CHAPTER 1

DISTILL: A MACHINE LEARNING APPROACH TO AB INITIO PROTEIN STRUCTURE PREDICTION

Gianluca Pollastri[a], Davide Baú and Alessandro Vullo
School of Computer Science and Informatics
UCD Dublin
Belfield, Dublin 4
Ireland
E-mail: {gianluca.pollastri|davide.bau|alessandro.vullo}@ucd.ie

We present Distill, a simple and effective scalable architecture designed for modelling protein C_α traces based on predicted structural features. Distill targets those chains for which no significant sequential or structural resemblance to any entry of the Protein Data Bank (PDB) can be detected. Distill is composed of: (1) a set of state-of-the-art predictors of protein structural features based on statistical learning techniques and trained on large, non-redundant subsets of the PDB; (2) a simple and fast 3D reconstruction algorithm guided by a pseudo-energy defined according to these predicted features.

At CASP6, a preliminary implementation of the system was ranked in the top 20 predictors in the Novel Fold hard target category. Here we test an improved version on a non-redundant set of 258 protein structures showing no homology to the sets employed to train the machine learning modules. Results show that the proposed method can generate topologically correct predictions, especially for relatively short (up to 100-150 residues) proteins. Moreover, we show how our approach makes genomic scale structural modelling tractable by solving hundreds of thousands of protein coordinates in the order of days.

1. Introduction

Of the nearly two million protein sequences currently known, only about 10% are human-annotated, while for fewer than 2% has the three-dimensional (3D) structure been experimentally determined. Attempts to

[a]To whom all correspondence should be addressed

predict protein structure from primary sequence have been carried out for decades by an increasingly large number of research groups.

Experiments of blind prediction such as the CASP series[1,2,3,4] demonstrate that the goal is far from being achieved, especially for those proteins for which no resemblance exists, or can be found, to any structure in the PDB[5] - the field known as *ab initio* prediction. In fact, as reported in the last CASP competition results[4] for the New Fold (NF) category, even the best predicted models have only fragments of the structure correctly modelled and poor average quality. Reliable identification of the correct native fold is still a long-term goal. Nevertheless, improvements observed over the last few years suggest that ab initio generated low-resolution models may prove to be useful for other tasks of interest. For instance, efficient ab initio genomic scale predictions can exploited to quickly identify similarity in structure and functions of evolutionary distant proteins[6,7].

Here we describe Distill, a fully automated computational system for ab initio prediction of protein C_α traces. Distill's modular architecture is composed of: (1) a set of state-of-the-art predictors of protein features (secondary structure, relative solvent accessibility, contact density, residue contact maps, contact maps between secondary structure elements) based on machine learning techniques and trained on large, non-redundant subsets of the PDB; (2) a simple and fast 3D reconstruction algorithm guided by a pseudo- energy defined according to these predicted features.

A preliminary implementation of Distill showed encouraging results at CASP6, with model 1 in the top 20 predictors out of 181 for GDT_TS on Novel Fold hard targets, and for Z-score for all Novel Fold and Near Novel Fold targets[6]. Here we test a largely revised and improved version of Distill on a non-redundant set of 258 protein structures showing no homology to the sets employed to train the machine learning modules. Results show that Distill can generate topologically correct predictions for a significant fraction of short proteins (150 residues or fewer).

This paper is organised as follows: in section 2 we describe the various structural features predicted; in section 3 we describe in detail the statistical learning methods adopted in all the feature predictors; in section 4 we discuss overall architecture of the predictive pipeline and the implementation and performances of the individual predictors; in section 5 we introduce the 3D reconstruction algorithm; finally in section 6 we describe the results of benchmarking Distill on a non-redundant set of 258 protein structures.

2. Structural Features

We call protein one-dimensional structural features (1D) those aspects of a protein structure that can be represented as a sequence. For instance, it is known that a large fraction of proteins is composed by a few well defined kinds of local regularities maintained by hydrogen bonds: helices and strands are the most common ones. These regularities, collectively known as protein secondary structure, can be represented as a string out of an alphabet of 3 (helix, strand, the rest) or more symbols, and of the same length of the primary sequence. Predicting 1D features is a very appealing problem, partly because it can be formalised as the translation of a string into another string of the same length, for which a vast machinery of tools for sequence processing is available, partly because 1D features are considered a valuable aid to the prediction of the full 3D structure. Several public web servers for the prediction of 1D features are available today, almost all based on machine learning techniques. The most popular of these servers [8,9,10,11] process hundreds of queries daily. Less work has been carried out on protein two-dimensional structural features (2D), i.e. those aspects of the structure that can be represented as two-dimensional matrices. Among these features are contact maps, strand pairings, cysteine-cysteine bonding patterns. There is intrinsic appeal in these features since they are simpler than the full 3D structure, but retain very substantial structural information. For example it has been shown [12] that correct residue contact maps generally lead to correct 3D structures.

In the remainder of this section we will describe the structural features predicted by our systems.

2.1. *One-dimensional structural features*

2.1.1. *Secondary structure*

Protein secondary structure is the complex of local regularities in a protein fold that are maintained by hydrogen bonds. Protein secondary structure prediction is an important stage for the prediction of protein structure and function. Accurate secondary structure information has been shown to improve the sensitivity of threading methods (e.g. [13]) and is at the core of most ab initio methods (e.g. see [14]) for the prediction of protein structure. Virtually all modern methods for protein secondary structure prediction are based on machine learning techniques[8,10], and exploit evolutionary information in the form of profiles extracted from alignments of multiple homologous sequences. The progress of these methods over the last 10 years has

been slow, but steady, and is due to numerous factors: the ever-increasing size of training sets; more sensitive methods for the detection of homologues, such as PSI-BLAST[15]; the use of ensembles of multiple predictors trained independently, sometimes tens of them[16]; more sophisticated machine learning techniques (e.g. [10]).

Distill contains the state-of-the-art secondary structure predictor Porter[11], described in section 4.

2.1.2. *Solvent Accessibility*

Solvent accessibility represents the degree to which amino acids in a protein structure interact with solvent molecules. The accessible surface of each residue is normalised between a minimum and a maximum value for each type of amino acid, and then reassigned to a number of classes (e.g. buried vs exposed), or considered as such. A number of methods have been developed for solvent accessibility prediction, the most successful of which based on statistical learning algorithms [17,18,19,20,21]. Within DISTILL we have developed a novel state-of-the-art predictor of solvent accessibility in 4 classes (buried, partly buried, partly exposed, exposed), described in section 4.

2.1.3. *Contact Density*

The contact map of a protein with N amino acids is a symmetric $N \times N$ matrix C, with elements C_{ij} defined as:

$$C_{ij} = \begin{cases} 1 & \text{if amino acid i and j are in contact} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We define two amino acids as being in contact if their mutual distance is less than a given threshold. Alternative definitions are possible, for instance based on different mutual C_α distances (normally in the 7-12 Å range), or on $C_\beta - C_\beta$ atom distances (normally 6.5-8 Å), or on the minimal distance between two atoms belonging to the side-chain or backbone of the two residues (commonly 4.5 Å).

Let $\lambda(C) = \{\lambda : Cx = \lambda x\}$ be the spectrum of C, $\mathcal{S}_\lambda = \{x : Cx = \lambda x\}$ the corresponding eigenspace and $\bar{\lambda} = \max\{\lambda \in \lambda(C)\}$ the largest eigenvalue of C. The principal eigenvector of C, $\bar{x}$, is the eigenvector corresponding to $\bar{\lambda}$. $\bar{x}$ can also be expressed as the argument which maximises the Rayleigh quotient:

$$\forall x \in \mathcal{S}_\lambda : \frac{x^T C x}{x^T x} \leq \frac{\bar{x}^T C \bar{x}}{\bar{x}^T \bar{x}} \tag{2}$$

Eigenvectors are usually normalised by requiring their norm to be 1, e.g. $\|x\|_2 = 1 \ \forall x \in \mathcal{S}_\lambda$. Since C is an adjacency (real, symmetric) matrix, its eigenvalues are real. Since it is a normal matrix ($A^H A = AA^H$), its eigenvectors are orthogonal. Other basic properties can also be proven: the principal eigenvalue is positive; non-zero components of $\bar{x}$ have all the same sign [22]. Without loss of generality, we can assume they are positive, as in [23]. We define a protein's Contact Density as the principal eigenvector of its residue contact map, multiplied by its corresponding eigenvalue: $\bar{\lambda}\bar{x}$.

