Eurographics Workshop on Visual Computing for Biology and Medicine (2015)
K. Bühler, L. Linsen, and N. W. John (Editors)

FoldSynth: Interactive 2D/3D Visualisation Platform for Molecular Strands

S. Todd[1], P. Todd[1], F. Fol Leymarie[1], W. Latham[1], L. A. Kelley[2], M. Sternberg[2], J. Hugues[3] and S. Taylor[3]

[1]Goldsmiths, University of London, Computing Dept., U.K.
[2]Imperial College London, Structural Bioinformatics Group, U.K.
[3]Oxford University, Computational Biology Research Group, U.K.

Abstract

FoldSynth is an interactive platform designed to help understand the characteristics and commonly used visual abstractions of molecular strands with an emphasis on proteins and DNA. 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 visualisation floating above a 2D triangular matrix representing distance constraints, contact maps or other features of residue pairs. As well as more conventional raster plots, contact maps can be shown with vectors representing the grouping of contacts as secondary structures. The 2D visualisation is also interactive and can be used to manipulate a molecule, define constraints, control and view the folding dynamically, or even design new molecules. While the 3D visualisation is more realistic showing a molecule representation approximating the physical behavior and spatial properties, the 2D visualisation offers greater visibility, in that all molecular positions (and pairings) are always in view; the 3D mode may suffer occlusions and create complex views which are typically hard to understand to humans.

Categories and Subject Descriptors (according to ACM CCS):
J.3 [Computer Applications]: Life and Medical Science—Biology and genetics

1. Introduction

FoldSynth is an interactive software platform which allows manipulations on an abstract 2D contact map representation to be expressed in realtime on a running physics-based 3D simulation of a molecule. It further allows experimentation with the dynamics and structure of complex molecular chains, with applications such as the folding and docking of proteins and DNA strings. It does this by combining: (a) various 3D renderings of molecules, (b) half of a 2D distance matrix or contact map, a classic representation in bio-chemistry of molecular strands (Fig. 1), (c) realtime physics-based simulation and (d) realtime interaction with all the previous features.

FoldSynth is based around a model of a simplified chain (of residues) using particles attracting and repulsing each other by springs; the model is derived from *Poing* [J*10]. There is one particle for each residue position along a strand. Secondary structures are explicitly represented as ribbons: alpha helices in red and beta sheets in green (Fig. 1). The main novel feature of *FoldSynth* is that it provides a very

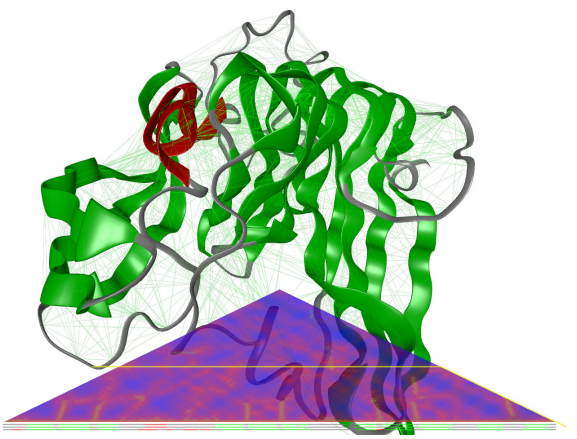

Figure 1: *Main* FoldSynth *visual mode: A protein from the PDB [B*00] was loaded and its springs are being partially used (which is shown and controlled on the distance triangular matrix via the horizontal yellow bar).*

high degree of interactivity with the model and the dynamics. It allows interactivity on the 3D model, on a 2D contact (or distance) map view, or on the physics-based simulation of a molecule.

2. Background

Other recent work in interactive molecular modeling does not match *FoldSynth* especially with respect to its dynamics and interactive features. For example, the CMView (or Contact Map View) package provides only the visualisation and analysis of static results [V*11]. In the space of social networks and on-line (serious) games, FoldIt has shown that interactivity is key in engaging users to study and manipulate scientific data such as molecular strands [C*10, E*12]. Other work involving interactive protein simulation until circa 2012 is reviewed by Férey *et al.* [F*12], while more recently the detailed survey by Kozlikova *et al.* covers most aspects of molecular visualisation (representations, rendering, simulations) [K*15].

A well established molecular visualisation platform is PyMOL, written in the Python programming language, and since 2010 supported by a commercial entity, Schrödinger LLP [Sch10]. More recently, there is a push to develop plugins and platforms on the basis of generic 3D software platforms from either (i) the graphics world — *e.g.* based on Autodesk's Maya [Iwa14] or the open source package Blender [A*12] — or (ii) the games industry with engines like Unity3D [L*13, P*15]. These recent efforts focus on the direct visualisation, and occasionally manipulation, of 3D models, are integrated within large software platforms (which may prove complex to master), and currently lack the dynamics and higher level interactive interface as provided by *FoldSynth* — for example, in PyMOL and CMView, although a contact map viewer is provided, it remains non-interactive and with limited visualisation options. Our approach allows users to comprehend and express relationships between secondary or higher level structures and the basic primary structure.

The goals of well-established gamification projects like FoldIt and EteRNA tend to be restricted to a particular application, *e.g.* to the problem of protein folding [C*10, E*12] in the case of FoldIt, and to the problem of the assembly of RNA molecules in the case of EteRNA [L*14a]. More recently Udock was introduced to address the problem of protein docking, again with an interface specialised to one very particular application [L*14b]. We have found that such gamification projects provide useful insights in what may work with a large community of users in terms of interactions when manipulating molecules of various types and representations.

We are interested in providing an interactive visualisation platform for a large scope of problems, including the manipulation of different molecular strands, as well as combining multiple molecules for docking. More closely related to our goals is the SketchBio project currently being developed at the University of North Carolina [W*14]. Its focus is on providing an interface to allow for the efficient staging of animations when creating scientifically accurate videos illustrating the functions of various large molecular complexes. SketchBio provides a two handed interaction mode of manipulation, low resolution surface renderings and drop shadows to help position molecules in 3D.

