JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

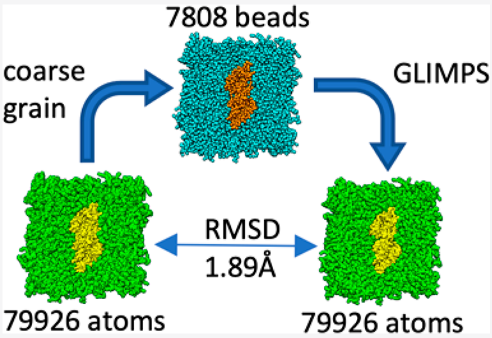
GLIMPS: A Machine Learning Approach to Resolution Transformation for Multiscale Modeling

Keverne A. Louison, Ian L. Dryden, and Charles A. Laughton*

Cite This: https://doi.org/10.1021/acs.jctc.1c00735 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We describe a general approach to transforming molecular models between different levels of resolution, based on machine learning methods. The approach uses a matched set of models at both levels of resolution for training, but requires only the coordinates of their particles and no side information (e.g., templates for substructures, defined mappings, or molecular mechanics force fields). Once trained, the approach can transform further molecular models of the system between the two levels of resolution in either direction with equal facility.



1. INTRODUCTION

A key component of almost all multiscale modeling applications is a tool that enables models of a system at one resolution (e.g., atomistic) to be transformed into their equivalent at another (e.g., coarse grained, CG). In most cases, the process of going from a higher resolution to a lower is quite straightforward: According to the rules of the particular coarse-graining approach, each low-resolution particle has its position (Cartesian coordinates) defined as the center of mass, or similar, of a defined subset of the higher resolution particles. This process is typically called “mapping”. However, the reverse transformation from a low-resolution representation of the system to a higher (backmapping) is of necessity a more complex procedure since there is, by definition, insufficient information in the low-resolution data alone to allow for an unambiguous assignment of the coordinates at the higher level of resolution. A wide variety of computational approaches to backmapping have been explored. Many are tied quite closely to certain simulation packages or coarse-graining schemes, making use of both their knowledge base and structure refinement facilities to achieve the resolution transformation.[1−6] More recently a number of approaches based on advanced machine learning (ML) methods (e.g., generative adversarial networks,[7,8] autoencoders,[9] Gaussian process regression, random forests,[10] Baysian inference[11]) have also been explored, though in general so far only demonstrated on very specific classes of problems and/or molecules of limited size.

As part of a current research project exploring ligand binding to the human beta-1 adrenoceptor, (β1-AR) we were interested in comparing the behavior of CG (Martini)[12] and atomistic simulations of the same system. In the course of this work, we discovered that combinations of relatively traditional supervised and unsupervised ML methods can provide a route to backmapping CG models to atomistic representations that requires no side information (e.g., substructure templates, molecular mechanics force fields), is not specific to any particular molecular type, is quick and easy to train, and is flexible with regards to the coarse-graining scheme.

In the next section, we describe the ML elements of our approach, which in addition to multivariate linear regression makes use of graph methods, principal component analysis (PCA), and restraint optimization. Then we evaluate the performance of the method when applied to multiscale (atomistic/Martini)[12] models of GPCR/lipid systems and compare with an established approach (the CG2AT2 code of Vickery and Stansfeld).[6]

2. METHODS

2.1. Molecular Data Representation. The ML methods discussed here include both supervised and unsupervised approaches and so require training and testing data sets at both levels of resolution. We define $\mathbf{X}_{\text{train}}$ as an $S{\times}N$ matrix of Cartesian coordinates corresponding to S training samples of a system of $N/3$ particles, and $\mathbf{Y}_{\text{train}}$ similarly as an $S{\times}M$ matrix of S samples of a system of $M/3$ particles. $\mathbf{X}_{\text{test}}$ and $\mathbf{Y}_{\text{test}}$ are vectors of length N and M, respectively. We assume that

Received: July 21, 2021

coordinate sets within $\mathbf{X}_{\mathbf{train}}$ and $\mathbf{Y}_{\mathbf{train}}$ have independently been least-squares fitted to remove any global translation and rotation, for example, by generalized Procrustes analysis,[13] and $\mathbf{X}_{\mathbf{test}}$ and $\mathbf{Y}_{\mathbf{test}}$ are also least-squares fitted to $\langle \mathbf{X}_{\mathbf{train}} \rangle$ or $\langle \mathbf{Y}_{\mathbf{train}} \rangle$ as appropriate.

2.2. Application of General Linear Models to Molecular Data. General linear models (GLM) use an $N \times M$ matrix $\mathbf{B}$ to predict $\mathbf{Y}$ from $\mathbf{X}$ (for simplicity we assume that $\mathbf{X}$ contains an element of constant value 1, so one row of $\mathbf{B}$ represents the intercepts):

$$\mathbf{Y}_{\mathbf{test}} \cong \mathbf{X}_{\mathbf{test}}\mathbf{B} \tag{1}$$

The elements of $\mathbf{B}$ can be estimated by multiple linear regression using the training sets $\mathbf{X}_{\mathbf{train}}$, $\mathbf{Y}_{\mathbf{train}}$. If $M < \min(N, S)$, then the predicted values of $\mathbf{Y}$ are linearly independent, but if $M > N$, then no matter how large the training set, the problem is under-determined and the values of $\mathbf{Y}$ are not linearly independent, which is the case of interest. To implement GLM methods here, we use the LinearRegression code from the Python scikit-learn package.[14]

