Journal of Mathematical Chemistry 26 (1999) 125–137

A self-assembling neural network for modeling polymers

Robert W. Harrison

Department of Microbiology, Kimmel Cancer Center, Thomas Jefferson University, Philadelphia, PA 19107, USA
E-mail: robert.harrison@acm.org

Received 13 May 1999

A central problem in modeling protein and other polymer structures is the generation of self-avoiding chains which obey *a priori* distance restraint information which could include a folding potential function. This problem is usually addressed with a lattice model or a torsion space model of the polymer. Exhaustive searches in these spaces are of necessity exponentially complex. A new computer algorithm for modeling polymers and polymeric systems is described. This algorithm is a randomized algorithm based on a self-assembling or Kohonen neural network. Given a defined chain topology, a defined spatial extent, and a prior probability distribution, it finds a set of coordinates which reproduce these properties. The convergence rate of the algorithm is linear with respect to the number of distance terms included. Modifications to the standard Kohonen algorithm to include a defined spatial metric, and a modified update rule improve the convergence of the standard algorithm and result in a highly efficient algorithm for building polymer models which are self avoiding and consistent with prior probability information and interatomic distance restraints.

1. Introduction

The computational problem of modeling protein or polymer structure from minimal data is difficult. The trivial algorithm of searching all the free internal torsion angles of the polymer is at best exponential in time and a member of the NP-hard class of problems. When there are sufficient data numerical techniques like distance geometry [5] and embedding or homotopy methods [2,7,8] can be effective. However, these approaches do not work well when the data are sparse or inconsistent. In this case there may not be a unique solution, and using a method which can produce a set of solutions consistent with the prior knowledge will be important. A new algorithm for searching conformational space is described in this paper. This algorithm, based on Kohonen neural networks [11], is efficient and well behaved with both highly determined problems and poorly determined problems.

Polymer modeling requires several assumptions. First, there is a defined chemical structure for the polymer. The chemical structure defines the spatial relationship between atoms on a small scale, but is not sufficient in itself for determining long range or large scale structure. Second, the polymer folds into a compact self-avoiding struc-

ture with a low probability of knots. Some statistical information about this compact structure may be known, such as radius of gyration or degree of crystallinity, and this information can be used to specify a prior distribution for the atomic positions. Third, some large scale information might be known about the structure. This could include quite detailed knowledge such as distance restraints derived from nuclear Overhauser effects (NOE) or known similar structures, but could also include more local data such as predicted secondary structure in proteins. An effective algorithm for modeling polymer structure must combine these sources of information and produce a structure or family of structures which reproduce the prior knowledge.

Kohonen networks work by choosing weights (node or atomic positions) so that the network spans a given spatial range. The networks are trained by trial against randomly chosen points from the spatial range by moving the node closest to the random point towards the random point and moving the neighbors of the node towards the random point as well. These networks are self-assembling because they automatically become ordered during the training phase. The specific choice of random points, the connection between nodes, and the specific update rule depend on the application of the network. For example, the solution to the traveling salesman problem can be approximated by using a closed circular loop for the node topology, choosing the cities as the random points, and using an appropriate update rule [9]. The traveling salesman problem is a classic NP-complete problem, and the performance of Kohonen networks on it shows that they are able to approximate solutions of NP-complete problems. When points are chosen from an area or volume with a uniform distribution and a linear chain is used, the chain fills the area or volume with a space-filling curve similar to a Koch or Peano curve. These curves are reminiscent of the way a polymer folds and suggest a natural isometry between the appropriate Kohonen network and polymer folding.

2. Methods

The algorithms presented below were implemented in AMMP [6] and tested on the systems described in the results section. The routines were written in C and compiled on either Digital Unix, GNUcc or Microsoft C++ for Windows 95/NT for testing on a DEC Alpha workstation, a generic Linux PC, or a Windows PC. The current version of the AMMP potential set, version SP4, was used [14,15]. The SP4 set is derived from the UFF potential set [12]. Distance restraints were implemented with a split biharmonic formalism. The potential was zero for distances between a lower and upper bound, and the deviation from the lower or upper bound had an independent quadratic force constant.

3. Algorithms

For clarity, the standard Kohonen algorithm is presented first, then the algorithm used for including distance restraints is developed and finally the modified Kohonen algorithm is presented.