Contact Density is a sequence of the same length as a protein's primary sequence. Recently [23] a branch-and-bound algorithm was described that is capable of reconstructing the contact map from the exact PE, at least for single domain proteins of up to 120 amino acids. Predicting Contact Densities is thus interesting: as one-dimensional features, they are significantly more tractable than full contact maps; nonetheless a number of ways to obtain contact maps from contact densities may be devised, including modifying the reconstruction algorithm in [23] to deal with noise, or adding Contact Densities as an additional input feature to systems for the direct prediction of contact maps (such as [24]). Moreover, Contact Densities are informative in their own right and may be used to guide the search for optimal 3D configurations, or to identify protein domains [25,26]. Contacts among residues, in fact, constrain protein folding and characterise different protein structures (see Figure 1), constituting a structural fingerprint of the given protein[27].

Distill contains a state-of-the-art Contact Density predictor [28].

2.2. *Two-dimensional structural features*

2.2.1. *Contact Maps*

Contact maps (see definition above), or similar distance restraints have been proposed as intermediate steps between the primary sequence and the 3D structure (e.g. in [29,30,24]), for various reasons: unlike 3D coordinates, they are invariant to rotations and translations, hence less challenging to predict by machine learning systems [24,31]; quick, effective algorithms exist to derive 3D structures from them, for instance stochastic optimisation methods [12,32], distance geometry [33,34], or algorithms derived from the NMR literature and elsewhere [35,36,37]. Numerous methods have been developed for protein residue contact map prediction [29,30,24,38] and coarse (secondary structure element level) contact map prediction [31], and some improvements are slowly occurring (e.g. in [38], as shown by the CASP6 experiment [39]).

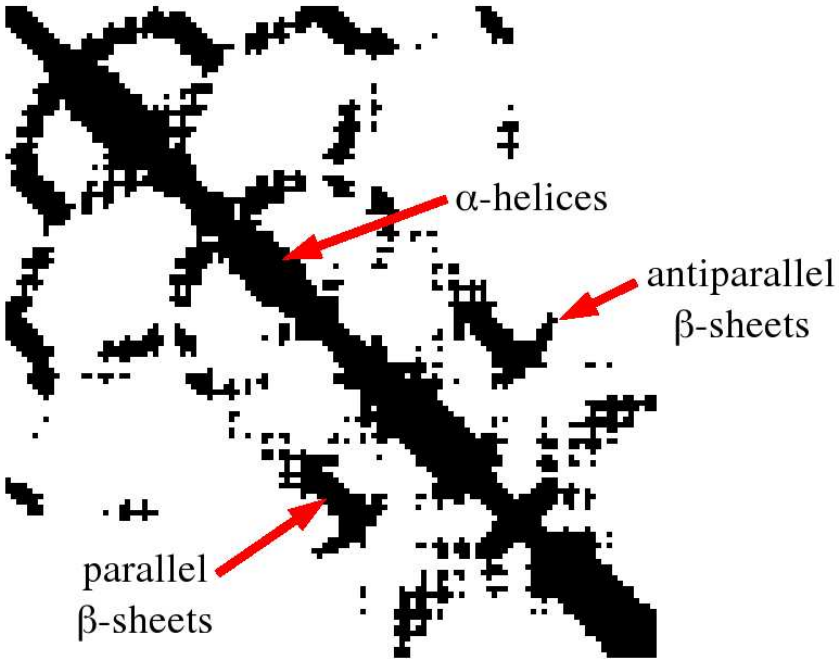

Fig. 1. Different secondary structure elements like helices (thick bands along the main diagonal) and parallel − or anti-parallel − β-sheets (thin bands parallel − or anti-parallel − to the main diagonal) are easily detected from the contact map.

Accurate prediction of residue contact maps is far from being achieved and limitations of existing prediction methods have again emerged at CASP6 and from automatic evaluation of structure prediction servers such as EVA [40]. There are various reasons for this: the number of positive and negative examples (contacts vs. non contacts) is strongly unbalanced; the number of examples grows with the squared length of the protein making this a tough computational challenge; capturing long ranged interactions in the primary sequence is difficult, hence grasping an adequate global picture of the map is a formidable problem.

The Contact Map predictor included in Distill relies on a combination of one-dimensional features as inputs and is state-of-the-art [28].

2.2.2. *Coarse Topologies*

We define the coarse structure of a protein as the set of three-dimensional coordinates of the N- and C-terminus of its secondary structure segments (helices, strands). By doing so, we: ignore coil regions, which are normally more flexible than helices and strands; assume that both strands and helices can be represented as rigid rods.

The actual coarse topology of a protein may be represented in a number of alternative ways: the map of distances, thresholded distances (contacts), or multi-class discretised distances between the centers of secondary structures [31,41]; the map of angles between the vectors representing secondary structure elements, or some discretisation thereof [41]. In each of these cases, if a protein contains M secondary structure elements, its coarse representation will be a matrix of $M \times M$ elements.

Although coarse maps are simpler, less informative representations of a protein structure than residue- or atom-level contact maps, they nonetheless can be exploited for a number of tasks, such as the fast reconstruction of coarse structures [41] and the rapid comparison and classification of proteins into structural classes [42].

Coarse contact maps represent compact sets of constraints and hence clear and synthetic pictures of the shape of a fold. For this reason, it is much less challenging to observe, and predict, long-range interactions between elements of a protein structure within a coarse model than in a finer one: a typical coarse map is composed by only hundreds of elements on a grid of tens by tens of secondary structure elements, while a residue-level contact map can contain hundreds of thousands or millions of elements and can typically be modelled only locally by statistical learning techniques.

For this reason, coarse maps can not only yield a substantial information compression with respect to residue maps, but can also assist in detecting interactions that would normally be difficult to observe at a finer scale, and contribute to improving residue maps, and structure predictions.

Distill contains predictors of coarse contact, multi-class distance and multi-class angle maps [41].

3. Review of Statistical Learning Methods Applied

3.1. *RNNs for undirected graphs*

A data structure is a graph whose nodes are marked by sets of domain variables, called labels. A skeleton class, denoted by the symbol #, is a set of unlabelled graphs that satisfy some topological conditions. Let $\mathcal{I}$ and $\mathcal{O}$

denote two label spaces: $\mathcal{I}^{\#}$ (resp. $\mathcal{O}^{\#}$) refers to the space of data structures with vertex labels in $\mathcal{I}$ (resp. $\mathcal{O}$) and topology #. Recursive models such as RNNs [43] can be employed to compute functions $\mathcal{T} : \mathcal{I}^{\#} \rightarrow \mathcal{O}^{\#}$ which map a structure into another structure of the same form but possibly different labels. In the classical framework, # is contained in the class of bounded DPAGs, i.e Directed Acyclic Graphs (DAGs) where each vertex has bounded outdegree (number of outgoing edges) and whose children are ordered. Recursive models normally impose causality on data processing: the state variables (and outputs) associated to a node depend only on the nodes upstream (i.e. from which a path leads to the node in question). The above assumption is restrictive in some domains and extensions of these models for dealing with more general undirected structures have been proposed [44,24,45].

