| JOURNAL | Structure |
|---------|-----------|
| ARTICLE TITLE | Coupling molecular dynamics and deep learning to mine protein conformational space |
| AUTHOR LIST | Matteo T. Degiacomi |

This is a paper under consideration at Cell Press and has not been peer-reviewed
Coupling molecular dynamics and deep learning to mine protein conformational space

Matteo T. Degiacomi *

Department of Chemistry, Durham University, South Road, DH1 3LE, UK

* matteo.t.degiacomi@durham.ac.uk
Summary

Flexibility is often a key determinant of protein function. In order to elucidate the link between their molecular structure and role in an organism, computational techniques such as molecular dynamics can be leveraged to characterize their conformational space. Extensive sampling is however required to obtain reliable results, useful to rationalize experimental data or predict outcomes before experiments are carried out. We demonstrate that a generative neural network trained on protein structures produced by molecular simulation can be used to obtain new, plausible conformations complementing and extending pre-existing ones. To demonstrate this, we show that a trained neural network can be exploited in a protein-protein docking scenario to predict large conformational changes upon binding. Overall, this work shows that neural networks can be used as an exploratory tool for the study of molecular conformational space.

**Keywords:** molecular modelling, deep learning, protein docking
Introduction

Function at the molecular level emerges from the arrangement of individual atoms and their associated dynamics. Specific interactions of simple molecules produce phenomena of increasing complexity, culminating with the finely tuned biological mechanisms that ultimately make life possible. Proteins are flexible molecules, and their dynamics are intimately connected to their function (Chu et al., 2013). The function can be modulated by conformational rearrangements in response to local environmental changes as diverse as changes in pH, temperature or electrostatic potential, as well as binding to specific ligands such as ions, small molecules, lipids or other proteins. As such, proteins should be seen not as a single, static structure, but as a conformational ensemble featuring more or less accessible states.

In order to gain information about protein molecular functions, a broad range of techniques have been developed to interrogate their structure. X-ray crystallography provides precious structural evidence at atomic resolution, but its drawback is that it locks proteins in a single well-defined conformation within a crystal lattice. Atomic information can also be obtained via NMR. This technique also informs about dynamics, and can sometimes identify multiple states, given that the molecule of interest is not too large. Other techniques report near-atomic or lower resolution data, describing for instance the shape of the protein (e.g. EM, IM/MS or SAXS), or measuring specific inter-atomic distances (e.g. chemical cross-linking). Even in light of this broad palette of techniques, studying the structure of a protein featuring multiple states is usually challenging, and even when a single conformation is present, its thermal fluctuations may render data interpretation arduous.

Computational methods such as molecular dynamics (MD) simulations aim at characterizing molecular conformational space by iteratively generating new structures on the basis of an initial, known atomic arrangement and a physical model of inter-atomic interactions. The structural
ensemble produced by such sampling can be used to rationalize results of performed experiments, or to predict their outcome before they are carried out. Since simulations provide a discretized view of a continuous space, they will always be subject to the risk of missing key details owing to under-sampling (Nemec and Hoffmann, 2017). This risk is usually low and acceptable when studying small or rigid proteins, and can be mitigated by running long simulations, or exploiting on enhanced sampling techniques. It becomes however increasingly severe the larger the molecular system under study and slower its dynamics. A common, and extreme, scenario where MD is (in most cases) unsuitable, is the prediction of the specific docking of multiple proteins into a complex. To tackle protein-protein docking, other sampling methods are typically exploited instead of relying on MD alone. These often utilize optimization engines exploring the roto-translational space of the binding partners, looking for the arrangements minimizing a specific scoring function (Kastritis and Bonvin, 2010; Zhang et al., 2016). It is acknowledged that accounting for molecular flexibility beyond the movement of amino acid side chains in docking processes is often key to producing suitable results (Zacharias, 2010) (Lensink et al., 2017). To account for conformational changes at backbone level, flexibility is typically accounted for by refining rigidly docked poses with methods such as energy minimization, MD or Monte Carlo (Schindler et al., 2015) (Dominguez et al., 2003), or docking pre-generated alternate conformations (Degiacomi and Dal Peraro, 2013; Marze et al., 2017). Overall, for any model scenario, let it be MD simulations or docking, it is critical for conformational spaces to be extensively sampled.

In this work we examine the use of deep learning, and more specifically generative neural networks, to enhance the sampling of molecular conformational space. While the usage of generative neural networks in image processing is widespread, their application to 3D point clouds is only a recent addition to the machine learning literature (Achlioptas et al., 2017). Proteins represent a particularly interesting application case in this area, since they do not feature a difficulty typical of raw 3D point clouds: as the position of their constituent atoms is constrained by covalent interactions, proteins
conformations can be seen as ordered sets of points. Here we use the conformations generated by one or more MD simulations as examples to train an autoencoder (Hinton and Salakhutdinov, 2006) (Rumelhart et al., 1986). We demonstrate that autoencoders can generate new, accurate protein conformations complementing pre-existing data produced by molecular dynamics simulations. We then show that a trained autoencoder coupled with a protein docking algorithm can be utilized to discover conformations close to the bound state, given an ensemble of structures sampling the unbound state.

