RESEARCH

Open Access

C2PO: an ML-powered optimizer of the membrane permeability of cyclic peptides through chemical modification

Roy Aerts[1,2†], Joris Tavernier[3†], Alan Kerstjens[4], Mazen Ahmad[5], Jose Carlos Gómez-Tamayo[5], Gary Tresadern[5] and Hans De Winter[1*]

Abstract

Peptide drug development is currently receiving due attention as a modality between small and large molecules. Therapeutic peptides represent an opportunity to achieve high potency, selectivity, and reach intracellular targets. A new era in the development of therapeutic peptides emerged with the arrival of cyclic peptides which avoid the limitations of parenteral administration via achieving sufficient oral bioavailability. However, improving the membrane permeability of cyclic peptides remains one of the principal bottlenecks. Here, we introduce a deep learning regression model of cyclic peptide membrane permeability based on publicly available data. The model starts with a chemical structure and goes beyond the limited vocabulary language models to generalize to monomers beyond the ones in the training dataset. Moreover, we introduce an efficient *estimator2generative* wrapper to enable using the model in direct molecular optimization of membrane permeability via chemical modification. We name our application *C2PO* (Cyclic Peptide Permeability Optimizer). Lastly, we demonstrate how a molecule correction tool can be used to limit the presence of unfamiliar chemistry in the generated molecules.

Scientific contribution: We provide an ML-driven optimizer application, named C2PO, that returns structurally modified cyclic peptides with an improved membrane permeability, one of the pivotal tasks in drug discovery and development. C2PO is a first-in-class application for cyclic peptide permeability amelioration, in that it converts a ML model into a generative optimizer of chemical structures. Additionally, through demonstration we incentivize the usage of an automated post-correction tool with a chemistry reference library to correct strange chemistry outputs from C2PO, a known issue for ML-generated chemical structures.

[†] Roy Aerts and Joris Tavernier have contributed equally to this work.

*Correspondence:
Hans De Winter
hans.dewinter@uantwerpen.be
[1] Laboratory of Medicinal Chemistry, Department of Pharmaceutical Sciences, University of Antwerp, Universiteitslaan 1, Wilrijk, 2610 Antwerp, Belgium
[2] Theory and Spectroscopy of Molecules and Materials, Department of Chemistry, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium
[3] Open Analytics NV, Jupiterstraat 20, 2600 Antwerp, Belgium
[4] Hyle, Antwerp, Belgium
[5] In Silico Discovery, Janssen Research & Development, Janssen Pharmaceutica N. V., Turnhoutseweg 30, B-2340 Beerse, Belgium

Introduction

Today, the dominant class of therapeutics are small organic molecules [1, 2]. They exhibit many advantages, among others, straightforward and low-cost synthesis, tunable bioactivity, and good membrane permeability and cell uptake [3]. However, due to their small size, they may be promiscuous ligands, possibly binding to unintended targets causing adverse side effects. Moreover, their small size makes them inherently suboptimal for disrupting large molecule interactions, e.g. protein–protein interactions [4]. Hypothetically, these drawbacks can be overcome by turning to peptide-based therapeutics which, due to their larger size, potentially exhibit a broader applicability and display higher selectivity [3]. Indeed, peptides have enjoyed increasing attention in the last decades, leading to an approximate 6% share in the FDA-approved drugs by mid-2022 [1].

An appealing subclass of peptides are cyclic peptides. By constraining the conformational flexibility of a peptide chain, the target affinity and selectivity can be further improved. Furthermore, cyclisation of the peptide structure increases protection against proteolysis, a weak point of linear peptide structures [5]. As such, cyclic peptides represent a unique class that unlocks the druggability of new targets [6]. Even though they do not comply with the conventional rule-of-five of Lipinski, their increased permeability relative to linear peptides can be attributed to their chameleonic propensity, *i.e.*, they adopt an open conformation when exposed to aqueous solutions, while converting to a closed conformation when entering hydrophobic environments [7–9]. Unfortunately, cyclisation and the associated chameleonic effect is not an infallible solution for all cases of poor cell permeability [10, 11].

A wide plethora of synthetic modifications are investigated and deployed to achieve favorable permeability properties [12, 13]. Common strategies include N-methylation [10, 11], substitution of amide bonds [14, 15], induction of steric occlusion through chemical modification [16], and alteration of the conformational population [17, 18]. However, estimating the effect of certain synthetic modifications on cyclic peptide permeability is not straightforward. An intermediate solution is to train a machine learning (ML) model on known permeability data to support decision-making. Recently, Li et al. released the CycPeptMPDB dataset of literature-collected permeabilities of cyclic peptides [19]. Since then, various contributions have published ML models to evaluate the permeability of cyclic peptides [20–27]. Such models can be used by medicinal chemists to obtain an indication of the cell permeability of envisioned novel peptides.

We herein present an ML-powered application that returns, given a starting structure, chemically modified cyclic peptides with an improved in-vitro permeability (Fig. 1). At the core of our application sits a molecular structure optimizer that is controlled by an underlying permeability ML model, which we present as *estimator-2generative*. As such, we introduce an alternative to the more commonly deployed direct generative ML models, where the structure-proposer is tied to a pre-trained ML model. We have named our model C2PO, an abbreviation for Cyclic Peptide Permeability Optimizer.

When it comes to optimizing or generating new chemical structures using ML, maintaining chemical validity often proves challenging, as models are not aware of the concept of chemical soundness. Typical ways of improving chemical validity of the generated molecules include: the post-filtering of incorrect or unwanted chemistry and re-iterating generative tasks until a satisfied number of generated molecules is collected, or the introduction of rules within the model to enforce the output of sane molecules. For the proper functioning of these approaches, one needs to explicitly define and implement valid chemistry, which is both delicate and time-consuming. Moreover, when imposed in the forms of restraints within the model itself, interference with the optimization process will occur. This reduction in generative flexibility can result in suboptimal optimization tasks. We circumvent these issues by taking an alternative route and attach a previously developed automated molecular correction tool as a post-processing step [28].

In the Results section of this work, the general outcomes of the molecular modification (the first step) and subsequent molecular correction (the second step) are expounded. This informs the reader about what one can expect from applying the proposed strategy. Subsequently, we take a deep dive in the Discussion section. In summary, our principal objectives are to incentivize the usage of the proposed *estimator2generative* model in all types of medicinal chemistry tasks and to showcase the elegance of using a subsequent dictionary-based correction protocol instead of hard-coding valid chemistry or post-filtering.

Method

ML estimation model

The target model is dictated to start with the chemical structure and to optimize the structure without being limited to using, for instance, amino acid vocabulary. The ML model of cyclic peptide membrane permeability was trained using the public CycPeptMPDB [19] database, pulled on June 1st, 2024. Eighty percent of in total 7,451 measurements in the dataset were used for training the model while the remaining entries were equally split into a test and a validation set. More information

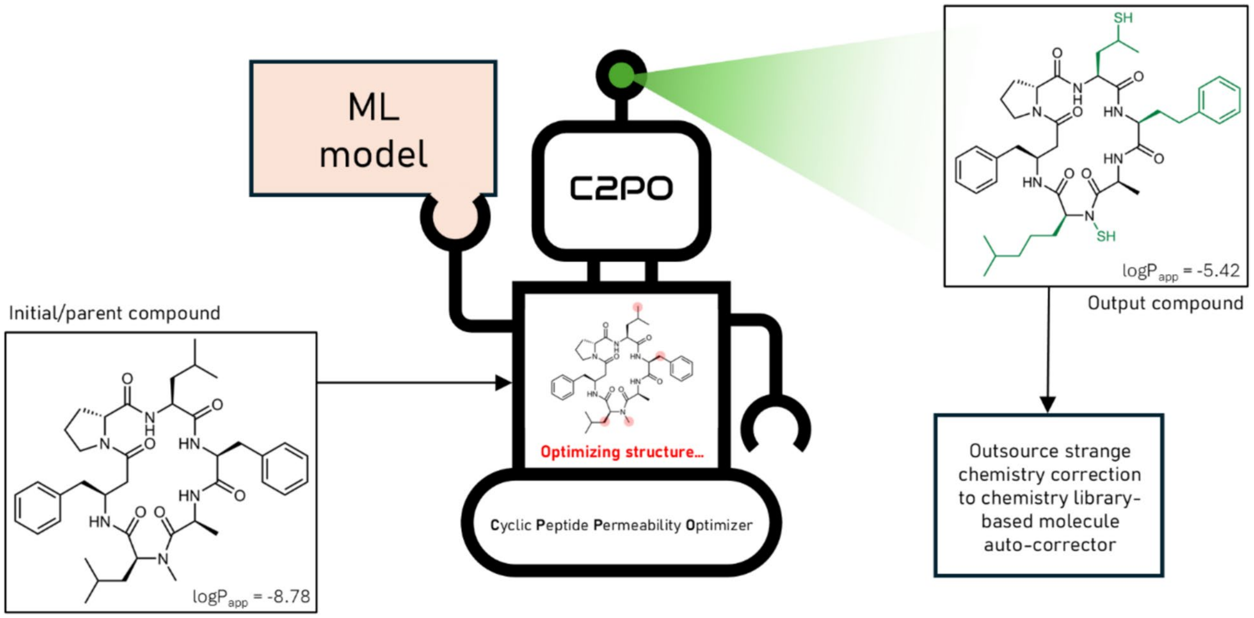
ML model
C2PO
Initial/parent compound
$\log P_{app}$ = -8.78
Optimizing structure...
Cyclic Peptide Permeability Optimizer
$\log P_{app}$ = -5.42
Output compound
Outsource strange chemistry correction to chemistry library-based molecule auto-corrector