A more general assumption is considered here: # is contained in the class of bounded-degree undirected graphs. In this case, there is no concept of causality and the computational scheme described in [43] cannot be directly applied. The strategy consists in splitting graphical processing into a *set* of causal "dynamics", each one computed over a plausible orientation of U.

More formally, assume $U = (V, E) \in \mathcal{I}^{\#}$ has one connected component. We identify a set of spanning DAGs $G_1, \ldots, G_m$ with $G_i = (V, E_i)$ such that:

- the undirected version of G_i is U
- $\forall\ v, u \in V\ v \neq u\ \exists\ i :\ (v, u) \in E_i^{\star}$ being $E_i^{\star}$ the transitive closure of E_i

and for each G_i, introduce a state variable X_i computed in the usual way. Fig.2 (left) shows a compact description of the set of dependencies among the input, state and output variables.

Connections run from vertices of the input structure (layer I) to vertices of the spanning DAGs and from these nodes to nodes of the output structure (layer O).

Using weight-sharing, the overall model can be summarized by $m + 1$ distinct neural networks implementing the output function $O(v) = g(X_1(v), \ldots, X_m(v), I(v))$ and m state transition functions $X_i(v) = f_i(X_i(ch_1[v]), \ldots, X_i(ch_k[v]), I(v))$. Learning can proceed by gradient-descent (back-propagation) due to the acyclic nature of the underlying graph. Within this framework, we can easily describe all contextual RNNs architecture developed so far. Fig.2 (center) shows that an undirected sequence is spanned by two sequences oriented in opposite directions. We

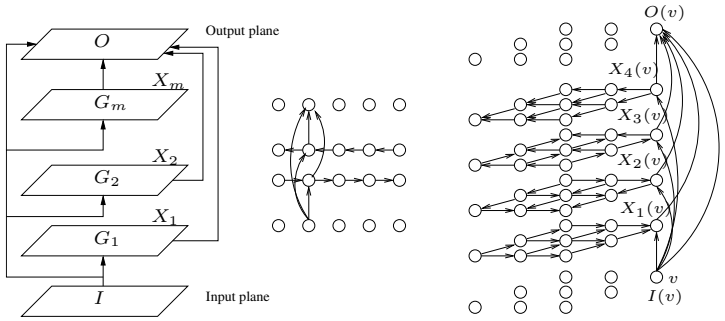

Fig. 2. (left): Contextual RNNs, dependencies among input, state and output variables. (center and right): processing of undirected sequences and grids with contextual RNNs (only a subset of connections are shown).

then obtain bi-directional recurrent neural networks [44] or 1D DAG-RNNs if we consider a straightforward generalisation from sequences to undirected graphs. For the case of two dimensional objects (e.g. contact maps), they can be seen as two-dimensional grids spanned by four directed grids oriented from each cardinal corner (Fig.2, right). The corresponding model is called 2D DAG-RNNs [24]. The 1D and 2D DAG-RNNs adopted in our architectures are described in more detail below.

3.2. *1D DAG-RNN*

In the 1D DAG-RNNs we adopt, connections along the forward and backward hidden chains span more than 1-residue intervals, creating shorter paths between inputs and outputs. These networks take the form:

$$\begin{aligned} o_j &= \mathcal{N}^{(O)}\left(i_j, h_j^{(F)}, h_j^{(B)}\right) \\ h_j^{(F)} &= \mathcal{N}^{(F)}\left(i_j, h_{j-1}^{(F)}, \ldots, h_{j-S}^{(F)}\right) \\ h_j^{(B)} &= \mathcal{N}^{(B)}\left(i_j, h_{j+1}^{(B)}, \ldots, h_{j+S}^{(B)}\right) \\ j &= 1, \ldots, N \end{aligned}$$

where $h_j^{(F)}$ and $h_j^{(B)}$ are forward and backward chains of hidden vectors with $h_0^{(F)} = h_{N+1}^{(B)} = 0$. We parametrise the output update, forward update and backward update functions (respectively $\mathcal{N}^{(O)}$, $\mathcal{N}^{(F)}$ and $\mathcal{N}^{(B)}$) using three two-layered feed-forward neural networks. In our tests the input associated with the j-th residue i_j contains amino acid information, and further one-dimensional information in some predictors (see section 4 for details). In all cases amino acid information is obtained from multiple

sequence alignments of the protein sequence to its homologues to leverage evolutionary information. The input presented to the networks is the frequency of each of the non-gap symbols, plus the overall frequency of gaps in each column of the alignment. I.e., if n_{jk} is the total number of occurrences of symbol j in column k, and g_k the number of gaps in the same column, the j^{th} input to the networks in position k is:

$$\frac{n_{jk}}{\sum_{v=1}^{u} n_{vk}} \tag{3}$$

for $j = 1 \dots u$, where u is the number of non-gap symbols while the $u+1^{th}$ input is:

$$\frac{g_k}{g_k + \sum_{v=1}^{u} n_{vk}} \tag{4}$$

In some of our predictors we also adopt a second filtering 1D DAG-RNN [11]. The network is trained to predict the structural feature given first-layer structural feature predictions. The i-th input to this second network includes the first-layer predictions in position i augmented by first stage predictions averaged over multiple contiguous windows. I.e., if $c_{j1}, \dots c_{jm}$ are the outputs in position j of the first stage network corresponding to estimated probability of residue j being labelled in class m, the input to the second stage network in position j is the array I_j:

$$\begin{aligned} I_j = (&c_{j1}, \dots, c_{jm}, \\ &\sum_{h=k_{-p}-w}^{k_{-p}+w} c_{h1}, \dots, \sum_{h=k_{-p}-w}^{k_{-p}+w} c_{hm}, \\ &\dots \\ &\sum_{h=k_p-w}^{k_p+w} c_{h1}, \dots, \sum_{h=k_p-w}^{k_p+w} c_{hm}) \end{aligned} \tag{5}$$

where $k_f = j + f(2w+1)$, $2w+1$ is the size of the window over which first-stage predictions are averaged and $2p+1$ is the number of windows considered. In the tests we use $w = 7$ and $p = 7$. This means that 15 contiguous, non-overlapping windows of 15 residues each are considered, i.e. first-stage outputs between position $j - 112$ and $j + 112$, for a total of 225 contiguous residues, are taken into account to generate the input to the filtering network in position j.

3.2.1. *Ensembling 1D DAG-RNNs*

A few two-stage 1D DAG-RNN models are trained independently and ensemble averaged to build each final predictor. Differences among models are introduced by two factors: stochastic elements in the training protocol, such as different initial weights of the networks and different shuffling of the examples; different architecture and number of free parameters of the models.

In [16] a slight improvement in secondary structure prediction accuracy was obtained by "brute ensembling" of several tens of different models trained independently. Here we adopt a less expensive technique: a copy of each of the models is saved at regular intervals (100 epochs) during training. Stochastic elements in the training protocol (similar to that described in [10]) guarantee that differences during training are non-trivial.

3.3. *2D DAG-RNN*

All systems for the prediction of two-dimensional structural features are based on 2D DAG-RNN, described in [24] and [31]. This is a family of adaptive models for mapping two-dimensional matrices of variable size into matrices of the same size.

