Case Study: Protein Folding using Homotopy Methods

james robert white

Since the late 1990s, the world of genomics has exploded. Beginning with the bacterium *Haemophilus influenzae* in 1995, researchers have sequenced the genomes of thousands of different organisms, which include dog, rat, mouse, and two versions of human.

The central dogma of molecular biology is that DNA is transcribed to RNA which is then translated to an amino acid sequence or *protein*. As the protein is translated, it folds into a distinct shape that determines its function. Under stable conditions, a protein will always fold the same way because its folded shape has minimum *free energy*.

A mutation in a gene at the DNA level has the potential to change the amino acid composition of the resulting protein. This new protein may not fold in the same way as the original, and the function of the protein may be destroyed. Many diseases are linked to protein misfolding such as: Alzheimer's, Huntington's, Parkinson's, and several types of cancer.

The pharmaceutical industry depends on knowing the structure of a folded protein. After determining the structure, they can create drugs to interact with a specific region of the protein. Current methods such as X-ray crystallography are tremendously expensive and time-consuming. It can take years to decipher the structure of a single protein.

In an effort to accelerate the pace of research, several groups are working to develop computational methods to determine the three-dimensional structure of a protein from its amino acid sequence. No sufficiently accurate methods have been developed yet, but databases of known structures are helping researchers gain insight into the major features of new proteins. The intuition is, if you know the structure of one protein, and you want to know how a similar protein folds, then use the structure you know as a starting guess.

This paradigm of computational protein folding fits the concept of homotopy very well. In this case study, we will use the Homotopy Optimization Method (HOM) to predict how chains of particles fold in two dimensions. In addition, we shall implement a new extension of HOM known as *Homotopy Optimization using Perturbations and Ensembles* or HOPE. The HOPE algorithm was developed by Daniel Dunlavy at the University of Maryland, and this case study parallels the experiments in his thesis. See pointer for additional information.

The Free Energy of 2-D Charged Particle Chains

Suppose that we have a chain of n charged particles in two dimensions. The charges on each particle can be +1 or -1 and each particle in the chain has a fixed bond length of $\overline{r}$. Then, the angles between all pairs of adjacent bonds define the structure of the chain. Let $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in \mathcal{R}^{2n}$ be the vector of coordinates in the xy plane with $\mathbf{x}_k \in \mathcal{R}^2$ representing the two-dimensional coordinates of the k^{th} particle. For example, the five particles $\mathbf{x}_1, ..., \mathbf{x}_5$, are represented in the following figure:

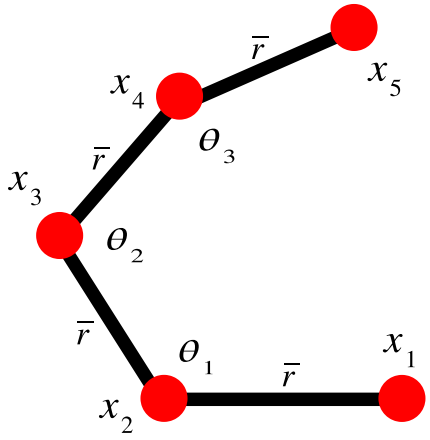

Figure 1: An example of particle structure with bond angles.

Let the first particle in the chain be fixed at ($\overline{r}$, 0) and the second particle be fixed at (0,0). Therefore, given an $n-2$ vector of angles, $\theta = (\theta_1, ..., \theta_{n-2})$, where $\theta_i \in [0, 2\pi)$ represents the bond angle between $\mathbf{x}_i$, $\mathbf{x}_{i+1}$, and $\mathbf{x}_{i+2}$, we can use the following recursion to compute the positions of every particle in the chain.

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{pmatrix} \cos(\theta_{k-2}) & -\sin(\theta_{k-2}) \\ \sin(\theta_{k-2}) & \cos(\theta_{k-2}) \end{pmatrix} (\mathbf{x}_{k-2} - \mathbf{x}_{k-1})$$

CHALLENGE 1 *To familiarize yourself with the recursion, write a function in* `MATLAB` *that plots the coordinates of a particle chain given a vector of bond angles. Let* $\overline{r} = 1.5$. *Recall that we fix the first and second particles at* ($\overline{r}$,0) *and* (0,0), *respectively. To get a balanced view of the chain, use* `axis equal` *when plotting. Plot the coordinates of the chain for the following input bond angle vectors:* θ_0 = `[pi/2; 7*pi/8; 7*pi/8]`, `[pi; pi/2; pi/3; pi/4; pi/5]`, *and* `[4*pi/5; 7*pi/5; 4*pi/5; 7*pi/5; 4*pi/5; 7*pi/5]`.