Standard Kohonen algorithm

The pseudo-code for the standard Kohonen network algorithm is presented below.

Initialization:
  Define a domain (*a region in area or space*)
  Define connections between nodes
  Define an internode weight function (Λ)
  Define a prior distribution for random number choice (*typically uniform*)
  Initialize the random number generator
  Set the node values (*or weights*) *to random values inside the domain*
  Set the relaxation constant k
Iterate until no change:
  Choose a random point in the domain (cp)
  Find the closest node
  Update the node values
    for the closest node
      $x := x + k(\text{cp} - x)$
    for its neighbors
      $x := x + k\Lambda(\text{cp} - x)$
  (*optional*) *reduce the value of* k

Λ is typically a bell-shaped function of the lexical distance between nodes. For example, in a one-dimensional network the nodes adjacent to node j, $(j+1, j-1)$, could be updated with weight $0.5k$ and the other nodes are not changed. In a two-dimensional network the neighbors of node (j,k) $(j \pm 1, k)$ and $(j, k \pm 1)$ would be updated.

This algorithm readily converges for linear chain topology [11,13]. However, when the chain topology is higher dimensional (i.e., 2D, 3D or fractal dimension), convergence can fail and result in a knotted solution. The standard algorithm cannot include information about distances between nodes; it can only include the information that nodes are linked together. This causes the standard algorithm to have both poor convergence to a desired spatial structure (i.e., the bond lengths are wildly inaccurate) and to fail to converge if the polymer model is higher than linear dimension (i.e., the structure is knotted). A higher than linear dimension (a fractal dimension) occurs when distance restraint information relates one part of a chain to another as in the use of NOE distance restraints to determine a protein structure from NMR data. These drawbacks are addressed with a modified update rule. However, in order to present the modified update rule, a graph relaxation algorithm for distance range constraints must be developed.

Graph relaxation algorithm for distance range constraints

A system of distance range constraints is a set of linear inequalities of the form

$$|x_i - x_j| \leqslant D_{ij},$$

where x_i, x_j are coordinates and D_{ij} is the distance between them. This linear system can be re-notated as a constraint graph [4] as long as the distances obey the triangle inequality. The coordinates (x) are the minimum path length to each vertex of the constraint graph. These can be found by the Bellman–Ford algorithm provided the constraint graph has no negative weight cycle (i.e., no combination of the D_{ij} is less than zero). The Bellman–Ford algorithm uses relaxation to find the solution. Relaxation examines each pair of vertexes and chooses the best local solution.

In the model building version of the Bellman–Ford algorithm the graph consists of the known distances between atoms due to bond lengths, covalent angles, van der Waals exclusions, and prior distance restraints. The relaxation step consists of looping through the atoms and for each atom moving all the other atoms to satisfy the distance terms. A damped relaxation was found to be more stable than full relaxation because the equations are vector equations rather than scalar.

Pseudo-code for the Bellman–Ford relaxation algorithm

Initialization:
  Define the distances to be met
  Define a damping constant K (*typically* 0.5 *when relaxation is used on its own*)
  Choose a "random" starting model
Iterate:
  For each atom in the model (x_i)
    find the other atoms in the model which have distance constraints
    if (*nonbonded* $<$ 4 Å *or bond, angle, distance restraint* $>$ *target value*) *then*
    move the atoms along the vector between them to satisfy the constraints
      Set the other atoms to obey any chirality terms
      $x_{\text{moved}} := x_{\text{old}} + K \cdot \text{error\_in\_target} \cdot (x_{\text{old}} - x_i)/|x_{\text{old}} - x_i|$

The iterations are repeated until the structure no longer shifts. This algorithm is much faster than the previously reported hybrid Krylov algorithm [7] because there are no line searches. However, run on its own it will produce a distorted structure and requires further optimization to produce a high quality model. The Bellman–Ford algorithm will build models from sets of distance constraints, but does not converge well from all possible starting points. There are some especially bad starting points, such as all atoms equal to zero or in a line, where the Bellman–Ford algorithm will converge very slowly. It is also difficult to add prior information such as a radius of gyration to the algorithm. The algorithm is linear in complexity when used with bond, angle, and distance restraints, but the algorithm will not produce a self-avoiding chain without checking all pairs of atoms. The algorithm is quadratic in complexity when finding a self-avoiding polymer chain because all pairs of atoms must be checked for self-avoidance. This can make it slow when there are many atoms. However, the relaxation algorithm used in the iteration forms the basis for the modified update step which makes the Kohonen network algorithm practical.

