Journal of Digital Information, Vol 5, No 1 (2004)

Unified Hyperstructures for Bioinformatics: Escaping the Application Prison

Adam Moore and Tim Brailsford
School of Computer Science and IT, University of Nottingham, NG8 1BB, UK
Email: {adam.moore, tim.brailsford}@nottingham.ac.uk

Abstract

The *Next Big Thing* in hypertext will be unifying different applications in bioinformatics through the ZigZag paradigm, allowing this field to live up to its promise of revolutionising the pharmaceutical industry. The paper outlines ZigZag, Ted Nelson's unique hyperstructural paradigm, and illustrates how, by examining a current bioinformatics task such as structure/binding prediction, the application of this novel paradigm has the potential to revolutionise bioinformatics completely by allowing a unified approach to a task currently fulfilled by fragmented data and applications.

1 Introduction

This paper proposes that the application of a new paradigm of hypertext, Nelson's ZigZag system, could potentially have a significant effect on the bioinformatics industry, becoming the *Next Big Thing*. This neatly illustrates a fundamental problem with the way information is currently manipulated in the workplace. It is important because, as bioinformatics is becoming increasingly central to modern healthcare, this is a paradigm shift that could change the world!

Bioinformatics may be defined as the use of computers and IT to solve problems in the life sciences, most importantly in the areas of molecular biology, genetics and biochemistry. These disciplines underpin almost all research in the pharmaceutical industry — a sector estimated by IMS Health (2002) to have a global value of $392 billion in 2001 and to be growing at an annual rate of about 12%. This industry is hugely important, both in economic and humanitarian terms. For example, the development of the first treatments for HIV, based on the HIV protease protein, were developed by modelling the molecule and then basing drug designs on key structural components (reviewed by Babine and Bender 1997). It has been estimated that bioinformatics has the potential to reduce both the cost of developing a new drug, and the time taken to do so, by about a third (Sood and Shukla 2003). Despite this huge potential, the lack of standardization for information storage and manipulation, and the lack of integration of activities of different players in the industry, has meant that the discipline of bioinformatics has yet to live up to its promise.

For example, Stein (2002) compared the current state of bioinformatics to the warring Italian city-states of the middle ages. These highly sophisticated societies typically had different legal, political and taxation systems, as well as differences in culture, language, weights and measures and currency. The Italian renaissance produced some of the most brilliant scientific thinkers of their (or indeed any) age. This great flowering of human intellect failed to translate into widespread technological or industrial development, however, largely because of the difficulty in overcoming these differences.

There is a striking parallel between this period of history and the current and ongoing bioinformatics revolution: modern bioinformatics data providers suffer from major differences in the mechanics of their work (and sometimes even in their world-view) that seriously damage scientific advancement. The conventional view is that this problem is solvable by standards committees and the availability of ever greater computing power, whereas the ultimate problem is more about the need for what Engelbart calls 'knowledge augmentation' (Engelbart and Lehtman 1988, Engelbart 1995). The problem is not tractable by simply increasing the computation cycles available, but by designing software that augments human intellect to address such complex problems better.

Bioinformatics is ultimately all about complexity, and is concerned with data sets that are uniquely vast, complex and almost infinitely interrelated. The majority of biochemical interactions that are involved in disease processes, and are therefore candidates for manipulation by drugs, are based around proteins. Proteins are constructed of 20 amino acid building blocks, with an average of about 300 per strand (i.e. there are theoretically about 20^{300} possible permutations). These strands, along with a variety of other substances, are then folded into complex and often mobile three-dimensional shapes. This is all controlled by the "software" of the human genome, which consists of about 3 billion nucleotide pairs (i.e. the fundamental unit of information) in about 30,000 genes.

The science of bioinformatics is aimed at making sense of all of this information, and using the resulting knowledge for such purposes as devising treatments for disease. A wide range of computationally intensive techniques - based on mainframes or, increasingly, grid technology - are used. Ultimately people need to be able to interpret this information. This is the aspect of bioinformatics that we focus on here.

The problem is that integration has to mean the ability to work with the different applications' functionality *at the same time* so that

- the user can view the combinations of data required by their task, rather than those that happen to be maintained by the same application
- the tools provided by the different applications are available at the same time, so the user is not disrupted bt changing to a different application to use a different tool

ZigZag is a system that excels at making the complex comprehensible by laying bare the underlying structure of information. It makes far greater provision for storing and visualising the fundamental interconnectedness of information than other, more conventional, methods of hypertextual representation. Rather than operate at one particular level of relationship expression (e.g. the syntactical scaffolding of XLink/XPointer or the semantic tuples of RDF), ZigZag hyperstructures allow the relationship between fundamental pieces of information/media along diverse lines of whatever implicit or explicit scaffolding the user or system may specify.