We adopt 2D DAG-RNNs with *shortcut connections*, i.e. where lateral memory connections span N-residue intervals, where $N > 1$. If $o_{j,k}$ is the entry in the j-th row and k-th column of the output matrix, and $i_{j,k}$ is the input in the same position, the input-output mapping is modelled as:

$$
\begin{aligned}
o_{j,k} &= \mathcal{N}^{(O)}\left(i_{j,k}, h^{(1)}_{j,k}, h^{(2)}_{j,k}, h^{(3)}_{j,k}, h^{(4)}_{j,k}\right) \\
h^{(1)}_{j,k} &= \mathcal{N}^{(1)}\left(i_{j,k}, h^{(1)}_{j-1,k}, .., h^{(1)}_{j-S,k}, h^{(1)}_{j,k-1}, .., h^{(1)}_{j,k-S}\right) \\
h^{(2)}_{j,k} &= \mathcal{N}^{(2)}\left(i_{j,k}, h^{(2)}_{j+1,k}, .., h^{(2)}_{j+S,k}, h^{(2)}_{j,k-1}, .., h^{(2)}_{j,k-S}\right) \\
h^{(3)}_{j,k} &= \mathcal{N}^{(3)}\left(i_{j,k}, h^{(3)}_{j+1,k}, .., h^{(3)}_{j+S,k}, h^{(3)}_{j,k+1}, .., h^{(3)}_{j,k+S}\right) \\
h^{(4)}_{j,k} &= \mathcal{N}^{(4)}\left(i_{j,k}, h^{(4)}_{j-1,k}, .., h^{(4)}_{j-S,k}, h^{(4)}_{j,k+1}, .., h^{(4)}_{j,k+S}\right) \\
& \qquad j,k = 1,\ldots,N
\end{aligned}
$$

where $h^{(n)}_{j,k}$ for $n = 1,\ldots,4$ are planes of hidden vectors transmitting contextual information from each corner of the matrix to the opposite corner. We parametrise the output update, and the four lateral update functions (respectively $\mathcal{N}^{(O)}$ and $\mathcal{N}^{(n)}$ for $n = 1,\ldots,4$) using five two-layered feed-forward neural networks, as in [31].

In our tests the input $i_{j,k}$ contains amino acid information, and structural information from one-dimensional feature predictors. Amino acid information is again obtained from multiple sequence alignments.

4. Predictive Architecture

In this section we briefly describe the individual predictors composing Distill. Currently we adopt three predictors of one-dimensional features: Porter (secondary structure), PaleAle (solvent accessibility), BrownAle (contact density); and two predictors of two-dimensional features: XStout (coarse contact maps/topologies); XXStout (residue contact maps). The overall pipeline is highlighted in figure 3.

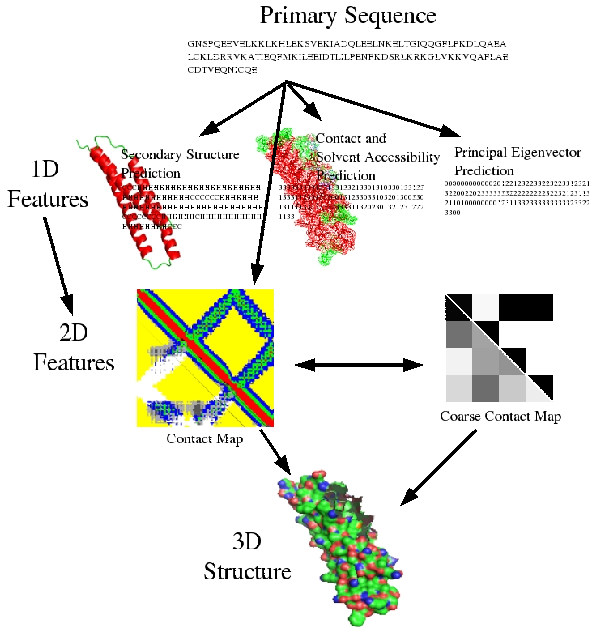

Fig. 3. **Distill**'s modelling scheme (http://distill.ucd.ie).

4.1. *Data set generation*

All predictors are trained on dataset extracted from the December 2003 25% pdb_select list[b]. We use the DSSP program [46] (CMBI version) to assign target structural features and remove sequences for which DSSP does not produce an output due, for instance, to missing entries or format errors. After processing by DSSP, the set contains 2171 protein and 344,653 amino acids (S2171).

We extract three distinct training/test protocols from S2171:

- Five-fold cross validation splits (5FOLD), in which test sequences are selected in an interleaved fashion from the whole set sorted alphabetically by PDB code (every fifth $+k$ sequence is picked). In this case the training sets contain 1736 or 1737 proteins and the test sets 435 or 434. The performances given on 5FOLD are effectively measured on the whole S2171, as each of its proteins appears once and only once in the test sets.
- The first fold of the above containing a training set of 1736 proteins (S1736) and a test set of 435 (S435).
- The same as the above, but containing only sequences of length at most 200 residues, leaving 1275 proteins in the training set (S1275) and 327 (S327) proteins in the test set.

Multiple sequence alignments for S2171 are extracted from the NR database as available on March 3 2004 containing over 1.4 million sequences. The database is first redundancy reduced at a 98% threshold, leading to a final 1.05 million sequences. The alignments are generated by three runs of PSI-BLAST [15] with parameters $b = 3000$, $e = 10^{-3}$ and $h = 10^{-10}$.

4.2. *Training protocols*

All RNNs are trained by minimising the cross-entropy error between the output and target probability distributions, using gradient descent with no momentum term or weight decay. The gradient is computed using the Back-propagation through structure (BPTS) algorithm (for which, see e.g. [43]). We use a hybrid between online and batch training, with $200 - 600$ (depending on the set) batch blocks (roughly 3 proteins each) per training set. Thus, the weights are updated $200 - 600$ times per epoch. The

[b] http://homepages.fh-giessen.de/˜hg12640/pdbselect

training set is also shuffled at each epoch, so that the error does not decrease monotonically. When the error does not decrease for 50 consecutive epochs, the learning rate is divided by 2. Training stops after 1000 epochs for one-dimensional systems, and 300 epochs for two-dimensional ones.

4.3. *One-dimensional feature predictors*

4.3.1. *Porter*

Porter[11] is a system for protein secondary structure prediction based on an ensemble of 45 two-layered 1D DAG-RNNs. Porter is an evolution of the popular SSpro[10] server. Porter's improvements include:

- Efficient input coding. In Porter the input at each residue is coded as a letter out of an alphabet of 25. Beside the 20 standard amino acids, B (aspartic acid or asparagine), U (selenocysteine), X (unknown), Z (glutamic acid or glutamine) and . (gap) are considered. The input presented to the networks is the frequency of each of the 24 non-gap symbols, plus the overall proportion of gaps in each column of the alignment.
- Output filtering and incorporation of predicted long-range information. In Porter the first-stage predictions are filtered by a second network. The input to this network includes the predictions of the first stage network averaged over multiple contiguous windows, covering 225 residues.
- Up-to-date training sets. Porter is trained on the S2171 set.
- Large ensembles (45) of models.

Porter, tested by a rigorous 5-fold cross validation procedure (set 5FOLD), achieves 79% correct classification on the "hard" CASP 3-class assignment (DSSP H, G, I → helix; E, B → strand; S, T, . → coil), and currently has the highest performance (over 80%) of all servers tested by assessor EVA[40].

4.3.2. *Pale Ale*

PaleAle is a system for the prediction of protein relative solvent accessibility. Each amino acid is classified as being in one of 4 (approximately equally frequent) classes: B=completely buried (0-4% exposed); b=partly buried (4-25% exposed); e=partly exposed (25-50% exposed); E=completely exposed (more than 50% exposed).

The architecture of PaleAle's classifier is an exact copy of Porter's

(described above). PaleAle's accuracy, measured on the same large, non-redundant set adopted to train Porter (5FOLD) exceeds 55% correct 4-class classification, and roughly 80% 2-class classification (Buried vs Exposed, at 25% threshold).

4.3.3. *Brown Ale*

BrownAle is a system for the prediction of protein Contact Density. We define Contact Density as the Principal Eigenvector (PE) of a protein's residue contact map at 8Å, multiplied by the principal eigenvalue. Contact Density is useful for the ab initio the prediction of protein of protein structures for many reasons:

- algorithms exist to reconstruct the full contact maps from the PE for short proteins [23], and correct contact maps lead to correct 3D structures;
- Contact Density may be used directly, in combination with other constraints, to guide the search for optimal 3D configurations;
- Contact Density may be adopted as an extra input feature to systems for the direct prediction of contact maps, as in the XXStout server described below;
- predicted PE may be used to identify protein domains [25].