FoldSynth emerged from a first collaboration with experts working on proteins and we initially developed our interface specifically to manipulate molecular strands formed by series of amino acids [L*10, T*11]. After gaining much experience with that application we started to work with another group of experts interested in DNA and RNA molecular strands. The results and the description of *FoldSynth* we detail here for the first time represent a summation of those experiences.

3. Physics and Inputs

The current implementation supports single or multiple chains, for studying single protein and DNA folding or multi–molecular 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 with the form rapidly adapting to reflect constraints set by the user. The implemented 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 constraints 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 damping factor to prevent uncontrolled vibrations.
4. Forces to 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 (or "anti-bunch").
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 reactivated.

Features 1, 3, 5 and 6 are derived from the Poing model [J*10], while we added features 2, 4 and 7. The user may interactively control the strengths of these various forces, and also create, destroy and disable springs and particles. This

spring model gives *FoldSynth* a dynamic behaviour we describe next.

3.1. 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 (a binding for the Open Graphics Library), and has been tested on latest Linux, MS Windows and Mac OS.

3.2. 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 molecular strand, *e.g.* a protein from a central resource such as the Protein DataBank (PDB) [B*00], either as a saved file or by web lookup. Alternatively, a user may type in 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 artificial molecular strands by "painting" these via a distance matrix or modify a loaded strand (§ 5.1). When a molecular strand is loaded from a known (protein or DNA) model, we also generate springs based on the given distances. *FoldSynth* will create springs based on any specified geometric distance threshold. 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 as discussed later.

4. Visualisations

FoldSynth provides a range of quality graphical visualisation modes and features. Each view can be coloured in various ways by: position in the sequence, amino acid type, charge or hydrophobicity, torsion angle, 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 or hide the part of the chain around selected particle(s). Hiding is useful in the context of surface representations, as it allows a hole to be made in the surface exposing interior details of the selected portion. We summarise below the different visualisation modes currently available.

Cartoon view: It shows a chain as the classic "cartoon" representation of chemical ribbons of the secondary structures [LH82], emphasising (α) helices and (β) sheets. A current structure is computed dynamically as the molecular strand folds.

Ball and stick view: It uses a ball representing each particle and sticks representing the chain segment along the backbone. Balls can be deformed into ellipsoids that show the current velocity of each particle.

Isosurface view: Shows each particle as a metaball [Bli82], with fast CPU and GPU implementations that allow realtime interactive use and blending into dynamic isosurfaces [LB06]. We use the following falloff polynomial curve to avoid both square root computations and normal field discontinuities: $f(r) = (1 - r^2)^3$, where r is the distance to a point in space. Our CPU implementation uses the marching cubes algorithm optimised by recursive subdivision — which makes it difficult to parallelise. Our current GPU implementation uses a "surface hunting" algorithm that fires rays from the surface of non-metaball spheres — which makes it easier to parallelise.

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 semi-transparent surfaces the folding mechanism. This type of visualisation also has good aesthetic value and artistic potential. A closely related visualisation effect has been recently proposed by Cockrell and Kantrowitz who emphasise the intermediate stages between extreme conformation positions of macromolecules, by generating a linking surface ribbon colored with a rainbow map [CK13]. This is used in a static rendition mode, while we emphasise the dynamic nature of the physics of molecular movements (Figs. 6 and 9.(c)).

Sequence view: Shows a spherical bead repetitively going through the chain which gives a good memento for the essential 1D nature of such molecular single strands, and allows us to display at once 1D, 2D, and 3D dynamic visualisation of the same molecular object.

Other visualisation features include: a current spring can be shown as a 3D line; particle labels can be displayed (by sequence position or residue type); graphs of various properties can be visualised 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*, *i.e.* whether a particle is close enough (below a threshold on distance) to another to be considered in contact. This is detailed next.

4.1. Interactive Distance Map

The distance map is an important part of the interaction. In biochemistry it provides a classic *static* view of the distances between particle pairs as a square symmetric matrix [TA04]. Using the symmetry property, we instead orient half of the matrix as a *perspective triangle* and display pairwise properties in a dynamic and interactive fashion. The 2D matrix

view is a standard view for such data and only provides orientation between this standard view and the 3D rendering. In this subordinate role the view has been tilted to maximise the space for the 3D dimension representation, which users have found a suitable compromise.

This triangular 2D surface can show the current distance (between particle pairs; yellow=close, blue=far), the target distance (as loaded from the databank), or the difference (stress) between the two (blue=wrong, red=correct; Fig. 2). It also shows the current springs (as white lines or points surrounded by a diamond shape, and projected flat down on the matrix). In our usual orientation the main diagonal line is along the bottom of the triangle with each particle represented by two 45 degree lines radiating up from the diagonal. These diagonal lines represent the row and column of the (half, symmetric) matrix corresponding to that particle. Each point on the matrix always represents a particle pair property (Fig. 2). *FoldSynth* can be set-up more conventionally with the matrix and 3D view in separate viewports, and with reduced perspective. This can improve detailed clarity but reduces the engaging visual impact, for example preventing the use of the height map.

Particle pairs may be selected by use of the mouse over the matrix. Hovering performs a preselection and clicking a selection. Individual particles are preselected and selected when the mouse is just below the main diagonal of the matrix. Selections and preselections are kept consistent between the main 3D view and the matrix view and fed-back in both views. Optionally, particles which are directly connected to the current selection by springs can be shown with a weaker highlight. *FoldSynth* can also show up to three per-particle properties in horizontal lines under the matrix main diagonal; these may be shown coloured or as graphs (Fig. 2).