In many potential applications the complexity of information is the result of human design, deliberate or otherwise. Bioinformatics is unusual in that extreme complexity is intrinsic to the discipline. Bioinformatics therefore provides a unique opportunity to explore the capabilities of the ZigZag approach to managing information-space, in the context of a discipline that is badly in need of such a new paradigm.

Although it is undoubtedly true that many problems in bioinformatics are only soluble by the application of vast amounts of processing power, it is also true that ultimately these problems have to be visualised for human interpretation. ZigZag is readily capable of representing complex structures in easily comprehensible forms, so it will extend the frontiers of the comprehensibility of biological information, that is, data structures will be browsable by humans who previously would have required what is essentially a proxy analysis by machine, using ever-increasing amounts of computing power.

It must also be realised that the solutions produced will only ever be as accurate or useful as the algorithms applied. This implies that such designed approaches to analysis are limited in scope, not allowing for the serendipitous association of results which this visualisation would facilitate. Therefore using ZigZag should allow a more 'lateral' approach to computational experimentation, instead of only relying on the application of analytical tools that are predicted to be useful.

This paper provides an overview of what we call the 'application prison', and briefly outlines the ZigZag hyperstructure. We describe a typical bioinformatics task in terms of both current techniques and our proposed alternative approach.

2 The tyranny of the "application"

Throughout their daily lives, computer users today are adrift in a sea of applications and files, which often overlap substantially in their information content. For example, address books in email and PIM applications, or the information in word processors or spreadsheets. Since these files are typically used by different applications, it is almost inevitable that the information contained within them is divergent in both the manner in which it is stored and its internal categorisation and structuring. All too often similar applications from different vendors use file formats that differ from those of their competitors sufficiently to ensure that moving information between systems is a painful experience.

Commonly the mechanisms that are used to represent and manipulate connections between such items of information consist of simple Web interfaces to underlying applications. Some simple linking to other online resources is often used to enhance the functionality of such systems. Typically these systems are essentially Web front-ends to file drops, where users can pick up a file deposited by another user (often in an arbitrary format). Sometimes interfaces to conversion software are provided, which will try to mix and match (with varying degrees of success) the requirements of one format with whatever the user requests. Files with important information missing may be translated into formats where that information is mandatory (for example, the position of hydrogen atoms on chemical structures) by conversion algorithms that employ 'best-guess' calculations. This adds uncertainty about the origin and veracity of the data. Moreover, since such uncertainty is cumulative the uncertainty (and therefore the concomitant reliability of the data) decreases as the conversions mount up.

While the problem of the application/file trap is rife in almost every area of computer use, it particularly exacerbates the problems involved in interpreting the complex systems in the world of bioinformatics. The cumbersome nature of disparate bioinformatics tools, and all the concomitant potential for losing information as data is transferred between them, makes the already difficult task of interpreting such systems far harder than it needs to be. This is aptly illustrated by Stein (2002), who described the simple task of retrieving all the new entries made to a bioinformatics database in a given week. Usually an individual researcher will write a simple script to automate this process by wrapping the various sequential steps into a single task. However, if any of the information resources changes its interface or format even slightly, all of these scripts may well have to be changed.

3 ZigZag approach to hyperstructure

The ZigZag paradigm of hypertext has been developed by Ted Nelson (Nelson 1998, Nelson 1999, Nelson 2001). It is a highly generalised information structure that is almost infinitely flexible, and capable of representing almost any conceivable type of data. A complete description of the ZigZag structure is beyond the scope of this paper; the reader is referred to the ZigZag tutorial (Nelson 1999).

In essence, however, ZigZag can be said to resemble a collection of lists that may intersect in an unlimited number of dimensions. Most forms of information may be represented as lists - this is fundamental to technologies as diverse as a telephone directory, the author list in a bibliography, or the atoms constituting a protein. The factor that is used to order a list (e.g. people's names, in the case of a telephone directory; the carbon backbone of a protein molecule in the case of bioinformatics) is the dimension that orders the data. In ZigZag, information is contained in cells (a primitive cell can contain any one form of data from a single keystroke to a number, an item of text or an entire image). Cells are ordered into dimensions, and there may be any number of these dimensions that may intersect in any number of places. In some respects, this is quite similar to a multidimensional spreadsheet, but with cells that are not bound into rigid rows and columns, rather they are interlinked and they can exhibit many complex behaviors, such as looping. Any given item of information is only ever stored once, but by "rotating the view" (i.e. varying the dimensions used in a representation) this information can be retrieved in whatever context it is required.