2.3. Application of Principal Component Analysis to Molecular Data. Although the application of PCA to molecular data is now quite well established, for completeness we outline the key elements of the process here. Principal component analysis of the $S \times N$ matrix $\mathbf{X}$ yields an $N \times L$ eigenvector matrix $\mathbf{W}$, where $L = \min(S, N)$, that transforms each row vector x_i, of $\mathbf{X}$ (each configuration) into a new vector of length L, usually called a scores vector, t_i:

$$t_i = x_i\mathbf{W} \tag{2}$$

or, over the whole set:

$$\mathbf{T} = \mathbf{XW} \tag{3}$$

While the total variance in the original data $\mathbf{X}$ may be arbitrarily distributed among the elements of $\mathbf{X}$, the PCA approach results in the first column of $\mathbf{T}$ explaining the maximum component of the variance, while subsequent columns capture in turn the maximal amount of the subsequently remaining variance. Because of the highly correlated nature of particle motions in a typical molecular system, these column variances tend to decrease rapidly along the series, and since each column mean is zero (from the mean centering done initially), later elements of each scores vector t_i also tend to zero. The matrix $\mathbf{W}$ is an orthogonal matrix, so its inverse is its transpose, and this permits the reverse transformation of a scores vector into a vector of Cartesian displacements:

$$x_i = t_i\mathbf{W}^{\mathrm{T}} \tag{4}$$

Since the later elements of t_i tend to zero, their contribution to x_i tends to zero, so we can truncate $\mathbf{W}$ and t_i to smaller number, M, of eigenvectors and scores, and

$$x_{iM} = t_{iM}\mathbf{W}^{\mathrm{T}}{}_{M} \cong x_i \tag{5}$$

that is, it is often possible to reconstruct the Cartesian coordinates of an observation from the first M of its L scores with good accuracy even if $M \ll L$. Here we use our own PCA code that wraps the Python scikit-learn implementation in a form tailored for molecular data.

2.4. Predicting Molecular Topology from Coordinate Data. Given sufficient samples of "good quality" coordinate data, it is possible to estimate the underlying molecular topology. For example, approximately invariant and short interparticle distances may signal bonded interactions. Knowledge of this topology can support additional methods to improve the accuracy of predicted molecular structures. If we assume that the coordinate set is for a single molecule, then, in graph terminology, it forms a single connected component. A short, but not necessarily minimal, spanning tree may be produced as follows. First, the mean $N \times N$ distance matrix $\mathbf{D}$ is calculated for the $S \times N$ coordinate matrix $\mathbf{X}$. Next an unconnected graph of N nodes is generated. The elements of $\mathbf{D}$ are then scanned in order of increasing value (i.e., starting with the shortest distance). For each $\mathbf{D}_{ij}$, if there is no current path (of any length) in the graph from i to j, an edge is added between them. The process continues until all nodes in the graph form a single connected component. Here we use the Python NetworkX package[15] to implement this process.

2.5. Predicting Coordinates from Invariant Geometry: Sprouting and Shaving. The topologies of molecules at full atomistic resolution are likely to contain many singly connected nodes, for example, each hydrogen atom. Typically, the positions of these atoms can be predicted with good accuracy from a knowledge of the positions of their closest "heavy" neighbors because of the stiffness of bond length and angle parameters. Put another way, a $\mathbf{Z}$-matrix for such an atom, if appropriately defined by its relationship to these neighbors, will be almost invariant. This property and process is of course what underpins the many methods available to add hydrogen atoms to heavy-atom only molecular structures (e.g., Reduce),[16] but here we implement the method without the requirement for any knowledge of the molecule's chemistry. First the training data are used to predict the graph, as described above. For each singly connected node i, the connected node j is found. The other nodes connected to j are then found. If there are at least two of them that are not themselves singly connected, then two are chosen, k and l. From examination of the training data, the mean bond length between i and j, the mean angle ijk, and mean (improper) torsion $ijkl$ are found and used to define the $\mathbf{Z}$-matrix entry for particle i. A high-resolution coordinate set $\mathbf{X}$ of size N may thus be decomposed into a $\mathbf{Z}$-matrix of length T for the terminal atoms plus a coordinate set $\mathbf{H}$ of size (N-3T) for the "heavy" atoms. Later on, new coordinate sets for just the (N-3T) heavy atoms may be expanded to all N atoms by application of $\mathbf{Z}$, since each row in $\mathbf{Z}$ defines the coordinates of a terminal atom by reference to coordinates of heavy atoms only. Here we refer to the former process as "shaving" and the latter as "sprouting". Our Python code to implement this leverages the MDTraj package[17] for fast calculation of bond angles and dihedrals.

2.6. Refining Coordinates from Invariant Geometry. In addition to the $\mathbf{Z}$-matrix-based approach for predicting the positions of terminal particles, we evaluate an elastic network-based approach[18] applicable to all particles. We use the graph discussed above to create an elastic network between all bonded particles, and also all pairs of particles bonded to a common neighbor (i.e., we assume all bond angles are almost invariant). Target lengths for each edge in the elastic network are taken from the training set's mean distance matrix, and all force constants set to a common value. The coordinates of crude predicted models are then relaxed using a truncated Newton method (Python package scikit-learn).