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 (*i.e.* parallel and near to the main matrix diagonal), and two consecutive strands of a *beta sheet* as a *vertical* or *horizontal* line for parallel and anti-parallel sheets. Showing the secondary structure as a line allows these patterns to be easily correlated with the secondary structure. The distance map also includes a magnifying glass view for seeing the current area of interest in more detail (Fig. 2, top). The matrix may be shown flat shaded, or as a height mapped mountain view (Fig. 2, middle). The matrix can also be used to represent other useful information between residue pairs: forces, energy measures, difference between the current structure and a target (Fig. 2, bottom), and other linear mappings.

Vector contact map view: The distance map may be further abstracted by grouping contact points into vector lines. These vectors can be automatically derived from the secondary structure, and we are working on simple image processing methods to derive them from the distance map. These vectors can operate as output, where they display an

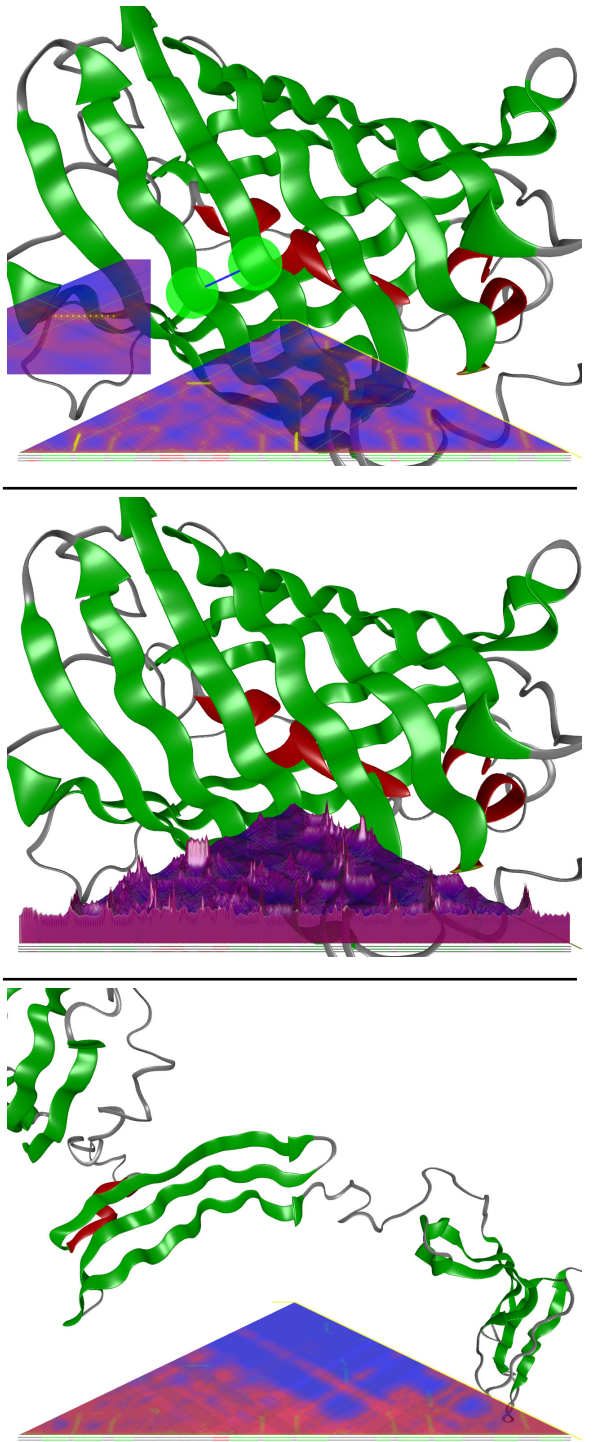

Figure 2: *Distance map views:* Top*: standard view (magnifying glass in center-left).* Middle*: 3D view (where matrix values are shown as heights forming a topographic map).* Bottom*: difference view between current and expected "target" structures, where red areas indicate correct distances, while blue shows incorrect values. The protein is partly unfolded in correspondence to blue areas.*

abstraction of the current springs or the current state of folding. Later paragraphs discuss how they can also operate as input, being used to define groups of springs and thus to impose secondary structures.

5. Interactions

There are three main types of interactions: *(i)* directly on the 3D viewport, *(ii)* by "hand gestures" on the (triangular) distance matrix, and *(iii)* via a searchable tree-menu GUI (Graphical User Interface) 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 users understand the various forces of the simulation, the ways they interact, and the effects they have. This is useful as a teaching tool, and has already given additional insight to professionals well aware of proteins, DNA, RNA and their 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 chemistry, useful in *FoldSynth*'s computer art and design applications (Fig. 3).

5.1. Drawing Distance Maps

In this mode the user has the capability of creating and modifying models by "drawing" on the distance map to create or destroy springs (forces). Through our research, we have progressively developed a variety of ways in which these operations can be performed.

(1) Simple painting on the distance map (free tool mode): where the GUI works like a traditional painting system. For example, a left mouse dragging gesture creates springs while right mouse dragging destroys these.

(2) Painting constrained to known secondary structures: there are modes for (a) alpha helices as horizontal lines just above the main matrix diagonal; (b) general helices or parallel sheets which are horizontal lines away from the main diagonal; and (c) anti-parallel sheets as vertical lines.