Three rules are fundamental to a ZigZag hyperstructure (generically, zzStructure):

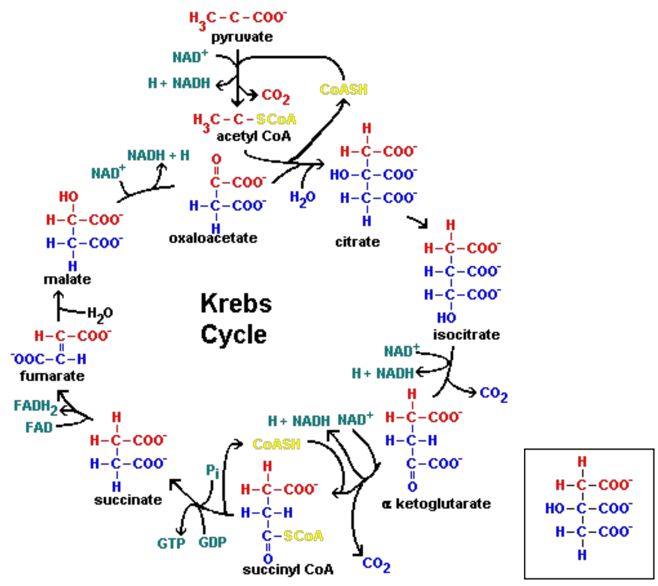
- There are no link types
- Each basic cell can contain only one type of information - text, sound, picture, video, etc.
- There are no one-to-many links, that is, each cell can have only one connection in the positive direction, and one in the negative direction, in any one dimension

Within ZigZag, all fundamental features are themselves encoded in ZigZag, and therefore, at the lowest level, all information everything is stored in cells (the cell being the sole primitive unit of ZigZag). Hence dimensions are themselves cells, and the connections merely represent the ordering of cells along any given dimension. This, plus the fact that there is only one link poswards and one negwards, ensures that all paths are non-branching, thus embodying the simplest possible mechanism for link traversal.

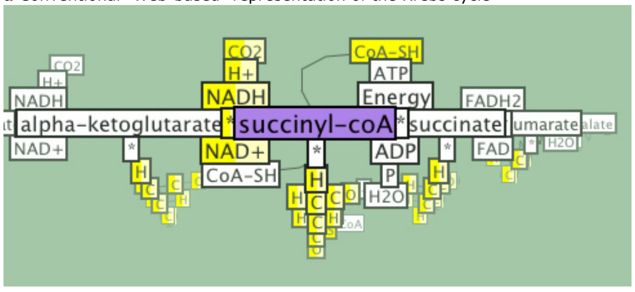
Taking the example of bioinformatics, an atom could be represented by a single cell. Information about that atom is stored by connecting it to appropriate dimensions (e.g. atomtype for the type of atom, X and Y for its coordinates). Further information about ZigZag and zzStructures can be found in the ZigZag tutorial.

To demonstrate the viability of representing biological information as zzStructures, we created a simple representation of some well known metabolic pathways using one of the current ZigZag prototypes (GZZ [see note]). It must be emphasised that this implementation of ZigZag does not have any of the custom views that we believe will eventually be essential for representing scientific information. However, even using the simple views that are available we can visualise these systems in a highly interactive and intuitive manner.

Figure 1 compares a static snapshot of a ZigZag representation of the Krebs Cycle (Figure 1b) with a 'conventional' Web-based representation (i.e. an animated GIF) of the same information (Figure 1a). The conventional representation is quite a complex diagram, because the information itself is complex. Nonetheless it is a simplification of the available information to avoid even greater complexity (for example, it tells the user nothing about the underlying chemistry or the diseases caused by dysfunctions in this system). The ZigZag version restricts the information available at any one time to the dimensions of interest, but movement within those dimensions and rotation between these and other dimensions are entirely under user control. Hence far more information is available in a form that is more easily 'digestible', being highly interactive and focused on the issue at hand.



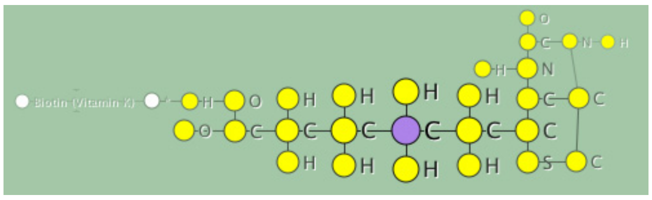
a Conventional "Web-based" representation of the Krebs cycle