Modified Kohonen algorithm

The modified Kohonen algorithm combines the standard Kohonen algorithm with the relaxation algorithm and gradient tracking to produce an algorithm which has defined spatial metric as well as a topological connectivity. The internode weight function (Λ) is transformed to enforce the spatial metric.

The pseudo-code for the modified Kohonen network algorithm is presented below.

Initialization:
  Define a domain (*a region in area or space*)
  Define connections between nodes
  Define a prior distribution for random number choice (*typically uniform*)
  Initialize the random number generator
  Set the node values (*or weights*) *to random values inside the domain*
  Set the relaxation constant k
Iterate until no change:
  Choose a random point in the domain
  Find the closest node (cp)
  Update the node values
    for the closest node
      $x := x + k(\text{cp} - x)$
    for its neighbors
      perform a relaxation step on the bond and distance terms from the closest node
      constrain the locations of the relaxed atoms
      perform one step of Steepest descent minimization on the whole structure

This algorithm converges to a structure which mostly has correct geometry and satisfies the distance data. It is generally sufficient to use bond, angle, chiral, and distance restraint terms during network training and to allow the nonbonded terms to be treated by the convergence of the network to a uniform density of atoms. In the absence of nonbonded potentials this algorithm is linear in complexity with respect to the number of atoms which makes it quite efficient. If the nonbonded terms are included, the algorithm is quadratic in complexity. When interrupted in mid-training, parts of the geometry will be distorted so that further energy minimization is required. This is performed with gradient optimization and limited use of molecular dynamics.

3.1. Least squares solution of distance terms

An alternative to the Bellman–Ford relaxation step is to use an analytic solution for the positions of the local atoms. When there are more than four independent distances from an atom to atoms of known position, there exists a unique least square error position for the atom. Structures can be modeled by successively solving the least square equations for each atom, in a manner which is exactly analogous to the Gauss–Seidel solution of linear systems of equations. While this local solution can be found numerically, it can also be determined by analytic means. Analytic techniques

to find the least square error position can also be used for the relaxation step in the Kohonen algorithm. For each distance the quadratic equation

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2,$$

where x, y, z are the coordinates, x_0, y_0, z_0 the fixed origin, and r the distance, can be written. Expanding the quadratic terms and grouping common powers results in the following linear equation in x^2, y^2, z^2, x, y, z:

$$\begin{pmatrix} 1 & 1 & 1 & -2x_0 & -2y_0 & -2z_0 \\ 1 & 1 & 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & 1 & 1 & -2x_2 & -2y_2 & -2z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & -2x_n & -2y_n & -2z_n \end{pmatrix} \begin{pmatrix} x^2 \\ y^2 \\ z^2 \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -r_0^2 \\ -r_1^2 \\ -r_2^2 \\ \vdots \\ -r_n^2 \end{pmatrix}.$$

Eliminating the x^2, y^2, z^2 terms from $n - 1$ equations

$$\begin{pmatrix} 1 & 1 & 1 & -2x_0 & -2y_0 & -2z_0 \\ 0 & 0 & 0 & 2x_0 - 2x_1 & 2y_0 - 2y_1 & 2z_0 - 2z_1 \\ 0 & 0 & 0 & 2x_0 - 2x_2 & 2y_0 - 2y_2 & 2z_0 - 2z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 2x_0 - 2x_n & 2y_0 - 2y_n & 2z_0 - 2z_n \end{pmatrix} \begin{pmatrix} x^2 \\ y^2 \\ z^2 \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -r_0^2 \\ r_0^2 - r_1^2 \\ r_0^2 - r_2^2 \\ \vdots \\ r_0^2 - r_n^2 \end{pmatrix}$$

