bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Automated interpretation of Cryo-EM density maps with convolutional neural networks

Philipp Mostosi[1,2], Hermann Schindelin[1], Philip Kollmannsberger[2], Andrea Thorn*[1]

[1] Institute of Structural Biology, Rudolf Virchow Center for Experimental Biomedicine, University of Würzburg, Josef-Schneider-Str. 2, 97080 Würzburg, Germany

[2] Center for Computational and Theoretical Biology, University of Würzburg, Campus Hubland Nord 32, 97074 Würzburg, Germany

Abstract

Haruspex is a fully convolutional neural network that automatically annotates both protein secondary structure and nucleotides in experimentally derived Cryo-EM maps. The network was trained on a carefully curated dataset of EMDB (Electron Microscopy Data Bank) entries. Haruspex enables users to identify folds and can be used to guide model building as well as validate structures.

In recent years, three-dimensional density maps reconstructed from single particle images obtained by electron cryo-microscopy (Cryo-EM) have reached unprecedented resolution. However, modelling an atomic structure to these maps remains difficult as researchers mostly rely on algorithms developed for crystallographic electron density maps, which are different in both their nature and error distribution. The first step in modelling a reconstruction map - assigning a fold to map regions - can be a major challenge. Parallel to the advances in Cryo-EM in the last decade, deep neural networks achieved remarkable image segmentation capabilities [1], making them the most powerful machine learning approach available. Convolutional neural networks (CNN) combine traditional image analysis with machine learning by cascading layers of trainable convolution filters. CNNs are thus exceptionally well suited for volume annotation and have been successfully applied to biological problems such as breast cancer mitosis recognition [2] and, in conjunction with encoder-decoder architectures, to volumetric data segmentation [3;4] .

In this work, we demonstrate that deep neural networks are capable of assigning macromolecular type, i.e. protein or nucleotides, and protein secondary structure elements to experimentally derived Cryo-EM maps. This can be utilized to drastically facilitate model building, to validate existing models and support the placement of known domain folds.

In low-resolution Cryo-EM maps, α-helices can often be discerned as long cylindrical elements. This has been exploited by the program *helixhunter* [5], which searches for prototype helices in reconstruction maps using a cross-correlation strategy. β-Strands are more difficult to identify as they are more variable in shape and therefore require morphological analysis [6]. A combination of these approaches led to the development of *SSEHunter* [7], which uses a density skeleton to detect

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

secondary structures. Deep learning poses an alternative approach and here we demonstrate that at resolutions of 4 Å or better, experimental data allow training a well-performing network for a multitude of specimens. Fully convolutional networks [8;3] allow swift segmentation map generation for objects of variable size and we employed a state of the art U-Net-style architecture [3]. The network processed 40^3 voxel segments with a voxel size of 1.0-1.2 $Å^3$ (covering a secondary structure element and its immediate surroundings) to annotate 20^3 voxel cubes (corresponding to the center of the input volume) with four channels containing the probabilities that the voxel is part of an α–helical or β-strand protein secondary structure element, RNA/DNA nucleotide, or other.

For network training, we pre-selected EMDB (Electron Microscopy Data Bank [9]) reconstruction maps with resolutions of 4 Å or better. From 576 entries (as of 15/2/2018), we picked 293 EMDB/PDB (Protein Data Bank [10]) pairs by three criteria: (1) Good fit between map and model; (2) presence of at least one annotated α–helix or β-sheet; (3) preference of higher resolution maps in case the same authors deposited several instances of the same macromolecular complex. Maps with severe misfits, misalignments, or models without corresponding reconstruction density (and vice versa) were omitted.

To generate ground truth data for network training, a python script was implemented to automatically annotate the reconstruction map according to the deposited structural model as α–helical, β-strand, nucleotide or other. The script combined the original annotations from PDBML files with secondary structure identified by DSSP [11] and STRIDE-like extension [12;13] (see Methods). If a secondary structure was identified, all voxels within 3 Å of backbone atoms were annotated accordingly. All voxels with density ≥1.0 σ (standard deviation of the map density distribution) but not within 5 Å of model atoms with density ≥1.0 σ were masked and did not contribute to the training. The voxel size of the reconstruction map was re-scaled to 1.1 Å, if outside [1.0; 1.2] Å. The maps were sliced into a total of 2183 segments á 70^3 $Å^3$ voxels, of which 110 segments (5%) were set aside for evaluation during network training. Each segment had to contain at least 100 atoms ≥1.0 σ, a backbone mean density of ≥3.0 σ, and at least 5% of the total segment volume annotated. The training data were augmented through on-GPU 90° rotations (24 possibilities), and by selecting a 40^3 voxel sub-segment at a random position. The network was trained for 40,000 steps with 100 segments employed per step.