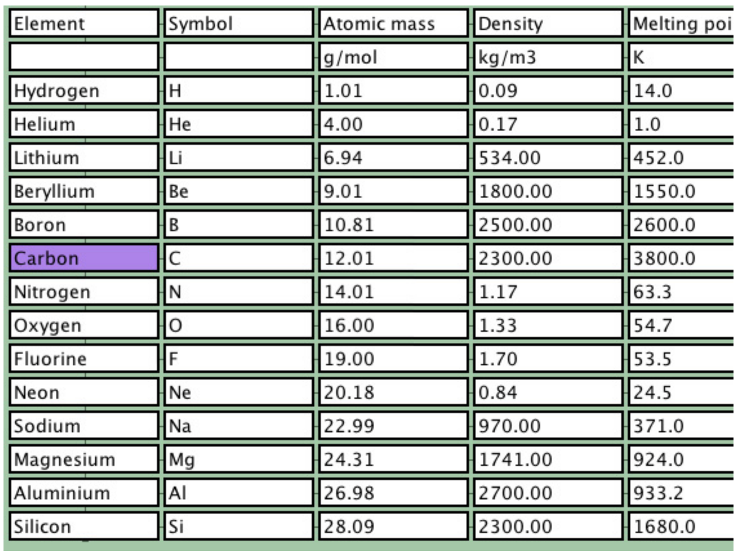
b ZigZag representation of the Krebs cycle

Figure 1. Comparing information in Web and ZigZag formats. The Krebs cycle is a key set of reactions defining mitochondrial respiration which is an essential part of any biochemist's knowledge: **a** picture and animated gif, a 'modern' way of representing the process (obtained, with permission, from http://www.people.virginia.edu/~rjh9u/krebs.html; **b** snapshot from the ZigZag representation of this process developed by the authors (an animated version of this, showing the CoA cycle being browsed gives a better impression of the interactive user experience). In the zzStructure, the full cycle is navigable by a user, providing direct experience of the processes involved. Information about chemical structures, other pathways inputs and outputs, corresponding diseases of dysfunction of this pathway and sufferers from those dysfunctions, are all available from a different set of interconnected dimensions navigable by the user (using keyboard commands, a mouse or a game controller)

Figure 2 is a part of the same prototype, demonstrating the results of switching dimensions and views of those dimensions. Figure 2a shows the chemical structure of one of the components of the metabolic process; Figure 2b shows information about the physical chemistry of the atoms involved; and Figure 2c shows the positions of the elements in the periodic table.

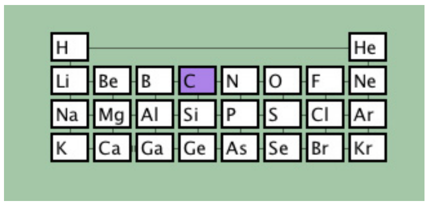


a



Element	Symbol	Atomic mass	Density	Melting poi
		g/mol	kg/m3	K
Hydrogen	H	1.01	0.09	14.0
Helium	He	4.00	0.17	1.0
Lithium	Li	6.94	534.00	452.0
Beryllium	Be	9.01	1800.00	1550.0
Boron	B	10.81	2500.00	2600.0
Carbon	C	12.01	2300.00	3800.0
Nitrogen	N	14.01	1.17	63.3
Oxygen	O	16.00	1.33	54.7
Fluorine	F	19.00	1.70	53.5
Neon	Ne	20.18	0.84	24.5
Sodium	Na	22.99	970.00	371.0
Magnesium	Mg	24.31	1741.00	924.0
Aluminium	Al	26.98	2700.00	933.2
Silicon	Si	28.09	2300.00	1680.0

b



c

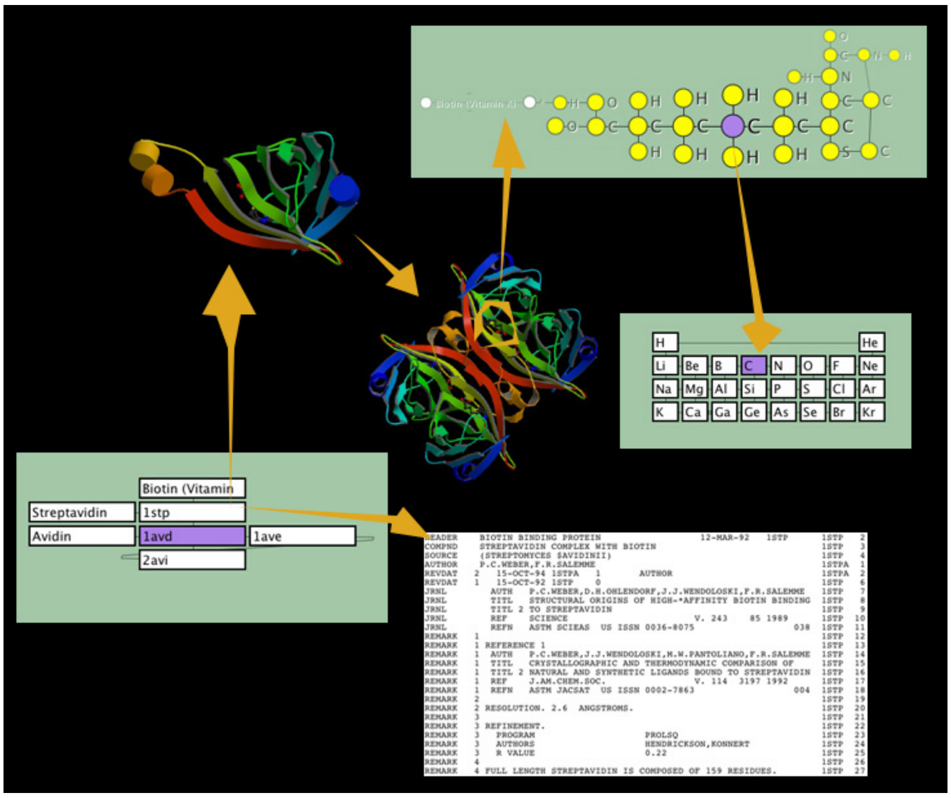
Figure 2. Representational connections in ZigZag illustrating some of the advantages of using zzStructures for bioinformatics: **a** chemical structure of biotin, otherwise known as vitamin K (note that the 2D structure has not been defined as a set of coordinates, but rather it is emergent from the cell connections, given the particular view that is being used); **b** as the zzStructure is linked to the periodic table, it is simple to gather information about any atom in the structure, by following the link; **c** the standard representation of the periodic table can also be presented, using the same cells organised along different dimensions