BrownAle predicts Contact Density in 4 classes. The class thresholds are assigned so that the classes are approximately equally numerous, as follows: N = very low contact density (0,0.04); n = medium-low contact density (0.04,0.18); c = medium-high contact density (0.18,0.54); C = very high contact density (greater than 0.54).

BrownAle's architecture is an exact copy of Porter's (described above). The accuracy of BrownAle, measured on the S1736/S435 datasets is 46.5% for the 4-class problem, and roughly 73% if the 4 classes are mapped into 2 (dense vs. non dense).

We have shown [28] that these performance levels for Contact Density prediction yield sizeable gains to residue contact map prediction, and that these gains are especially significant for long-ranged contacts, which are known to be both harder to predict and critical for accurate 3D reconstruction.

4.4. *Two-dimensional feature predictors*

4.4.1. *XXStout*

XXStout is a system for the prediction of protein residue contact maps. Two residues are considered in contact if their C-αs are closer than a given threshold. XXStout predicts contacts at three different thresholds: 6Å, 8Å and 12Å. The contact maps are predicted as follows: protein secondary structure, solvent accessibility and contact density are predicted from the sequence using, respectively, Porter, PaleAle and BrownAle; ensembles of two-dimensional Recursive Neural Networks predict the contact maps based on the sequence, a 2-dimensional profile of amino-acid frequencies obtained from a PSI-BLAST alignment of the sequence against the NR, and predicted secondary structure, solvent accessibility and contact density. The introduction of contact density as an intermediate representation improves significantly the performances of the system. XXStout is trained the S1275 set and tested on S327. Tables 1 and 2 summarise the performances of XXStout on S327. Performances are given for the protein length/5 and protein length/2 contacts with the highest probability, for sequence separations of at least 6, at least 12, and at least 24, in CASP style [3]. These performances compare favourably with the best predictors at the latest CASP competition [28].

Table 1. XXStout. Top protein length/5 contacts classification performance as: precision%(recall%)

separation	≥ 6	≥ 12	≥ 24
8Å	46.4% (5.9%)	35.4% (5.7%)	19.8% (4.6%)
12Å	89.9% (2.3%)	62.5% (2.0%)	49.9% (2.2%)

Table 2. XXStout. Top protein length/2 contacts classification performance as: precision%(recall%)

separation	≥ 6	≥ 12	≥ 24
8Å	36.6% (11.8%)	27.0% (11.0%)	15.7% (9.3%)
12Å	85.5% (5.5%)	55.6% (4.6%)	43.8% (4.9%)

4.4.2. *XStout*

XStout is a system for the prediction of coarse protein topologies. A protein is represented by a set of rigid rods associated with its secondary structure elements (α-helices and β-strands, as predicted by Porter). First, we employ cascades of recursive neural networks derived from graphical models to predict the relative placements of segments. These are represented as distance maps discretised into 4 classes. The discretisation levels ((0Å,10Å),(10Å,18Å),(18Å,29Å),(29Å,∞)) are statistically inferred from a large and curated data set. Coarse 3D folds of proteins are then assembled starting from topological information predicted in the first stage. Reconstruction is carried out by minimising a cost function taking the form of a purely geometrical potential. The reconstruction procedure is fast and often leads to topologically correct coarse structures, that could be exploited as a starting point for various protein modelling strategies [41]. Both coarse distance maps and a number of coarse reconstructions are produced by XStout.

5. Modelling Protein Backbones

The architecture of Fig. 3 is designed with the intent of making large scale (i.e. genomic level) structure prediction of proteins of moderate ($length \leq 200$ AA) and possibly larger sizes. Our design relies on the inductive learning components described in the previous sections and is based on a pipeline involving stages of computation organised hierarchically. For a given input sequence, first a set of flattened structural representations (1D features) is predicted. These 1D features, together with the sequence are then used as an input to infer the shape of 2D features. In the last stage, we predict protein structures by means of an optimisation algorithm searching the 3D conformational space for a configuration that minimises a cost. The cost is modelled as a function of geometric constraints (pseudo energy) inferred from the underlying set of 1D and 2D predictions (see section 4).

Inference of the contact map is a core component of the pipeline and is performed in $O(|w|n^2)$ time, where n is the length of the input sequence and $|w|$ is the number of weights of our trained 2D-DAG RNNs (see section 4.4.1). In section 5.3, we illustrate a 3D reconstruction algorithm with $O(n^2)$ time complexity. All the steps are then fully automated and fast enough to make the approach suitable to be applied to multi-genomic scale predictions.

5.1. *Protein representation*

To avoid the computational burden of full-atom models, proteins are coarsely described by their main chain alpha carbon (C_α) atoms without any explicit side-chain modelling. The bond length of adjacent C_α atoms is restricted to lie in the interval 3.803 Å$\pm$ 0.07 in agreement with the experimental range ($D_B = 3.803$ is the average observed distance). To mimic the minimal observed distance between atoms of different amino acids, the excluded volume of each C_α is modelled as a hard sphere of radius $D_{HC} = 5.0$ Å(distance threshold for hard core repulsion). Helices predicted by Porter are modelled directly as ideal helices.

5.2. *Constraints-based Pseudo Energy*

The pseudo-energy function used to guide the search is shaped to encode the constraints represented by the contact map and by the particular protein representation (as described above).

Let $\mathcal{S}_n = \{r_i\}_{i=1\ldots n}$ be a sequence of n 3D coordinates, with $r_i = (x_i, y_i, z_i)$ the coordinates of the i-th C_α atom of a given conformation related to a protein p. Let $\mathcal{D}_{\mathcal{S}_n} = \{d_{ij}\}_{i<j}, d_{ij} = \|r_i - r_j\|_2$, be the corresponding set of $n(n-1)/2$ mutual distances between C_α atoms. A first set of constraints comes from the (predicted) contact map which can be represented as a matrix $C = \{c_{ij}\} \in \{0,1\}^{n^2}$. The representation of protein models discussed in the previous paragraph induces the constraints $\mathcal{B} = \{d_{ij} \in [3.733, 3.873], |i-j| = 1\}$, encoding bond lengths, and another set $\mathcal{C} = \{d_{ij} \geq D_{HC}, i \neq j\}$ for clashes. The set $\mathcal{M} = C \cup \mathcal{B} \cup \mathcal{C}$ defines the configurational space of physically realisable protein models.

The cost function measures the degree of structural matching of a given conformation $\mathcal{S}_n$ to the available constraints. Let $\mathcal{F}_0 = \{(i,j) \mid d_{ij} > d_T \wedge c_{ij} = 1\}$ denote the pairs of amino acid in contact according to C but not in $\mathcal{S}_n$ ("false negatives"). Similarly, define $\mathcal{F}_1 = \{(i,j) \mid d_{ij} \leq d_T \wedge c_{ij} = 0\}$ as the pairs of amino acids in contact in $\mathcal{S}_n$ but not according to C ("false positives"). The objective function is then defined as:

$$\begin{aligned} \mathrm{C}(\mathcal{S}_n, \mathcal{M}) = \alpha_0 \{1 + \sum_{(i,j)\in\mathcal{F}_0} (d_{ij}/D_T)^2 + \sum_{(i,j):d_{ij}\notin\mathcal{B}} (d_{ij} - D_B)^2\} \\ + \alpha_1 |\mathcal{F}_1| + \alpha_2 \sum_{(i,j):d_{ij}\notin\mathcal{C}} e^{(D_{HC} - d_{ij})} \end{aligned} \quad (6)$$

Note how the cost function is based only on simple geometric terms. The combination of this function with a set of moves allows the exploration of

the configurational space.