2.7. Initializing the Resolution Transformation Method. Astute readers will have noticed an apparent chicken-and-egg conundrum here: This ML method requires a training set

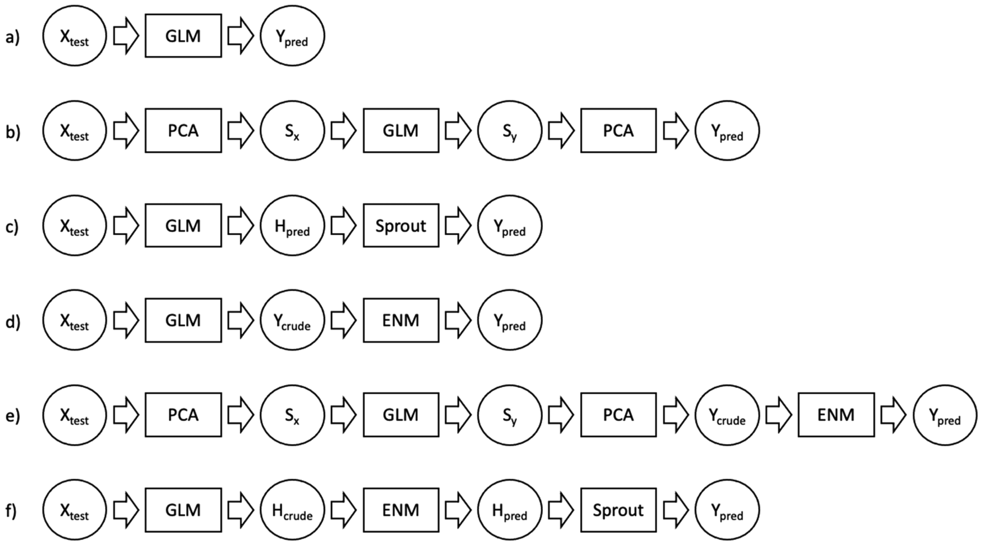

a) X_{test} GLM Y_{pred}
b) X_{test} PCA S_x GLM S_y PCA Y_{pred}
c) X_{test} GLM H_{pred} Sprout Y_{pred}
d) X_{test} GLM Y_{crude} ENM Y_{pred}
e) X_{test} PCA S_x GLM S_y PCA Y_{crude} ENM Y_{pred}
f) X_{test} GLM H_{crude} ENM H_{pred} Sprout Y_{pred}

Figure 1. ML-based backmapping pipelines evaluated in this work. Items in circles are data, and items in boxes are data transformation processes. X_{test}: low-resolution data; Y_{pred}: high-resolution predicted coordinates; S_x: low-resolution PCA scores; S_y: high-resolution PCA scores; H_{pred}: high-resolution heavy atom coordinates; Y_{crude}: approximate (intermediate) high-resolution predicted coordinates; H_{crude}: approximate (intermediate) high-resolution heavy atom coordinates; GLM: general linear model; PCA: principal component analysis; Sprout: terminal atom prediction algorithm; ENM: elastic network energy minimization.

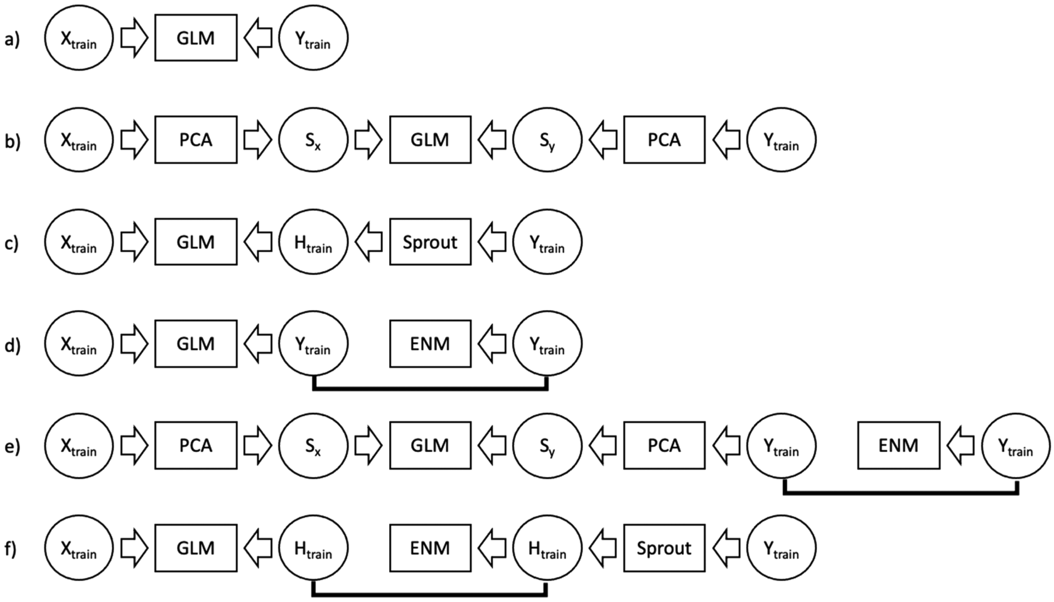

a) X_{train} GLM Y_{train}
b) X_{train} PCA S_x GLM S_y PCA Y_{train}
c) X_{train} GLM H_{train} Sprout Y_{train}
d) X_{train} GLM Y_{train} ENM Y_{train}
e) X_{train} PCA S_x GLM S_y PCA Y_{train} ENM Y_{train}
f) X_{train} GLM H_{train} ENM H_{train} Sprout Y_{train}

Figure 2. Data flows for the training of each of the ML-based backmapping pipelines illustrated in Figure 1. Dark bars indicate where the same data are used to train two steps. Key to symbols in Figure 1.

of matched pairs of structures and high and low resolution - how are these to be generated? This is the one point where side information is required. It is assumed that there is an independent approach to performing the forward mapping transformation, that is, to convert a model at high resolution to one at low resolution. This is typically not a major issue; coarse graining procedures are typically conceptually and computationally straightforward, in contrast to the reverse transformation which is the key aim of the present approach. The parametrization of a new resolution transformation ML pipeline thus begins by taking an ensemble of representative structures at high resolution and applying the already-existing forward mapping approach to generate their low-resolution counterparts. There is no requirement that the molecular systems used for training are the same as those which will be ultimately backmapped, only that the training set(s) between them feature all of the molecular components present in the target system (in the sense of whole molecules, submolecular fragments would not suffice).