After training, the network was tested on an independent set of 167 EMDB maps (selected by the same criteria as training data and deposited after February 2018). Virus and ribosome structures were omitted from the test set: viruses' symmetry definition can disagree between map and model and symmetry-averaged maps exhibit particular features; ribosomes are very common and may hence bias the network. For evaluation, we investigated residues with mean backbone densities ≥1.0 σ and compared the predicted secondary structure on a per-residue basis with the one derived from the deposited PDB model. Using this criterion, the network achieved similar performance on training, evaluation and test data. Over all test maps, there were 74.1% true positives r_p (correctly predicted residues), 18.9% false positives f_p and 4.4% false negatives f_n, resulting in a median recall rate $r_p(r_p + f_n)^{-1}$ of 94.2% and precision $r_p(r_p + f_p)^{-1}$ of 79.2%. As a typical example the human ribonuclease P holoenzyme (EMDB entry 9627) illustrates the power of our approach (Fig. 1), which

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

is not only able to accurately predict the RNA vs. protein distribution in this complex but also correctly assigns secondary structure elements in the protein areas with few exceptions. While the high number of false positives was worrying at first, inspection of the test cases revealed that false positives were often elements closely resembling helices, sheets or nucleotides (see Fig. 2). In particular, semi-helical structures, β-hairpin turns and residues belonging to polyproline type II (P_{II}) helices [14] were misclassified as α-helical and loosely parallel structures were frequently misclassified as β-strands.

Haruspex was trained for resolutions as low as 4 Å, and with the current rate of resolution increase in published maps, by 2021, the average resolution may well be 3.5 Å. Irrespective of this, we will extend our approach to lower resolution data in the future; low resolution experimental maps with a well-matching model, however, are difficult to obtain. This obstacle has previously been faced by Si et al. [15] (*SSELearner*) and Li et al. who developed machine learning approaches for protein secondary structure prediction in Cryo-EM maps (but not nucleotides) [16], and consequently resorted partly to simulated maps generated with pdb2mrc [17]. Simulated maps may lack the error structure and processing artefacts found in experimentally derived reconstruction densities. Si et al. tested their support vector machine on 10 simulated maps of relatively small structures (<40 kDa) and, as available data were still very limited in 2012, only 13 experimental maps paired with individually selected training maps. Haslam et al. [18] used a 3D U-Net, which was trained on 25 simulated and 42 experimental maps between 3-9 Å resolution to predict helices and sheets obtaining an *F1* score $2(\text{recall}^{-1} + \text{precision}^{-1})^{-1}$ between 0.79 and 0.88. However, the network was only tested on six simulated maps and one experimentally derived map. We used a total of 293 experimentally derived maps in a semi-automated process to provide a more realistic training environment. The amount of newly released high resolution structures in conjunction with our processing infrastructure allowed us to test our network performance on a representative set of 167 unique depositions. In addition, we identify nucleotides, which to our knowledge has not been attempted before. Ribosomes, spliceosomes and polymerases all contain substantial amounts of DNA/RNA nucleotides and are among the most common specimens studied by single-particle Cryo-EM. In addition, β-turns, poly-proline and membrane detergent regions might be desirable additions in future versions of the network.

We show that a neural network can be used to automatically distinguish between nucleotides and protein and to assign the two main protein secondary structure elements in experimentally derived Cryo-EM maps. This technique will render the process of protein structure determination faster and easier. Haruspex was trained on a carefully curated ground truth dataset based on experimental data from EMDB. The pre-trained network can be straightforwardly applied to predict structures in newly reconstructed Cryo-EM density maps, and will be refined and adapted as new data become available. Besides guidance for model building and domain placements, the network may also be useful for model validation due to its high median recall and precision rates of 94.2% and 79.2%, respectively. The trained network and documentation are available from gitlab.com/phimos/haruspex.