We shall declare the distance between two particles $\mathbf{x}_i$ and $\mathbf{x}_j$ to be $r_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||_2$. Now that we have a way to construct the chain's position from its

internal bond angles, we may define the potential energy function, $E : \mathcal{R}^{n-2} \rightarrow \mathcal{R}$ as:

$$E(\theta) = E_{vdw}(\theta) + E_{el}(\theta),$$

where

$$E_{vdw}(\theta) = \sum_{i=1}^{n-3} \sum_{j=i+3}^{n} \epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - 2 \left(\frac{\sigma}{r_{ij}} \right)^{6} \right),$$

and

$$E_{el}(\theta) = \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} \frac{q_i q_j}{r_{ij}}.$$

$E_{vdw}(\theta)$ is the van der Waals equation, which is the sum of all interactions between particles along the chain. $E_{el}(\theta)$ is the equation for electrostatic potential, and q_i equals the charge of particle $\mathbf{x}_i$ (+1 or -1). Note that although θ is the input of the energy equation, it is nowhere to be found on the right side. We must use the recursion to calculate the coordinates given the vector of bond angles in order to solve this equation.

Figure 2 shows a chain of 40 particles of alternating charges which has a very high energy value. A natural chain would quickly unfold from this state because it seeks its minimum free energy.

So, given a chain of charged particles, we are trying to find $\theta_{opt} \in \mathcal{R}^{n-2}$ such that

$$E(\theta_{opt}) = \min_{\theta} E(\theta).$$

CHALLENGE 2 *Use the code from the previous challenge to write a function which calculates $E(\theta)$ of a chain of particles assuming all particles are positively charged. Let eplison = 0.4 and sigma = 3.6 in the van der Waals equation. Find the energy of a chain of particles with each of the following starting values: θ_0 =* `[pi/2; 7*pi/8; 7*pi/8]`, `[pi; pi/2; pi/3; pi/4; pi/5]`, *and* `[4*pi/5; 7*pi/5; 4*pi/5; 7*pi/5; 4*pi/5; 7*pi/5]`.

Defining a Homotopy

To design a proper homotopy for the free energy function, we must think about which terms are actually changing in the equations as we transition from the equation for our initial chain, $E^0(\theta)$, to our final chain, $E^1(\theta)$. We assume that the initial and final chains both have the same number of particles. Therefore, the only thing that changes from one molecule to the next are the charges on each particle. This means we do not need to define the homotopy with respect to $E_{vdw}(\theta)$. We are only concerned with deforming the initial charges q^0 to the final charges q^1. For clarity, we consider q_k^0 to be the charge of the k^{th} particle in the initial chain.

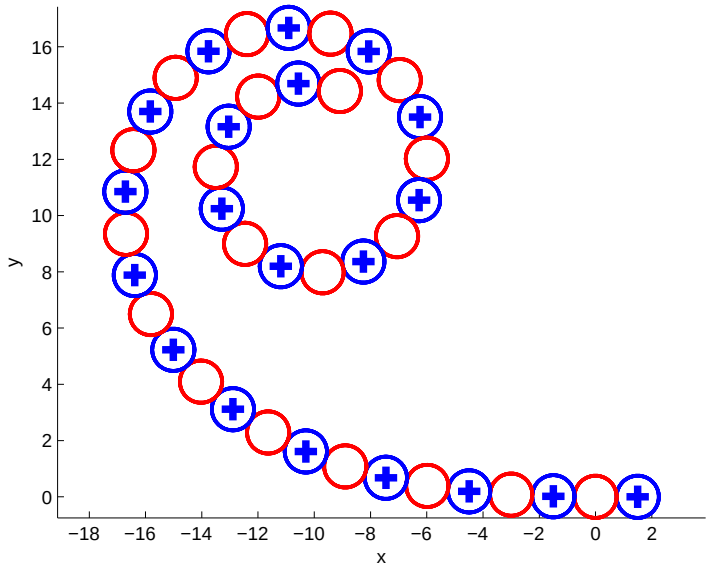

Figure 2: A chain of particles organized in a structure with energy 3.6320e+03. This is nowhere near the natural state of the chain, it just looks cool.