4 A scenario in biological information space

To illustrate the potential that ZigZag has in bioinformatics, we consider a task that is commonly undertaken in bioinformatics research: the creation of candidates for the structure of a new biologically active molecule, as the first stage in the design of a new drug. We first explain the process, and how it is done using today's software. We then explain how it could be done using a bioinformatics software suite based on ZigZag.

Most drugs work by mimicking a natural association between a biologically important protein and some other molecule. At a molecular level, this molecule attaches to a specific place (the active site) on the surface of the protein. To design a potential drug candidate requires prediction of the chemical structure of a molecule that will mimic this natural interaction. The starting point is normally a known molecular structure, consisting of a protein together with a small molecule bound to its active site. To design a drug, the usual approach is to make changes to the chemical structure of the candidate molecule and its orientation with respect to the active site (see, for example, Kitchen 2004). Since there are literally astronomical numbers of permutations, a heuristic approach is usually taken and this requires substantial CPU resources. The problem is vastly larger when, as in some cases, the conformation of the protein changes during the interaction or as a result of extraneous circumstances.

Using modern research techniques, the approach taken is to obtain a protein structure, usually from a public domain source (such as the Protein Data Bank). This is then imported into an electronic simulation environment, where calculations are performed for each atom and its associated bonds. These calculations describe the properties of each atom, or group of atoms, within the context of the simulation. The structure of the isolated protein is then generated by subtracting the candidate molecule from the overall structure. This protein is then replicated many times and a variation of the candidate molecule, in terms of both structure and orientation, is inserted into each instance. Depending on the number of candidate molecules used, this process is repeated anything from a few thousand to few billion times. This provides a starting point for the simulation of each target molecule, the simulation being of the changes in atomic behavior over time. These simulations are then run until a predefined set of terminating conditions is reached. Figure 3 is a conceptual visualisation of how a bioinformatics hyperstructure constructed using the ZigZag paradigms described may appear.



```
HEADER    BIOTIN BINDING PROTEIN                         12-MAR-92   1STP      1STP   2
COMPND    STREPTAVIDIN COMPLEX WITH BIOTIN                                     1STP   3
SOURCE    (STREPTOMYCES $AVIDINII)                                             1STP   4
AUTHOR    P.C.WEBER,F.R.SALEMME                                                1STPA  1
REVDAT   2   15-OCT-94 1STPA    1       AUTHOR                                 1STPA  2
REVDAT   1   15-OCT-92 1STP     0                                              1STP   6
JRNL        AUTH   P.C.WEBER,D.H.OHLENDORF,J.J.WENDOLOSKI,F.R.SALEMME          1STP   7
JRNL        TITL   STRUCTURAL ORIGINS OF HIGH-*AFFINITY BIOTIN BINDING         1STP   8
JRNL        TITL 2 TO STREPTAVIDIN                                             1STP   9
JRNL        REF    SCIENCE                            V. 243    85 1989        1STP  10
JRNL        REFN   ASTM SCIEAS  US ISSN 0036-8075                        038   1STP  11
REMARK   1                                                                     1STP  12
REMARK   1 REFERENCE 1                                                         1STP  13
REMARK   1  AUTH   P.C.WEBER,J.J.WENDOLOSKI,M.W.PANTOLIANO,F.R.SALEMME         1STP  14
REMARK   1  TITL   CRYSTALLOGRAPHIC AND THERMODYNAMIC COMPARISON OF            1STP  15
REMARK   1  TITL 2 NATURAL AND SYNTHETIC LIGANDS BOUND TO STREPTAVIDIN         1STP  16
REMARK   1  REF    J.AM.CHEM.SOC.                     V. 114  3197 1992        1STP  17
REMARK   1  REFN   ASTM JACSAT  US ISSN 0002-7863                        004   1STP  18
REMARK   2                                                                     1STP  19
REMARK   2 RESOLUTION. 2.6  ANGSTROMS.                                         1STP  20
REMARK   3                                                                     1STP  21
REMARK   3 REFINEMENT.                                                         1STP  22
REMARK   3   PROGRAM                    PROLSQ                                 1STP  23
REMARK   3   AUTHORS                    HENDRICKSON,KONNERT                    1STP  24
REMARK   3   R VALUE                    0.22                                   1STP  25
REMARK   4                                                                     1STP  26
REMARK   4 FULL LENGTH STREPTAVIDIN IS COMPOSED OF 159 RESIDUES.               1STP  27
```