results in the $n - 1$ degree linear equations

$$\begin{pmatrix} 2x_0 - 2x_1 & 2y_0 - 2y_1 & 2z_0 - 2z_1 \\ 2x_0 - 2x_2 & 2y_0 - 2y_2 & 2z_0 - 2z_2 \\ \vdots & \vdots & \vdots \\ 2x_0 - 2x_n & 2y_0 - 2y_n & 2z_0 - 2z_n \end{pmatrix} \begin{pmatrix} x \\ y \\ x \end{pmatrix} = \begin{pmatrix} r_0^2 - r_1^2 \\ r_0^2 - r_2^2 \\ \vdots \\ r_0^2 - r_n^2 \end{pmatrix}$$

or

$$Ax = b.$$

These can be solved by least squares, with simplest approach being the construction of the normal equations ($A^{\mathrm{T}}Ax = a^{\mathrm{T}}b$, where $^{\mathrm{T}}$ indicates matrix transpose), followed by the Moore–Penrose inverse. (Since the rank of the normal equation is at most three, the use of highly sophisticated methods like the singular value decomposition is somewhat unnecessary.) Operationally it is useful to "polish" the solution with a small amount of numerical optimization, since the simultaneous solution of the distance terms is sometimes not possible. This operation is exactly analogous to iterative improvement of the solution of linear systems of equations.

It is possible to construct an overdetermined problem. For example, in homology modeling distance restraints may be derived for many more pairs of atoms than are required to determine the structure. The Bellman–Ford relaxation algorithm can perform badly when this is the case, because it may be unable to sort out a local neighborhood

and relax the model to an unknotted form. In this case the least squares solution will out-perform the simple Bellman–Ford relaxation. However, the least squares algorithm on its own, like the Bellman–Ford relaxation step on its own, does not efficiently solve distance restraint problems akin to those presented in this paper. For example, with G protein, the least squares algorithm converges more slowly and results in poorer quality geometry.

3.2. Energy minimization

After building the models with the modified Kohonen network, the structures were further refined to produce the final models. A minimal annealing procedure was used where energy minimization combined with relatively short runs of molecular dynamics at moderate temperatures.

4. Results

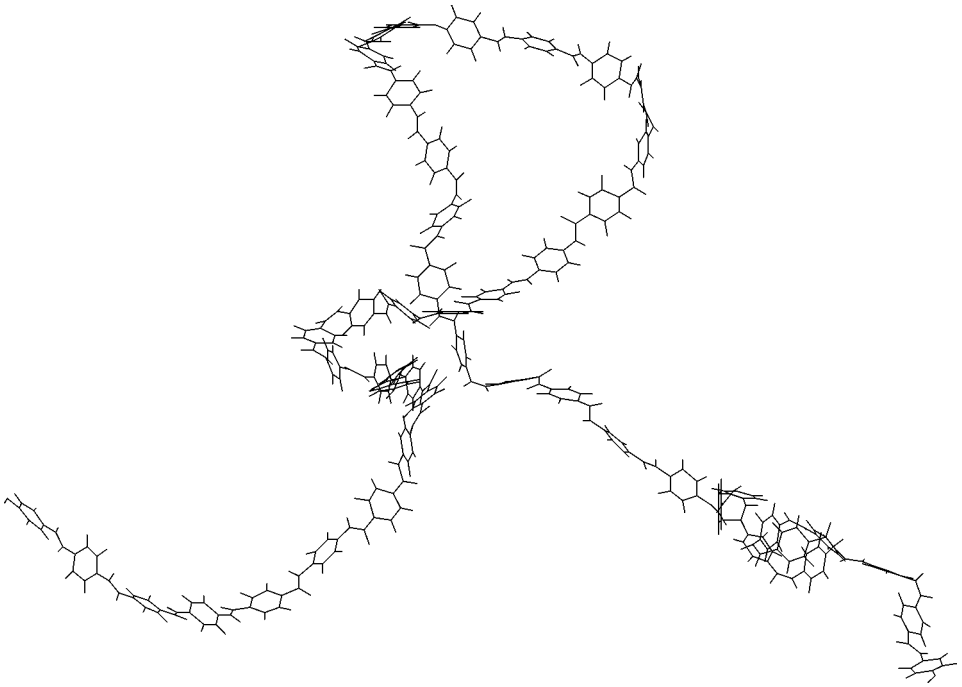
Several examples will be used in order to demonstrate the utility of the modified Kohonen neural net.