Thus, we define the homotopy function as:

$$H(\theta, \lambda) = \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} \frac{(\lambda q_i^1 + (1-\lambda) q_i^0)(\lambda q_j^1 + (1-\lambda) q_j^0)}{r_{ij}} + E_{vdw}(\theta)$$

The Homotopy Optimization Method (HOM)

Recall in Chapter 24, we discussed the homotopy method for solving nonlinear equations. The following variation allows us to use the homotopy function to solve an unconstrained minimization problem. To illustrate the application to our problem, we use our homotopy equation in the description.

Algorithm 1 HOM Algorithm

Begin with a homotopy function $H(\theta, \lambda)$, and a local minimizer of $H(\theta, 0)$, θ^0.

Set $\lambda = 0$, and let m be some integer such that $m \geq 1$.

for $k = 1, ..., m$

Increase λ by $1/m$.

Using your favorite algorithm, minimize $H(\theta, \lambda)$ starting with $\theta^{(k-1)}$, obtaining $\theta^{(k)}$.

end

At the end, we have computed $\theta^{(m)}$ which minimizes $H(\theta, 1)$, so $\theta^{(m)}$ minimizes our original problem.

CHALLENGE 3 *For the free energy equation, let $\sigma = 3.6$, $\epsilon = 0.4$, and $\overline{r} = 1.5$.*
(a) Use the Quasi-Newton BFGS method (`fminunc`) to find a global minimizer of the energy of the molecule: q^0 = [+1 +1 +1 +1 +1 +1]. Turn off the 'LargeScale' option for (`fminunc`). The minimum energy of this chain is 0.7741. Give an example of a starting value for which Quasi-Newton does not converge to a global minimizer, and one example for which it does converge to a global minimizer.
(b) Implement the HOM algorithm to solve for the minimum energy of the molecule q^1 = [-1 -1 +1 +1 +1 +1] using the global minimizer of part (a) as your starting value and q^0 as your starting molecule. Use Quasi-Newton BFGS for minimization. Let m = 10, so the method takes 10 steps. After finishing the algorithm, plot (side-by-side) the initial molecule at its global minimizer, and the final molecule at its minimizer. Mark negatively charged particles with a red circle, and positively charged particles with a blue circle with a '+' in the center. At the top of the plots, give the minimum energy found for that chain.
Do the same starting with the non-global minimizer from (a).
The global minimum energy of q^1 is -3.4178. Did you find a global minimizer in both cases?

The HOPE Method

Despite the improvement of HOM over standard Newton-like methods, we usually seek the global minimizer of the problem. HOM effectively creates a set of points which converge to a local minimizer, but if it is not the global minimizer, then the structure of the chain of particles will not be correct.

In an effort to improve the probability of finding the true global minimizer, there exists an extension of HOM that generates an *ensemble* of paths which converge to different local minimizers simultaneously. The HOPE algorithm takes a set of minimizers found from a previous step, and creates several perturbations of each minimizer. Then, it minimizes the problem starting at each of these perturbed versions and selects a set of points that generate the lowest *unique* values of the function.

For the perturbation of each minimizer in our problem, we denote a function $\xi : \mathcal{R}^N \to \mathcal{R}^N$, where $\xi(\theta)$ is a stochastically perturbed version of the minimizer θ. As shown below, there are a few additional variables that must be considered in the HOPE algorithm. c_{max} is an integer describing the maximum size of an ensemble allowed for each step. This is important because if we had no upper bound on ensemble size, then it would increase exponentially. $\hat{c}$ is the number of perturbed versions of a minimizer to create for each step.