**Results**

In order to generate low-dimensional representations of proteins’ conformational space, we exploit an autoencoder (see Figure 1). This is a type of neural network that, via unsupervised learning, attempts to first compress and then decompress a multidimensional input, so that the difference between input and output is minimized. The network is first trained by providing it with a collection of alternative molecular conformations. Its performance is then tested with a new set of conformations not previously used for training. The first part of the network, called the “encoder”, passes the input signal (flattened Cartesian coordinates) thorough a series of hidden layers containing a decreasing amount of neurons. This ultimately produces a low-dimensional representation of the input molecular structure, called the “latent vector”. The values of the latent vector then become the input for a second series of hidden layers, this time with increasing amount of neurons, called the “decoder”. The decoder expands the latent vector into an output that should be as similar as possible to the initial molecular structures passed through the encoder. In sum, the encoder allows casting a protein conformational space (possible atom positions in 3D space) into a non-linear, low-dimensional representation (latent space), whereas the decoder can convert the coordinates of such low dimensional space into specific protein conformations. We note that our network architecture does not feature the ubiquitously used convolution operator, as it cannot be used to process 3D points.
Learning and assessing protein conformational space

While any coordinate within the latent space is associated to an atomic arrangement, not every coordinate will correspond to a plausible molecular structure. On one hand, while in general selecting values close to regions of the latent space sampled during the training of the network will usually yield physically plausible model, we observed no correlation between the physical plausibility of a model and its distance in the latent space from points used as training examples. This is because distances in latent space do not usually linearly correlate with distances in the 3-dimensional Cartesian space. Generally, it is hard to determine what the dimensions in latent space represent in terms of motion in the three dimensional space. Given the difficulty in predicting whether a coordinate in the latent space will be associated with a physically plausible structure, we defined two methods to determine whether a model generated by the autoencoder should be considered as valid or not.

To automatically assess whether structures generated from the latent space should be considered plausible, we tested their ability to fool another learning algorithm, trained to determine whether an atomic arrangement provided as input is protein-like or not. To do so, we adopted a Random Forests (RF) classifier (Breiman, 2001) trained on two classes of data: structures extracted from an MD simulation (“correct”) and structure from the same simulation, with a small amount of noise added to atomic coordinates (see SI and Figure S1). Furthermore, we also defined two heuristic criteria, seen as necessary conditions, to quickly assess whether models feature atoms excessively far from all the others (“stretching”, more than 1.8 Å away from each other atom), or excessively close to any other (“compression”, less than 0.1 Å from another atom).

We first selected a small pool of proteins of different size and shape, to assess whether the autoencoder would be able to reconstruct their conformational space, and the RF classifier to distinguish between good and bad ones. For this test, we selected Malate dehydrogenase (MDH, a
33 kDa protein forming homodimers), αB crystallin (a rigid 18 kDa homodimer), Phospholipase A2 (13 kDa, featuring two long random coil regions, 15 and 25 amino acids-long respectively) and Encephalitis virus envelope glycoprotein (an elongated 43 kDa protein, approximately a cylinder three times longer than wide). For all proteins, we ran >200 ns long MD simulations, and trained the autoencoder to compress their conformational space (backbone and Cβ atoms) down to a 2-dimensional space. Since, unlike the general case of 3D point clouds, proteins are represented by ordered sets of points, root mean square deviation (RMSD) is a usable metric to assess the similarity between autoencoder-generated models and their MD-generated counterpart. In all cases, test structures could be reconstructed with an RMSD < 1.5 Å from the original, a deviation smaller than that observed within the MD simulations (see Table 1). We then assessed whether generated test cases featured secondary structure elements consistent with those in input structures. To do so, we assigned the position of all missing atoms, and determined the secondary structure of each test structure and encoded-decoded counterpart. Comparing the two indicated that, in each case, >93% of amino acids in generated proteins had correct secondary structure elements (see Table 1). In order to verify whether the reconstructions could be considered as new plausible protein conformations, we tested them using both our scoring function and the RF classifier. In all cases >98% of reconstructed structures were considered valid by the classifier, and all but one case had >97% structures featuring no stretching or compression (see Table 1). This shows that the autoencoder is capable of producing protein conformations close to the conformational space provided as input.

