FoldSynth: A physics-based interactive visualisation platform for proteins and other molecular strands

P. Todd Goldsmiths
S. Todd Goldsmiths
F. Fol Leymarie* Goldsmiths
W. Latham Goldsmiths
B. Jefferys UCL
L. Kelley Imperial

ABSTRACT

FoldSynth is an interactive multimedia platform designed to help people understand the characteristics of molecular strands with an emphasis on proteins. It uses a simple model of molecular forces to give real time interactive animations of the folding and docking processes. The shape of a molecular strand is shown as a 3D visualization floating above a 2D triangular matrix representing distance constraints, contact maps or other features of residue pairs. The 2D visualization is also interactive and can be used to manipulate a molecule, define constraints, control and view the folding dynamically, or even design new molecules.

1 INTRODUCTION

FoldSynth is based around a simplified amino-acid chain model (derived from the Poing model [2]) using particles attracting and repulsing each other by springs. There is one particle for each amino acid residue. Secondary structures are explicitly represented as ribbons: alpha helices in red and beta sheets in green.

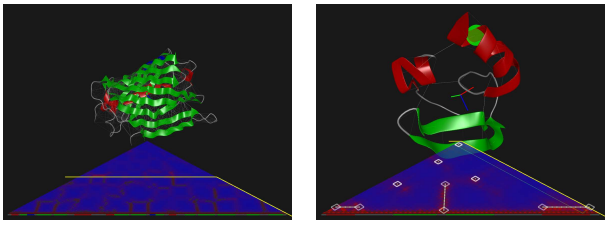

Figure 1: Main FoldSynth visual mode: Left: A protein from the PDB was loaded and its springs are being partially used. Right: An artificial fold is being created ("painted") from the 2D triangular matrix.

FoldSynth is the most interactive software platform to include physics and to allow experimentation with the dynamics of complex molecular chains, with applications such as protein folding (Fig. 1) and docking (Fig. 2). It does this by combining (a) the more classical 3D renderings of molecules, (b) a 2D distance matrix or contact map, (c) realtime simulation and (d) realtime interaction with all the previous features. Other recent solutions in that space lag behind FoldSynth especially with respect to its dynamics and interactive features performance. E.g., the recently released CMView package provides only static results, with limited functionalities in comparison to FoldSynth and depends on a separate 3D graphics engine (PyMol) [4]. In the space of social networks and on-line (serious) games, FoldIt has showed that interactivity is key in engaging users to study and manipulate scientific data such as molecular strands [1]. In comparison to FoldSynth, FoldIt focuses on 3D classical renderings and thus lacks a comparable 2D interface making it harder to interact with residue pairing features. FoldIt goals are also presently exclusively oriented to the problem of protein folding, while we are interested in a larger scope of problems, including the manipulation of other molecular strands, and combining multiple molecules for docking.

*University of London, UK, e-mail:ffl@foldsynth.com

2 BASIC FEATURES

The current implementation supports single or multiple chains, for studying single protein folding or multi protein docking. Connectivity for these backbone chains is maintained by springs; several of the graphics views assume the chain model which naturally creates a dynamic system, a central aspect of FoldSynth. The model is simple enough that it can easily be understood, and can show interesting dynamics and interactive behaviours.

The model includes the following basic features. (1) two sets of springs: (i) between each particle pair connecting consecutive amino acids, and (ii) between arbitrary particle pairs — all springs have a set (individual) length, and operate as repulsive when particles are too close, or attractive when they are too distant; (2) torsion springs attempt to enforce a dihedral angle between four successive particles; (3) a general damping factor to prevent uncontrolled vibrations; (4) simple forces that emulate electrostatic and hydrophobic effects; (5) random impacts on particles to coarsely simulate a water solution which keeps the system active and help avoid some local minima; (6) clash detection and repulsion; (7) a general repulsive force between all particles within a certain distance, such that when there are very few springs active the model opens out, which gives a chance for the chain to take up a new folded configuration when springs are reactived. The user may interactively control the strengths of these various forces; and also create, destroy and disable springs and particles.

3 DYNAMICS

The user has direct control over the dynamics simulation speed, which is typically played as close to realtime as possible. Each step is limited by particle forces and velocities to ensure a (reasonably) stable simulation. The final observed latency is mainly a function of the processing capacity of the hardware, but is typically not a problem on a laptop with a good graphics card. NB: Our implementation is in Java using JOGL for graphics, and has been tested on latest Linux, MS Windows and Max OS X.

4 SOURCE DATA

The particles and springs for a FoldSynth model may be set up in various ways. The most common way is to load a known protein from a central resource such as the Protein DataBank (PDB), either as a saved file or by web lookup. Alternatively, a user may type an amino acid sequence or allow the system to generate a random sequence of given length. In each case, the sequence of amino acids and associated backbone springs are created in the model. The user may also graphically define from scratch ("paint") artificial molecular strands, or modify a loaded strand.

Where a protein is loaded from a known protein model, we also generate springs based on the known distances. FoldSynth can create springs based on any geometric distance. At one extreme, we have springs only for contact pairs in the known folded structure. At the other extreme, we have springs between every pair of particles,

however far apart they lie in the real structure. The spring details may also be modified interactively and dynamically.

5 Views

FoldSynth provides quality graphics and sonifications. These are mostly based on traditional protein visualization techniques and are also important to the use of FoldSynth in computer arts and installations. Each view can be coloured in various ways: by position in the sequence, by amino acid type, by charge or hydrophobicity, by torsion angle, or by secondary structure. Where these properties change during the dynamic simulation, we can colour the view based on the current value, the target value, or the difference between them. Also, transparency can be used to make it easier to view the part of the chain around selected particle(s). We only summarize these here as they are not central to the interactive theme of this presentation.