Fig. 1 A visualization of the Cyclic Peptide Permeability Optimizer (C2PO). The application accepts a cyclic peptide structure and improves the permeability by mutating the chemical structure. C2PO bases its optimizations on a pre-trained machine learning (ML) estimation model. ML-driven applications have the tendency to (occasionally) propose strange chemistry. We let a chemical library-based auto-correction tool identify on foreign chemistry and subsequent correction, instead of a manual evaluation by experts. The auto-corrector tool used here is the one published by Kerstjens and De Winter [28]. The depicted structures are real examples extracted from the case study performed in this contribution (vide infra; first campaign in Fig. 4). The model outputs multiple optimized structures. Here the best so-called offspring molecule is depicted (top right). The N-S bond in the output compound's structure might be identified as questionable chemistry. However, the autocorrection application ruled this structure to be familiar. The initial compound was selected from the public CycPeptMPDB database(CycPeptMPDB ID: 3109)

on the presence of multiple permeability entries of the same peptide and the dataset split strategy can be consulted in the Supplementary Information (see Additional file 1). A Graph Transformers deep learning architecture was used due to the reported state-of-the-art performance of this class of deep learning (DL) architecture across various applications [29]. A depiction of the ML estimation model can be consulted in Fig. 2. The model architecture follows the framework from GRAPHGPS [30] which provides a combination of the local information from message-passing with the global information from the multi-head attention. The graphs are generated starting from the SMILES using RDKit [31]. Random walk positional encoding [32] is used to encode the positional information. The range of the global attention is controlled by an exponential decay as proposed in Gradformer [33]. The code is implemented using Pytorch [34] and PyTorch Geometric [35]. More details on the model architecture, hyperparameters and training settings are presented in the Supplementary Information (see Additional file 1).

Estimator2generative optimization wrapper

In natural language processing robustness of models is often improved when trained using adversarial examples [36]. These are examples where small changes in the data change the outcome of the model. Instead of trying to trick the model to change the outcome, we used these techniques to optimize the molecules with respect to a desired value of a property of interest (see Fig. 2, middle panel, for conceptually understanding the optimization stage). We based our optimization routine on the HotFlip algorithm [36]. This algorithm approximates the best possible flip of two tokens based on one neural network function evaluation (forward pass) and one backward pass. The adversarial loss (in our case, the desired loss) can be approximated by:

$$\mathcal{L}\left(y_{\mathrm{des}}, f\left(\mathrm{t}_{\mathrm{des}}\right)\right) \approx \mathcal{L}\left(y_{\mathrm{des}}, f(\mathrm{t})\right) + \nabla\mathcal{L}\left(y_{\mathrm{des}}, f(\mathrm{t})\right)\left(\mathrm{t}_{\mathrm{des}} - \mathrm{t}\right) \quad (1)$$

with y_{des} being the desired value, $\mathrm{t}_{\mathrm{des}}$ the unknown desired molecule, y the current property value and t the current molecule. This loss is then minimized with respect to $\mathrm{t}_{\mathrm{des}}$ by:

$$\arg \min_{\mathrm{t}_{\mathrm{des}}} \nabla\mathcal{L}\left(y_{\mathrm{des}}, f(\mathrm{t})\right)\left(\mathrm{t}_{\mathrm{des}} - \mathrm{t}\right) \quad (2)$$

Practically, for our graph model we used the embeddings of the atom numbers in the graph encoder as t and $\mathrm{t}_{\mathrm{des}}$ in Eqs. 1 and 2. We limit the flips to a restricted number of possible atom numbers, avoiding the potential

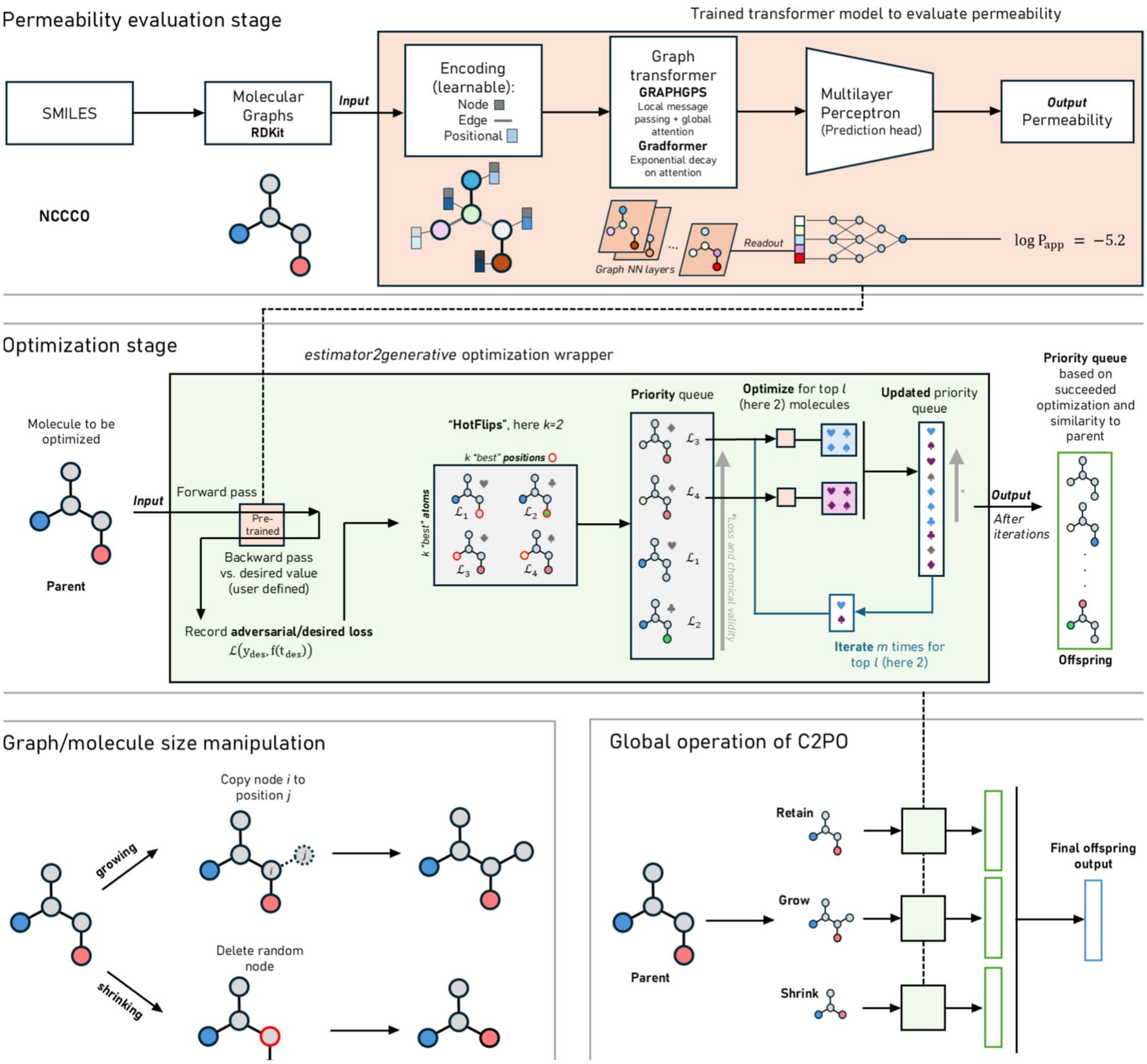

Fig. 2 An overview of the inner workings of C2PO (Cyclic Peptide Permeability Optimizer), visualizing the method description in the main text. At the heart sits a ML estimation model (top panel), trained on the CycPeptMPDB dataset [19]. Input SMILES are converted to molecular graphs, which are steered through a graph transformer model to estimated permeabilities. The trained transformer model is deployed in the optimization stage (middle panel). The estimator2generative optimization of a molecule using the permeability ML model of the top (orange box) is depicted in the green box using a fictive example. The optimizer retains the graph size, limiting the search space. Therefore, graphs are manipulated to either grow or shrink (bottom left). To explore various molecular sizes, C2PO can be operated using parallel optimization tracks (user defined) and collects the results from all parallel optimizations in a single pool of offspring molecules (bottom right). All molecular depictions and choices of parameters are fictive, merely serving as illustration of the algorithm's flow