In our tests, we have observed cases of trained autoencoders featuring undesirable mode collapse, whereby the network learns to associate the average protein structure to each input conformation (see Figure S2). In particular, the odds of mode collapse were higher for training sets of large and rigid (i.e. with structures in the training set having a low average RMSD) proteins. Thus, we observe that autoencoders perform best for the characterization of flexible proteins.
Representing a multi-state system

As a harder test case, we selected MurD, a 47 kDa ATP-driven ligase responsible for the biosynthesis of a bacterial peptidoglycan precursor (UDP-N-acetylmuramoyl-L-alanyl-D-glutamate). MurD has been crystallized in both open, unbound, state (PDB: 1E0D (Bertrand et al., 2000)) and closed state (PDB: 3UAG (Bertrand et al., 1999)), bound to GDP and UDP-N-acetylmuramoyl-L-alanyl-D-alanine. Comparing these structures shows that ligand binding causes one of the three globular domains of MurD (residues 300 to 439) to rearrange (see Figure 2A). We performed three MD simulations: one of the open state, one of the closed state, and one of the closed state with its ligands removed (hereon called closed-apo). In 200 ns the first two simulations maintained their specific domains arrangement, while the third transitioned from the closed to the open state, sampling conformations not observed in neither the closed nor the open simulations (see Figure 2B).

We trained the autoencoder with structures from only open and closed simulations, asking it to compress their 5124 degrees of freedom down to a 2-dimensional latent vector (see Figure 1). Subsequent testing with a set of 100 randomly selected structures not used for training revealed that the RMSD (backbone and β carbon) between input and reconstructed structures was 0.9 ± 0.2 Å, with a best case of 0.7 Å and a worst case of 1.7 Å. We then tested whether the autoencoder trained with only open and closed conformation would be able to correctly reconstruct 1000 structures from the closed-apo conformation. Comparison of input and encoded-decoded structures revealed an average RMSD of 2.2 ± 0.5 Å, with the best case of 0.8 Å and the worst case of 3.4 Å. Importantly, we were also able to obtain structures at the transition between closed and open state, i.e. different from anything submitted as example for training process. These were reconstructed with an RMSD of 1.9 ± 0.3 Å (see Figure 2CD).
Docking proteins with flexibility

As a final test case, we selected the HIV-1 hexameric capsomer, for which the crystal structure of both hexamer (PDB: 3MGE (Pornillos et al., 2010)) and monomer (PDB: 1E6J (Monaco-Malbet et al., 2000)) have been solved. The RMSD between the monomeric structure and subunits within the assembled state is very large, 10.5 Å. Interestingly, when the unbound state of HIV-1 capsomer is studied by MD, conformations featuring higher similarity to the bound states are found. We have previously shown that a docking algorithm implemented using our POWER optimization environment can automatically select these conformations as most suitable candidates to generate hexameric models (Degiacomi and Dal Peraro, 2013). Although promising, the limitation of such an approach, is that the docking algorithm will provide results only as good as the best structure within the provided conformational ensemble. Furthermore the optimization engines currently available within POWER are not profiled to handle discretized search space cases, meaning that we can expect their performance to be suboptimal.

We trained an autoencoder with a 2-dimensional latent vector on a µs-long simulation of the unbound state. 100 randomly selected test structures were subsequently reconstructed with an RMSD of only 1.5 ± 0.5 Å (Figure 3A) from the original. We then characterized the whole latent space by generating all structures on a 200x200 sampling grid. Within the produced models we found that several were both acceptable (no penalty from both scoring function and random forests classifier, see Figure 3B), and with an RMSD from the bound state smaller than the best model available within the simulation (2.7 Å, against 3.2 Å from the MD simulation, see Figure 3C). This observation shows that the latent space trained with unbound structures can describe protein conformations more suitable to form a complex than any of the structures it was trained with.

We then assessed whether POWER would be able to identify such structures in the process of docking six HIV-1 monomers into a complex. For this, we developed a new POWER module,
leveraging on the trained autoencoder to generate candidate structures for docking. Thus, the search
space to generate the HIV-1 hexamer according to a circular symmetry was six-dimensional: two
coordinates within the 2-dimensional latent space, three rotation angles to determine the orientation
of each subunit within the complex, and one radius of the circular assembly (see Methods). The
docking protocol produced 151 models, the best of which had a backbone RMSD of 3.7 Å from
complex (see Figure 3D). Interestingly, the subunits composing the complex had an RMSD of 2.9
Å from the known bound state, lower than the best structure present in the MD simulation (3.2 Å).
This result shows that, in cases where conformational selection plays a major role, coupling
autoencoders and general optimization algorithms may help in predicting the structure of a protein
assembly with subunits undergoing substantial conformational changes.