Acknowledgements

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

This work was supported by the DFG (project TH2135/2-1), the High Performance Computing Cloud of Würzburg University (DFG project 327497565) and the Rudolf Virchow Center for Experimental Biomedicine. We would like to thank Bettina Böttcher, Niko Grigorieff, Tom Burnley and Jola Mirecka for fruitful discussions; and Bernhard Fröhlich for great computational support.

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Methods

Training Data

We queried the Electron Microscopy Data Bank (EMDB) for all single particle Cryo-EM maps with a resolution ≤4 Å, for which corresponding protein models were available in the Protein Data Bank in Europe (PDBe), yielding 576 map and model pairs as of February 2018. We filtered these EMDB/PDB pairs by the following three criteria: (1) Good fit between map and model; (2) presence of at least one annotated α-helix or β-sheet; and (3) preference of the highest resolution maps in case the same authors deposited several instances of the same macromolecular complex. Maps with severe misfits, misalignments, or models without corresponding reconstruction densities, and vice versa, were discarded. After applying these criteria, we retained 293 map/model pairs for generating the training data.

To extract secondary structure information from the PDB data, we developed a custom parser for the PDBML [19] format based on xmltodict [20]. To obtain additional secondary structure information, we implemented a variant of the DSSP algorithm [11] without strand direction, and a torsion angle based secondary structure detection inspired by STRIDE [13]: annotated or DSSP-detected secondary structures were extended by neighbouring amino acids if they matched the same Ramachandran profile.

Annotation of reconstruction maps

For every entry pair, the augmented model was then superimposed on the map and all voxels within 3 Å of a Cα or C,N,O-backbone atom, or, in the case of nucleotides, within 3 Å of any atom, were assigned the respective class (helix, sheet or nucleotide) if their value was higher than ½ of the average backbone density of the helix, sheet or nucleotide in question. Secondary structures with a backbone standard deviation of <2 σ and atoms without secondary structure assignment were labelled as "empty" to exclude incorrectly modelled, misfitted, or flexible structures. For some training data pairs (e.g. virus capsids), only small or partial protein models were deposited for large Cryo-EM maps, resulting in well-defined high-density regions without model coverage. These regions will not get annotated and will result in false positives if the network tries to predict the actual structure. To mitigate this, all voxels with density ≥1.0 σ but not within 5 Å of a model atom with density ≥1.0 σ were masked as unmodeled density and hence did not contribute to training.

Since our network generated a single class label as output, the reconstruction density of the secondary structures must be converted to a strict assignment to one of the three classes in order to be used as training examples. For each secondary structure, the reconstruction map density was multiplied by the backbone standard deviation and rescaled to an output density between zero and one (corresponding to 0.5 and 1.0 times the average backbone density of the local secondary structure element) for each label type. The highest channel value determined the voxel class. If multiple channels shared the same value, sheets took precedence over nucleotides, which took precedence over helices. Voxels where all channel values were below 0.01 were assigned the "empty" class. Finally, reconstruction maps were rescaled to a voxel size of 1.1 Å, if they were outside of [1.0; 1.2] Å.

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Generation of training segments

To generate the 70^3 voxel sized segments needed for training, candidate volumes were sampled from the entire map, and segments with a mean backbone density <3.0 σ, less than 5% annotated volume, or less than 100 atoms with standard deviation ≥1.0 σ were discarded. This resulted in altogether 2183 training segments, of which 110 segments (5%) were held back for evaluation during training. To generate additional segments for training, we applied rotations in steps of 90° around all three axes, resulting in 24 rotated versions of each segment that could all be used as separate training volumes since the convolutional network is not rotation-invariant. Segments were further augmented during training by using a randomly translated 40^3 sub-cube for each step.