bias in the optimization when the model chooses chemical elements rarely seen in training. Since flipping atoms generally leads to invalid molecules, we only flip once instead of allowing multiple flips at once as originally described in the HotFlip algorithm [36]. Instead, we use one backward pass to flip to the best k atoms at the best k positions (see Fig. 2, middle panel, with $k = 2$ as example). The resulting molecules are placed on a priority queue based on their desired loss. The desired loss for invalid molecules in this iteration is increased by the maximum loss of the valid molecules such that invalid molecules are placed after the valid molecules on the priority queue for each iteration. Next, we try to improve for a given number of iterations the top l molecules on this queue (see

Fig. 2, middle panel, with $l = 2$ as example) and add to it the newly optimized molecules, like beam search. After the iterative procedure (dark blue iteration loop m in Fig. 2, middle panel), all the molecules from the optimization are placed in a new and separate priority queue with the Tanimoto similarity to the original compound subtracted from the desired loss in increasing order. The approach described so far does not allow the molecule to grow or shrink.

To broaden the search space of our optimization, we manipulated the graph data directly and used simple techniques to grow and shrink the graph while staying as close as possible to the original graph. Note that this may result in graphs that no longer represent correct molecules. To grow the graph, we choose a node at position i randomly and place its duplicate at position j. The node features are identical, and the new node was then connected to the same nodes as its original, copying the edge features. This generally will lead to invalid molecules but can be considered as an intermediate step to find even better optimizations. To allow for graph shrinkage, we randomly delete nodes from the graph. Here we base ourselves on the index of the graph node. We use the simple heuristic that it is likely that two nodes next to each other in the list of nodes are connected in the molecule. Therefore, when we remove a random node from the graph, we replace the deleted node in its edges with the previous node, in a way collapsing the deleted node with the previous node.

Note that the graphs may no longer represent correct molecules, but these are given to the optimization routine to broaden the search space. At the end of the routine (after the optimization iterations), however, graphs will be converted to SMILES, representing the final output of the optimization. For the output chemical structures to make sense we pass them through a RDKit validity check. Invalid molecules are discarded. Chirality is lost in the process and outputs will, therefore, not contain stereochemical information.

Case study setup

All permeability values are reported in terms of the logarithm of the permeability velocity ($\log P_{app}$). The values range between −10 and −4, which is equivalent to 1.0×10^{-10} and 1.0×10^{-4} cm/s, respectively. From all cyclic peptides in CycPeptMPDB, 700 (~10%) compounds that were classified as having low permeability ($\log P_{app}$ of −6.5 or less), were randomly drawn as starting points for the permeability improvement through molecular modification with the above-described optimization model. These drafted systems approximately span the entire chemical space represented by the CycPeptMPDB dataset, as detailed in the Supplementary Information (see Additional file 1). The starting point of each campaign is referred to as the *parent* compound. The model is instructed to return a maximum of 20 *offspring* compounds per campaign, with a desired permeability value of -4.5 as objective. It should be noted that the exact, absolute value of this threshold is not critical and does not have any specific chemical rationale (see Discussion). As our model is gradient-based, the desired value merely serves to provide the optimizer with a direction, with the goal of shifting 'low' permeable peptides towards 'high' permeable ones. Only modifications of the peptide side chains were allowed. RDKit was used to identify the atom indices constituting the peptide backbone through the SMARTS pattern search "[C;X4;H1,H2][CX3](=O)[NX3][C;X4;H1,H2][CX3](=O)". The peptide backbone macrocycle was preserved by retaining cycle sizes consisting of 11 atom members or more using RDKit's GetRingInfo() routine. The optimization algorithm was only allowed to change existing elements to the chemical elements C, N, O, Cl, F, Br, and S. Upon altering the cyclic peptide, the model was permitted to return offspring molecules up to a net removal or insertion of five atoms. The intrinsic optimization parameters k, l, and m (see Fig. 2, middle panel) were all set to five. All settings mentioned can be changed by the user. The Tanimoto similarity measure was used to estimate structural similarity throughout this contribution.

Molecule auto-correction

During the last step in our workflow, a dictionary-based molecular auto-correction tool is applied to inquire about the validity of a molecule and to subsequently correct invalid molecules [28]. The output SMILES formats from the first permeability optimization step are fed to the tool. The drug-like molecules of ChEMBL31 [37] were used as reference to create the dictionary of chemical validity. For the creation of said dictionary we set the circular atomic environment radius to 1. The molecule auto-correction algorithm is implemented as a tree search, with vertices representing molecules. The input molecule serves as the root of the tree. At each iteration the tree is built up by selecting a molecule and enumerating some of its analogs through systematic application of graph-based perturbations [38]. Which molecules are selected for expansion, as well as which analogs are enumerated, is governed by a set of policies. We refrained from exploring the available types of policies, and performed molecular correction using the default policies provided by the developers [28].

Results

ML estimation model

Underneath the cyclic peptide optimizer sits a trained ML estimation permeability engine. Hence, the optimizer's meaningfulness relies on the quality of the permeability estimation when presented with a chemical structure. Figure 3 depicts the comparison between the true (abscissa) and the estimated permeability (ordinate) for all cyclic peptides contained in CycPeptMPDB in separate subplots for the train (80%), validation (10%) and test (10%) splits. Visually, the datapoints in the scatterplots of Fig. 3 are concentrated around the equality diagonal (black line), which is indicative of a meaningful ML estimation model. This is reflected in the R^2, Pearson correlation (r), and the mean average error (MAE) values of respectively 0.61, 0.78, and 0.37 for the test set. Following a comparative analysis with related studies [20–27], detailed in the Supplementary Information (see Additional file 1), we conclude to have a state-of-the-art model. In this contribution we are mainly concerned with improving permeability upon chemical modification, and, within this context, utilizing this ML estimation model is justified. Finally, certain cyclic peptides were assigned a value of −10 for permeability by the authors of the CycPeptMPDB work, as the experimental detection limit did not allow for a proper permeability determination [19]. We moved this lower bound from −10 to −8.5 prior to model training and deployment to have the arbitrary set values closer to the other ones. Our estimation model cannot handle those cases properly, as can be observed in Fig. 3 as a vertical line at −8.5. Other cases of structured datapoints on vertical lines can be observed for instance at −7. This is due to decisions made by either the original reporters of the experimental values or the way the authors of CycPeptMPDB interpreted the results. We do not delve deeper into the specific reasons, as we estimated its effect to be minimal for our model and case study. However, we should note that this has also been observed for other experimental datasets, such as the binding affinity values reported in ChEMBL [39]. Here, we decided to accept the noise that is added as a consequence of retaining these datapoints in the dataset.