**Discussion**

Herein, for the first time we have used autoencoders trained on structures from MD simulations as a
tool for enriching the sampling of molecular conformational space. We have shown that the low-
dimensional space encoded by an autoencoder can be used to produce new molecular structures that
are, from a geometric perspective, plausible. Overall, this shows that the dimensionality reduction
capabilities of autoencoders are superior to those of other popular approaches, such as Principal
Components Analysis (PCA). This is connected to the non-linear nature of the autoencoder’s latent
space, giving them an enhanced capability of correctly representing the movement of covalently
bonded atoms as they explore the molecular conformational space. Since it is non-trivial to predict
whether a coordinate in the latent space will be associated with a plausible protein structure, we
propose quick methods to assess whether a newly generated protein three-dimensional structure can
be considered as geometrically valid. Testing generated models with a Random Forests classifier
demonstrates that the autoencoder is capable of producing protein structures considered correct by a
learning algorithm trained to detect minimal discrepancies from ordinary atomic arrangements.
Remarkably, we were able to show that the latent space described by a trained autoencoder can be mined to extract information usable in contexts where extensive sampling is critical. As an application example, in this work we have shown how it can be exploited to generate structures usable in protein-protein docking scenario. In this context, we propose a model scoring function that is effective when used by an optimization engine geared to minimize continuous fitness functions. The core idea behind all currently presented protein docking applications within POWER is that, in order to rapidly create a first coarse subunits arrangement, an accurate energy function is not necessary. Suitable complexes generated in such manner can be refined with more expensive computational techniques in a second step. Exploiting structures generated by neural networks for docking perfectly fits within this philosophy: as the position of backbone and beta carbon atoms can be faithfully predicted, missing side chains atoms can be derived and refined *a posteriori* on the basis of standard force field parameters, as demonstrated in our models secondary structure analysis. Further applications of this approach lay in areas where experimental data report on ensemble quantities, such as chemical cross-linking or FRET.

In our tests, we have observed that training the autoencoder on rigid protein structures may to lead to mode collapse, whereas networks capable of producing a range of different structures can be obtained in the case of flexible proteins. Although this shows that autoencoders work appropriately in cases where they can actually be most useful, mode collapse remains undesirable, and recent research in the machine learning community has been dedicated to tackling this phenomenon (Arjovsky et al., 2017). Collecting further test cases will enable determining whether there exists a single neural network architecture and training protocol yielding the best performance, or if *vice versa* custom solutions (i.e. testing multiple possible network structures and training protocols) are required, on a per case scenario, to obtain optimal performance. Given a sufficiently large dataset, it may be possible to train a general neural network for molecular modelling, that could be quickly trained via reinforcement learning to tackle a specific conformational space sampling problem.
Acknowledgements

We wish to thank Drs. Noura Al-Moubayed and Chris Willcocks for useful discussions, and Dr. Valentina Erastova and Lucas Rudden for critically reviewing the manuscript. The work was supported by the Engineering and Physical Sciences Research Council (grant EP/P016499/1).

Declaration of Interests

The author declares no competing financial interest.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’16), pp. 265–284.

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. (2017). Learning Representations and Generative Models for 3D Point Clouds. arXiv Prepr. 1707.02392 1–22.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. Proc. 34th Int. Conf. Mach. Learn. 1–32.

Bertrand, J.A., Auger, G., Martin, L., Fanchon, E., Blanot, D., Le Beller, D., Van Heijenoort, J., and Dideberg, O. (1999). Determination of the MurD mechanism through crystallographic analysis of enzyme complexes. J. Mol. Biol. 289, 579–590.

Bertrand, J.A., Fanchon, E., Martin, L., Chantalat, L., Auger, G., Blanot, D., Van Heijenoort, J., and Dideberg, O. (2000). “Open” structures of MurD: Domain movements and structural similarities with folylpolyglutamate synthetase. J. Mol. Biol. 301, 1257–1266.

Breiman, L. (2001). Random Forests. Mach. Learn. 45, 5–32.

Chollet, F. (2015). Keras.
Chu, X., Gan, L., Wang, E., and Wang, J. (2013). Quantifying the topography of the intrinsic energy landscape of flexible biomolecular recognition. Proc. Natl. Acad. Sci. 110, E2342–E2351.

Degiacomi, M.T., and Dal Peraro, M. (2013). Macromolecular symmetric assembly prediction using swarm intelligence dynamic modeling. Structure 21.

Dominguez, C., Boelens, R., and Bonvin, A. (2003). HADDOCK: a protein–protein docking approach based on biochemical or biophysical Information. J. Am. Chem. Soc. 1731–1737.

Eswar, N., Webb, B., Marti-Renom, M.A., Madhusudhan, M.S., Eramian, D., Shen, M.Y., Pieper, U., and Sali, A. (2007). Comparative protein structure modeling using Modeller. Curr Protoc Protein Sci 2, 15–32.

Hinton, G.E., and Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. Science (80). 313, 504–507.

Kabsch, W., and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22, 2577–2637.

Kastritis, P.L., and Bonvin, A.M.J.J. (2010). Are scoring functions in protein-protein docking ready to predict interactomes? Clues from a novel binding affinity benchmark. J. Proteome Res. 9, 2216–2225.

Lensink, M.F., Velankar, S., and Wodak, S.J. (2017). Modeling protein–protein and protein–peptide complexes: CAPRI 6th edition. Proteins Struct. Funct. Bioinforma. 85, 359–377.