2.8. Alternative Resolution Transformation Workflows. The simplest resolution transformation procedure investigated here involves just the GLM step; however, other workflows tested the value of adding a variety of pre- and postprocessing steps (Figure 1).

The simple GLM workflow ("GLM", Figure 1a) requires matrix **B** to be estimated from the training data $\mathbf{X}_{\mathbf{train}}$, $\mathbf{Y}_{\mathbf{train}}$ by multiple linear regression. Since for backmapping, $M > N$, the columns of the $N{\times}M$ matrix **B** are not linearly independent, so

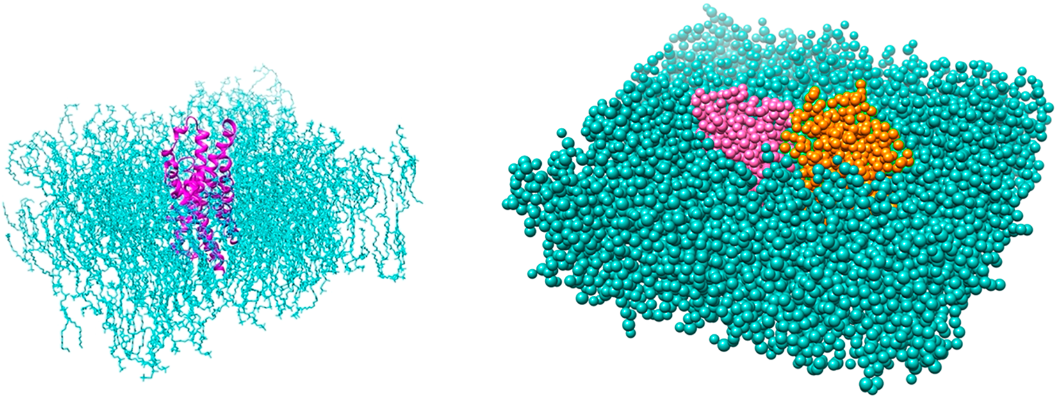


Figure 3. Examples from the training (left) and test (right) data used to evaluate the resolution transformation methods, to illustrate the size and nature of the system. The training set features of one molecule of β1-AR (pink) in a DPPC bilayer (cyan), while the test set features a β1-AR dimer (pink and orange) in a DPPC bilayer.

Table 1. Performance of ML-Based Pipelines Applied to Backmapping Martini Models of DPPC[a]

molecule	options	$\langle \mathrm{RMSD}_{\mathrm{fg}} \rangle$[b]	$\langle \mathrm{RMSD}_{\mathrm{cg}} \rangle$[c]	$\langle \mathrm{RMSE}_{\mathrm{bonds}} \rangle$[d]	$\langle \mathrm{RMSE}_{\mathrm{angles}} \rangle$[e]
DPPC	none	1.35	0.10	0.63	37.2
DPPC	PCA	1.38	0.25	0.67	35.4
DPPC	ENM	1.65	0.62	0.25	22.7
DPPC	Sprout	1.52	0.05	0.25	26.2
DPPC	ENM + Sprout	1.69	0.50	0.11	14.2
DPPC	Sprout + PCA	1.53	0.18	0.26	27.3
DPPC	PCA + ENM	1.68	0.53	0.23	21.7
DPPC	all	1.70	0.53	0.11	14.1

[a]Predicting coordinates of 130 atoms from those of 12 CG beads, training set of 1000 samples, independent test set of 100 samples. [b]Mean RMSD (Å) between true and predicted atomistic structures in the test set. [c]Mean RMSD (Å) between true CG structures in the test set and those generated by forward-mapping the predicted atomistic structures. [d]Mean value of the root-mean-square error in predicted bond lengths (Å). [e]Mean value of the root-mean-square error in predicted bond angles (°).

limiting the potential of the approach to accurately model $\mathbf{Y}_{\mathbf{test}}$, given $\mathbf{X}_{\mathbf{test}}$.

As a potential route to mitigating this, we next explored a workflow ("PCA-GLM-PCA", Figure 1b) in which **X** and **Y** are first subjected to PCA, so the GLM step maps the N scores of **X** (assuming $N \geq$ number of training samples, S) onto the top N scores of Y, and matrix **B** is fully determined. We hypothesized that moving the dimensionality expansion process from the GLM to the inverse transform of the PCA of **Y** (predicting M Cartesian coordinates from N scores) might work better as PCA is designed for such tasks.

The next workflow ("GLM-Sprout", Figure 1c) explored the potential of the "shave" and "sprout" approach described in the Methods section, by which the M **Y** coordinates are separated into a constant **Z** matrix representation for T particles plus a smaller Cartesian coordinate matrix **H** for the remaining (M-3T) coordinates. Now the GLM step has to map the N CG coordinates onto a smaller number of fine-grained ones, and the approach is likely to be more performant; indeed, it might be that (M-3T) < N, and matrix **B** is fully determined.

An elastic network-based coordinate refinement step may be added into each of the above workflows (Figure 1d−f), creating the "GLM-ENM", "PCA-GLM-PCA-ENM", and "GLM-ENM-Sprout" options, respectively.

2.9. Training the Resolution Transformation Pipelines.

All resolution transformation pipelines can be trained using just matched pairs of training coordinate sets at the two levels of resolution; no additional information, for example, molecular topologies or force field parameters, is required (Figure 2).

