top) and performance metrics (bottom) of the five best FP models. The models are ordered by their increasing loss. The epoch indicates the point where the smallest loss was observed.

| Model | Radius | Fingerprint size | pr1 | pr2 | pr3 | droppr | actpr | merge | ff1 | ff2 | ff3 | dropff | actff | Batch size |
|-------|--------|------------------|-----|-----|-----|--------|-------|-------|-----|-----|-----|--------|-------|------------|
| 1     | 1      | 1024             | 1024| 512 | 128 | 0.373  | ReLU  | Stack | 512 | 0.358| ELU | 128 |
| 2     | 2      | 2048             | 2048| 2048| 128 | 0.409  | ReLU  | Add   | 128 | 256 | 0.230 | ReLU | 64 |
| 3     | 3      | 2048             | 1024| 2048| 1024| 0.675  | ReLU  | Add   | 128 | 64  | 0.559 | ELU | 256 |
| 4     | 2      | 4096             | 2048| 2048| 256 | 0.285  | ReLU  | Stack | 256 | 256 | 0.052 | ReLU | 64 |
| 5     | 2      | 2048             | 512 | 512 | 1024| 0.392  | ReLU  | Add   | 2048| 0.592| ELU | 64 |

| Model | Loss | Accuracy | Epoch |
|-------|------|----------|-------|
| 1     | 0.072| 0.972    | 14    |
| 2     | 0.087| 0.971    | 28    |
| 3     | 0.088| 0.965    | 28    |
| 4     | 0.089| 0.971    | 8     |
| 5     | 0.089| 0.974    | 10    |
Supplementary Table 4: Configurational parameters (top) and performance metrics (bottom) of the five best GCN models. The models are ordered by their increasing loss. The epoch indicates the point where the smallest loss was observed.

| Model | pr1 | pr2 | pr3 | droppr | actpr | Merge | ff1 | ff2 | ff3 | dropff | actff | actgather | Batch size |
|-------|-----|-----|-----|--------|-------|-------|-----|-----|-----|---------|-------|------------|------------|
| 1     | 64  | 256 | 0   | Tanh   | Add   | 512   | 512 | 512 | 0   | 0.410   | ReLU   | ReLU       | 128        |
| 2     | 64  | 256 | 0   | ReLU   | Add   | 512   | 512 | 512 | 0   | 0.409   | ReLU   | ReLU       | 128        |
| 3     | 128 | 256 | 0   | ReLU   | Add   | 1024  | 1024| 1024| 0   | 0.409   | ReLU   | Tanh       | 128        |
| 4     | 128 | 256 | 0   | Tanh   | Add   | 512   | 1024| 512 | 0.424| ReLU    | ReLU   | ReLU       | 128        |
| 5     | 256 | 0   | 0   | ReLU   | Stack | 512   | 1024| 0   | 0.340| ReLU    | Tanh   | 128        |

| Model | Loss | Accuracy | Epoch |
|-------|------|----------|-------|
| 1     | 0.095| 0.968    | 51    |
| 2     | 0.099| 0.971    | 45    |
| 3     | 0.104| 0.963    | 63    |
| 4     | 0.105| 0.966    | 36    |
| 5     | 0.106| 0.964    | 57    |
Supplementary Figure 2: Performance metrics of the ensemble models over a ten-fold cross-validation experiment. The height of the bars shows the average value on the ten validation sets. The black error bars correspond to ± one unit of standard deviation.

Supplementary Figure 3: Score histogram (left) and cumulative distribution function (right) of the 71 invalid combinations for carbamazepine.
5 Coformer clustering

The clustering of the coformers was done with Ward’s hierarchical clustering method [23], according to the procedures described in Ref. 1. The adjacency matrix of the coformer network (see section 1), which essentially contains all determined cocrystals in the CSD, is transformed into a similarity matrix, where coformer similarity for a pair of coformers $i$ and $j$ is defined with the Jaccard index [24]:

$$s_{i,j} = \frac{n_i \cap n_j}{n_i \cup n_j},$$

(1)

where the neighbors $n_i$ of coformer $i$ can be found from the set of nodes (or coformers) $N$ as:

$$n_i = \{ j \in N | A_{i,j} = 1 \}.$$  

(2)

For the purpose of hierarchical clustering, the similarity matrix was converted into dissimilarity matrix ($d_{i,j} = 1 - s_{i,j}$) and was resized to include only the 75 most popular coformers. These were determined based on their degree ($= |n_i|$) in the adjacency matrix. Ward’s clustering method works in an agglomerative fashion, repeatedly merging coformers or clusters thereof that are least dissimilar (or closest/most similar) in larger clusters. The method starts by placing all coformers in separate clusters or singletons, which were subsequently agglomerated into clusters. The distance to a cluster $p$ containing multiple coformers is calculated as:

$$d(p, q) = \sqrt{\frac{|q| + |s|}{|q| + |s| + |t|}d(q, s)^2 + \frac{|q| + |t|}{|q| + |s| + |t|}d(q, t)^2 - \frac{|q|}{|q| + |s| + |t|}d(s, t)^2}$$

(3)

where $p$ is the cluster as a result from merging clusters $s$ and $t$, and $q$ is one of the remaining clusters. The distances at which two clusters were joined were recorded and illustrated as a dendrogram (Figure 4).