(3) A vector graphics approach where lines are drawn arbitrarily over the matrix to create springs, with snapping constraints to encourage them to conform to either parallel or anti-parallel orientation with respect to the main diagonal. In contrast to raster contact maps, these vectors emphasise the notion that the set of contacts making up a secondary structure element is a coherent entity in itself. They allow subsequent manipulations to be more flexible, as by dragging the ends of an entire line. Some lines may remain completely passive, for visually marking up features of interest. The extent of a structure and its position in the context of the whole molecule can thus be controlled (Fig. 3.(b,c,d))

Some of these same interactions may also be applied to particles selected directly in the 3D view, with commands for creating helices and sheets of a controllable number of strands. When the user interacts with either the 2D or 3D views, corresponding feedback is given as to which parts of the model are involved in the interaction. Selecting or preselecting particles in the 3D view will cause the rows and columns in the 2D matrix relating to those particles to be highlighted, while hovering over a point in the matrix will cause the two particles corresponding to that point to be highlighted, along with their rows and columns. Additionally, a weaker highlight may be applied to any particles that are directly connected to the current selection by springs. Clicking on a vector line causes the set of particles involved in that structural formation to be selected, with corresponding visualisation as described. This visual feedback allows the user to understand which parts of the model are being manipulated, and which other structures will become connected or disconnected as a result of manipulations.

5.2. Masking and disclosure of springs and particles

The second main form of interaction gives the ability to control which springs and particles are active at a given time. For that purpose, we provide tools for drawing rectangular regions onto the 2D matrix, aligned orthogonally to the main diagonal. These act to either *mask* (disable, Fig. 4) or *disclose* (explicitly re-enable where another mask is active) any contacts in the corresponding region. The user can then study the contribution of these contacts to the overall form as the model folds and unfolds in response to their manipulation.

Two additional special cases of these tools exist. One allows the user to mask all springs that are beyond a controllable sequence distance apart. This allows the progressive introduction of increasingly high level structures, as at first only turns, soon followed by α-helices and subsequently other high level structures including the isolated contacts that define the tertiary level structures emerge. Another allows the gradual disclosure of particles along the backbone. In this version of the tool, non-disclosed particles are entirely removed from the active model rather than merely having their contacts disabled. This can be seen as approximating the gradual emergence of a protein from a ribosome during synthesis. As the chain grows, structures are formed subject to any other masks that may be active.

6. Applications: From Proteins to DNA

We first developed *FoldSynth* for proteins and to study interactively protein folding. More recently we started to apply the *FoldSynth* software to understanding the 3D topology as a result of interactions in the genome as assayed by methods such as 3C [D*02], 4C [vdW*12], 5C [D*06],

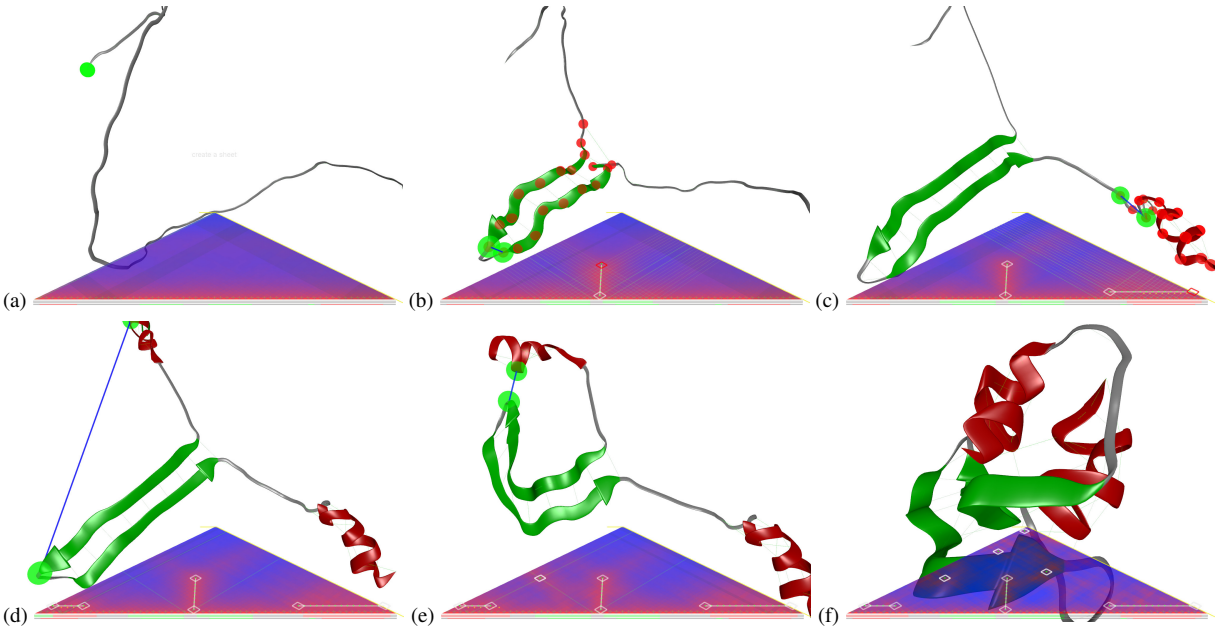

Figure 3: *Interacting by drawing secondary structure:* (a) *initial unfolded position;* (b) *draw a fold, vertical line on map;* (c) *draw a helix, horizontal line to the right and near the main matrix diagonal (at the bottom);* (d) *2nd helix drawn (bottom left); prepare to add a point connection;* (e) *point connection made (middle left);* (f) *a few more point connections complete the molecule.*