4.1. Kevlar

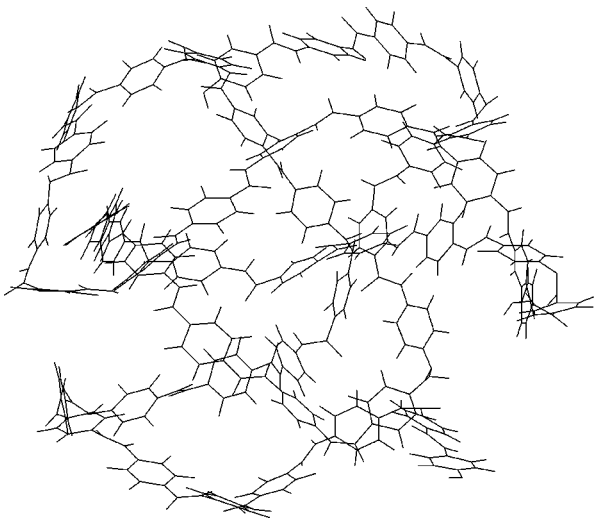
Kevlar is a synthetic polymer made from repeating units of p-dibenzoic acid and p-diamino-benzene. This nonbiological example is included to demonstrate how neural networks can be used to study the self-assembly of simple polymers from a description of the local chemical structure. In complex biological polymers, this description is generally not very interesting, but in synthetic and simple polymers the statistical description of polymer structure rather than the structure of any individual conformer is the essential question. The effect of choosing a larger prior distribution is shown in figures 1(a) and (b) where uniform distributions with a 70 Å radius and a 15 Å radius were used during the training. This allowed the generation of extended and compact self-avoiding nonknotted polymer chains. Changing the parameters of the prior distribution, or using a series of different seeds for the random number generation will result in a family of related structures.

4.2. Naphthalene

In addition to the study of polymers the self association of monomeric molecules can be studied with the Kohonen network. The algorithm is useful for studying molecular assemblies without making prior assumptions of crystallinity or long-range order. Figure 2 shows a 180 Å “wire” made of naphthalene with this algorithm. Face-to-face packing is seen in the “wire” and in larger bulk models. This approach can be used to study the onset of order in quasi-crystalline systems.



(a)



(b)

Figure 1. (a) An extended conformation of a kevlar 24-mer. It was generated by the Kohonen algorithm with a large prior distribution. (b) A compact self-avoiding nonknotted conformation of the kevlar 24-mer. Generating compact self-avoiding conformations is a more difficult problem than generating extended conformations. The Kohonen algorithm has converged to one. These figures show the algorithm converges to self-avoiding chains *without* the use of explicit nonbonded terms in the potential energy function.

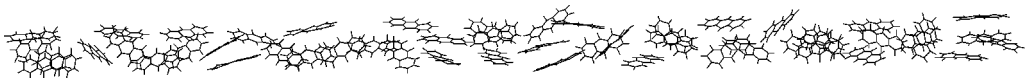


Figure 2. A 180 Å "wire" of naphthalene is shown. It was generated with a long cylindrical prior and shows typical face-to-face packing. Similar self-associations are seen when a compact prior is used.

4.3. *Cellulose fiber bundles*

The construction of models for amorphous quasi-ordered bundles of self-avoiding, self-organizing polymers is a difficult task since it is not only necessary to be self-avoiding, but to also avoid other fibers in the bundle. The Kohonen algorithm is well suited to this task. To show this cellulose bundles were modeled from a description of the covalent geometry and the constraints of being an extended chain with regular conformation. Bundles of 10 20-mers of 1′,4′-polyglucose were generated for two different bundle geometries. The 20-mers were forced to be extended with a 100 Å distance restraint between the initial O1 atom and the final O4 atom. The conformational regularity was enforced with a variation of the swarm algorithm of Huber and van Gunsteren [10] where the O6–O6($I + 1$), O6($I + 1$)–O6($I + 2$) distances were restrained to be the same as the average values in all the sugars. The O6–O6($I + 2$), O6($I + 2$)–O6($I + 4$) distances were also restrained.