For the "GLM" pipeline (Figure 2a), the elements of B are fit to $\mathbf{X}_{\mathbf{train}}$ and $\mathbf{Y}_{\mathbf{train}}$ by multiple linear regression. For the "PCA-GLM-PCA" pipeline (Figure 2b), the two PCA models are determined independently from $\mathbf{X}_{\mathbf{train}}$ and $\mathbf{Y}_{\mathbf{train}}$, then each is used to transform the associated coordinates to sets of scores, $\mathbf{S}_{\mathbf{X}}$ and $\mathbf{S}_{\mathbf{Y}}$. The **B** matrix of the GLM is then fitted to these. For the "GLM-Sprout" pipeline (Figure 2c), first the sprout algorithm is parametrized using the high-resolution data set $\mathbf{Y}_{\mathbf{train}}$, then it is applied to that data to generate a "shaved" version of the training set, $\mathbf{H}_{\mathbf{train}}$. Finally, the GLM is fitted to this and $\mathbf{X}_{\mathbf{train}}$. Pipelines containing elastic network minimization steps are trained in the same way, using $\mathbf{Y}_{\mathbf{train}}$ or $\mathbf{H}_{\mathbf{train}}$ as appropriate (Figure 2d−f).

2.10. Performance Metrics.

With a test set $\mathbf{X}_{\mathbf{train}}$ of low-resolution samples and the matched $\mathbf{Y}_{\mathbf{train}}$ high-resolution set, the most obvious performance metric is the RMSD between predicted structures, $\mathbf{Y}_{\mathbf{pred}}$, and $\mathbf{Y}_{\mathbf{train}}$. However, this is not really fair because the forward mapping of any **Y** within a certain envelope could generate the same **X**, so even if $\mathbf{Y}_{\mathbf{pred}}$ deviates from $\mathbf{Y}_{\mathbf{test}}$, it is not necessarily "wrong". Therefore, we also take the $\mathbf{Y}_{\mathbf{pred}}$ structures, forward-map them to $\mathbf{X}_{\mathbf{pred}}$, and measure the RMSD between $\mathbf{X}_{\mathbf{pred}}$ and $\mathbf{X}_{\mathbf{test}}$ (which legitimately should ideally be zero).

In addition to RMSD, we also measure how well the reconstruction procedures generate structures with accurate bond lengths and angles, since we find that these metrics and low RMSD do not always correlate.

Table 2. Performance of ML-Based "GLM-ENM-Sprout" Pipelines Applied to Backmapping Martini Models of DPPC as a Function of the Size of the Training Data Set

molecule	options	$\langle \text{RMSD}_{\text{fg}} \rangle$[a]	$\langle \text{RMSD}_{\text{cg}} \rangle$[b]	$\langle \text{RMSE}_{\text{bonds}} \rangle$[c]	$\langle \text{RMSE}_{\text{angles}} \rangle$[d]
DPPC	$N_{\text{train}} = 500$	1.69	0.51	0.12	13.7
DPPC	$N_{\text{train}} = 100$	1.76	0.54	0.14	14.4
DPPC	$N_{\text{train}} = 50$	1.93	0.69	0.17	14.5
DPPC	$N_{\text{train}} = 20$	2.14	0.95	0.20	15.3
DPPC	$N_{\text{train}} = 10$	2.76	1.87	0.19	15.0

[a]Mean RMSD (Å) between true and predicted atomistic structures in the test set. [b]Mean RMSD (Å) between true CG structures in the test set and those generated by forward-mapping the predicted atomistic structures. [c]Mean value of the root-mean-square error in predicted bond lengths (Å). [d]Mean value of the root-mean-square error in predicted bond angles (°).

3. RESULTS

Our training data (Figure 3) came from a 200 ns MD all-atom simulation of a molecule of the human β1-AR in a DPPC bilayer (348 molecules of lipid). Full details of the simulation protocols are included in the Supporting Information. The simulation provided us with a set of 501 snapshots and thus a total of 501 conformations of the protein and 174,348 conformations of DPPC. Our test data (Figure 3) was 10 snapshots taken from an independent all-atom simulation of a dimer of β1-AR in a bilayer of 546 DPPC lipids.

Both all-atom trajectories were converted to Martini CG equivalents using a version of "Martinize"[19] adapted to handle DPPC, then all four trajectories were split into separate test or training sets for the protein and for DPPC. Though in total over 174, 000 conformations of DPPC were potentially available in the training set and 5460 in the test set, we cut this down to just the first 1000 of these for the training set and 100 for the test set.

3.1. Backmapping Performance for DPPC.

We began by exploring the performance of our approach using the DPPC data. The atomistic model of DPPC consists of 130 atoms, while the Martini model has just 12 beads. The test set of 100 conformations is structurally diverse: At the all-atom level, the mean RMSD between conformations is 4.83 Å. This obviously presents a significant challenge for any backmapping approach; however, we do have in this case the availability of a training set of where the number of observations (1000) is much greater than the number of coordinates (390) we seek to predict.