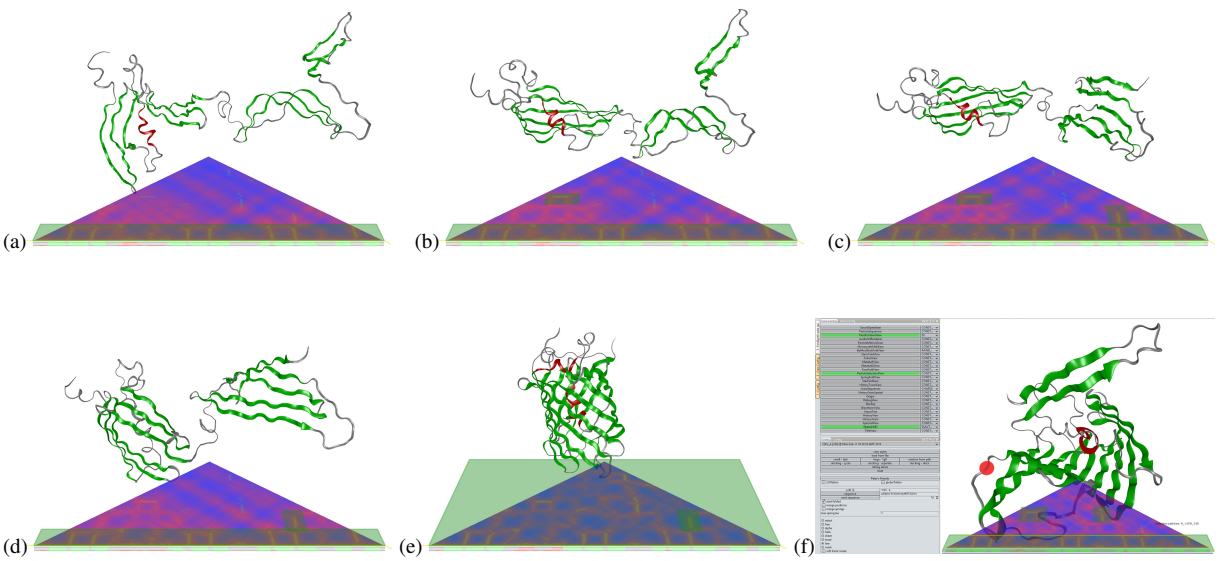

Figure 4: *Masks being used to selectively view the effects of different groups of contacts residues:* (a) *protein with only short secondary constraints disclosed by a mask (long rectangle near the main diagonal/bottom of matrix);* (b) *mask for another group of contacts brings parts of the protein together;* (c) *mask for another contact group;* (d) *remove mask in* (b)*, to see the effect of just contact groups in* (a) *and* (c)*;* (e) *extend the original mask found in* (a) *to disclose all contacts.* (f) *Fuller view at the GUI with menus.*

Hi-C [LA*09] and Capture-C [H*14] experiments. This allows the scientists to understand the Topologically Associated Domains (TADs) which are normally only represented within a 2D view in a genome browser. For example, our first experiments very quickly gave a much more concrete idea of the TADs linked with a particular enhancer and gene associated with blood formation. In some ways the use is very much as for proteins, but in other ways there is a large difference. The similarity is in the interaction and dynamics, and the use of (symmetric) distance matrices. The main differences arise because the size of DNA strands makes any real-time graphics on the full strand awkward, and interactive dynamics more challenging.

FoldSynth can read interaction information which are represented as genome coordinates and their interaction strength. In the datasets we have worked with so far each particle represents a window of 5000 residues; but even with the 5000 summary datasets this still gives over 18000 particles with over 7 million recorded contact probabilities. To handle such long strands *FoldSynth* can operate on a small segment or pair of segments at a time. The segments can realistically hold up to 1000 particles, or circa 5% of the total strand. The entire dataset is however loaded to make it faster to move between segments, and the user can then select a required segment. The segment choice may be made in a companion custom built Large Matrix Viewer (*LMV*), described below, that shows the matrix at multiple zoom levels. We intend later to offer directly within *FoldSynth* the option to group larger particles than those in the raw data, thus showing a broader but less detailed view of the strand. *LMV* already performs such aggregation. Unlike with proteins we have not had available any real 3D data for DNA strands. The contact probabilities are used to create springs in the *FoldSynth* model, and these can help the dynamics to find possible 3D conformations, very much as when *FoldSynth* performs interactive folding on proteins. As with proteins, the user can experiment with model parameters such as the transformation between raw contact probabilities and associated spring strengths and lengths. It is also possible to use the various masking options on the 2D matrix to perform "what if" experiments on the importance of various contacts on the overall structure, or the relative importance of local and more distant contacts.

We can annotate the distance matrix or the 3D view colouring by the structure information contained in .BED files.[†] Up to 3 BED files at a time can be loaded. These can all be displayed under the matrix in the same way secondary structures are displayed for proteins; and one at a time can be used for colouring the 3D view. Selection feedback (on either the matrix or the 3D form) shows both particle number/id and any BED information associated with that particle (Fig. 5).

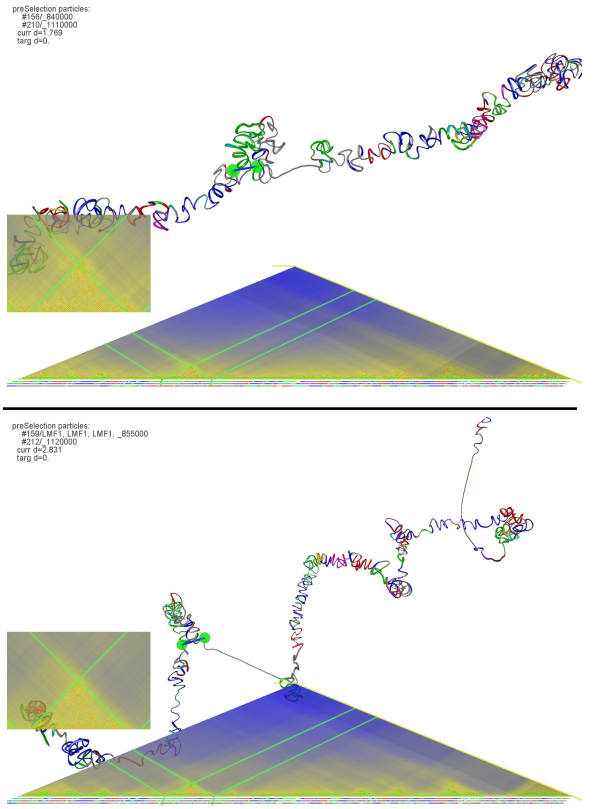

Figure 5: Top*:* FoldSynth *displaying a segment from a DNA strand. The cursor is used to select a TAD in the matrix view; the zoomed region of the matrix is also shown (to the left, above the matrix area); the TAD is visible in the 3D view and information about the selected data shown at the top left.* Bottom*: This is similar to the previous view but with an additional anti-bunch force in the dynamics. The TADs remain bunched, but that is made more obvious by the relatively spread nature of the overall strand segment.*