Because the model required the alignment of highly asymmetric objects, the calculation was performed by first building a self-avoiding model in a large volume and then shrinking that volume down to the final one. This produces a series of models which mimic condensation to an ordered state. Pictures of the compact and extended fibrils are shown in figure 3. Regular, helical chains bundled into an ordered fibril are produced for both staggered and unstaggered geometries. The helicity and regular structure are a consequence of the local geometry of the sugars.

4.4. *G protein from experimental NOE data*

Proteins are a particularly important class of polymers. Unlike simple polymers, the twenty different monomers (amino acids) in proteins force the proteins to fold into highly specific structures. Often protein structure is solved from distance data derived from NMR, and the Kohonen algorithm should converge from this data. To demonstrate convergence, G protein (pdb1pga.ent) data were taken from the benchmark set which is supplied with XPLOR [1,3]. The data were divided into short-range restraints which related adjacent residues and long-range restraints which related more distant residues. The algorithm readily converges when supplied with complete data.

An interesting quirk was observed when modeling protein structures. As reported previously [8], the distance restraints are unique up to the definition of a hand. Distance restraints can form either a left-handed or a right-handed system, and from the viewpoint of the distances both solutions are correct. Therefore, the algorithm can converge to either a left- or right-handed solution. Two strategies were evaluated for this problem. These strategies were to either find a right-handed random structure and

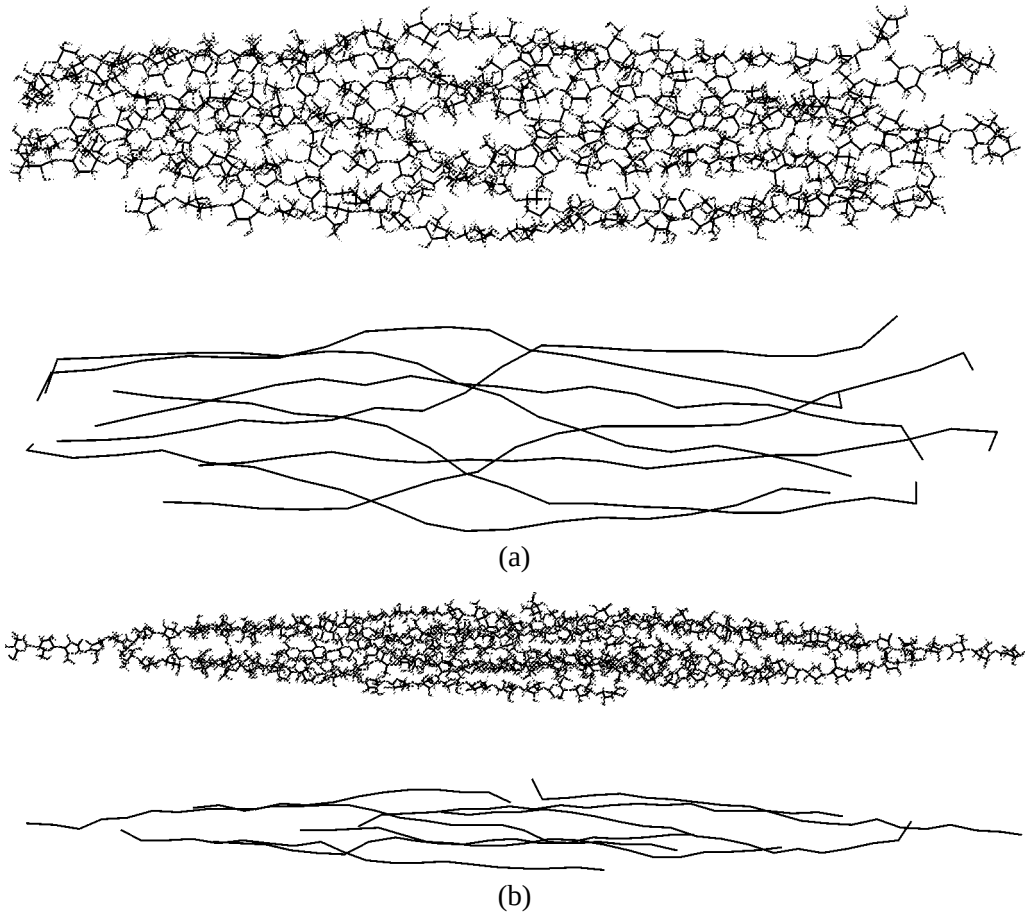


Figure 3. A compact (a) and an extended (b) cellulose fibril generated with the Kohonen algorithm. For visual clarity both the all-atom and C1 trace structures are shown. The models have helically intertwined strands of twenty-long poly-1′,4′-glucose sugars. This shows that the Kohonen algorithm cannot only generate single self-avoiding chains, but multiple self-avoiding chains and ensembles of the chains which obey geometric prior information. The helicity and and approximate internal symmetry is a consequence of the molecular geometry and not a directly imposed restrain on the system.