Maier, J.A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K.E., and Simmerling, C. (2015). ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. J. Chem. Theory Comput. 11, 3696–3713.

Marze, N.A., Burman, S.S.R., Sheffler, W., and Gray, J.J. (2017). Efficient Flexible Backbone Protein-Protein Docking for Challenging Targets. bioRxiv 223511.

Monaco-Malbet, S., Berthet-Colominas, C., Novelli, A., Battaï, N., Piga, N., Cheynet, V., Mallet,
F., and Cusack, S. (2000). Mutual conformational adaptations in antigen and antibody upon complex formation between an Fab and HIV-1 capsid protein p24. Structure 8, 1069–1077.

Nemec, M., and Hoffmann, D. (2017). Quantitative Assessment of Molecular Dynamics Sampling for Flexible Systems. J. Chem. Theory Comput. 13, 400–414.

Phillips, J.C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R.D., Kalé, L., and Schulten, K. (2005). Scalable molecular dynamics with NAMD. J Comput Chem 26, 1781–1802.

Pornillos, O., Ganser-Pornillos, B.K., Banumathi, S., Hua, Y., and Yeager, M. (2010). Disulfide Bond Stabilization of the Hexameric Capsomer of Human Immunodeficiency Virus. J. Mol. Biol. 401, 985–995.

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

Schindler, C.E.M., de Vries, S.J., and Zacharias, M. (2015). iATTRACT: Simultaneous global and local interface optimization for protein-protein docking refinement. Proteins Struct. Funct. Bioinforma. 83, 248–258.

Wang, J., Wang, W., Kollman, P.A., and Case, D.A. (2006). Automatic atom type and bond type perception in molecular mechanical calculations. J. Mol. Graph. Model. 25, 247–260.

Zacharias, M. (2010). Accounting for conformational changes during protein-protein docking. Curr. Opin. Struct. Biol. 20, 180–186.

Zhang, Q., Feng, T., Xu, L., Sun, H., Pan, P., Li, Y., Li, D., and Hou, T. (2016). Recent Advances in Protein-Protein Docking. Curr. Drug Targets 17, 1586–1594.
STAR Methods

CONTACT FOR REAGENT AND RESOURCE SHARING

Further information and requests for data will be fulfilled by the author, Matteo T. Degiacomi (matteo.t.degiacomi@durham.ac.uk)

METHOD DETAILS

Molecular dynamics simulations

Simulations were run with using the Amber ff14SB (Maier et al., 2015) on the NAMD (Phillips et al., 2005) molecular dynamics engine. For MurD closed state, ligands were parameterized using ANTECHAMBER (Wang et al., 2006). All atom types were found within existing ffSB14 parameters. Furthermore, the crystal structure of both MurD closed and open state featured two gaps (residues 221 to 224 and 242 to 244 for the closed, and 221, 222 and 183 to 188 in the open). We added all those missing regions using MODELLER (Eswar et al., 2007). The MurD closed-apo state was produced by removing the ligands from the closed state.

All proteins were solvated with TIP3P water and the resulting boxes neutralized with addition of Na⁺ and Cl⁻ ions. The resulting systems energy minimized with 2000 conjugate gradient steps. We then performed 0.5 ns simulation with 2 fs time step (restraining every covalent bond with SHAKE) in the NPT ensemble, with all protein’s alpha carbons constrained by a harmonic potential. In all simulations Langevin dynamics were used to impose a temperature of 300 K, using a damping of 1 ps. A constant pressure of 1 Atm was imposed via a Langevin piston having a period of 200 fs, and a decay of 50 fs. The systems were then further equilibrated in the NVT ensemble for 1 ns, after which production runs of variable length (between 0.1 and 1 μs) in the NPT ensemble were performed. In all simulation steps, particle mesh Ewald was used to treat long range electrostatic interactions, a cutoff distance of 12 Å was set on van der Waals interactions.
Autoencoder design

The autoencoder was developed in Python 3.5 using the Keras package (Chollet, 2015), with Tensorflow backend (Abadi et al., 2016). In order to identify a suitable autoencoder structure and training protocol, we performed a systematic test using training and test sets generated from our MD simulations. From each simulation, one frame every 100 ps was extracted, selecting only Ca, C, N, and Cβ atoms to represent the proteins’ backbone and sidechains directions. In the case of MurD, we combined 1913 structures from the open state and 2607 for the closed one. The coordinates of each dataset were first preprocessed: each simulation was aligned (by minimizing the RMSD from the first structure), and shifted so that atoms would only have positive coordinates. Finally coordinates were normalizing between zero and one. The first and last layer of the autoencoder were N-dimensional, i.e. one dimension per protein degree of freedom. We tested two different networks, one featuring 3 encoding and 3 decoding layers (hereon “3-layer autoencoder”), and one featuring 5 encoding and 5 decoding layers (hereon “3-layer autoencoder”), all with 20% dropout. In both cases, we used a RELU activation function for each layer but the last one that was set as sigmoid, and used a binary cross-entropy as loss function.