We can also read and integrate WIG files which show numerical information of other assays across the genome and use these to annotate the 2D matrix view and to change the 3D view: for example by modifying the strand radius according to the .WIG value. In some cases there are alternate datasets that contain different experimental contact probabilities on the same DNA strand. *FoldSynth* can load a pair of raw data sets, and smoothly move between the two. This movement modifies springs based on interpolating probabilities, and the dynamics then generates alternative 3D structures. The smooth movement between these structures gives a very intuitive way of comparing the implications of the two underlying raw datasets (Fig. 6). This deformation is similar

[†] Details on the BED and WIG formats can be found at: https://genome.ucsc.edu/FAQ/FAQformat.html

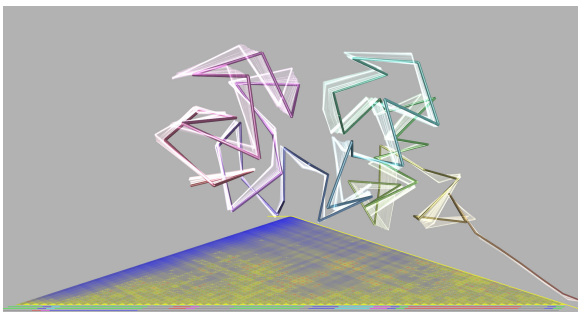

Figure 6: *Difference between two conformations, shown as a* history trace *view. The solid coloured form shows one of the configurations, and the fairly hard white form the other, while the intermediate pale area shows the change between conformations. The difference between the conformations is too small to show up well as two separate images. This conformation change is usually viewed as a realtime animation change in* FoldSynth.

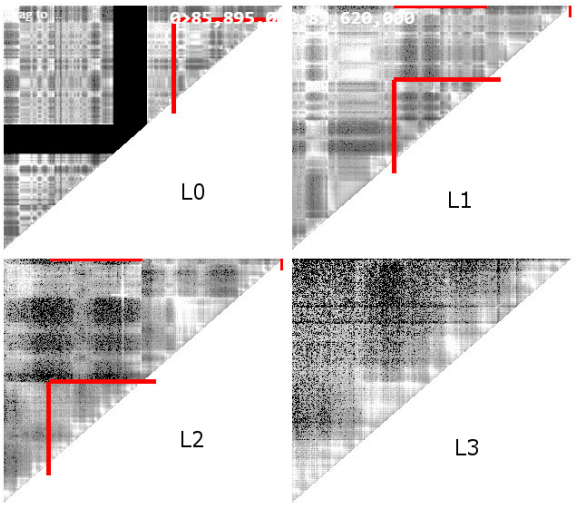

Figure 7: *A single segment selected in the* LMV*: the user makes selections near the main diagonal of the matrix at each level, thus selecting a single segment of the DNA chain.*

to experiments on proteins with bistable conformations, or the deformation of a protein as it takes part in a docking.

6.1. Large Matrix Viewer (*LMV*)

The large matrix viewer (*LMV*) is a separate WebGL browser based program that can be used in its own right, or federated with *FoldSynth* to select regions to process. *LMV* shows the matrix derived from the raw data at four zoom levels. Interactive user selection at any zoom level will change the region showed at all the higher zoom levels. To zoom in, a user will typically select the approximate region of interest in the outermost zoom level (L 0, entire strand), then refine the selection at the next zoom level (L 1), then at L 2, so that the final level (L 3) can be set up very precisely (Fig. 7). Currently the zoom levels are fixed with only coarse user control on the ratio between these. The zooming for detailed levels and aggregation needed to show lower levels (*e.g.* the matrix for the entire strand) are implemented in graphics shaders in the GPU (using WebGL and GLSL). For systems with midrange discrete GPUs this means the entire interaction and redisplay process can take part at 60 fps, and the selection can be made continuously by mouse dragging. For less powerful systems the redisplay after a new selection may take a significant part of a second, and it is easier to make each selection by a discrete mouse click.

It is very common to display an object and a zoom. Use of multiple simultaneous zoom levels is less common, but very valuable with such large datasets. As far as we are aware the *LMV* is unique in offering simultaneous zoom levels with such high interactive speeds. When a final selection has been made, that selection may be transferred to *FoldSynth* for processing, for example by drag-drop. Two kinds of selection may be made. Where the selection is along the diagonal of the matrix it represents a single segment of the DNA strand. *FoldSynth* processing will then show the possible folding of that segment (Fig. 8). Where the selection is more internal to the matrix it effectively identifies two segments of the DNA strand. The higher matrix levels will be segmented; with the two triangular regions representing the contacts within each segment, and the square region representing the contacts between the two segments. *FoldSynth* processing will then show the folding of each of the segments, and also the way the two segments contact each other. This is exactly equivalent to viewing of docked or multi-chain proteins. Where a selection pair is made, the large matrix view currently limits both segments to being the same size. That is not inherent in the design and will need a small UI change to overcome. *FoldSynth* already permits two segments of different sizes.