Permeability optimization

Now, we turn to the main outcome of this contribution, namely improvement of permeability by mutating the chemical structure of cyclic peptides, driven by our developed *estimator2generative* optimizer. Figures 4 and 5 are illustrations of the permeability progressions per individual campaign and in property distributions, respectively. In Fig. 4, each vertical line represents a campaign, where the parent and offspring molecules are assigned a dark and light grey color, respectively. In total, 13,043 offspring molecules were produced. Based on the Tanimoto similarity, we identified that there are seldom cases (227 or 1.74%) where the offspring molecule is not changed with respect to the starting parent compound. In general, most proposed cyclic peptides did exhibit an ameliorated permeability (higher $\log P_{app}$). The violin plots in Fig. 5 also display this upward trend when relating the offspring with the parent set of compounds. Nevertheless, the model does also return cases where optimization failed (lower $\log P_{app}$), albeit these cases are a minority. The optimization algorithm tries to find atom flips that improve permeability, but there are cases where no or only a few better structures are found. In those cases, the algorithm does return worse offspring compounds. However, improvements in permeability do not always translate to successful campaigns. Namely, when analyzing permeability values, one is principally concerned with categorizing molecules as low or high permeable, with the threshold approximately around $\log P_{app}$ of −6.0 [19, 40]. A campaign could be considered successful if at least one offspring molecule is classified as highly permeable.

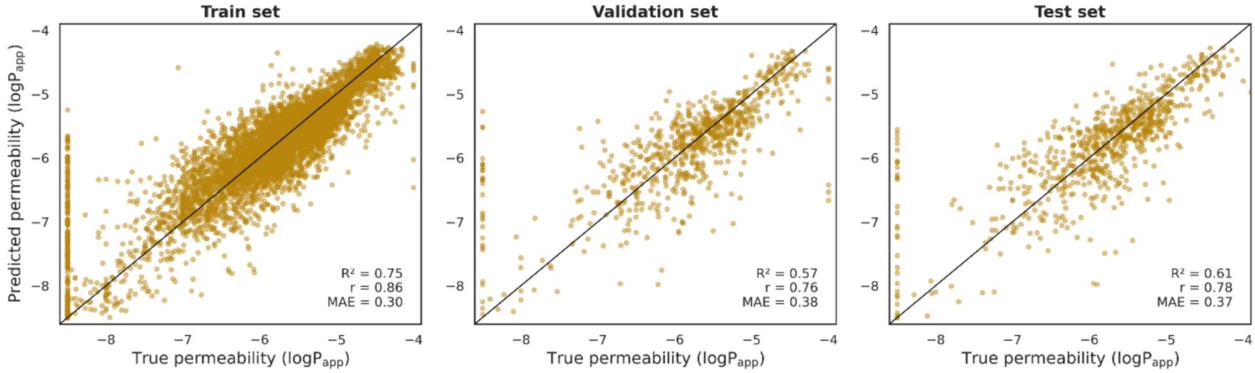

Fig. 3 The true versus ML-estimated permeability for each of the dataset splits CycPeptMPDB. The estimations were performed using our estimation model, where eighty percent of the data was used for training and ten percent each for validation and testing

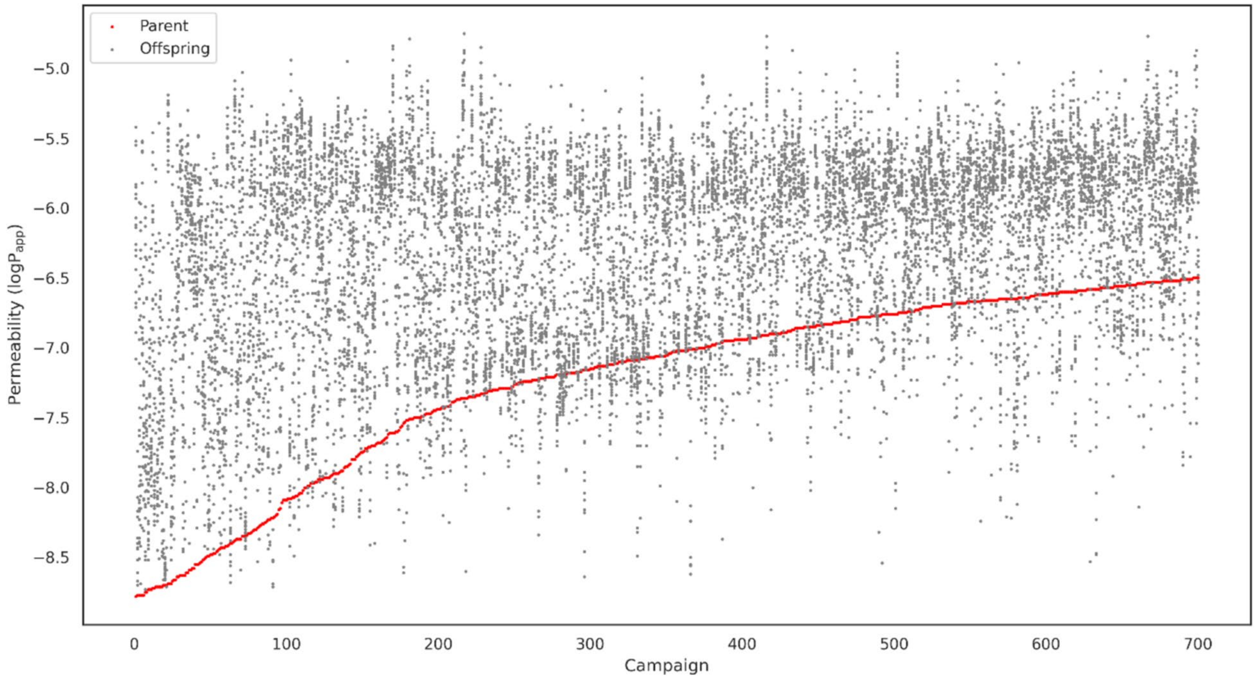

Fig. 4 The permeability values for each individual campaign (vertically oriented). The campaign's parent and offspring molecules have a red and grey color, respectively. The campaigns are sorted according to increasing logP$_{app}$ value. The chemical structures of the parent and best offspring molecule of campaign 1 are visualized in Fig. 1

By using this threshold, we determined that in our case study 538 (76.86%) campaigns and 5,389 (42.05%) actual offspring molecules can be considered successful. When inspecting the offspring violin plot in Fig. 5, the average (boxplot) can be found around a LogP$_{app}$ of −6, and the largest density is found towards −5.6.

Besides the permeability property, we also wanted to gauge the complexity of the structural changes that were introduced in the algorithm. Figures 6 and 7 are provided to shed light on this. The two figures are identical, apart from the properties loaded as heatmap. Figures 6 and 7 depict the improved permeability versus the measure of structural change, expressed as the Tanimoto similarity between each offspring and parent compound. This similarity measure equals 1 for identical molecules. Additionally, we investigated the permeability optimization in light of the starting molecular weight and molecular weight changes, and this is detailed in the Supplementary Information (see Additional file 1). We do not observe any relation between the extent of structural transformations and the improvement of permeability. The heatmap in Fig. 6 shows the absolute permeability outcomes for each of the offspring peptides, where indeed, a better outcome logically relates to a stronger permeability jump. Another interesting viewpoint is shown in Fig. 7, which focusses on the parents' absolute permeabilities. When deploying an improvement algorithm, it preferably behaves in such a way that it recognizes the extent of optimization needed and acts accordingly. Imagine a scenario where one starts from a strongly unfavorable permeability (e.g., a logP$_{app}$ of −8.0). The algorithm will have to cover a larger value range than a compound that is already close to the −6.0 border. The blue points in Fig. 7 are the unfavorable starting points and are subjected to strong permeability changes (a higher permeability improvement). Moreover, the model does not necessarily achieve this by applying stronger chemical modifications, as we mentioned before. This implies that the model can identify efficient transformations, and that improvement is not a mere consequence of an increased number of chemical modifications.