Random Forests classifier

In a classification problem, determining whether an arrangement of atoms corresponds to a plausible protein conformation or not can be interpreted as a classification problem. A protein with N atoms can be represented as a single point in a 3xN dimensional space. The classifier divides this space into two regions, corresponding to plausible and not plausible protein conformations, respectively. To tackle this problem, we adopted a Random Forests (RF) classifier. In short, RF is composed of an ensemble of decision trees (“weak learners”), trained on the basis of examples of known class. After training, RF classifies an input structure on the basis of the most voted within all its decision trees. Preliminary tests indicated that using more than 50 estimators did not improve the
classifier’s performance. We therefore set 50 as number of learners, while maximal tree depth was left unbound. For every molecular dynamics simulation described in Main Text, 90% of structures were selected as a training set, and the remaining 10% as test set. For each simulation, as for the autoencoder, only Cα, C, N, and Cβ were considered. To generate examples of unsuitable protein conformations, we altered the coordinates of half or protein conformations by adding random noise to their positions (i.e. adding small displacements to each of them).

**Protein docking**

To dock six HIV-1 subunits into a complex with POWER, we adopted a docking protocol previously described (Degiacomi and Dal Peraro, 2013). In summary, in order to build a circular hexamer, the search space was defined as three rotation angles defining the orientation of each monomer, the radius of the circle, and two dimensions defining coordinates in the latent space described by the autoencoder. To assemble a specific model, POWER would first require the autoencoder to generate the structure associated to a specific coordinate in the latent space, and then assemble the model according to desired rotations and radius values. Docking accounted only for atoms generated by the autoencoder, i.e. Cα, C, N and Cβ. The fitness function to be minimized featured a sum of terms including a 9-6 Lennard Jones potential to avoid steric clashes of backbone atoms, an error function assessing the matching with experimental data, as well as two terms, $S_{\text{stretch}}$ and $S_{\text{compress}}$, assessing the quality of the subunit generated by the autoencoder (‘stretching’ and ‘compression’, as defined in main text). These two latter components were designed to be continuous, in order to drive the optimization algorithm towards regions of the conformational space yielding plausible models. Let $\mathbf{d}$ the N dimensional vector reporting, for each atom, the distance to the closest neighbor.

\[
S_{\text{stretch}}(\mathbf{d}) = \begin{cases} 0 & \text{if } \max(d_{s}) < 2 \\ \max(\mathbf{d}) - 2 & \text{otherwise} \end{cases} \quad (1)
\]

\[
S_{\text{compress}}(\mathbf{d}) = \begin{cases} 0 & \text{if } \min(\mathbf{d}) > 0.1 \\ 5 - \min(\mathbf{d}) \times 50 & \text{otherwise} \end{cases} \quad (2)
\]
Data used to guide the docking process was based on two loose distance restraints, i.e. a disulfide bridge (between C42 and C54) and a salt bridge (between Glu212 and Lys140) between neighbouring dimers. These had to be both plausible, i.e. have their respective amino acids at <5 Å. All models generated by POWER with fitness function smaller than zero (no clash, matching experimental data, plausible subunits) were accepted and clustered via a hierarchical clustering, using a 2 Å cutoff. The model shown in Figure 3D had the positions of all missing atoms reconstructed by Amber’s tleap tool.

QUANTIFICATION AND STATISTICAL ANALYSIS

Random Forests benchmarking
To assess how accurately the RF classifier could discriminate between perturbed and unperturbed structures, we performed multiple training runs with variable levels of random noise (see Figure S1). We found that in all cases the classifier can discriminate with >99% accuracy a true structure from an altered one when noise level as low as 0.5 Å are applied (i.e., every degree of freedom altered with a random number uniformly distributed between 0 and 0.5 Å).

Autoencoder benchmarking
For protein modelling test cases, we assessed the performance of both the 3-layer and 5-layer autoencoder using different batch sizes (50, 100, 200, 300 examples per batch), sizes of the latent vector (2, 3, 4 neurons) and optimizer (Adam, SGD). Autoencoders performance was assessed in terms of RMSD of reconstructed test structures against original ones. For each combination of these parameter and network, we trained the resulting autoencoder three times for 500 epochs, for a total of 144 independent training runs per case (see Figures S4 and S4). At every run, 100 randomly selected structures were separated and used for testing. We found that the Adam optimizer outperforms SGD, that 2 encoding neurons are in most cases sufficient to yield good performance,
that 5 layers-deep networks perform marginally better, and that batch size does not have a significant effect on the overall performance. In main text we report the performance of the best 5-layers encoder and decoder trained with Adam, with a 2-dimensional latent vector, i.e. the highest data compression level we tested. For each run, we also calculated the pairwise RMSD of all structures used as test set, and that of the resulting reconstructed protein structure, to determine whether the autoencoder can reproduce the diversity in examples provided as input (see Figure S2). Finally, we compared the secondary structure of encoded-decoded structures with their input counterparts. To do so, we assigned all missing atoms using Amber’s *tleap* tool, and calculated the resulting model’s secondary structure using DSSP (Kabsch and Sander, 1983). DSSP assigns secondary structure of each amino acid to one of eight different categories (seven kinds of structure, plus random coil).