6 Experimental procedures for cocrystal synthesis and characterization

RS-ketoprofen (48 mg; TCI Europe NV, > 98% purity) and carbamazepine (47 mg; Aldrich, ≤ 100% pure) were ground in the presence of 20 µL acetonitrile (i.e. liquid-assisted grinding, LAG) for 30 minutes at 25 Hz with a Retsch MM 400. A white powder was harvested, which was subsequently analyzed by powder X-ray diffraction (PXRD). For powder diffraction analysis,
Supplementary Figure 4: Powder diffraction patterns of RS-ketoprofen (red), carbamazepine (blue), and the co-ground powder (green). The simulated PXRD pattern of the cocrystal structure (black), confirming that the phase obtained by LAG is the cocrystal.

Samples were thinly applied on a zero-background (557)-silicon wafer. The diffractograms were measured on a Panalytical Empyrean diffractometer in Bragg-Brentano geometry using CuKα radiation from a sealed LFF tube and a PIXcel3D 1x1 detector. The powder patterns of the obtained mixture (shown in Supplementary Figure 4) was different from its constituents and known polymorphs in the CSD, indicating the formation of a new phase.

Approximately 10 mg of the co-ground powder with racemic ketoprofen was dissolved in 1 mL of methanol and left to slowly evaporate, yielding colourless block-like crystals suitable for single-crystal X-ray diffraction after 4 days.

The structure of the cocrystal is presented in Supplementary Figure 5 and its simulated powder diffraction pattern is also shown in Supplementary Figure 4, confirming that the phase obtained by LAG is indeed the discovered cocrystal. Hydrogen bonding details (Supplementary Table 5) and the crystallographic data (Supplementary Table 6) of the cocrystal are presented below.
Supplementary Figure 5: ORTEP plot of the cocrystal. As the structure is disordered, only the major conformation ('A') of carbamazepine is shown.

Supplementary Table 5: H-bonding details of the cocrystal. Besides the interaction between the acid and amide groups on the coformers, involving one hydrogen (H21A) of the amide group, the ketone group of ketoprofen (C13 = O03) forms a hydrogen bond to the second hydrogen (H21B) on the amide.

| Donor   | Acceptor | ARU    | D - H  | H...A   | D...A   | D - H...A   |
|---------|----------|--------|--------|---------|---------|-------------|
| O01     | -H01     | .O20A  | 1555.02| 0.972(17)| 1.598(17)| 2.5572(14)  | 168.7(13)  |
| N21A    | -H21A    | .O02   | 1555.01| 0.88    | 2.14    | 2.9823(16)  | 161.0       |
| N21A    | -H21B    | .O03   | 4565.01| 0.88    | 2.32    | 3.1454(15)  | 155.9       |

*: Reflections were measured on a Bruker D8 Quest diffractometer with sealed tube and Triumph monochromator (λ = 0.71073 Å). Software package used for the intensity integration was Saint (v8.38a, Bruker AXS Inc., Madison, Wisconsin, USA). Absorption correction was performed with SADABS [25]. The structures were solved with direct methods using SHELXT [26]. Least-squares refinement was performed with SHELXL-2014 [27] against $|F_h|^2$ of all reflections. Non-hydrogen atoms were refined freely with anisotropic displacement parameters. Hydrogen atoms were placed on calculated positions or located in difference Fourier maps. All calculated hydrogen atoms were refined with a riding model.
Supplementary Table 6: Crystallographic data\textsuperscript{a} of the cocrystal containing RS-ketoprofen and carbamazepine.

| Crystal data |  |
|--------------|---|
| Chemical Formula | C\textsubscript{16}H\textsubscript{14}O\textsubscript{3} \cdot C\textsubscript{15}H\textsubscript{12}N\textsubscript{2}O |
| \( M_r \) | 490.54 |
| Crystal system, space group | Monoclinic, \( P2_1/n \) |
| Temperature (K) | 150 |
| \( a, b, c (\text\AA) \) | 16.9943 (7), 7.7147 (3), 19.6426 (8) |
| \( \beta (\text\degree) \) | 97.9814 (16) |
| \( V (\text\AA}^3) \) | 2550.32 (18) |
| \( Z \) | 4 |
| Radiation type | \( \text{MoK}\alpha \) |
| \( \mu (\text{mm}^{-1}) \) | 0.09 |
| Crystal size (mm) | 0.33 \times 0.30 \times 0.10 |