Chemical structure sanity check

To date, it remains a major challenge to develop generative models that propose correct chemistry, even when they are trained on chemical data. Herein, we apply an automated molecular correction tool. In the first step, it diagnoses foreign chemical features within molecules using an underlying chemical reference dictionary. When foreign hits are present, a subsequent molecular correction is performed. The dictionary check ruled 2,931 out of the 12,816 generated offspring molecules as chemically invalid, being a significant portion (22.9%) of all the generated offspring compounds. The

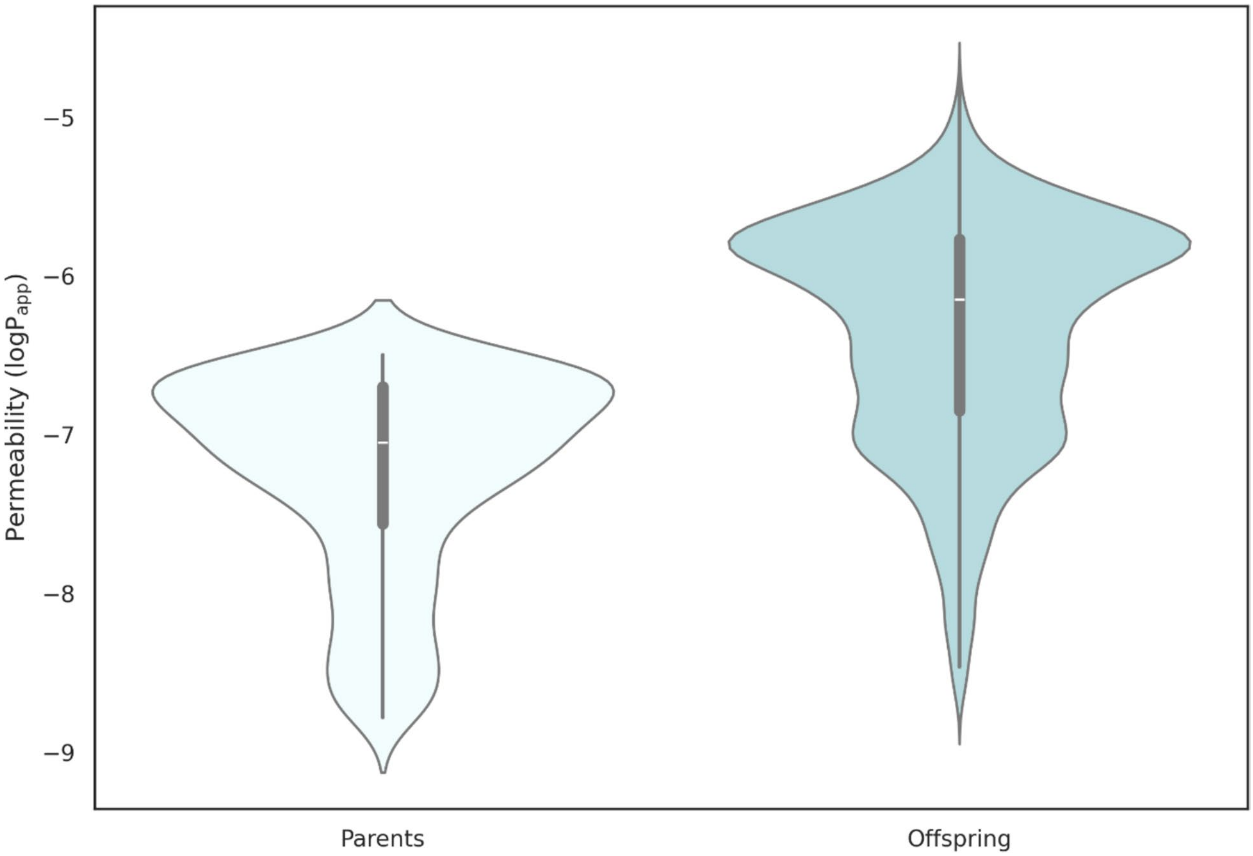

Fig. 5 Violin plots of the permeability values of the 700 parent molecules and the 12,816 offspring molecules. These plots hold boxplot information and inform about the value distribution through a kernel density plot

downside of our strategy is that an additional chemical modification step is undertaken, potentially counteracting the initial permeability optimization task. Figure 8 tracks the permeability change upon correction. The most important observation here is the high and linear correlation between pre- and post-correction permeability, and that there is no clear trend between scatter points falling below or above the identity diagonal. This indicates that the correction process does not systematically make permeability worse or better. Five scenarios manifest:

1) the pre-correction offspring's permeability was successfully optimized (above -6.0), and remains optimized after molecular correction (north-eastern quadrant, I);
2) the pre-correction offspring's permeability optimization was not successful, but falls in the desirable permeability range upon correction (north-western quadrant, II);
3) both the pre- and post-corrected molecules fall in the low permeability range (south-western quadrant, III);
4) the desired permeability exhibited by the offspring compound is lost upon molecular correction (south-eastern quadrant, IV);
5) the molecular correction mutates the chemical structure but returns the parent compound of the campaign.

Cases that fall in the northern quadrants (I and II) are considered positive outcomes, whereas the southern quadrants represent ultimately (III and IV) failed cases. Scenario 1 is the desired scenario, as these campaigns display the desired functioning of the two tools. The cases in scenario 2 are mere products of coincidence, as the correction tool is unaware of the permeability property. Scenario 3 cases align with a logical outcome: one cannot expect that the second step, merely focused on molecular correction, co-improves the permeability. Lastly, scenario 4 are cases where the structural correction counteracts the initial permeability amelioration

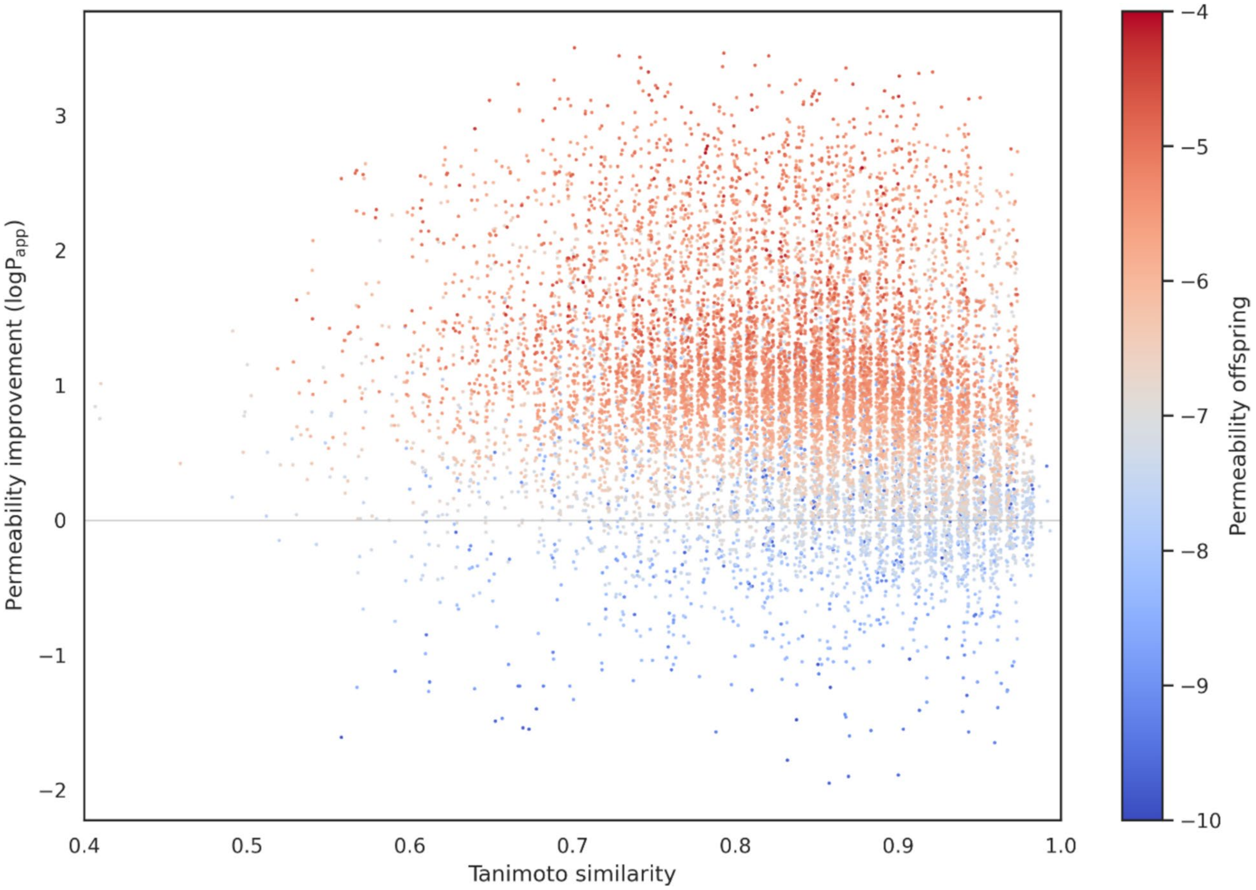

Fig. 6 The Tanimoto similarity between the offspring and parent molecule versus the permeability change. The heatmap is colored according to the absolute permeability value of the offspring molecules

and is highly undesirable. The number of cases for each of five scenarios are 881 (scenario 1), 190 (scenario 2), 1,468 (scenario 3), 280 (scenario 4), and 112 (scenario 5). For the two eastern quadrants, most cases fall into the desired quadrant I. However, one should keep in mind that negative scenarios because of attaching the two tools together do occur (scenario 4). Lastly, Fig. 8 portrays the similarity heatmap between the structure of the corrected offspring molecule and the campaign's parent compound. No correlation between this similarity and the quadrants are observed.