6.2. Case study: Contact frequency for DNA

Contact frequency data between DNA strands within the nucleus is at present effectively limited to static 2D representations. This is a very difficult and suboptimal human interface with these data as it requires the operator to try and interpret dynamic functional 3D relationships from static 2D representations. A real world example comes from the Oxford team of scientists who gained valuable insights into a current problem using existing whole genome interaction datasets when these were rendered into 3D space using *FoldSynth*. A functional interaction was known between the regulatory elements of the Alpha globin genes and the NME4 gene widely spaced regions on chromosome 16 and which was active only in blood cells [L*09]. It was unclear from existing 2D representations why these two regions should interact

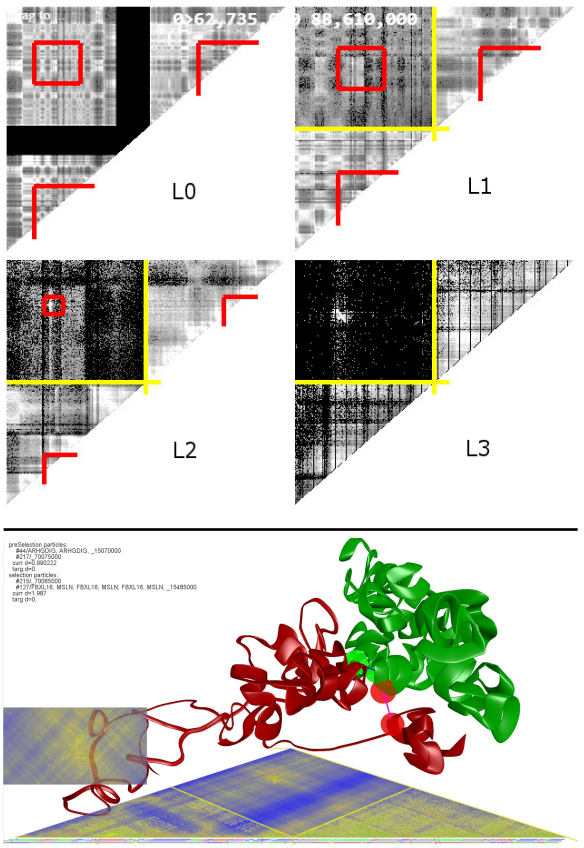

Figure 8: Top: LMV *is used to select a pair of segments. The range was chosen to bring out a high spot visible in the full view (L0), with the selection refined in the more zoomed views; by the detailed view at L3 the individual contacts from the raw data are visible. The yellow lines show the segmentation of the matrix into 2 intra-segment triangles and an inter-segment square.* Bottom*: Corresponding* FoldSynth *view which shows the 2 segments in red and green, where the latter is more compact as it has more contacts. Text information (including* IDs *and* .BED *data) on selected areas are shown at the top left.*

in this way when many other functional regions were spaced in between these two functional elements. When rendered into 3D space it became evident that these two regions were consistently physically proximal even in non-blood cells and hence appeared optimally positioned to interact when these functional elements became active in blood cells.

7. Conclusion

We have discussed an interactive graphical tool for exploring molecular strand structures, with interactions both directly on the 3D representation and via a 2D map — of various distances, of contacts, and derived features shared between pairs of residue particles. The distance map is in fact absolutely critical to provide a connection for users familiar with this common rendering with the new 3D view and is a regularly used feature. In our recent experiments with long DNA strands we have found that to understand how the non-coding genome interacts with and controls the expression of the coding genome, their 3D relationship must be reliably determined.

FoldSynth provides a unique way for scientists and students to visualise not just the final structure of these molecular strands, but also to explore the folding processes that lead to that structure, and to play with different aspects of that folding process in a way that is highly tangible. The model is simplified to permit realtime interaction and simulation as well as to allow free formation of unlikely or impossible forms that more realistic constraints would disallow (Fig. 9).

FoldSynth also provides vector graphics lines to represent collections of springs on the contact map; these lines correspond to various secondary structure features. Representing them as such graphically has an analogous effect on the comprehensibility of contact map plots to the well established cartoon representations of the same structural features in 3D models. Direct interaction with these higher level features is a particularly powerful way of helping the user understand the initially unintuitive way in which 2D matrix plots relate to 3D form and could even be used by more advanced users to sketch designs for potential protein structures. In our ongoing work we are applying similar techniques to protein docking (Fig. 10). We expect to add more statistical methods that will permit a molecular strand (protein or DNA) to fold autonomously, but still providing interaction with the methods so that the system leads to a better understanding of such methods and where they apply. We are also considering better interfaces, to make it easier to mix and match the capabilities of *FoldSynth* with the features of other systems.

References

[A*12] ANDREI R. M., ET AL.: Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics 13*, 4 (2012). 2

[B*00] BERMAN H., ET AL.: The protein data bank. *Nucleic Acids Research 28*, 1 (January 2000), 235–242. 1, 3

[Bli82] BLINN J. F.: A generalization of algebraic surface drawing. *ACM Trans. on Graphics 1*, 3 (July 1982), 235–256. 3

[C*10] COPPER S., ET AL.: Predicting protein structures with a multiplayer online game. *Nature 466* (Aug. 2010), 756–60. 2

[CK13] COCKRELL G. M., KANTROWITZ E. R.: Viewmotions Rainbow: A new method to illustrate molecular motions in proteins. *J. Molec. Graphics & Modelling 40* (2013), 48–53. 3

[D*02] DEKKER J., ET AL.: Capturing chromosome conformation. *Science 295*, 5558 (2002), 1306–11. 5

[D*06] DOSTIE J., ET AL.: Chromosome conformation capture carbon copy (5C). *Genome Research 16* (2006), 1299–1309. 5