Network Architecture

We use a 3D U-Net architecture with a single input channel (reconstruction density) and an input layer size of 40^3 voxels, shown in supplementary Fig. 1. The encoding branch consisted of two 3x3x3 convolutional layers with 32 and 64 feature channels, respectively, followed by 2x2x2 max-pooling layers. Another convolutional layer with 128 feature channels followed by 2x2x2 max-pooling layer finally resulted in an 8^3 cube with 128 feature channels at the deepest layer of the network. This cube was passed through another convolutional layer with the same data padding in order to preserve its dimensions. A fully connected layer was considered, but not chosen due to its high memory and performance cost. The decoding branch of the U-Net was made of two blocks, each consisting of a deconvolution followed by two 3x3x3 convolutions (128 feature channels in the first, 64 and 32 channels in the second block to restore symmetry) with concatenated sections of the corresponding layer in the encoding part. The output part consists of a final 1x1x1 convolution followed by a soft-max output layer. The output layer reproduced the central 20^3 voxel cube of the input layer in four annotation channels representing co-dependent probabilities for the four classes (helix, sheet, nucleotide, empty) summing up to one. The highest channel value determined the predicted class. Implementation was realized using Tensor Flow [21].

Network Training

The network was trained for 40,000 steps on training batches of 100 random segment pairs per step, using ADAM stochastic optimization [22] with a learning rate of 0.001, β_1 = 0.9, β_2 = 0.999 and ε = 0.1. Error assignment for backpropagation was performed using cross-entropy loss, where the target class was represented in one-hot encoded binary format (1 for the target class, 0 for the other three classes). To account for class imbalance, voxels were weighted according to overall class occurrence in the training data. Furthermore non-true negatives were weighted 16-fold stronger than true negatives due to an overabundance of the latter.

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

193 **References:**

194 [1] Ciresan, D. C., Meier, U., Masci, J. & Schmidhuber, J. (2011). *The 2011 International Joint*
195 *Conference on Neural Networks*. 1918–1921.

196 [2] Cireşan, D. C., Giusti, A., Gambardella, L. M. & Schmidhuber, J. (2013). *Medical Image Computing*
197 *and Computer-Assisted Intervention – MICCAI 2013*, Vol. pp. 411–418. Springer, Berlin, Heidelberg.

198 [3] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. (2016). *International*
199 *Conference on Medical Image Computing and Computer-Assisted Intervention*, Vol. pp. 424–432.
200 Springer.

201 [4] Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel,
202 Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T., Prinz, M.,
203 Palme, K., Simons, M., Diester, I., Brox, T. & Ronneberger, O. (2019). *Nature Methods*. **16**,.

204 [5] Jiang, W., Baker, M. L., Ludtke, S. J. & Chiu, W. (2001). *J Mol Biol*. **308**, 1033–1044.

205 [6] Kong, Y., Zhang, X., Baker, T. S. & Ma, J. (2004). *Journal of Molecular Biology*. **339**, 117.

206 [7] Baker, M. L., Ju, T. & Chiu, W. (2007). *Structure*. **15**, 7.

207 [8] Shelhamer, E., Long, J. & Darrell, T. (2017). *IEEE Transactions on Pattern Analysis and Machine*
208 *Intelligence*. **39**, 640–651.

209 [9] Tagari, M., Newman, R., Chagoyen, M., Carazo, J.-M. & Henrick, K. (2002). *Trends in Biochemical*
210 *Sciences*. **27**, 589.

211 [10] Berman, H., Henrick, K. & Nakamura, H. (2003). *NSMB*. **10**, 980.

212 [11] Kabsch, W. & Sander, C. (1983). *Biopolymers: Original Research on Biomolecules*. **22**, 2577–
213 2637.

214 [12] Frishman, D. & Argos, P. (1995). *Proteins: Structure, Function, and Bioinformatics*. **23**, 566–579.

215 [13] Tronrud, D. E., Berkholz, D. S. & Karplus, P. A. (2010). *Acta Crystallographica Section D Biological*
216 *Crystallography*. **66**, 834–842.

217 [14] Hollingsworth, S. A. & Karplus, P. A. (2010). *BioMolecular Concepts*. **1**,.

218 [15] Si, D., Ji, S., Nasr, K. A. & He, J. (2012). *Biopolymers*. **97**, 698–708.

219 [16] Li, R., Si, D., Zeng, T., Ji, S. & He, J. (2016). 2016 IEEE International Conference on Bioinformatics
220 and Biomedicine (BIBM), Vol. pp. 41–46.