Discussion

From the presented results, we can conclude that the overall framework herein performs as desired. The model does propose permeability-optimized cyclic peptides for most of the cases. It behaves in line with such a model, namely it suggests structure efficient ways of modifying cyclic peptides towards improving permeabilities and suggests potential improvement paths according to the extent of required optimization (strong permeability improvement for strongly unfavorable starting points). Then, most of the offspring cases, submitted to automated molecular correction, do not see the optimized permeability annulled. This workflow, in its current form, can be deployed to assist in improving permeabilities of cyclic peptides. It should be emphasized that this contribution did not pursue obtaining as many successful optimization campaigns as possible. It is up to the user to deploy this method to accomplish specific objectives.

Our core estimation model is not a perfect permeability value estimator, and caution should be practiced when interpreting the results. We demonstrated that our estimation model performs in line with related models, without pushing the limit to outperform them. Our current model can potentially be ameliorated by adopting a multimodal approach, *i.e.*, feeding the ML model with different data structures and molecular information. This

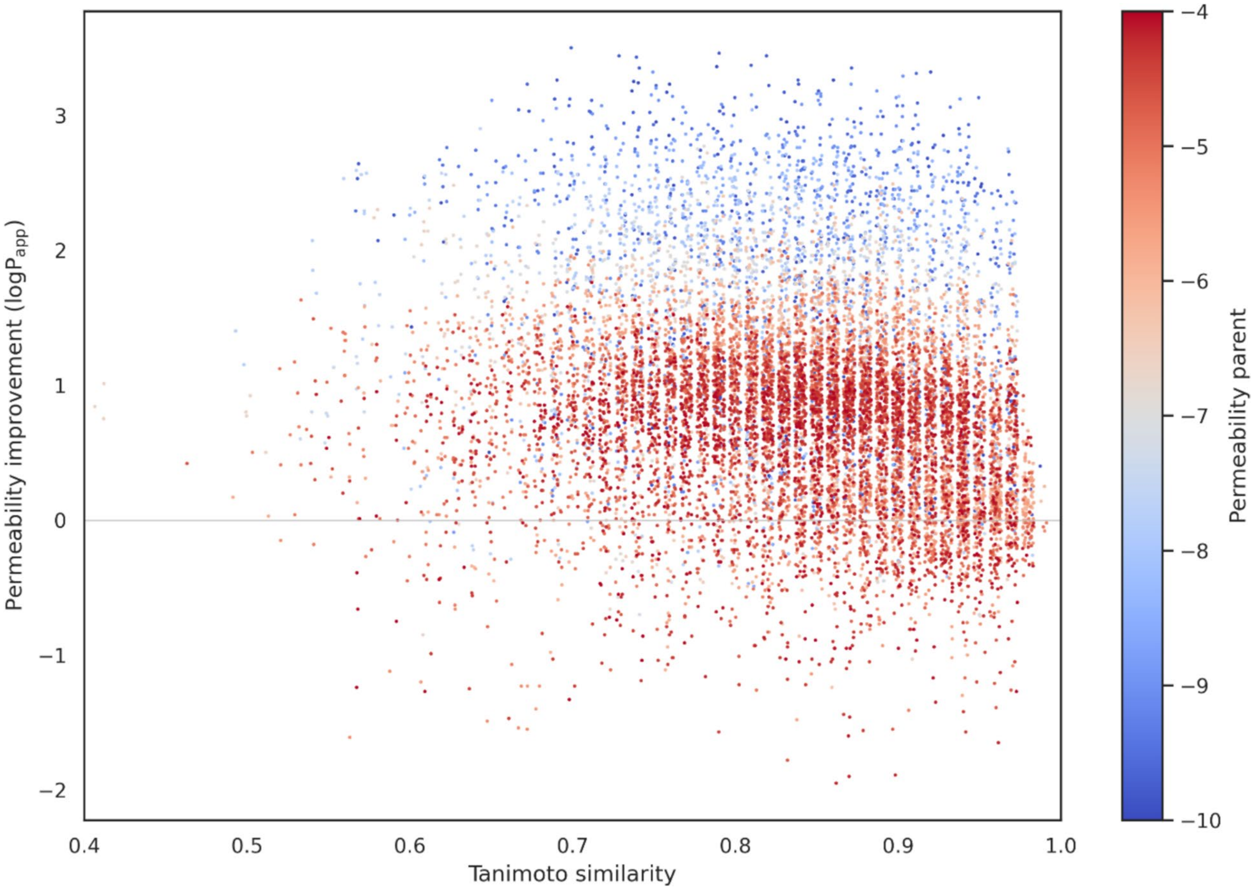

Fig. 7 The Tanimoto similarity between the offspring and parent molecule versus the permeability change. The heatmap is colored according to the absolute permeability value of the parent molecules

is beyond the scope of the present work. In addition, we tried to conduct an independent validation of our permeability estimator model in case of the so-called optimized offspring. This validation failed, as expounded in the Supplementary Information (see Additional file 1). The advised manner to consider the absolute values outputted is to use them to categorizing molecules in low or high permeability classes [40]. Here, we set the threshold at −6.0 $\log P_{app}$ for pragmatic reasons. We envision our methods to be used by chemists as an idea generator. When presented with a structure, the model suggests a set of optimized molecules. Those that surpass the provisioned threshold may be valid options, and, based on the exact output permeability value, hints are provided as to rankings within this set of potentially valid options. Again, be aware that this ranking is fallible.

We understand that our case study does not necessarily reflect the objectives of future users. Here, only side chains are suggested by the algorithm and modifications are allowed to increase or reduce the number of atoms by five (see Methodology). These factors can be changed according to the desires of the user. Firstly, our model supports flexibility as to instructing the model in retaining and/or modifying certain types or specific parts of starting molecules, for instance, allowing changes in the backbone of peptides or to keep the principal parts of the cyclic peptide for binding to the target. For instance, it is possible for cyclic moieties (cyclic systems in the side chains, disulfide-cyclized peptides, etc.) to be disrupted in our presented case study. If desired, these moieties can be protected through different settings. Secondly, the pool of chemical elements the model can use to replace existing atoms can be specified, although one should be aware that the underlying estimation model is trained on compounds containing specific chemical elements. For example, one can opt to allow carbon-atom insertions only. Thirdly, often times generated molecules remain highly like the starting point. The net atom insertions or deletions can be specified. Fourthly, the model can be retrained using different data if desired, in case the user has a custom dataset or when an extended or alternative dataset is published in the future. Lastly, it is advised that