[E*12] EIBEN C. B., ET AL.: Increased diels-alderase activity

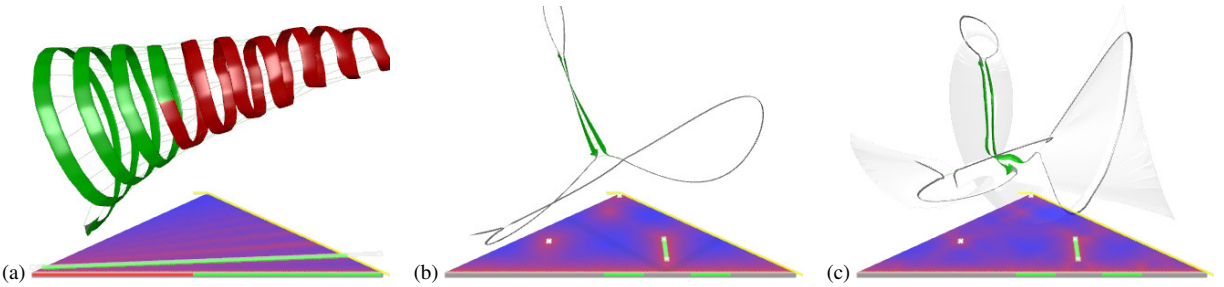

Figure 9: *Unreal form design: (a) conical spiral ribbon obtained by painting a set of force constraints at a diagonal orientation on the distance map; (b) looping structure enforced by imposing a position constraint at the tip of the triangular area of the distance map, and including one beta sheet (straight line constraint drawn perpendicularly to the main basis of the distance map) as well as one additional pinch area (constraints drawn on the left side of the distance map); (c) visualisation of the "history view" as the form moves and deforms under its dynamics.*

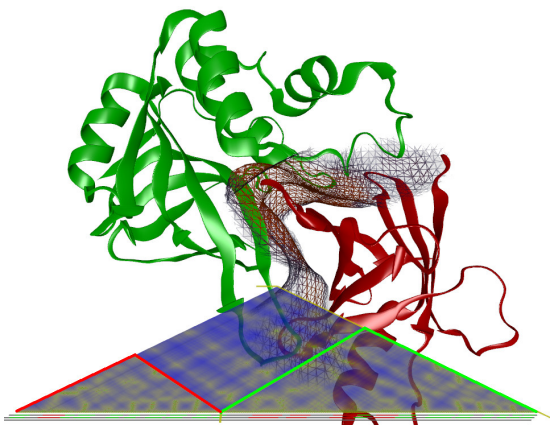

Figure 10: *Illustration of the use of* FoldSynth *in molecular* docking *mode: a pair of proteins from the PDB were loaded and docking initiated. A medial surface (of distances) is shown (as a mesh) in between the proteins which offers an additional visual guide of the progression of the dynamic process.*

through backbone remodeling guided by Foldit players. *Nature Biotechnology 30* (2012), 190–2. 2

[F*12] FÉREY N., ET AL.: Advances in human-protein interaction. In *Protein-Protein Interactions: Computational & Experimental Tools*. InTech, 2012, ch. 2, pp. 27–64. 2

[H*14] HUGHES J. R., ET AL.: Analysis of hundreds of cis-reg. landscapes at high resolution in a single, high-throughput experiment. *Nature Genetics 46*, 2 (2014), 205–12. 7

[Iwa14] IWASA J.: Crafting a career in molecular animation. *Molecular Biology of the Cell 25*, 19 (2014), 2891–3. 2

[J*10] JEFFERYS B., ET AL.: Protein folding requires crowd control in a simulated cell. *Journal of Molecular Biology 397*, 5 (2010), 1329–38. 1, 2

[K*15] KOZLIKOVA B., ET AL.: Visualization of biomolecular structures: State of the art. In *Eurographics Conference on Visualization (EuroVis)* (2015), pp. 61–81. 2

[L*09] LOWER K. M., ET AL.: Adventitious changes in long-range gene expression caused by polymorphic structural variation and promoter competition. *PNAS 106*, 51 (2009), 21771–6. 8

[L*10] LEYMARIE F. F., ET AL.: Foldsynth. Poster (T16) at EMBO VizBi Workshop, 2010. 2

[L*13] LV Z., ET AL.: Game on, science — how video game technology may help biologists tackle visualization challenges. *PLoS One 8*, 3 (2013). 2

[L*14a] LEE J., ET AL.: RNA design rules from a massive open laboratory. *PNAS 111*, 6 (2014), 2122–7. 2

[L*14b] LEVIEUX G., ET AL.: Udock: interactive docking entertainment system. *Faraday Discussions 169* (2014), 425–41. 2

[LA*09] LIEBERMAN AIDEN E., ET AL.: Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science 326*, 5950 (2009), 289–93. 7

[LB06] LOOP C., BLINN J. F.: Real-time GPU rendering of piecewise algebraic surfaces. *ACM Trans. on Graphics 25*, 3 (2006), 664–70. 3

[LH82] LESK A. M., HARDMAN K. D.: Computer–generated schematic diagrams of protein structures. *Science 216*, 4545 (1982), 539–40. 3

[P*15] PÉREZ S., ET AL.: Three-dimensional representations of complex carbohydrates and polysaccharides. *Glycobiology 25*, 5 (2015), 483–91. 2

[Sch10] SCHRÖDINGER, LLC: The PyMOL molecular graphics system. www.pymol.org, 2010. 2

[T*11] TODD P., ET AL.: Foldsynth. Poster at 1st IEEE BioVis Symposium, 2011. 2

[TA04] TAYLOR W., ASZODI A.: *Protein Geometry, Classification, Topology and Symmetry*. CRC Press, 2004. 3

[V*11] VEHLOW C., ET AL.: CMView: Interactive contact map vis. & analysis. *Bioinformatics 27*, 11 (2011), 1573–4. 2

[vdW*12] VAN DE WERKEN H., ET AL.: 4C technology: Protocols and data analysis. *Methods in Enzymology 513* (2012), 89–112. 5

[W*14] WALDON S. M., ET AL.: SketchBio: A scientist's 3D interface for molecular modeling and animation. *BMC Bioinformatics 15*, 334 (2014). 2