221 [17] Tang, G., Peng, L., Baldwin, P. R., Mann, D. S., Jiang, W., Rees, I. & Ludtke, S. J. (2007). *Journal of*
222 *Structural Biology*. **157**, 38–46.

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

[18] Haslam, D., Zeng, T., Li, R. & He, J. (2018). Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, Vol. pp. 628–632. New York, NY, USA: ACM.

[19] Westbrook, J., Ito, N., Nakamura, H., Henrick, K. & Berman, H. M. (2004). *Bioinformatics*. **21**, 988–992.

[20] Blech, M. xmltodict.

[21] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. & Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

[22] Kingma, D. P. & Ba, J. (2014). *ArXiv Preprint ArXiv:1412.6980*.

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

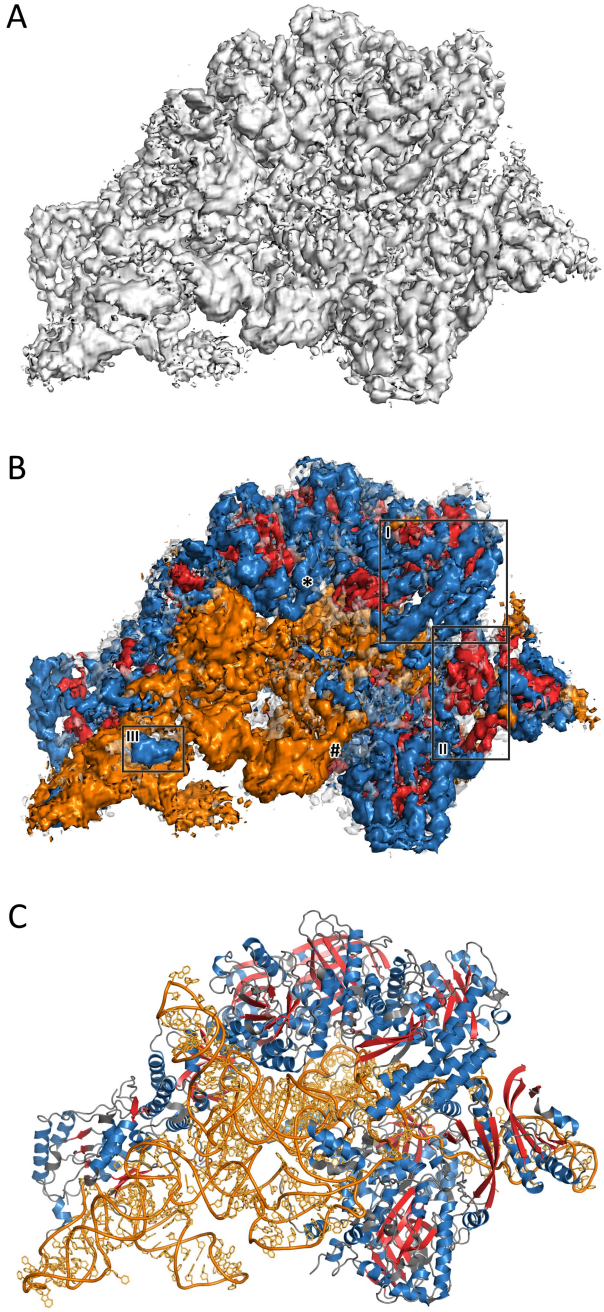

Figure 1. Haruspex Annotation. A. Reconstruction map for the human Ribonuclease P holoenzyme (EMDB entry 9627). Manual assignment of secondary structure features can be difficult, in particular if the composition of a macromolecular complex is unknown. The shown surface corresponds to σ = 0.04 with no carving. **B.** Secondary structure as identified by our network in the map, projected onto the surface. Orange corresponds to nucleotides; blue to helices; red to sheets and transparent grey were not assigned any secondary structure. This was a fairly typical test case with 70.5% true positives, 18.8% false positives and 10.7% false negatives. Recall was 86.8% and precision 79.0%. Region (I) depicts a well-predicted α-helical structure, (II) a β-sheet and (III) RNA misinterpreted as

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

an α-helix. **C.** Model (PDB 6AHU) for comparison. The regions depicted in Fig. 2C and 2D are marked # and *, respectively.