Results are shown in Table 1. The basic "GLM" pipeline can regenerate atomistic models of DPPC from their CG equivalents with a mean RMSD of just 1.36 Å, and these atomistic models are entirely consistent with the CG representation (can be mapped back to CG structures with a mean RMSD from the test conformation of just 0.12 Å). However, inspection of the models shows they have poor geometry, particularly for the hydrogen atoms, as the bond length and angle error metrics show. We explored Lasso[20] and Ridge[21] regression enhancements but saw no significant improvement in performance (results not shown). Surprisingly, the PCA-GLM-PCA pipeline performs slightly worse, even though the GLM step now has the apparently easier task of predicting just 30 scores (3N-6) from 30 scores, rather than 390 coordinates from 36 coordinates; the PCA inverse transform of those 30 scores to 390 coordinates suffers from considerable inaccuracy, even though 30 scores are enough to capture 86% of the total variance for the training set. We did evaluate the option of training a GLM model to predict more than 30 atomistic PC scores from the 30 CG scores, but since the system is thus inevitably underdetermined and the atomistic PCs not linearly independent, we were not surprised when no better performance was obtained (results not shown). The "GLM-ENM" pipeline results in models with somewhat improved internal geometry but not by much; the crude models from the GLM step are frequently too bad start points for the truncated Newton algorithm in the ENM stage to converge. The "GLM-Sprout" pipeline performs better: The geometry of the models is as good as achieved with the ENM step, without the associated drift away from the underpinning Martini conformation. This makes sense, the "shaving" process reduces the atomistic model from 130 atoms to 48, simplifying the task for the GLM step which now has to predict just 144 coordinates from 36. Now if we add the ENM step back in, creating the "GLM-ENM-Sprout" pipeline, the reduced number and improved quality of the crude coordinates presented to the minimizer enables it to perform much better, in turn presenting even better "shaved" coordinates to the final "sprout" step. The resulting atomistic models show very good structural metrics with just a small cost to the RMSD metrics. As seen above with the "GLM-ENM" pipeline, the ENM process does result in some drift of the structure away from the CG reference (RMSD CG around 0.5 Å), but this still is, in absolute terms, a modest error. More detailed analysis of the relatively modest bond angle metric reveals that small numbers of poorly predicted values skew the result somewhat, for example, for this "GLM-ENM-Sprout" pipeline, 50% of bond angles are predicted to within 7° (data not shown). Reversing the order of the Sprout and ENM steps in the pipeline (so the ENM step is last) results in marginally worse performance (data not shown). As expected, based on its indifferent contribution noted before, adding a PCA step into the pipeline offers no advantages over the corresponding pipelines that omit it.

The GLM-ENM-Sprout pipeline is computationally efficient, despite the inclusion of the iterative refinement step, and the backmapping process takes about 23 ms per DPPC conformation.

Using the optimal "GLM-ENM-Sprout" pipeline, we then examined the importance of the size of the training set (Table 2). We see that the accuracy of the reconstructed coordinates decreases steadily as the training set gets smaller, though the geometric quality of the structures is less affected; the Sprout and ENM steps can be parametrized to good accuracy with very few training structures. Using more than 1000 observations in the training set provided no enhancement in performance.

3.2. Backmapping Performance for β1-AR.

Backmapping the protein structure represents some different challenges for this methodology. For a start, the problem is larger: A molecule of the protein is 4473 atoms or 628 CG

Table 3. Performance of ML-Based Pipelines Applied to Backmapping Martini Models of β1-AR[a]

molecule	options	$\langle RMSD_{fg} \rangle$[b]	$\langle RMSD_{cg} \rangle$[c]	$\langle RMSE_{bonds} \rangle$[d]	$\langle RMSE_{angles} \rangle$[e]
β1	none	2.58	1.07	1.23	23.6
β1	PCA	2.58	1.07	1.24	23.6
β1	ENM	1.89	1.20	0.44	23.0
β1	Sprout	2.01	1.07	0.43	14.9
β1	Sprout + ENM	1.72	1.09	0.07	7.32
β1	Sprout + PCA	2.07	1.07	0.43	14.9
β1	PCA + ENM	1.92	1.19	0.55	24.4
β1	all	1.72	1.09	0.07	7.27

[a] Predicting coordinates of 4473 atoms from those of 628 CG beads, training set of 501 samples, independent test set of 20 samples. [b] Mean RMSD (Å) between true and predicted atomistic structures in the test set. [c] Mean RMSD (Å) between true CG structures in the test set, and those generated by forward-mapping the predicted atomistic structures. [d] Mean value of the root-mean-square error in predicted bond lengths (Å). [e] Mean value of the root-mean-square error in predicted bond angles (°).

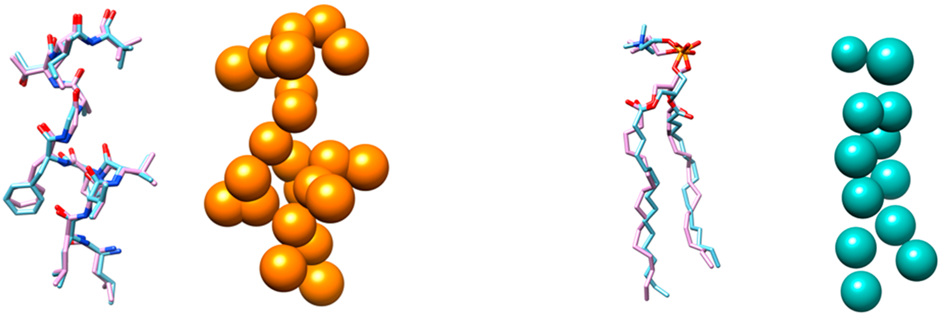


Figure 4. Example of the performance of the GLIMPS resolution transformation method. Left: Actual (blue) and predicted (pink) atomistic models for a section of the protein compared to the CG representation (orange). Right: Actual (blue) and predicted (pink) atomistic models for a representative DPPC lipid molecule compared to the CG representation (cyan).

particles. The test structures are less divergent, the mean RMSD between them is just 1.78 Å, but all are significantly different from the structures in the training set (mean RMSD between test and training set 3.4 Å). In addition, there are now just 501 samples in the training set. Despite these differences, the performance of the different pipelines matches very well what was observed for the DPPC test case (Table 3). Adding Sprout or ENM to the pipeline improves the performance over the basic GLM method, and adding both is even better. The GLM-ENM-Sprout pipeline rebuilds CG models of the protein back to full atomistic models (including hydrogens) with a mean RMSD error of just 1.72 Å and even smaller errors in bond lengths and angles than was obtained for DPPC. As expected for this much larger system, the reconstruction process is more computationally demanding at about 1.2 s per protein conformation.