Figure 3. A conceptual visualisation of a possible ZigZag hyperstructure for bioinformatics, using the proposed ZigZag structures showing the dimensions of files attached to streptavidin and biotin. Also shown are an output of the original ASCII file, and a representation of the file's 3D coordinates, first as the single sub-unit, then, in the centre, of the whole structure, which is composed of four identical sub-units (these views are taken from output produced by the PDB). One of the biotin molecules is highlighted and its structure within the zzSpace displayed, and a reference link is shown to the periodic table where the values used for the chemical structure calculations will be found

Having predicted the behaviours of each candidate molecule, these behaviours are then evaluated for their biological efficacy, in terms of binding and fit between the active site and the candidate molecule. The most promising candidates are then passed on to the next stage of investigation, which might be more detailed simulation, chemical evaluation or experimental testing in biological systems.

Therefore the steps of a typical example procedure may be (noting that this is not a fixed *recipe*, but rather one example of a procedure that has many subtle permutations):

- download a starting structure
- transform that starting structure into a suitable format for the simulation environment
- generate candidate protein molecule
- generate candidate binding molecules
- generate starting structures
- run simulation
- evaluate simulation results

4.1 Current technology: the application prison

A typical approach to this, using current technology, is initially to download the structure of a system of interest over the Internet. These structures are processed by software known as a "force-field type generator" that uses graph theory to predict the type of each atom and chemical bond within the context of the structure (i.e. what chemical bonding atoms participate in, and what they are bound to). Owing to the practical limitation of CPU resources, artificial constraints must be placed on the extent of the graph, and thus the output from this process is inevitably unreliable, dictated by the complexity of the algorithm used, and not simply the feasibility of the graph/structure of the result. This means that the output structure must be visually inspected using a 3D rendering engine, and the process may need to be repeated until, in the researcher's judgement, a usable structure has been obtained.

Candidate molecules are generated, using candidate generation software. This involves placing the original molecule (without the protein bound to it) into a file, and then replacing parts of this with other chemical structures. In addition to this the molecule is rotated so that different orientations are considered. This process is then repeated to generate each candidate molecule, until all chemically viable permutations are exhausted. Candidate molecules are then processed by adding them back into the file that contains the typed protein and discarding any that contain structures that are not simply meaningless, but what chemists call 'nonsensical'.

Each candidate (i.e. the chemical structure and orientation) is run within a simulation environment until the termination conditions are satisfied. The simulation environment is far too computationally demanding to run effectively on individual desktop computers so it is implemented either by means of grid technology, or by batch processing on a mainframe.

The output of the simulation, for each candidate, is the structure reached at the time that the termination conditions are satisfied. These output structures are then evaluated using a series of techniques, usually implemented using different analysis software. First, structures that do not display any association at all between the candidate molecule and the protein are eliminated; as are structures that form associations that are not related to the active site. Finally, associations between the remaining candidate molecules and the active site are compared with each other, to evaluate the "tightness of fit" and the overall strength of the association.

Once the modelling process has yielded a number of promising candidates (by this stage there should be less than 100) then the relative merits of these candidates need to be assessed using some type of visualisation technique. The techniques used to do this vary widely, but typically they might involve such things as: overlaying graphs describing properties of the interaction; superimposition of "ghost" representations of several alternative candidate molecules in the active site; or the calculation of various indices of the "quality" of the associations.

In practice, this type of procedure involves the use of many different applications that implement a number of different design paradigms. Shifting data between these applications is often cumbersome, and writing custom Perl scripts to move data between applications is a common task that bioinformatics researchers undertake. Although it is rarely done in practice, in principle it is possible to integrate the functionalities of all of these different applications into a single "molecular modelling" suite. Even so, this is a convenience rather than a fundamental change. Such suites are integrated in the same way that Office suites are integrated: they are essentially a bundle of different programs using different design paradigms. It is (usually) easier to move data between them, but the fundamental problem, that the application restricts free access to and interpretation of the data, remains.