**DATA AND SOFTWARE AVAILABILITY**

All data and software presented in this work is freely available for download in Durham University repository DRO-DATA [placeholder link].

**KEY RESOURCES TABLE**

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---------------------|--------|------------|
| Software and Algorithms | Powell software and manual | http://lbm.epfl.ch | PowellER |
| protein                  | PDB | d.o.f. a | training structures (#) b | MD RMSD (Å) c | classifier accuracy d | classifier accept. e | scoring function accept. f | test structures RMSD (Å) g | test structures sec. struct. h |
|-------------------------|-----|----------|--------------------------|---------------|-----------------------|----------------------|--------------------------|----------------------------|--------------------------------|
| Malate dehydrogenase*   | 1MLD        | 7344     | 2811                      | 2.88          | 100%                  | 100%                 | 100%                     | 1.00±0.13                   | 95.9±1.5                       |
| αB crystallin           | 2WJ7        | 1857     | 2829                      | 4.79          | 99.6%                 | 99%                  | 100%                     | 1.12±0.24                   | 93.3±1.9                       |
| Phospholipase A2        | 1POA        | 1286     | 4690                      | 3.84          | 98.5%                 | 98%                  | 94%                      | 0.89±0.15                   | 96.4±1.7                       |
| Envelope glycoprotein   | 1SVB        | 4632     | 2191                      | 6.21          | 97.3%                 | 98%                  | 99%                      | 1.67±0.35                   | 94.0±1.2                       |
| MurD, closed            | 3UAG        | 5124     | 2507                      | 3.80          | 99.6%                 | 100%                 | 100%                     | 1.14±0.22                   | 94.9±1.5                       |
| MurD, open              | 1E0D        | 5124     | 1813                      | 5.77          | 100%                  | 100%                 | 100%                     | 1.43±0.50                   | 93.9±1.3                       |
| MurD, closed + open     | 3UAG, 1E0D  | 5124 | 4320                      | 10.22         | 100%                  | 100%                 | 100%                     | 0.90±0.16                   | 94.4±2.0                       |
| HIV-1                   | 1E6J        | 2481     | 6142                      | 17.84         | 99.5%                 | 99%                  | 92%                      | 1.47±0.50                   | 96.0±2.1                       |

Table 1: performance of learning algorithms on all tested protein structures. All cases reported here feature proteins encoded in a 2-dimensional latent space and, besides the case indicated with *, without mode collapse (see Figure S2 and S3). a degrees of freedom, i.e. 3 times the amount of atoms. b Quantity of structures used to train the autoencoder. c Maximal RMSD within all structures in the simulation, as an indicator of the amount of structural variability in the protein conformational space. d Random Forests classifier accuracy after training with molecular dynamics structures (positive examples) and structures having 0.5 Å uniform noise applied on each degree of freedom (negative examples). e Percentage of encoded-decoded test structures accepted by the Random Forests classifier. f Percentage of encoded-decoded test structures having a scoring function penalty equal to zero. g RMSD of test set structures against their encoded-decoded counterparts. h Percentage of secondary structure elements in test structures (as defined by DSSP (Kabsch and Sander, 1983)) consistent with those in input structures.
Figure 1: autoencoder structure. The autoencoder is a neural network composed of two parts: an encoder (in light gray, with and input layer having a size equal to the degrees of
freedom of the protein provided as input, followed by four hidden layers, with a decreasing amount of neurons noted in white) and a decoder (four dark gray hidden layers, followed by a last layer having same size as the input). The first reduces protein atomic coordinates to a position in a low-dimensional space (blue points in the graph representing the so-called “latent space”) while the second converts such coordinates into a protein structure. The autoencoder is trained to encode-decode structures so that the difference between input (dark blue proteins) and output (light blue proteins) is minimized. After training, the decoder alone can be used to generate new protein structures (in red) from any coordinate within the latent space. The scatter plot represents structures from MurD open, closed, and closed-apo simulations.
Figure 2: reconstruction of MurD closed-to-open transition. (A) For all MurD simulations, we calculated the angle between the centre of mass of domain 4 (resid 299-437), and two centres of mass of the distal ends of domain 3 (residues 120-230 and 230-299, respectively, shown with arrows) (B) The closed bound state (blue), and the open unbound one (red) preserve their specific inter-domain orientations, whereas removing ligands from the closed states leads the protein to convert into the open state (black). (C) Histogram of RMSD of encoded-decoded closed-apo structures with respect of the original structure. (D) We trained the autoencoder with frames from open and closed state, and asked it to encode-decode structures from the closed-apo simulation. For each reconstruction, we calculated the
RMSD with respect of the input. We report the RMSD with respect of the domains angle. The closed state is more accurately reconstructed than the open state. Frames in white regions have an opening angle different from anything seen in both open (red zone) and closed (blue zone) simulation. They are nevertheless reconstructed with an RMSD around 2 Å.
Figure 3: protein-protein docking using structures dynamically generated by the autoencoder. (A) Quality of 100 test set structures generated by the autoencoder trained with a simulation of HIV-1 capsomer’s (pdb: 1E6J) monomeric state. The histogram shows the distribution of RMSD between original and reconstructed structure. Above, three examples of reconstructions are shown (in blue the original structures, in red the predicted ones). Only atoms generated by the autoencoder, i.e. Cα, C, N and Cβ, are shown. Examples show the best (0.5 Å RMSD), average (1.5 Å), and worst (3.0 Å) models produced. (B) 2-dimensional latent space coloured as a function of the model quality each position is
associated to, calculated as a sum of our compression and stretching heuristics. Large regions of the latent space correspond to plausible structures. The gray circle represents the encoded position of HIV-1 capsomer’s monomeric crystal structure. The blue circle represents the position of a monomer extracted from hexameric HIV-1 capsomer structure (not used for training, pdb: 3MGE). Its associated reconstruction features a small amount of deformation. Small red circles represent the positions of structures selected by the docking algorithm as good candidates of HIV-1 capsomer’s bound state. The red circle shows the position of the monomer used to build the model of HIV-1 capsomer with smallest RMSD from the known crystal structure. (C) 2-dimensional latent space coloured as a function of the RMSD against HIV-1 capsomer’s bound state each position is associated to. The RMSD between unbound and bound state is large, while a region close to the position of the bound state is associated to models with a small RMSD (<3 Å) from it. (D) Superimposition of HIV-1 capsomer’s hexameric crystal structure and the best model generated by our docking algorithm leveraging on structures generated on-demand by the trained autoencoder.
Coupling molecular dynamics and deep learning to sample
protein conformational space