change it to meet the distance restraints or to meet the distance restraints and then determine which hand of solution was found. With geometrically complete data, such as high quality NOE data, the strategy of meeting the distance restraints first was found to be faster. Therefore, for this test two passes of neural net training were performed. The first found an approximate solution to the distance data for a racemic mix of amino acids, and the second used that solution after the chirality was evaluated and the hand of the structure inverted if needed.

The distance restraints were introduced with an arbitrary force constant of 100 kcal/mol/Å^2. The final structure had no serious violations of the distance terms with a total energy of 80 kcal/mol for 922 terms. Therefore, as far as the experimental

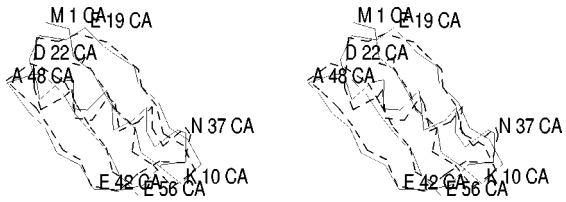


Figure 4. The structure of G protein (pdb1pga.ent) and the structure found by the Kohonen algorithm with NMR distance restraints (dashed line).

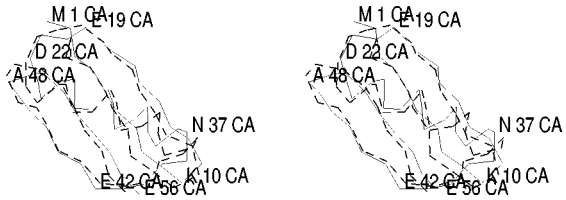


Figure 5. The structure of G protein and the structure found by the Kohonen algorithm with NMR long-range and synthetic short-range restraints (dashed line).

data are concerned, the resulting structure is correct. The structure (figure 4) has an RMSD on α-carbons with respect to pdb1pga.ent of 1.76 Å. A large part of this error is a misplaced N-terminus (residue 1 is in error by 8 Å, residue 2 by 4 Å, and residue 3 by 0.5 Å). When two residues from the N-terminus are excluded, the RMSD drops to 1.168 Å which implies that the experimental structure has been recovered.

4.5. G protein from long-range NOE data and synthetic secondary structure data

Knowledge of the secondary structure alone was not sufficient for the algorithm to converge to a recognizably correct structure. The long-range NOE data alone were not sufficient to produce an accurate local structure (RMSD 3.10 Å). However, when combined with long-range experimental data, enforcing secondary structure via "synthetic" distance restraints was sufficient to solve the structure. The same script used for the all-experimental data example was used, with the file names and distance restraints changed. The RMSD against pdb1pga.ent is 1.85 Å with most of the error in the N-terminal two residues (figure 5).

4.6. Efficiency

The Kohonen neural network algorithm is efficient. This can be seen from both a theoretical viewpoint, discussed below, and from the CPU times required. The G protein example required approximately 30 minutes on a 233 Mhz Pentium II processor under Windows NT4.0. The four-dimensional embedding and related homotopy algorithms we had implemented earlier [7,8] require several hours (6–12; the exact time depends critically on the schedule for removing the fourth dimension or relaxing the

homotopy) for the same run and do not converge to as good a solution as the neural network.

Theoretical analysis of the complexity of the algorithms shows why it is efficient. The algorithm scales linearly with the number of trial points and the cost of updating the local atomic positions (i.e., the internode weight function Λ). That is, if N_p random points are chosen consistent with the prior distribution, then N_p evaluations of Λ are required. The bond, angle, chirality, torsion, and distance restraint terms all scale linearly with the number of atoms N_a. Since the algorithm determines a self-avoiding structure without the use of superlinear complexity terms like the nonbonded potentials, they can be ignored until required for the final optimization of the structure. This results in an overall complexity of $O(N_p N_a)$ which is linear in both atom number and number of random trials. At worst, if all pairs of atoms are required for the internode weight function, the complexity of the algorithm would be quadratic in the number of atoms, which compares well with the exponential complexity required for a direct search algorithm and is the same complexity as gradient optimization or molecular dynamics on the whole potential set.