4.2 A new paradigm for bioinformatics

In the ZigZag information space, all bioinformatics data - and indeed all data, from bibliographic data to email - is stored in cells. These cells are ordered along dimensions that are, by convention, named with a "d." prefix. For example, X, Y and Z atomic coordinates would be stored on the d.x, d.y and d.z dimensions, respectively. Discrete constructs in ZigZag-space that contain data of a specific type are called zzCules, and the visualisation is created using one of many different views.

A powerful aspect of ZigZag that is of great importance to the type of combinatorial problem described above is that it is programmable. ZigZag contains a programming language called NuZZL (Nelson 2004), a dataflow language that has some similarity to Prograph (Smedley 1998). NuZZL uses cells to contain data items that are passed along the dimensions of the program. This means that a zzCule can contain code that describes its own behavior. One type of NuZZL program, called a Weavebot, can create new zzCules.

All of this provides a lot of functionality that makes sophisticated bioinformatics data manipulation feasible. The interlinked nature of ZigZag makes the integration of such manipulations relatively easy, certainly much easier than the approach of isolating aspects of the functionality into individual applications. In ZigZag a program may also be referred to as an 'applitude', that is, a discrete process running inside the ZigZag environment. Applitudes are *not* applications, but rather part of the seamless fabric of the ZigZag environment.

The initial stage of the task is to download the molecular structures over the Internet and parse them into a ZigZag data structure. To do this the unique identifying ID of the molecule is entered into a cell and then the appropriate 'download and import' applitude is run to fetch and parse the file.

The original molecule is then placed on the d.original dimension. To transform the starting structure into one suitable for simulation, the type of each atom is then predicted using graph theory and this information is placed along the d.forcefield dimension for each atom and each bond. The typed structure will then be emergent from this (i.e. the typing process will create a zzCule that contains it).

To generate the candidate structures, places where substitutions will occur are then attached on d.substitution, and a candidate set is then generated as a summation of candidates on d.substitution, d.rotateX, d.rotateY and d.rotateZ (these last three dimensions containing the orientation of the candidate). A standard set of rotations could be used to process each candidate, and the candidate zzCules are then created from these dimensional connections by a Weavebot. The candidate zzCules (i.e. from d.candidate) are then serialized and handed off to the grid for processing.

The result zzCules will be returned on to d.complete and the status of grid jobs will be monitored on d.grid. The zzCules returned from the grid are then evaluated and attached to one of d.nobind (for candidates where there was no binding), d.noactive (for candidates that were bound, but not to the active site) and d.active (where there is binding to the active site). The parameters that are required for the visualisation will be generated as a result of membership of d.active.

Finally, the visualisation is provided by ZigZag views. We believe that four new views will be necessary:

- a view to display 3D atomic structures, with functionality similar to that found in applications such as RasMol
- a view to display 2D atomic structures, with functionality similar to that found in applications such as JChemPaint or the ChemDraw plug-in
- histogram view to display various numeric data representations (line charts, scatter plots, etc.)
- a report view to manage the process (particularly for managing grid jobs).

Again, the power of ZigZag is that these new views are not developments of new applications, but simply built out of the visualisation primitives that are part of the overall space. A lot of the visualisation is provided by the very nature of ZigZag. There would, for example, be no need for a spreadsheet or a database in this system, as this is functionality provided *de facto* for data stored using ZigZag.

This novel approach to the manipulation of bioinformatics data offers a number of advantages over conventional approaches. First, it is unified. There is no need to switch between paradigms of representation for different parts of the process, something that is cumbersome, time-consuming and risks loss of information due to incompatibilities of the underlying information models, also leading to cognitive stress and overload among researchers, decreasing the amount of work that can be done due to the overhead required from switching between different application paradigms.

Second, several aspects of the process will essentially be emergent. The NuZZL programming language will make feasible the creation of zzCules that are self-processing and, to some extent, self-analysing.

Third, the process will be far more transparent to researchers, partly because of the single paradigm, but also because there will always be the possibility of exploring any of the dimensions present at any stage in the process (or indeed of creating new ones, if an idea strikes). Hence the complexity becomes manageable because it may readily be temporarily restricted to the aspect that the user is interested in. Multivariate complexity can be vastly simplified by viewing it in terms of intersecting dimensions.

Finally, this approach will offer means of analysis that are impossible, or at least difficult, using current technologies. For example, it will be capable of doing comparisons of both sets and partially ordered sets, because a partial set is a valid ZigZag structure, whereas it is not usually a valid spreadsheet or database structure.

5 Conclusions

We have described the problems we believe to be inherent in the current approaches to bioinformatics tasks. The lack of a unified structure for the information involved and the applications working with this data lead to cognitive dissonance and therefore inefficiency in the application of direction to the tasks. This is a problem of the application of sentience, rather than a limit of computation cycles. We proposed using ZigZag, a single universe of multidimensional interconnectedness to alleviate this problem. Using ZigZag hyperstructures, tasks become a movement of views over the underlying data, rather than switching of priority between different applications with different aspects of the data and the underlying process.