Matteo T. Degiacomi *

Chemistry Department, Durham University, South Road, DH1 3LE, UK

*matteo.t.degiacomi@durham.ac.uk
Supplementary Figures

Figure S1, related to Table 1: performance of Random Forests (RF) classifier in discriminating real protein structures from altered ones. For each dataset tested in this work, we assess the accuracy of the RF classifier in discriminating between protein structures extracted from molecular dynamics simulations, and structures having had their atomic positions altered with uniformly distributed noise. Noise levels between having maxima ranging from 0.05 to 1 Å were tested (i.e., every degree of freedom is altered by a random amount uniformly distributed between 0 and a desired maximum). In all tested cases, the classifier can discriminate with >99% accuracy real structures with structures altered with a 0.5 Å noise. With lower noise levels the performance decays, converging to 50% (random choice) when noise levels under 0.2 Å are applied. In this application we selected the smallest noise level yielding high classification accuracy, i.e. 0.5 Å.
Figure S2, related to Table 1: comparison of diversity in structures of all test sets, against their autoencoder-generated counterparts. For each autoencoder trained in this work, we calculated the pairwise RMSD of all structures used as test set, and that of the resulting reconstructed protein structures to characterize their diversity. The scatter plots report, for each protein, the mean and standard deviation of each test set (red points) and autoencoder-generated structures (black points). Regions associated to partial (defined as 50% of test set mean value) or full (25% of test set mean value) mode collapse are shown with
light and dark grey, respectively. Percentage of training cases featuring full mode collapse are, per protein dataset: 1MLD: 100%, 2WJ7: 33.3%, 1POA: 55.5%, 1SVB: 91.6%, 3UAG: 94.4% 1E0D: 33.3%, 3UAG+1E0D: 31.9% 1E6J: 56.9%. Mode collapse takes place more often for large and rigid proteins. We observe that, in the absence of mode collapse, standard deviations of generated and original structure datasets are comparable. Mean RMSD is on the other hand always equal or lower in generated sets, indicating that the autoencoder features a “damping” of most extreme protein movements.
Figure S3, related to Table 1: performance of test phases on each attempted autoencoder training on the first four cases reported in Table 1. On the x-axis, specific values for three explored parameters are reported i.e., from top to bottom: dimension of latent space (2, 3 or
4), batch size (50, 100, 200 or 300) and depth of encoder and decoder (3 or 5 layers). For each parameter combination, three independent trainings were performed each using a different set of structures randomly selected from the input dataset. On the y-axis, the RMSD between 100 input and encoded-decoded test structures is reported. Data reported in black with a large circle (indicating the mean RMSD), featured no mode collapse, as defined in Figure S2. Data indicated in black, with a small black circle featured partial mode collapse, data indicated in grey features mode collapse.
Figure S4, related to Table 1: performance of test phases on each attempted autoencoder training on the second four cases reported in Table 1. Figure legend as per Figure S3.