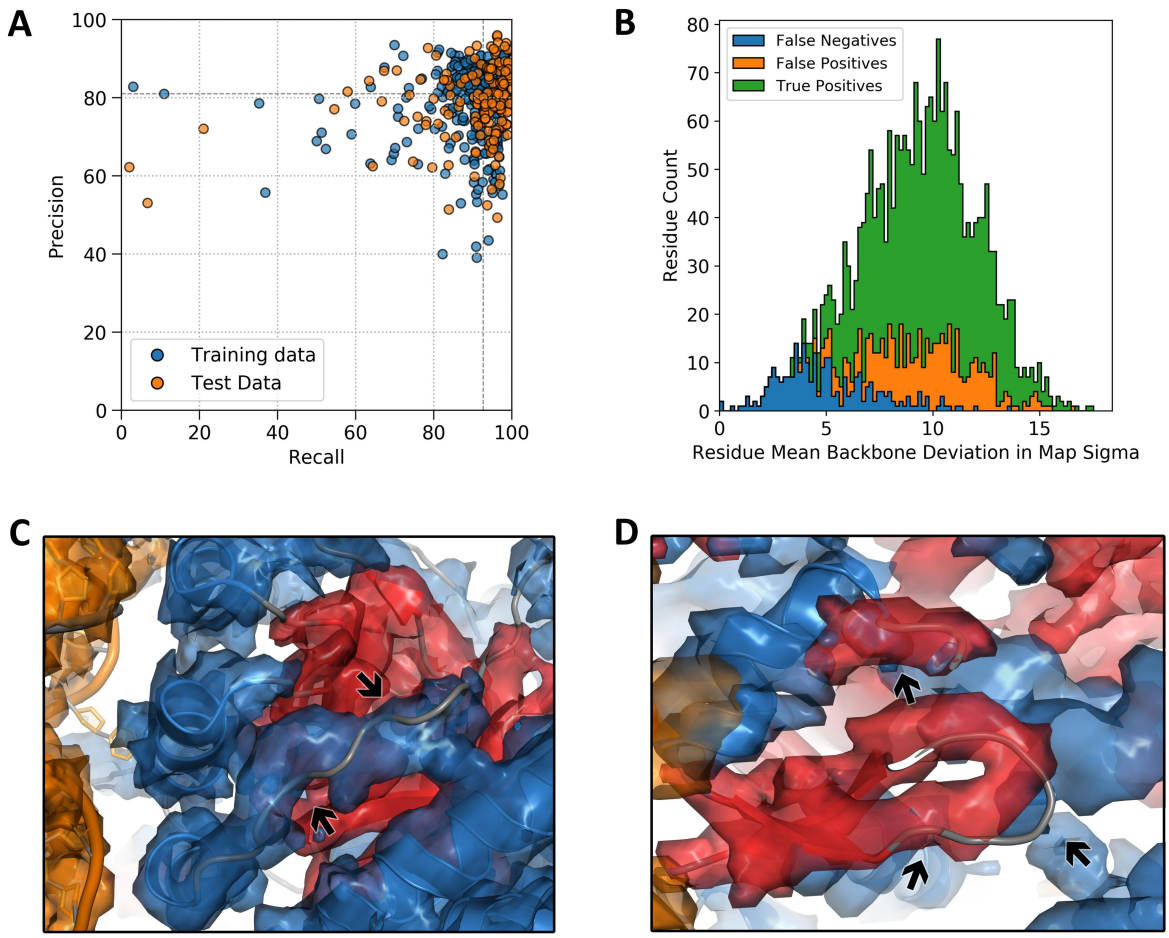

Figure 2. Network performance. A. Network precision vs. recall rates, with one marker per EMDB entry (test set markers are orange, training set markers blue). Both perform similarly well. **B.** Frequency vs. map σ level for EMDB 9627 on a per-residue basis: True positives (green), false positives (orange) and false negatives (blue). This plot is typical: false negatives often occur in low density map regions. **C.** α-Helical false positives (PDB 6AHU, J131 – 139): The model partly occupies the conformational space of a polyproline type II helix (P_{II}), which is often misinterpreted as α-helical and may have been modelled incorrectly (given that the model does not completely fit the density). **D.** False positives in a β-sheet (6AHU, B215-B221). The deposited model does not maintain the hydrogen bonding that defines β-sheets; to the network, however, the fold still 'looks' like a β-sheet and a third segment (top) is also assumed to be part of it.