Currently, much of the proposed solution is a specification, rather than a working implementation. As described in section 3, however, we have already created some simple bioinformatics prototypes. We have used ZigZag to represent molecular structures and metabolic pathways, and have created NuZZL programs to perform simple bioinformatics calculations. This demonstration is very simple in comparison with the type of useful scenario outlined previously. However, it does show that moderately complex biological systems can be rendered into seemingly simple and intuitive representations using this approach, and that all of this could be achieved without using any custom bioinformatics views.

This work is at a very early stage, but we are confident it is feasible. If ZigZag for bioinformatics lives up to the promise of this vision, then it could have a profound effect upon the practice of bioinformatics - and by extension upon the healthcare industry. This might just be a very big thing indeed!

Acknowledgements

The authors would like to thank Professor Ted Nelson for his help and advice, and for making development versions of ZigZag available to us. The Leverhulme Trust sponsored a Visiting Professorship for Ted Nelson, which greatly aided the initial development of this work. We are also most grateful to all in the Web Technology Group at the University of Nottingham, for their support and for many helpful discussions. Also, the referees who made many useful and constructive comments.

References

Babine, R.E. and Bender, S.L. (1997) "Molecular Recognition of Protein-Ligand Complexes: Applications to Drug Design". *Chemical Reviews*, 97 (5), 1359-1472

Engelbart, D.C. (1995) "Towards augmenting the human intellect and boosting our collective IQ". *Communications of the ACM*, 38 (8), 30-33

Engelbart, D.C. and Lehtman, H. (1988) "Working Together". *Byte Magazine*, December, 245-252 http://www-sul.stanford.edu/depts/hasrg/histsci/ssvoral/engelbart/append1-ntb.html

IMS Health (2002) IMS World Review http://www.ims-global.com/products/sales/review.htm

Kitchen, D.B. (2004) "A computer-assisted drug design primer". Albany Molecular Research, Inc. http://www.albmolecular.com/company/departments/medchem/molmod/mm03.shtml

Nelson, T.H. (1998) "What's On My Mind?" (ZigZag overview article). *First Wearable Computer Conference*, Fairfax, VA, May 12-13 http://www.xanadu.com.au/ted/zigzag/xybrap.html

Nelson, T.H. (1999) "Welcome to ZigZag" (the ZigZag tutorial). http://xanadu.com/zigzag/tutorial/ZZwelcome.html

Nelson, T.H. (2001) "Deeper Cosmology, Deeper Documents" (technical briefing). *12th ACM conference on Hypertext and Hypermedia*, Århus, Denmark, August 14-18

Nelson, T.H. (2004) "Cosmology for a different computer universe: data model, mechanisms, virtual machine and visualization infrastructure". *Journal of Digital Information*, Special Issue on Future Visions of Common-Use Hypertext, Vol. 5, No. 1

Smedley, G. (1998) "Visual Programming with Prograph". *Dr Dobb's Journal*, September, 76

Sood, K. and Shukla, R. (2003) "Bioinformatics: Ready or Not For Prime Time?" *LA Vox*, 5 (24) http://www.larta.org/LAVox/ArticleLinks/2003/030616_bioinformatics.asp

Stein, L.D. (2002) "Bioinformatics - Building a Nation from a Land of City States". *O'Reilly Bioinformatics Technology Conference*, Tucson, AZ, January 28-31. See B. Stewart, "A Land of City States", report on Lincoln Stein's Keynote, 30 January http://www.oreillynet.com/pub/a/network/2002/01/29/bioday2.html

Links

Protein Data Bank http://www.rcsb.org/pdb

RasMol http://www.umass.edu/microbio/rasmol/

JChemPaint http://jchempaint.sourceforge.net/

ChemDraw http://chemstore.cambridgesoft.com/software/product.cfm?pid=93

Notes

There are now several implementations of ZigZag (see http://xanadu.com/zigzag/ for announcements). The most important of these are:

- GZigZag (GZZ), an open source Java implementation (http://www.nongnu.org/gzz/)
- ZZZ, an OpenGL version for Windows, Macintosh and Linux that Ted Nelson is currently alpha-testing.

GZZ has far more advanced views then ZZZ, and is more stable, but it is not currently undergoing development. It is intended that in the future this will form the basis for an open source bioinformatics software suite for use in research. In the long term, ZZZ would be a suitable basis for any commercial spin-offs that might ultimately ensue. ZZZ is currently less polished but it has an active user and development community and is currently maintained and being directly driven by Ted Nelson.

ZigZag is covered by United States Patent 6,262,736, July 17, 2001 "Interactive connection, viewing, and manoeuvering system for complex data", and is a registered trademark of Ted Nelson.