5.3. *Optimisation Algorithm*

The algorithm we used for the reconstruction of the coordinates of protein C_α traces is organised in two sequential phases, *bootstrap* and *search*.

The function of the first phase is to *bootstrap* an initial physically realisable configuration with a self-avoiding random walk and explicit modelling of predicted helices. A random structure is generated by adding C_α positions one after the other until a draft of the whole backbone is produced. More specifically, this part runs through a sequence of n steps, where n in the length of the input chain. At stage i, the position of the i-th C_α is computed as $r_i = r_{i-1} + d\frac{r}{|r|}$ where $d \in [3.733, 3.873]$ and r is a random direction vector. Both d and r are uniformly sampled. If the i-th residue is predicted at the beginning of an helix all the following residues in the same segment are modelled as an ideal helix with random orientation.

In the *search* step, the algorithm refines the initial bootstrapped structure by global optimisation of the pseudo-potential function of Eq. 6 using local moves and a simulated annealing protocol. Simulated annealing is a good choice in this case, since the constraints obtained from various predictions are in general not realisable and contradictory. Hence the need for using a "soft" method that tries to enforce as many constraints as possible never terminating with failure, and is robust with respect to local minima caused by contradictions. The search strategy is similar to that in [12], but with a number of modifications. At step t of the search, a randomly chosen C_α atom at position $r_i^{(t)}$ is displaced to the new position $r_i^{(t+1)}$ by a crankshaft move, leaving all the others C_α atoms of the protein in their original position (see Figure 4). Secondary structure elements are displaced as a whole, without modifying their geometry (see Figure 5). The move in this case has one further degree of freedom in the helix rotation around its axis. This is assigned randomly, and uniformly distributed. A new set of coordinates $\mathcal{S}^{(t+1)}$ is accepted as the best next candidate with probability $p = min(1, e^{\Delta C/T^{(t)}})$ defined by the annealing protocol, where $\Delta C = C(\mathcal{S}^{(t)}, \mathcal{M}) - C(\mathcal{S}^{(t+1)}, \mathcal{M})$ and $T^{(t)}$ is the temperature at stage t of the schedule.

The computational complexity of the above procedure depends on the maximum number of available steps for the annealing schedule and the number of operations required to compute the potential of Eq.6 at each step. The computation of this function is dominated by the $O(n^2)$ steps required

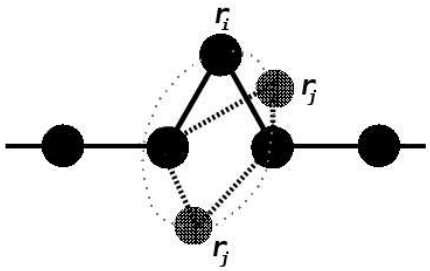

Fig. 4. Crankshaft move: the i-th C_α at position r_i is displaced to postition r_j without moving the others atoms of the protein.

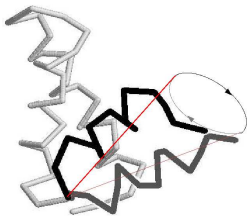

Fig. 5. Secondary structure elements are displaced as a whole, without modifying their geometry.

for the comparison of all pairwise mutual distances with the entries of the given contact map. Note however that the types of move adopted allow to explicitly avoid the evaluation of the potential for each pair of positions. In the case of a residue crankshaft move, since only one position of the structure is affected by it, ΔC can be directly computed in $O(n)$ time by summing only the terms of Eq.6 that change. For instance, in the case of the terms taking into account the contact map, the displacement of one C_α changes only a column and a row of the map induced by the configuration, hence the effect of the displacement can be computed by evaluating the $O(n)$ contacts on the row and column affected. The complexity of evaluating the energy after moving rigidly a whole helix is the same as moving all the amino acids on the helix independently. Hence, the overall cost of a search

is $O(ns)$ where n is the protein length and s is the number of residues moved during the search. In practice, the number s necessary to achieve convergence is proportional to the protein length, which makes the search complexity quadratic in n. A normal search run for a protein of length 100 or less takes a few tens of seconds on a single state-of-the-art CPU, roughly the same as computing the complex of 1D and 2D feature predictions.

6. Reconstruction Results

The protein data set used in reconstruction simulations consists of a non redundant set of 258 protein structures showing no homology to the sequences employed to train the underlying predictive systems. This set includes proteins of moderate size (51 to 200 amino acids) and diverse topology as classified by SCOP (all-α, all-β, α/β, $\alpha+\beta$, surface, coiled-coil and small).

In all the experiments, we run the annealing protocol using a non linear (exponential decay) schedule with initial (resp. final) temperature proportional to the protein size (resp. 0). Pseudo energy parameters are set to $\alpha_0 = 0.2$ (false non-contacts), $\alpha_1 = 0.02$ (false contacts) and $\alpha_2 = 0.05$ (clashes), so that the conformational search is biased towards the generation of compact clash-free structures and with as many of the predicted contacts realised.

Our algorithm is first benchmarked against a simple baseline that consists in predicting models where all the amino acids in the chain are collapsed into the same point (center of mass). We run two sets of simulations: one where the reconstructions are based on native contact maps and structural features; one where all the structural features including contact maps are predicted. Using contact maps of native folds allows us to validate the algorithm and to estimate the expected upper bound of reconstruction performance. In order to assess the quality of predictions, two measures are considered here: root mean square deviation (RMSD); longest common sequence (LCS) averaged over four RMSD thresholds (1, 2, 4 and 8 Å) and normalised by the sequence length.

For each protein in the test set, we run 10 folding simulations and average the distance measures obtained over all 258 proteins. In Figure 6, the average RMSD vs sequence length is shown for models derived from true contact maps (red crosses) and from predicted contact maps (green crosses), together with the baseline computed for sequences of the same length of the query. With true (reps. predicted) contact maps, the RMSD averaged over all test chains is 5.11 Å(resp. 13.63 Å), whereas the LCS1248

measure is 0.57 (resp. 0.29). For sequences of length up to 100 amino acids, the reconstruction algorithm using predicted contact maps obtains an an average LCS1248 of 0.43.

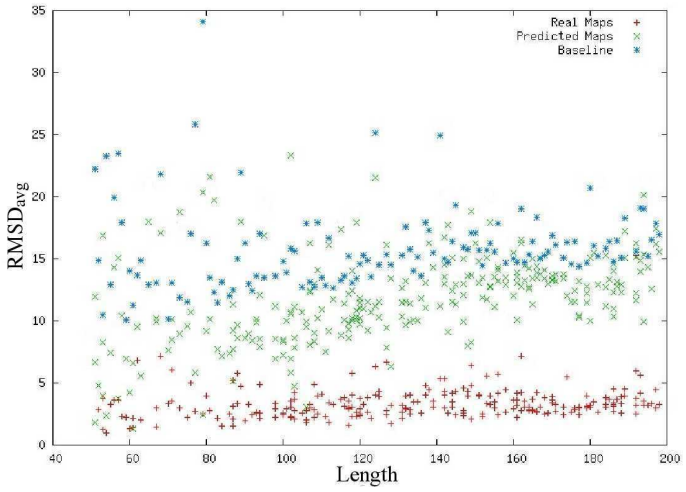

Fig. 6. Average RMSD vs sequence length.

In a second and more detailed suite of experiments, for each protein in the test set, we run 200 folding simulations and cluster the corresponding optimal structures at final temperature after 10000 iterations. The clustering step aims at finding regions of the configurational space that are more densely populated, hence are likely to represent tight minima of the pseudo-energy function. The centroid of the n^{th} cluster is the configuration whose q^{th} closest neighbour is at the smallest distance. After identifying a cluster, its centroid and q closest neighbours are removed from the set of configurations. Distances are measured by RMSD (see below), and a typical value of q is between 2 and 20. In our experiments, the centroids of the top 5 clusters are on average slightly more accurate than the average reconstruction. A protein is considered to be predicted with the correct topology if at least one of the five cluster centroids is within 6.5 Å RMSD to the native structure for at least 80% of the whole protein length ($LCS(6.5) \geq 0.8$).