bioRxiv preprint doi: https://doi.org/10.1101/644476; this version posted May 21, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

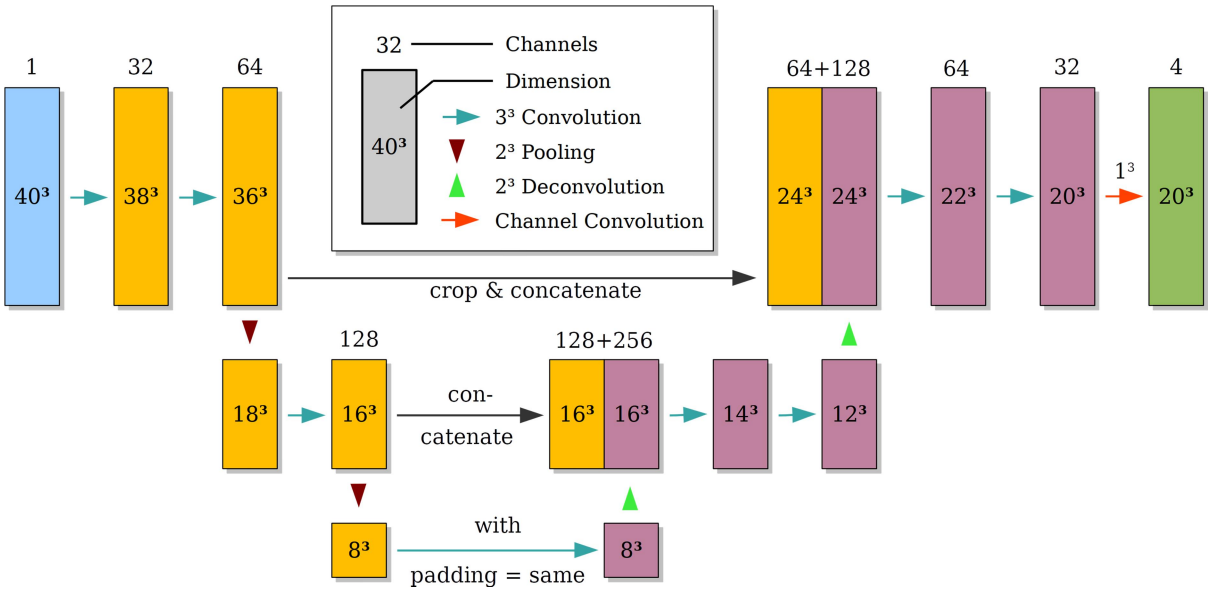

Figure 3 (Methods). Haruspex neural network architecture. The network consists of multiple interconnected layers, shown as rectangular boxes. We employed a state-of-the-art U-Net-like encoder-decoder architecture [4], a subclass of so-called fully-connected networks where spatial information and object details are encoded, reduced by pooling layers and then recovered again with up-sampling or transpose convolutions. The term U-Net arises from the U-like shape of the data flow. The layers are connected by convolution and pooling operations (arrows). Layer height represents the level of abstraction: lower layer data, generated by pooling operations, contain more abstract representations of the map. Input data (blue) is fed into the downconvolutional arm (yellow) in order to extract valuable information, which is then combined with previously discarded information through concatenations in the upconvolutional arm (purple) to compute annotated output data (green) for a subsection (20^3) of the input volume (40^3). Our network consists of two encoder blocks, containing altogether three convolutional layers (3x3x3) and two pooling layers (2x2x2). This is followed by two decoder blocks, one with upconvolution followed by two 3x3x3 convolutions and 128 feature channels, and one with upconvolution followed by two 3x3x3 convolutions with 64 and 32 feature channels, with concatenated sections of the corresponding layer in the encoding part. The output part consists of a final 1x1x1 convolution followed by a soft-max output layer. This results in 13 layers in total (12 + 1 convolution at bottom). The network is trained end-to-end by comparing the predicted class of each voxel to the annotated EMDB model using cross-entropy loss, propagating the error back through the network, and adapting the network weights to iteratively minimize the error.