The **cartoon view** shows a chain as a cartoon view of the secondary structures, emphasising (α) helices and (β) sheets. A current structure is computed dynamically as the protein folds. The **ball and stick view** uses a ball representing each particle and sticks representing the chain segment along the backbone. Balls can be deformed into ellipses that show the current velocity of each particle. An **isosurface view** showing each particle as a metaball; with fast CPU and GPU implementations that allow realtime interactive use. A **history view** shows a temporary trace of each moving particle. When the chain is in a fairly stable state, this illustrates well molecular vibrations. As the chain folds, it traces as 3D surfaces the folding mechanism. This type of visualisation also has good aesthetic and artistic potential. A **sequence view** shows a spherical bead repetitively going through the chain: this gives a good memento for the essential 1D nature of such chains, and allows us to display at once 1D, 2D, and 3D dynamic visualisation of the same molecular object.

Other visualization features include: the current spring can be shown as a 3D line; particle labels (by sequence position or residue type); and graphs of various properties, such as the x,y,z values along the backbone, or a spectral analysis of the vibrations of the chain or of a selected subpart. A **distance map** shows distance measures between particle pairs or other features related to such pairwise relations, such as contacts. This is detailed next. FoldSynth also includes various sonifications, not detailed here.

5.1 Distance Map View

The distance map is an important part of the interaction. It provides a fairly conventional view of the distances between particle pairs [3]. We usually orient it as a perspective triangle. It can show the current distance, the target distance (as loaded from the databank), or the difference (stress) between the two. It also shows the current springs. In our usual orientation the main diagonal lines along the bottom of the triangle and each particle is represented by two 45 degree lines radiating up from the diagonal. Each point on the matrix represents a particle pair [figure].

Particle pairs may be selected by use of the mouse over the matrix. Hovering performs a preselection and clicking a selection. Selections and preselections are kept consistent between the main 3D view and the matrix view and fed-back in both views. FoldSynth can show up to three per particle properties in lines under the matrix. These may be shown coloured or as graphs.

There are well known patterns that show up in the matrix view. For example, an alpha helix shows up as a line just above the base, and two consecutive strands of a sheet as a vertical or horizontal line for parallel and antiparallel sheets. Showing the secondary structure as a line along the base allows these patterns to be easily correlated with the secondary structure.

The matrix may be shown flat shaded, or as a height mapped mountain view The matrix can also be used to represent other useful information between residue pairs: forces, energy measures, other linear mappings.

6 Interactions

There are three main types of interactions in FoldSynth: (i) directly on the 3D viewport, (ii) by "hand gestures" on the distance matrix, and (iii) via a tree-menu GUI showing the various properties and their values and features, e.g. the relative strength of the various forces in the simulation.

The highly interactive and dynamic nature gives a feeling of play. However, FoldSynth is also capable of helping understand the various forces of the simulation, the ways they interact, and the effects they have. This could be useful as a teaching tool, and has already given additional insight to professionals already well aware of proteins and protein folding. There is an important caveat here. Seeing is believing, and what is being seen is the effect of our simplified dynamic model; while this gives very useful insight, it is important not to allow our understanding to become too distorted by details of the behaviour. It is also possible to create artificial structures far from the realm of proteins, useful in FoldSynth's computer art applications.

6.1 Drawing Distance Maps

In this mode the user has the capability of creating models by "drawing" on the distance map to create or destroy springs (forces). There are three main ways to do so. (1) Simple painting on the distance map (free tool mode): where the GUI works like a traditional painting system. (2) Painting constrained to known secondary structures. (3) A vector graphics approach where lines are drawn arbitrarily over the matrix to create springs. These lines are more flexible, as once drawn they may be edited by moving the ends or dragging the entire line (Fig. 1, right). Also, we can associate properties to lines to turn the associated springs on and off, or to create different kinds of forces. Some of these same interactions may also be applied directly on the 3d view.

6.2 Controlling disclosure of springs and particles

The second main form of interaction is the ability to control the range of springs and particles in effect at a given time; we call this disclosure. This will be detailed in another paper.

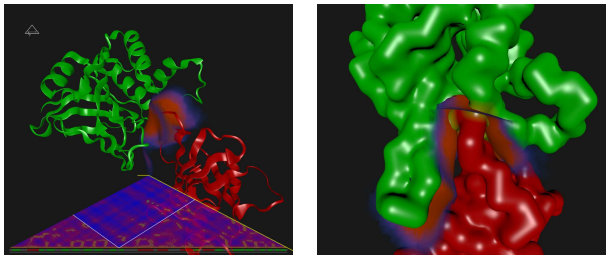

Figure 2: Two views of dynamic protein docking in FoldSynth.

References

[1] S. Copper et al. Predicting protein structures with a multiplayer online game. *Nature*, 466:756–60, Aug. 2010.

[2] B. Jefferys et al. Protein folding requires crowd control in a simulated cell. *Journal of Molecular Biology*, 397(5):1329–1338, April 2010.

[3] W. Taylor and A. Aszodi. *Protein Geometry, Classification, Topology and Symmetry*. CRC Press, 2004.

[4] C. Vehlow et al. CMView: Interactive contact map visualization and analysis. *Bioinformatics*, 27(11):1573–4, 2011.