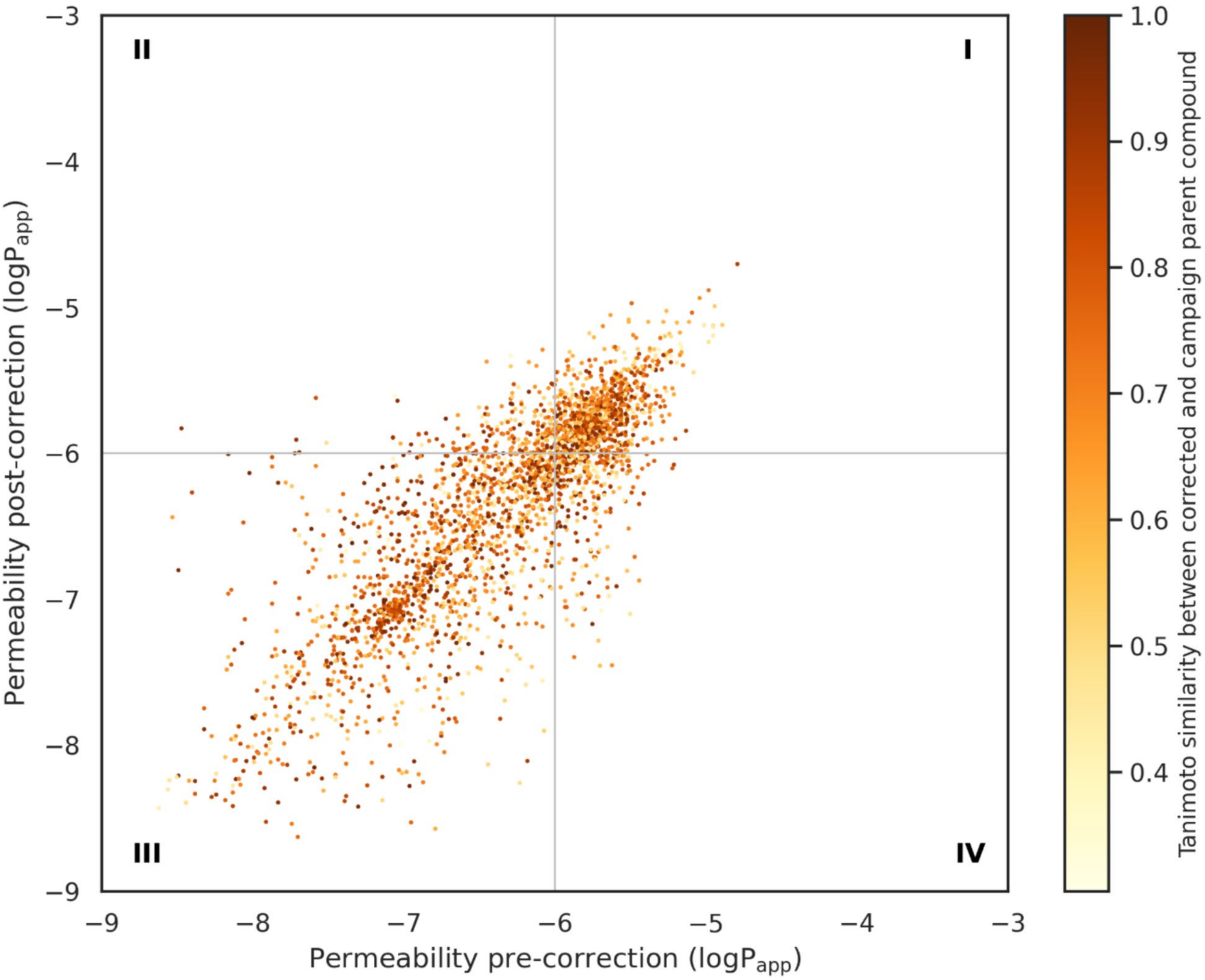

Fig. 8 The permeability value pre- and post-correction with the molecular auto-correction method. The heatmap is colored according to the Tanimoto similarity between the corrected and the parent compound

the optimization parameters *k*, *l*, and *m* are altered to satisfy the needs of the study. In our case, we obtained a decent result with an ad hoc choice of these parameters, and did not systematically probe the optimization performance versus the values of parameters *k*, *l*, and *m*.

Then, consider the second step in our workflow: the molecular correction powered by a chemical validity dictionary. In most cases the post-correction of successfully optimized offspring compounds (from step 1 in the workflow) preserves the desired permeability. However, there are scenarios where permeability worsens upon chemical correction, causing them to return to the low permeability category. Nevertheless, post-applying the auto-correction tool is appealing due to its speed (no development needed, quick execution upon inquiry) and lack of interference with the optimization engine of the model. The flip of the coin of this black-box tool is that one relies heavily on the database used within the dictionary. Here we utilized the ChEMBL dataset as in the original study of the molecular auto-correction tool, which is meaningful chemistry in our opinion. The user has the option to customize the underlying dictionary by feeding it with a dataset of own selection [28].

On a more technical note, the current two independent steps of molecular optimization and correction can be interfaced more deeply. As both software packages are capable of manipulating RDKit *mol* objects, it becomes possible to directly send compounds as *mol* objects from C2PO to the auto-correction tool. This means that an intermediate rendering and validity check by RDKit is skipped (transferring full responsibility for structural validity to the molecular auto-correction tool), potentially leading to a higher throughput of the optimization to the correction tool and a better overall result. An even deeper connection between the two applications can be established by defining the ML permeability estimation model as an objective function for setting up an *explicit objective preservation* selection policy within the autocorrection tool. As such, the algorithm will be simultaneously optimizing the permeability objective and chemical correctness. The downside of any of the integrations mentioned is that the two applications are strictly tied together (no intermediate SMILES outputs), removing their independent functioning and deployment. Here we opted to keep the two steps separate, but it is important to be aware of the possibility of linking tools together through different data formats, as is the case here.

Finally, for researchers that will use the C2PO framework to tackle different optimization questions, including other chemistry and molecular properties, it is advised to play around with all the settings to reach an adequate performance. The parameters we utilized here are not necessarily transferable to other scientific questions.

Conclusion

Generally, cyclic peptides lack adequate membrane permeability to be developed into medicines. We propose C2PO (Cyclic Peptide Permeability Optimizer), an application that improves permeability by modifying the chemical structure of a given cyclic peptide. C2PO is ML-driven, trained on the experimental CycPeptMPDB dataset, and can be categorized in the *estimator2generative* optimization paradigm (Fig. 2). However, ML-based applications that output chemical structures, as is the case here, have the tendency of (occasionally) proposing strange chemistry. This is attributable to the loss of chemical knowledge, although it is generally considered to be implicitly learned. Therefore, we opted for checking and correcting (where needed) the outcomes of C2PO using a chemistry library-based autocorrection application in a subsequent step.

This contribution provides insights in what one can expect when applying the two above-mentioned applications. 700 permeability optimization campaigns were launched, where only the peptide side chains were allowed to be modified. In general, we observed optimization for many of the campaigns, meaning that bad permeability starting points were optimized to structures with an estimated permeability above the threshold of −6.0 $\log P_{app}$. In the chemical correctness check step, we identified that a substantial portion of the output structures needed correction. The autocorrection tool modified those, and we tracked how the optimized permeability altered upon chemical correction of the structures (the chemical corrector is not aware of the initial permeability improvement task). Various scenarios took place, but the most important one was that for many campaigns the second step did not counteract the initial permeability optimization.

At the end of our contribution, we discussed in more detail the results obtained by informing the reader about the way to and not to use our model and workflow. Moreover, our C2PO application has some flexibility in setup, allowing the users to perform simulations according to their own needs. We did not pursue the best possible optimization result in this work and, instead, focused on providing insights into the basic capabilities of the applications presented. Nevertheless, we inform about ways of improving the overall performance of permeability optimization and molecular autocorrection.

Finally, with this work we hope to raise a general interest in adopting *estimator2generative* optimizer strategies for tackling chemical problems, as well as deploying chemistry-library driven applications for post-correcting molecular structures generated through ML.

Abbreviations

C2PO	Cyclic peptide permeability optimizer
ML	Machine learning
FDA	U.S. Food and Drug Administration
DL	Deep learning
MAE	Mean average error

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-025-01109-x.

Additional file 1. C2PO: an ML-optimizer of the membrane permeability of cyclic peptides through chemical modification. This file contains: multiplate entries in CycPeptMPDB, the data splitting strategy, the estimation model hyperparameters, a chemical structure space analysis of the parent selection, a comparison analysis with related evaluation models, a relation analysis between molecular weight and permeability gain, and a relation analysis between physicochemical descriptors typically related to permeability.

Author contributions

Conceptualization: RA, JT, MA, HDW; Methodology: RA, JT, MA, AK; Software: JT, MA, AK. Investigation: RA, JT, MA; Writing – Original Draft Preparation: RA, JT, MA; Writing – Review & Editing: all authors. Supervision: MA, JCG-T, GT, HDW.

Funding

This work was partly funded by the Flanders innovation & entrepreneurship (VLAIO) project HBC.2021.112.