| Data collection |  |
|-----------------|---|
| Diffractometer | Bruker D8 Quest Apex3 |
| Absorption correction | Multi-scan \textit{SADABS} 2016/2: Krause, L., Herbst-Irmer, R., Sheldrick G.M. & Stalke D., J. Appl. Cryst. 48 (2015) 3-10 |
| \( T_{\text{min}}, T_{\text{max}} \) | 0.712, 0.746 |
| No. of measured, independent and observed [\( I > 2\sigma(I) \)] reflections | 37024, 6340, 5535 |
| \( R_{\text{int}} \) | 0.024 |
| \( (\sin \theta/\lambda)_{\text{max}}(\text{\AA}^{-1}) \) | 0.667 |

| Refinement |  |
|-------------|---|
| \( R[F^2 > 2\sigma(F^2)], wR(F^2), S \) | 0.040, 0.110, 1.05 |
| No. of reflections | 6340 |
| No. of parameters | 392 |
| No. of restraints | 48 |
| H-atom treatment | H atoms treated by a mixture of independent and constrained refinement |
| \( \Delta\rho_{\text{max}}, \Delta\rho_{\text{min}} (\text{e\text\AA}^{-3}) \) | 0.26, -0.24 |
| Special remarks | Carbamazepine is disordered. |

References

[1] Devogelaer, J. J., Meekes, H., Vlieg, E. & de Gelder, R. Cocrystals in the Cambridge Structural Database: a network approach. \textit{Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials} \textbf{75}, 371–383 (2019).

[2] Devogelaer, J.-J. \textit{et al.} Cocrystal design by network-based link prediction. \textit{CrystEngComm} \textbf{21}, 6875–6885 (2019).

[3] O’Boyle, N. M. \textit{et al.} Open Babel: An open chemical toolbox. \textit{Journal of Cheminformatics} \textbf{3}, 33 (2011).

[4] Todeschini, R., Consonni, V., Mannhold, R., Kubinyi, H. & Folkers, G. \textit{Molecular Descriptors for Chemoinformatics: Volume I: Alphabetical Listing / Volume II: Appendices, References}. Methods and Principles in Medicinal Chemistry (Wiley, Hoboken, USA, 2009).
Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **28**, 31–36 (1988).

Mauri, A., Consonni, V., Pavan, M. & Todeschini, R. DRAGON software: an easy approach to molecular descriptor calculations. *MATCH* **56**, 237–248 (2006).

Durant, J. L., Leland, B. A., Henry, D. R. & Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences* **42**, 1273–1280 (2002).

Rogers, D. & Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **50**, 742–754 (2010).

Duvenaud, D. K. *et al.* Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems* **28**, 2224–2232 (Curran Associates, Inc., Red Hook, USA, 2015).

Ramsundar, B. DeepChem. [https://deepchem.io/](https://deepchem.io/).

Landrum, G. A. RDKit: Open Source Cheminformatics. [http://www.rdkit.org](http://www.rdkit.org).

Chollet, F. Keras. [https://keras.io/](https://keras.io/).

Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. [http://arxiv.org/abs/1502.03167](http://arxiv.org/abs/1502.03167) (2015).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15**, 1929–1958 (2014).

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).

Altae-Tran, H., Ramsundar, B., Pappu, A. S. & Pande, V. Low Data Drug Discovery with One-Shot Learning. *ACS Central Science* **3**, 283–293 (2017).

Bengio, Y., Courville, A. & Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**, 1798–1828 (2013).

Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. [http://arxiv.org/abs/1412.6980](http://arxiv.org/abs/1412.6980) (2017).

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE* **104**, 148–175 (2016).

Bergstra, J., Yamins, D. & Cox, D. D. Making a Science of Model Search. [http://arxiv.org/abs/1209.5111](http://arxiv.org/abs/1209.5111) (2012).

Bergstra, J. Hyperopt: Distributed Hyperparameter Optimization. [https://github.com/hyperopt/hyperopt](https://github.com/hyperopt/hyperopt).

Bergstra, J., Bardenet, R., Bengio, Y. & Kégl, B. Algorithms for Hyper-parameter Optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS’11, 2546–2554 (Curran Associates, Inc., Red Hook, USA, 2011).

Ward, J. H. J. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* **58**, 236–244 (1963).
[24] Jaccard, P. The distribution of the flora in the alpine zone.1. *New Phytologist* **11**, 37–50 (1912).

[25] Krause, L., Herbst-Irmer, R., Sheldrick, G. M. & Stalke, D. Comparison of silver and molybdenum microfocus X-ray sources for single-crystal structure determination. *Journal of Applied Crystallography* **48**, 3–10 (2015).

[26] Sheldrick, G. M. SHELXT Integrated space-group and crystal-structure determination. *Acta Crystallographica Section A: Foundations and Advances* **71**, 3–8 (2015).

[27] Sheldrick, G. M. Crystal structure refinement with SHELXL. *Acta Crystallographica Section C: Structural Chemistry* **71**, 3–8 (2015).