4.7. Summary: Unanswered questions and future directions

The ability of the algorithm to converge from sparse or minimal data was demonstrated. The absolute convergence to a unique structure from any sparse data set was not. It is still necessary to define the appropriate geometric conditions that are required for a data set to be uniquely convergent. By changing the seed of the pseudorandom number generator a family or ensemble of structures can be generated if the data are not sufficient to uniquely determine the structure.

The algorithm can be applied to problems in molecular modeling where distance information and prior probability distributions are known. While it is well adapted to polymeric problems, it readily handles systems of independent molecules. The application of this algorithm to homology modeling and *ab initio* protein folding is being developed.

Acknowledgements

This work was partially supported by National Institutes of Health AI/GM41380 and the Elsa U. Pardee foundation.

References

[1] A. Achari, S.P. Hale, A.J. Howard, G.M. Clore, A.M. Gronenborn, K.D. Hardman and M. Whitlow, 1.67 Angstroms X-ray structure of the B2 immunoglobin binding domain of streptococal protein G and comparison to the NMR structure of the B1 domain, Biochem. 31 (1992) 10449.

[2] T.C. Beulter and W.F. van Gunsteren, Molecular dynamics free energy calculation in four dimensions, J. Chem. Phys. 101 (1994) 1417–1422.

[3] A.T. Brünger, G.M. Clore, A.M. Gronenborn and M. Karplus, Three-dimensional structure determination of proteins by molecular dynamics with interproton distance restraints: Application to Crambin, Proc. Natl. Acad. Sci. USA 83 (1986) 3801–3805.
[4] T.H. Cormen, C.E. Leierson and R.L. Rivest, *Introduction to Algorithms* (McGraw Hill, New York, 1990) pp. 540–543.
[5] J. De Vlieg and W.F. van Gunsteren, Combined procedures of distance geometry and molecular dynamics for determining protein structure from nuclear magnetic resonance data, Methods Enzymol. 202A (1991) 268–300.
[6] R.W. Harrison, Stiffness and energy conservation in molecular dynamics: an improved integrator, J. Comp. Chem. 14 (1993) 1112–1122.
[7] R.W. Harrison, D. Chatterjee and I.T. Weber, Analysis of six protein structures predicted by comparative modeling techniques, Proteins: Struct. Funct. Genet. 23 (1995) 463–471.
[8] R.W. Harrison, C.C. Reed and I.T. Weber, Analysis of comparative modeling predictions for CASP2 targets 1, 3, 9 and 17, Proteins: Struct. Funct. Genet. Suppl. 1 (1997) 68–73.
[9] J. Hertz, A. Krogh and R.G. Palmer, Introduction to the theory of neural computation, in: *Santa Fe Institute Studies in Complexity Lecture Notes*, Vol. 1 (Addison-Wesley, Redwood City, CA, 1991) pp. 244–246.
[10] T. Huber and W.F. van Gunsteren, SWARM-MD: Searching conformational space by cooperative molecular dynamics, J. Phys. Chem. A102 (1998) 5937–5943.
[11] T. Kohonen, *Self-Organizing Maps*, 2nd ed. (Springer-Verlag, Berlin, 1997).
[12] A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddard III and W.M. Skiff, UFF a full periodic table force field for molecular mechanics and molecular dynamics simulations, J. Am. Chem. Soc. 114 (1992) 10024–10035.
[13] R. Rojas, *Neural Networks* (Springer-Verlag, Berlin, 1996) pp. 393–405.
[14] I.T. Weber and R.W. Harrison, Molecular mechanics calculations on HIV-1 protease with peptide substrates correlate with experimental data, Protein Engrg. 9 (1996) 679–690.
[15] I.T. Weber and R.W. Harrison, Molecular mechanics calculations on Rous sarcoma virus protease with peptide substrates, Prot. Sci. 6 (1997) 2365–2374.