Data availability

The code, hyperparameters and the weights of the final model are uploaded to the GitHub repository [https://github.com/UAMCAntwerpen/C2PO/] (https:/github.com/UAMCAntwerpen/C2PO). The data generated for the case study presented herein is also published in the repository. Python notebooks are provided to run custom model training and permeability optimization tasks.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 12 June 2025 Accepted: 12 October 2025
Published online: 11 November 2025

References

1. Sharma K, Sharma KK, Sharma A, Jain R (2023) Peptide-based drug discovery: current status and recent advances. Drug Discov Today 28:103464. https://doi.org/10.1016/j.drudis.2022.103464
2. Pereira AJ, de Campos LJ, Xing H, Conda-Sheridan M (2024) Peptide-based therapeutics: challenges and solutions. Med Chem Res 33:1275–1280. https://doi.org/10.1007/s00044-024-03269-1
3. Wang L, Wang N, Zhang W et al (2022) Therapeutic peptides: current applications and future directions. Sig Transduct Target Ther 7:1–27. https://doi.org/10.1038/s41392-022-00904-4
4. Smith MC, Gestwicki JE (2012) Features of protein–protein interactions that translate into potent inhibitors: topology, surface area and affinity. Expert Rev Mol Med 14:e16. https://doi.org/10.1017/erm.2012.10
5. Merz ML, Habeshian S, Li B et al (2024) De novo development of small cyclic peptides that are orally bioavailable. Nat Chem Biol 20:624–633. https://doi.org/10.1038/s41589-023-01496-y
6. Verdine GL, Walensky LD (2007) The challenge of drugging undruggable targets in cancer: lessons learned from targeting BCL-2 family members. Clin Cancer Res 13:7264–7270. https://doi.org/10.1158/1078-0432.CCR-07-2184
7. Sethio D, Poongavanam V, Xiong R et al (2023) Simulation reveals the chameleonic behavior of macrocycles. J Chem Inf Model 63:138–146. https://doi.org/10.1021/acs.jcim.2c01093
8. Wieske LHE, Atilaw Y, Poongavanam V et al (2023) Going viral: an investigation into the chameleonic behaviour of antiviral compounds. Chem Eur J 29:e202202798. https://doi.org/10.1002/chem.202202798
9. Ramelot TA, Palmer J, Montelione GT, Bhardwaj G (2023) Cell-permeable chameleonic peptides: exploiting conformational dynamics in *de novo* cyclic peptide design. Curr Opin Struct Biol 80:102603. https://doi.org/10.1016/j.sbi.2023.102603
10. Ohta A, Tanada M, Shinohara S et al (2023) Validation of a new methodology to create oral drugs beyond the rule of 5 for intracellular tough targets. J Am Chem Soc 145:24035–24051. https://doi.org/10.1021/jacs.3c07145
11. Dougherty PG, Sahni A, Pei D (2019) Understanding cell penetration of cyclic peptides. Chem Rev 119:10241–10287. https://doi.org/10.1021/acs.chemrev.9b00008
12. Di L (2015) Strategic approaches to optimizing peptide ADME properties. AAPS J 17:134–143. https://doi.org/10.1208/s12248-014-9687-3
13. Di L, Artursson P, Avdeef A et al (2020) The critical role of passive permeability in designing successful drugs. ChemMedChem 15:1862–1874. https://doi.org/10.1002/cmdc.202000419
14. Hosono Y, Uchida S, Shinkai M et al (2023) Amide-to-ester substitution as a stable alternative to N-methylation for increasing membrane permeability in cyclic peptides. Nat Commun 14:1416. https://doi.org/10.1016/j.bmc.2017.08.031
15. Ghosh P, Raj N, Verma H et al (2023) An amide to thioamide substitution improves the permeability and bioavailability of macrocyclic peptides. Nat Commun 14:6050. https://doi.org/10.1038/s41467-023-41748-y
16. Nielsen DS, Hoang HN, Lohman R-J et al (2014) Improving on nature: making a cyclic heptapeptide orally bioavailable. Angew Chem Int Ed Engl 53:12059–12063. https://doi.org/10.1002/anie.201405364
17. Bhardwaj G, O'Connor J, Rettie S et al (2022) Accurate *de novo* design of membrane-traversing macrocycles. Cell 185:3520-3532.e26. https://doi.org/10.1016/j.cell.2022.07.019
18. Faris JH, Adaligil E, Popovych N et al (2024) Membrane permeability in a large macrocyclic peptide driven by a saddle-shaped conformation. J Am Chem Soc 146:4582–4591. https://doi.org/10.1021/jacs.3c10949
19. Li J, Yanagisawa K, Sugita M et al (2023) Cycpeptmpdb: a comprehensive database of membrane permeability of cyclic peptides. J Chem Inf Model 63:2240–2250. https://doi.org/10.1021/acs.jcim.2c01573
20. Xu X, Xu C, He W et al (2024) HELM-GPT: de novo macrocyclic peptide design using generative pre-trained transformer. Bioinformatics 40:btae364. https://doi.org/10.1093/bioinformatics/btae364
21. Geylan G, Maria LD, Engkvist O et al (2024) A methodology to correctly assess the applicability domain of cell membrane permeability predictors for cyclic peptides. Digit Discov 3:1761–1775. https://doi.org/10.1039/D4DD00056K
22. Cao L, Xu Z, Shang T et al (2024) Multi_cycgt: a deep learning-based multimodal model for estamating the membrane permeability of cyclic peptides. J Med Chem 67:1888–1899. https://doi.org/10.1021/acs.jmedchem.3c01611
23. Wang Z, Chen Y, Ye X, Sakurai T (2024) CyclePermea: Membrane Permeability Prediction of Cyclic Peptides with a Multi-Loss Fusion Network. In: 2024 International Joint Conference on Neural Networks (IJCNN). pp 1–8
24. Yu Y, Gu M, Guo H et al (2024) MuCoCP: a priori chemical knowledge-based multimodal contrastive learning pre-trained neural network for the prediction of cyclic peptide membrane penetration ability. Bioinformatics 40:btae473. https://doi.org/10.1093/bioinformatics/btae473
25. Tan X, Liu Q, Fang Y et al (2024) Peptide permeability across diverse barriers: a systematic investigation. Mol Pharm 21:4116–4127. https://doi.org/10.1021/acs.molpharmaceut.4c00478
26. Feller AL, Wilke CO (2025) Peptide-aware chemical language model successfully predicts membrane diffusion of cyclic peptides. J Chem Inf Model 65:571–579. https://doi.org/10.1021/acs.jcim.4c01441
27. Wang Z, Chen Y, Shang Y et al (2025) Multicycpermea: accurate and interpretable prediction of cyclic peptide permeability using a multimodal image-sequence model. BMC Biol 23:63. https://doi.org/10.1186/s12915-025-02166-2
28. Kerstjens A, De Winter H (2024) Molecule auto-correction to facilitate molecular design. J Comput Aided Mol Des 38:10. https://doi.org/10.1007/s10822-024-00549-1
29. Dwivedi VP, Bresson X (2021) A generalization of transformer networks to graphs. https://arxiv.org/abs/2012.09699
30. Rampášek L, Galkin M, Dwivedi VP, et al (2023) Recipe for a general, powerful, scalable graph transformer https://doi.org/10.48550/arXiv.2205.12454
31. Landrum G (2013) RDKit: a software suite for cheminformatics, computational chemistry, and predictive modeling. Greg Landrum 8:5281
32. Dwivedi VP, Luu AT, Laurent T, et al (2022) Graph neural networks with learnable structural and positional representations https://arxiv.org/abs/2110.07875
33. Liu C, Yao Z, Zhan Y, et al (2024) Gradformer: graph transformer with exponential decay https://arxiv.org/abs/2404.15729
34. Paszke A, Gross S, Massa F, et al (2019) PyTorch: an imperative style, high-performance deep learning library https://arxiv.org/abs/1912.01703
35. Fey M, Lenssen JE (2019) Fast graph representation learning with PyTorch geometric https://arxiv.org/abs/1903.02428
36. Ebrahimi J, Rao A, Lowd D, Dou D (2018) HotFlip: White-Box Adversarial Examples for Text Classification. arXiv:171206751 [cs]
37. Gaulton A, Bellis LJ, Bento AP et al (2012) ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Res 40:D1100–D1107. https://doi.org/10.1093/nar/gkr777
38. Kerstjens A, De Winter H (2023) A molecule perturbation software library and its application to study the effects of molecular design constraints. J Cheminform 15:89. https://doi.org/10.1186/s13321-023-00761-5
39. Landrum GA, Riniker S (2024) Combining IC50 or Ki values from different sources is a source of significant noise. J Chem Inf Model 64:1560–1567. https://doi.org/10.1021/acs.jcim.4c00049
40. Zhu C, Jiang L, Chen T-M, Hwang K-K (2002) A comparative study of artificial membrane permeability assay for high throughput profiling of drug absorption potential. Eur J Med Chem 37:399–407. https://doi.org/10.1016/S0223-5234(02)01360-0

Publisher's Note