3.3. Backmapping Performance for a Combined Protein–Lipid System.

Finally, having produced optimized and trained backmapping pipelines for both proteins and lipids independently, we evaluated the performance of our method when applied to backmapping entire snapshots of the β1-AR dimer + lipid system from CG to all-atom representations (Figure 4). We compared the performance of our approach (GLM-ENM-Sprout, henceforth referred to as "GLIMPS") with the CG2AT2 method of Vickery and Stansfeld. Each of our 10 test CG snapshots was backmapped using both methods. For our GLIMPS pipeline, we used the Python-based command line tools we have developed as part of this project, and which are freely available (see Supporting Information). The full CG2AT2 pipeline includes a final NVT molecular dynamics step, but for fairer comparison, we omitted this, stopping the process after the energy minimization step (option "-o none"). We also investigated what happened to models generated by GLIMPS when they were further subjected to a similar force field-based energy minimization step (same Gromacs code and force field as used by CG2AT2).

Results are shown in Table 4. In terms of overall accuracy of reconstruction, the two approaches show very comparable performance. Bond and angle metrics are somewhat better for CG2AT2, but this is to be expected as the method includes a force field-based energy minimization step. If the same force field-based refinement step is added to the end of the GLIMPS pipeline, the quality of the resulting structures matches that of CG2AT2.

Table 4. Comparison of the GLIMPS ML Pipeline and CG2AT2 Method Applied to Backmapping Martini Models of β1-AR Dimer/DPPC Lipid Systems[a]

component	method	$\langle RMSD_{fg} \rangle$[b]	$\langle RMSE_{bonds} \rangle$[c]	$\langle RMSE_{angles} \rangle$[d]
DPPC	GLIMPS	1.68	0.11	14.1
DPPC	CG2AT	1.92	0.01	4.9
DPPC	GLIMPS + EM	1.77	0.01	2.0
β1	GLIMPS	1.73	0.08	7.9
β1	CG2AT	1.57	0.01	2.2
β1	GLIMPS + EM	1.58	0.01	1.8

[a] An independent test set of 10 samples. The training data were the same as that used above. Metrics for GLIMPS models after an additional force-field-based energy minimization step are also shown. [b] Mean RMSD (Å) between true and predicted atomistic structures in the test set. [c] Mean value of the root-mean-square error in predicted bond lengths (Å). [d] Mean value of the root-mean-square error in predicted bond angles (°).

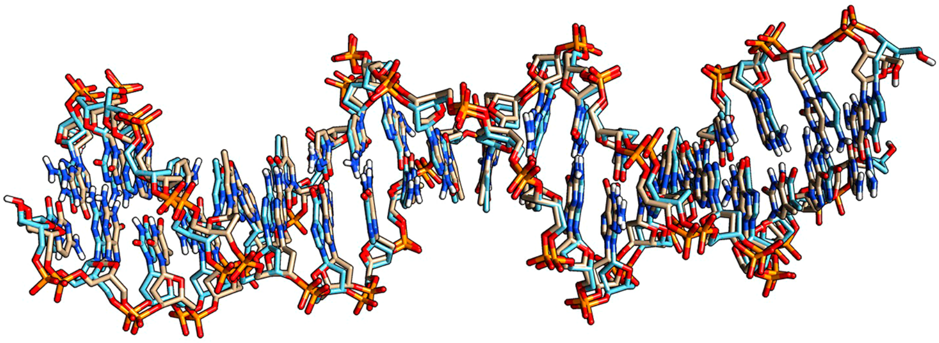


Figure 5. Example of the performance of GLIMPS in back mapping a helical parameter-based CG representation (133 parameters) of a 20mer DNA duplex to an atomistic model (1266 atoms). Brown: actual structure; blue: predicted structure; RMSD 0.9 Å.

3.4. Testing the Generality of GLIMPS: Backmapping DNA Structures from Noncoordinate-Type Coarse Grained Data.

The GLIMPS approach is very flexible; indeed there is no requirement that the CG model that is back mapped is even one based on the Cartesian coordinates of beads. To test this, we took 7061 snapshots from an atomistic MD simulation of a DNA 20mer duplex (unpublished data) and used cpptraj[22] to calculate the base pair step helical parameters (shift, slide, rise, tilt, roll, twist + Zp) for each step in each snapshot, that is, generating a CG model of the duplex (1266 atoms) consisting of 133 helical parameters. Saving a random 50 of the data samples as the test set, we then trained a "GLM-Sprout-ENM" pipeline with the remainder. The trained back-mapper could generate atomistic models for the test set from the helical parameter representations with a mean error of 0.69 Å. An example of one of the poorer-predicted models (RMSD 0.9 Å) is shown in Figure 5.

4. CONCLUSIONS

We have identified a ML-based approach to backmapping CG models of molecular structures to finer grained ones that requires only matched sets of coordinates at the two resolutions as a training data set. Despite having no explicit knowledge of molecular structure or force fields, or any rules that were used in the original forward-mapping process, the method can reconstruct atomistic models of protein−lipid systems from Martini CG representations with an accuracy close to that achieved by tools that do (here, CG2AT2). While tools such as CG2AT2 have the advantage that they generate not only sets of atomistic coordinates but also complementary topology and parameter files, and also that in effect they come "pre-trained" for common membrane simulation components, GLIMPS has the advantage that it can be applied to systems containing nonstandard components, such as ligands or cofactors, with very little additional effort. As long as a method is on hand that can convert the atomistic model of this component to the CG form, GLIMPS can learn the backmapping and apply it to independent data without any end user intervention. More generally, GLIMPS is easily applied to backmapping any CG model, not just a Martini one. We envisage GLIMPS as a particularly useful tool in multiscale simulation scenarios where there is a need to run a simulation at one level of representation for a time, then convert to another level and run for a time, and then convert back and run longer, etc. We have developed a Python library and set of command-line tools to implement the GLIMPS procedure; this is freely available (see Supporting Information).