Table 3. Number of topologically correct ($LCS(6.5) \geq 0.8$) predicted models.

Cluster size	*All*	*Short*	*Medium*	*Long*
2	26/258	22/62	3/103	1/93
3	32/258	24/62	7/103	1/93
10	29/258	23/62	6/103	0/93
20	32/258	26/62	6/103	0/93

Table 4. Percentage of correctly predicted topologies with respect to sequence length and structural class.

Length	α	β	$\alpha+\beta$	α/β	Surface	Coiled-coil	Small
all	20.3	4.0	7.3	6.3	33.3	66.7	16.7
short	64.7	25.0	35.7	33.3	60.0	60.0	25.0
medium	6.3	0	2.8	11.8	0	100	0
long	0	0	0	0	0	-	-

A first set of results is summarised in table 3, where the proteins are divided into separate categories based on the number of models in each cluster (2, 3, 10 and 20) and length: from 51 to 100 amino acids (small), between 100 and 150 amino acids (medium) and from 150 to 200 amino acids (long). Table 3 shows for each combination of length and cluster size the number of proteins in the test set for which at least one of the five cluster centroids is within 6.5 Å of the native structure over 80% of the structure. From the table it is evident that correctly predicted topologies are restricted to proteins of limited size (up to 100-150 amino acids). Distill is able to identify the correct fold for short proteins in almost half of the cases, and for a few further cases in the case of proteins of moderate size (from 100 to 150 residues).

In table 4, we group the results for 20 dimensional clusters according to the SCOP assigned structural class and sequence length. For each combination of class and length, we report the fraction of proteins where at least one of the five cluster centroids $LCS(6.5) \geq 0.8$ to the native structure. These results indicate that a significant fraction of α-helical proteins and those lacking significant structural patterns are correctly modelled. Reliable identification of strands and the corresponding patterns of connection is a major source of difficulty. Nevertheless, the reconstruction pipeline identifies almost correct folds for about a third of the cases in which a short protein contains a significant fraction of β-paired residues.

Figures 7, 8, 9 contain examples of predicted protein models from native

and predicted contact maps.

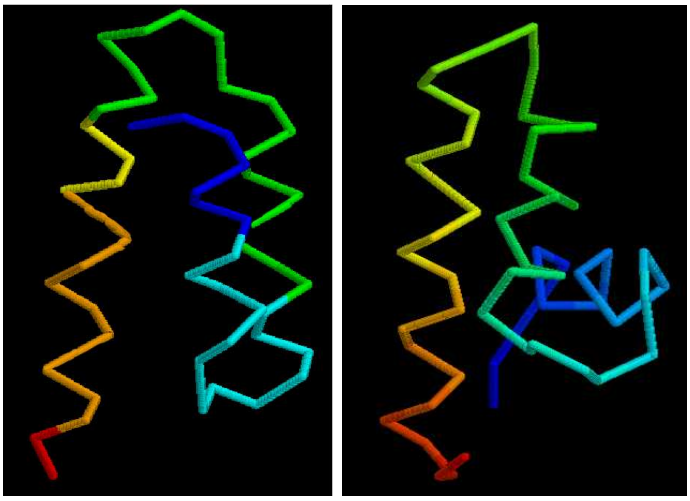

Fig. 7. Examples of reconstruction, protein 1OKSA (53 amino acids): real structure (left) and derived protein model from predicted contact map (right, RMSD = 4.24 Å).

7. Conclusions

In this chapter we have presented Distill, a modular and fully automated computational system for ab initio prediction of protein coarse models. Distill's architecture is composed of: (1) a set of state-of-the-art predictors of protein features (secondary structure, relative solvent accessibility, contact density, residue contact maps, contact maps between secondary structure elements) based on machine learning techniques and trained on large, non-redundant subsets of the PDB; (2) a simple and fast 3D reconstruction algorithm guided by a pseudo energy defined according to these predicted features.

Although Distill's 3D models are often still crude, nonetheless they may yield important information and support other related computational tasks. For instance, they can be effectively used to refine secondary structure and contact map predictions[47] and may provide a valuable source of information to identify protein functions more accurately than it would be possible by sequence alone[7]. Distill's modelling scheme is fast and makes genomic scale

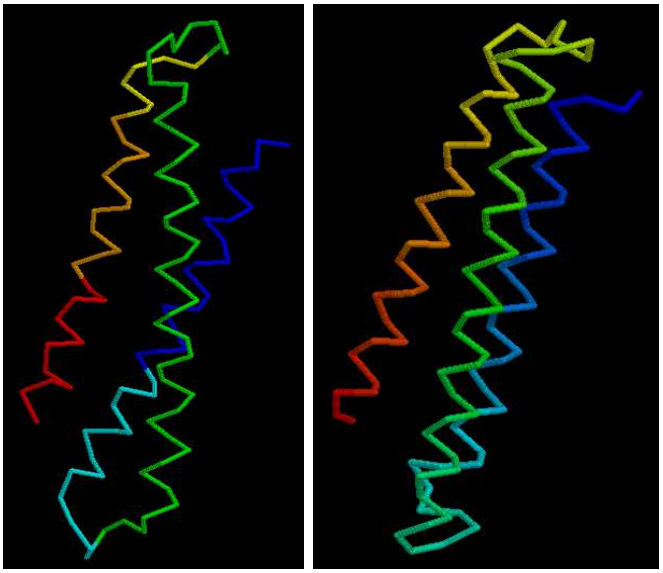

Fig. 8. Example of reconstruction, protein 1LVF (106 amino acids): real structure (left) and derived protein model from predicted contact map (right, RMSD = 4.31 Å).

structural modelling tractable by solving hundreds of thousands of protein coordinates in the order of days.

8. Acknowledgement

This work is supported by Science Foundation Ireland grants 04/BR/CS0353 and 05/RFP/CMS0029, grant RP/2005/219 from the Health Research Board of Ireland, a UCD President's Award 2004, and an Embark Fellowship from the Irish Research Council for Science, Engineering and Technology to AV.

References

1. C.A. Orengo, J.E. Bray, T. Hubbard, L. Lo Conte, and I.I. Sillitoe. Analysis and assessment of ab initio three-dimensional prediction, secondary structure, and contacts prediction. *Proteins: Structure, Function and Genetics*, 37(S3):149–70, 1999.
2. AM Lesk, L Lo Conte, and TJP Hubbard. Assessment of novel fold targets in CASP4: predictions of three-dimensional structures, secondary structures,

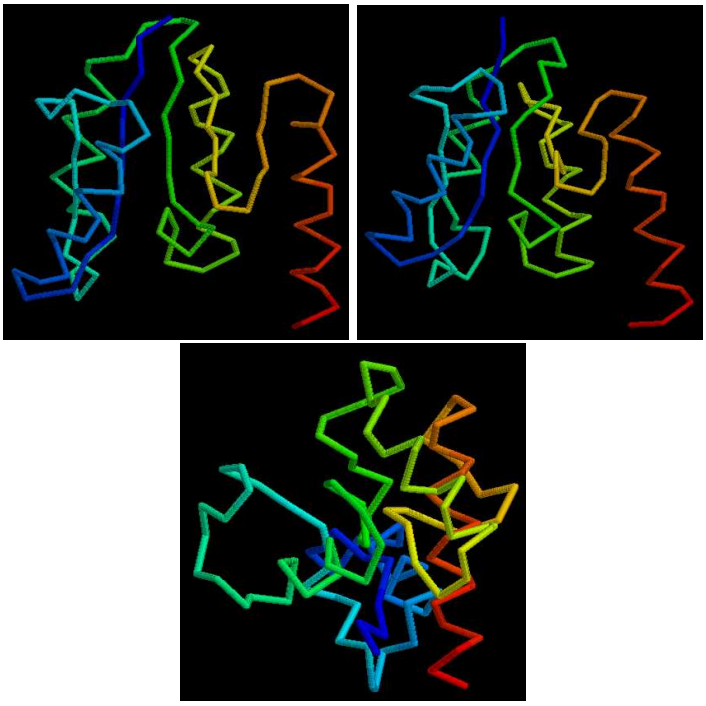

Fig. 9. Examples of reconstruction, protein 2RSL (119 amino acids): real structure (top-left), predicted model from true contact map (top-right, RMSD = 2.26 Å) and predicted model from predicted contact map (bottom, RMSD = 11.1 Å).

function and genetics. *Proteins: Structure, Function and Genetics*, S5:98–118, 2001.