Algorithm 2 HOPE Algorithm

Begin with a homotopy function $H(\theta, \lambda)$, and a local minimizer of $H(\theta, 0)$, θ^0.
Set $\lambda = 0$, $c^{(0)} = 0$, and let m, c_{max}, and $\hat{c}$ be integers such that $m \geq 1$, $c_{max} \geq 1$, and $\hat{c} \geq 0$.
for $k = 1, ..., m$
  Increase λ by $1/m$.
  for $i = 1, ..., c^{(k-1)}$
    Minimize $H(\theta, \lambda)$ starting with $\theta_i^{(k-1)}$, obtaining $\theta_{i,0}^{(k)}$.
    if $\hat{c} > 0$ **then**
      for $j = 1, ..., \hat{c}$
        Minimize $H(\theta, \lambda)$ starting with $\xi(\theta_i^{(k-1)})$, obtaining $\theta_{i,j}^{(k)}$.
      end
    end
  end
  $c^{(k)} = \min\{c^{(k-1)}(\hat{c}+1), c_{max}\}$
  $\theta_1^{(k)}, ..., \theta_{c^{(k)}}^{(k)}$ = the "best" (unique) local minimizers out of $\theta_{i,j}^{(k)}, i = 1, ..., c^{(k-1)}, j = 0, ..., \hat{c}$
end
At the end, we choose the minimizer with the lowest function value of $\theta_i^{(m)}, i = 1, ..., c^{(m)}$ to be our output.

By creating an ensemble of different minimizers, the HOPE algorithm increases the likelihood of finding the true global minimizer. This method is very effective for problems that have many local minima, but due to the number of minimization calls, there may be a strain on computational resources. Tuning the variables appropriately for each situation is essential.

CHALLENGE 4 *(a) For the HOPE algorithm, if c_{max} was ∞, what would be the value of c after the k^{th} iteration of the program? What values of $\hat{c}$ and c_{max} reduce the HOPE algorithm to HOM?*
(b) Consider our problem of folding, and choose a stochastic permutation function which will be used to generate perturbed versions of minimizers found in previous steps of HOPE. Discuss why you chose this function over other possibilities.
(c) Implement the HOPE algorithm with your permutation function from (b). Let $m = 5$, $\hat{c} = 3$, and $c_{max} = 6$. Begin by finding a local minimizer of the energy for:
*q^0 = [+1 +1 +1 +1 +1 +1 +1 +1 +1 +1 +1 +1] given a starting guess of θ_0 = [9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10; 9*pi/10]. Then, use HOPE to find a minimizer for the energy of*
q^1 = [+1 +1 +1 -1 -1 -1 +1 +1 +1 -1 -1 -1].
Turn off the 'LargeScale' option for (`fminunc`). After finishing the algorithm, plot (side-by-side) the initial molecule at its local minimizer, and the final molecule at its minimizer. Mark negatively charged particles with a red circle, and positively charged particles with a blue circle with a '+' in the center. At the top of the plots, give the minimum energy found for that chain.
(d) Using the same input parameters as in part (c), compare your results with the HOM function from the previous challenge. Which method resulted in the better structure prediction? Can HOPE ever do worse than HOM? Why?

POINTER.
The HOPE method was developed at the University of Maryland by Dunlavy in 2005.
For extensive results on 2-D folding of charged particles, see
Dulavy, D. (2005) Homotopy Optimization Methods and Protein Structure Prediction. PhD thesis - University of Maryland - College Park.
For information on the extension of HOPE to 3-D advanced protein folding problems, see
Dunlavy, D. *et al.* (2005) HOPE: A Homotopy Optimization Method for Protein Structure Prediction. *Journal of Computational Biology*, **12**, 1275-1288.
For other information on computational protein folding:
Li, H. *et al.* (1996) Emergence of Preferred Structures in a Simple Model of Protein Folding. *Science* **273**, 666-669.
Radford, S. and C. Dobson. (1999) From Computer Simulations to Human Disease: Emerging Themes in Protein Folding. *Cell*, **97**, 291-298.
and http://folding.stanford.edu/.
The Critical Assessment of Techniques for Protein Structure Prediction (CASP) is a competition held every two years in which groups try to predict a structure of a new protein whose true folding shape has been secretly and experimentally determined. See http://predictioncenter.org/casp7/