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.1c00735.

Details of the molecular dynamics simulation methods; and details of the freely available GLIMPS Python software (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Charles A. Laughton − *School of Pharmacy, University of Nottingham, Nottingham NG7 2RD, United Kingdom;* orcid.org/0000-0003-4090-3960; Email: charles.laughton@nottingham.ac.uk

Authors

Keverne A. Louison − *School of Pharmacy, University of Nottingham, Nottingham NG7 2RD, United Kingdom*

Ian L. Dryden − *School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, United Kingdom;* Present Address: Department of Mathematics and Statistics, Florida International University, 11200 SW 8th Street, Miami, Florida 33199, USA

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.1c00735

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

K.L. was supported by the BBSRC Doctoral Training Partnership BB/M008770/1. Compute resources were in part supported by EPSRC grant EP/P011993/1.

■ REFERENCES

(1) Rzepiela, A. J.; Schäfer, L. V.; Goga, N.; Jelger Risselada, H.; De Vries, A. H.; Marrink, S. J. Reconstruction of Atomistic Details from Coarse-Grained Structures. *J. Comput. Chem.* **2010**, *31* (6), 1333−1343.

(2) Wassenaar, T. A.; Pluhackova, K.; Böckmann, R. A.; Marrink, S. J.; Tieleman, D. P. Going Backward: A Flexible Geometric Approach to Reverse Transformation from Coarse Grained to Atomistic Models. *J. Chem. Theory Comput.* **2014**, *10* (2), 676−690.

(3) Machado, M. R.; Pantano, S. SIRAH Tools: Mapping, Backmapping and Visualization of Coarse-Grained Models. *Bioinformatics* **2016**, *32* (10), 1568−1570.

(4) Lombardi, L. E.; Martí, M. A.; Capece, L. CG2AA: Backmapping Protein Coarse-Grained Structures. *Bioinformatics* **2016**, *32* (8), 1235−1237.

(5) Badaczewska-Dawid, A. E.; Kolinski, A.; Kmiecik, S. Computational Reconstruction of Atomistic Protein Structures from Coarse-Grained Models. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 162−176.

(6) Vickery, O. N.; Stansfeld, P. J. CG2AT2: An Enhanced Fragment-Based Approach for Serial Multi-Scale Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **2021**, *17*, 6472.

(7) Stieffenhofer, M.; Wand, M.; Bereau, T. Adversarial Reverse Mapping of Equilibrated Condensed-Phase Molecular Structures. *Mach. Learn. Sci. Technol.* **2020**, *1* (4), 045014.

(8) Li, W.; Burkhart, C.; Polińska, P.; Harmandaris, V.; Doxastakis, M. Backmapping Coarse-Grained Macromolecules: An Efficient and Versatile Machine Learning Approach. *J. Chem. Phys.* **2020**, *153* (4), 041101.

(9) Wang, W.; Gómez-Bombarelli, R. Coarse-Graining Auto-Encoders for Molecular Dynamics. *npj Comput. Mater.* **2019**, *5* (1), 1−9.

(10) An, Y.; Deshmukh, S. A. Machine Learning Approach for Accurate Backmapping of Coarse-Grained Models to All-Atom Models. *Chem. Commun.* **2020**, *56* (65), 9312−9315.

(11) Peng, J.; Yuan, C.; Ma, R.; Zhang, Z. Backmapping from Multiresolution Coarse-Grained Models to Atomic Structures of Large Biomolecules by Restrained Molecular Dynamics Simulations Using Bayesian Inference. *J. Chem. Theory Comput.* **2019**, *15*, 3344−3353.

(12) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; De Vries, A. H. The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *J. Phys. Chem. B* **2007**, *111* (27), 7812−7824.

(13) Dryden, I. L.; Mardia, K. V. *Statistical Shape Analysis, with Applications in R*, 2nd ed.; John Wiley & Sons, Ltd: Chichester, 2016.

(14) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (85), 2825−2830.

(15) Hagberg, A. A.; Schult, D. A.; Swart, P. J. Exploring Network Structure, Dynamics, and Function Using NetworkX. Proceedings from the *7th Python in Science Conference*, August 19−24, 2008; Pasadena, CA; Varoquaux, G., Vaught, T., Millman, J., Eds.; SciPy: Austin, TX, 2008; pp 11−15.

(16) Word, J. M.; Lovell, S. C.; Richardson, J. S.; Richardson, D. C. Asparagine and Glutamine: Using Hydrogen Atom Contacts in the Choice of Side-Chain Amide Orientation. *J. Mol. Biol.* **1999**, *285* (4), 1735−1747.

(17) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L. P.; Lane, T. J.; Pande, V. S. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophys. J.* **2015**, *109* (8), 1528−1532.

(18) Schröder, G. F.; Brunger, A. T.; Levitt, M. Combining Efficient Conformational Sampling with a Deformable Elastic Network Model Facilitates Structure Refinement at Low Resolution. *Structure* **2007**, *15* (12), 1630−1641.

(19) De Jong, D. H.; Singh, G.; Bennett, W. F. D.; Arnarez, C.; Wassenaar, T. A.; Schäfer, L. V.; Periole, X.; Tieleman, D. P.; Marrink, S. J. Improved Parameters for the Martini Coarse-Grained Protein Force Field. *J. Chem. Theory Comput.* **2013**, *9* (1), 687−697.

(20) Tibshirani, R. Regression Shrinkage and Selection via the Lasso: A Retrospective. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2011**, *73* (3), 273−282.

(21) Hoerl, A. E.; Kennard, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **2000**, *42* (1), 80.

(22) Roe, D. R.; Cheatham, T. E. PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. *J. Chem. Theory Comput.* **2013**, *9*, 3084.