3. J Moult, K Fidelis, A Zemla, and T Hubbard. Critical assessment of methods of protein structure prediction (casp)-round v. *Proteins*, 53(S6):334–9, 2003.
4. J Moult, K Fidelis, A Tramontano, B Rost, and T Hubbard. Critical assessment of methods of protein structure prediction (casp)-round vi. *Proteins*, Epub 26 Sep 2005, in press.
5. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucl. Acids Res.*, 28:235–242, 2000.
6. JJ Vincent, CH Tai, BK Sathyanarayana, and B Lee. Assessment of casp6 predictions for new and nearly new fold targets. *Proteins*, Epub 26 Sep 2005, in press.
7. R Bonneau, CE Strauss, CA Rohl, D Chivian, P Bradley, L Malmstrom, T Robertson, and D Baker. De novo prediction of three-dimensional struc-

tures for major protein families. *Journal of Molecular Biology*, 322(1):65–78, 2002.
8. DT Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, 292:195–202, 1999.
9. B Rost and C Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.
10. G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47:228–235, 2002.
11. G. Pollastri and A. McLysaght. Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*, 21(8):1719–20, 2005.
12. M Vendruscolo, E Kussell, and E Domany. Recovery of protein structure from contact maps. *Folding and Design*, 2:295–306, 1997.
13. DT Jones. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.*, 287:797–815, 1999.
14. P Bradley, D Chivian, J Meiler, KMS Misura, CA Rohl, WR Schief, WJ Wedemeyer, O Schueler-Furman, P Murphy, J Schonbrun, CEM Strauss, and D Baker. Rosetta predictions in casp5: Successes, failures, and prospects for complete automation. *Proteins*, 53(S6):457–68, 2003.
15. SF Altschul, TL Madden, and AA Schaffer. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucl. Acids Res.*, 25:3389–3402, 1997.
16. TN Petersen, C Lundegaard, M Nielsen, H Bohr, J Bohr, S Brunak, GP Gippert, and O Lund. Prediction of protein secondary structure at 80% accuracy. *Proteins: Structure, Function and Genetics*, 41(1):17–20, 2000.
17. B. Rost and C. Sander. Conservation and prediction of solvent accessibility in protein families. *Proteins: Structure, Function and Genetics*, 20:216–226, 1994.
18. H. Naderi-Manesh, M. Sadeghi, S. Arab, and A. A. Moosavi Movahedi. Prediction of protein surface accessibility with information theory. *Proteins: Structure, Function and Genetics*, 42:452–459, 2001.
19. M. H. Mucchielli-Giorgi, S. Hazout, and P. Tuffery. PredAcc: prediction of solvent accessibility. *Bioinformatics*, 15:176–177, 1999.
20. J. A. Cuff and G. J. Barton. Application of multiple sequence alignments profiles to improve protein secondary structure prediction. *Proteins: Structure, Function and Genetics*, 40:502–511, 2000.
21. G. Pollastri, P. Fariselli, R. Casadio, and P. Baldi. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins*, 47:142–235, 2002.
22. N. Biggs. Algebraic graph theory. second edition. 1994.
23. M. Porto, U. Bastolla, H.E. Roman, and M. Vendruscolo. Reconstruction of protein structures from a vectorial representation. *Phys.Rev.Lett.*, 92:218101, 2004.
24. G. Pollastri and P. Baldi. Prediction of contact maps by recurrent neural network architectures and hidden context propagation from all four cardinal corners. *Bioinformatics*, 18, Suppl.1:S62–S70, 2002.

25. L. Holm and C. Sander. Parser for protein folding units. *Proteins*, 19:256–268, 1994.
26. U. Bastolla, M. Porto, H.E. Roman, and M. Vendruscolo. Principal eigenvector of contact matrices and hydrophobicity profiles in proteins. *Proteins: Structure, Function, and Bioinformatics*, 58:22–30, 2005.
27. P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins: Structure, Function and Genetics*, (S5):157–62, 2001.
28. A Vullo, I Walsh, and G Pollastri. A two-stage approach for improved prediction of residue contact maps. *BMC Bioinformatics*, in press.
29. P. Fariselli and R. Casadio. Neural network based predictor of residue contacts in proteins. *Protein Engineering*, 12:15–21, 1999.
30. P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, 14(11):835–439, 2001.
31. P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures – dag-rnns and the protein structure prediction problem. *Journal of Machine Learning Research*, 4(Sep):575–602, 2003.
32. D.A. Debe, M.J. Carlson, J. Sadanobu, S.I. Chan, and W.A. Goddard. Protein fold determination from sparse distance restraints: the restrained generic protein direct monte carlo method. *J. Phys. Chem.*, 103:3001–3008, 1999.
33. A. Aszodi, M. J. Gradwell, and W. R. Taylor. Global fold determination from a small number of distance restraints. *J. Mol. Biol.*, 251:308–326, 1995.
34. E.S. Huang, R. Samudrala, and J.W. Ponder. Ab initio fold prediction of small helical proteins using distance geometry and knowledge-based scoring functions. *J. Mol. Biol.*, 290:267–281, 1999.
35. J. Skolnick, A. Kolinski, and A.R. Ortiz. Monsster: a method for folding globular proteins with a small number of distance restraints. *J. Mol. Biol.*, 265:217–241, 1997.
36. P.M. Bowers, C.E. Strauss, and D. Baker. De novo protein structure determination using sparse nmr data. *J. Biomol. NMR*, 18:311–318, 2000.
37. W. Li, Y. Zhang, D. Kihara, Y.J. Huang, D. Zheng, G.T. Montelione, A. Kolinski, and J. Skolnick. Touchstonex: Protein structure prediction with sparse nmr data. *Proteins: Structure, Function, and Genetics*, 53:290–306, 2003.
38. R.M. McCallum. Striped sheets and protein contact prediction. *Bioinformatics*, 20, Suppl. 1:224–231, 2004.
39. Casp6 home page.
40. V.A. Eyrich, M.A. Marti-Renom, D. Przybylski, M.S. Madhusudan, A. Fiser, F. Pazos, A. Valencia, A. Sali, and B. Rost. Eva: continuous automatic evaluation od protein structure prediction servers. *Bioinformatics*, 17:1242–1251, 2001.
41. G Pollastri, A Vullo, P Frasconi, and P Baldi. Modular dag-rnn architectures for assembling coarse protein structures. *Journal of Computational Biology*, in press.
42. CA Orengo, AD Michie, S Jones, DT Jones, Swindells MB, and Thornton

JM. Cath - a hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997.

43. P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Trans. on Neural Networks*, 9:768–86, 1998.
44. P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937–946, 1999.
45. A Vullo and P Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20(5):653–659, 2004.
46. W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.
47. A Ceroni, P Frasconi, and G Pollastri. Learning protein secondary structure from sequential and relational data. *Neural Networks*, 18(8):1029–39, 2005.