Statistical Data Mining's Challenges in Bioinformatics

Diego Kuonen appeals to data miners and bioinformaticians to trust each other and to collaborate in unlocking secrets of the living cell

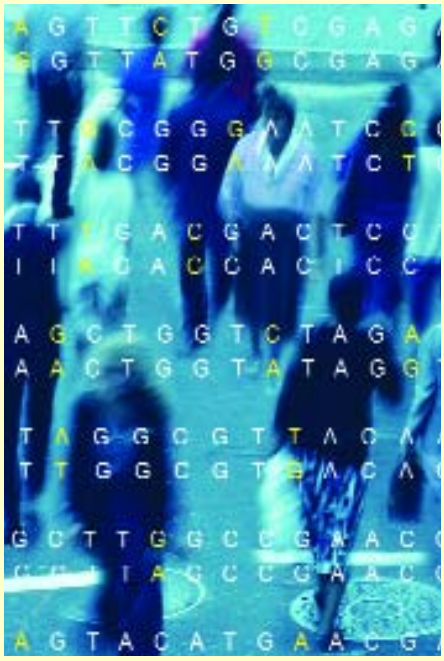
From a statistical data miner's perspective, most bioinformaticians tend to be ignorant of statistical data mining, are too impatient for solutions, expect the statistical data miners to know the solutions long before they have any data, will only use the latest algorithms, ignore software unless it is easy to use, and are always updating the data files. On the other hand, from a bioinformatician's perspective, statistical data miners do not understand the biological questions, take too long to come up with answers, moan about sample size and replication, speak a different scientific language, speak different programming languages, and use dreadful software. Hence, bioinformaticians and statistical data miners tend to criticise each other.

However, the field of bioinformatics, like statistical data mining, concerns itself with learning from data. Statistical data mining is fundamental to what bioinformatics is really trying to achieve. There is the opportunity for an immensely rewarding synergy between bioinformaticians and data miners.

The genome is the total amount of genetic information that an organism possesses; it is the biological 'program' for making the organism. Genes in turn are made from DNA, which stores information in the form of a sequence of nucleotide bases, a string of four letters (Adenine, Cytosine, Guanine and Thymine). There are about three billion such letters in the human genome. A sequence example is: TTCAGCCGATATCCTGGTCAGATTCTCTAAGTC GGCTATAGGACCAGTCTAAGAGA. The human genome has about 30,000 to 35,000 genes, segments of DNA that contain the recipe to make proteins. Proteins are the crucial molecules that do most of a cell's work.

The 'central dogma of molecular biology' states that, in a cell, information flows from the nuclear DNA, to RNA, to protein synthesis. Proteins in their turn are built from amino acids (whose recipes are contained in the genetic code of the DNA), and a protein sequence can be represented by a string made up from 20 different letters, each standing for an amino acid. Stored digitally in computers worldwide are trillions of sequences: that is, trillions of pieces of information that need to be turned into knowledge. The mountain of information that is, for example, the draft sequence of the human genome may be impressive, but without interpretation that is all it remains: a mass of data.

Identifying and interpreting interesting patterns hidden within the immense list of bases that constitute a genome is a critical goal. This is one of the topics of bioinformatics. More generally,



'U.S. Department of Energy Human Genome Program'

bioinformatics is the science of storing, extracting, organising, analysing, interpreting, and utilising information from biological sequences and molecules. Bioinformatics merges new technologies, such as sequence and transcriptome analysis, with computer science and advanced statistical (data mining) methods to organise, analyse and interpret data.

One of the most basic operations in bioinformatics involves searching for similarities, or homologies, between a (newly) sequenced piece of DNA and previously sequenced DNA segments from various organisms ('pair-wise sequence alignments'). Finding near-matches allows

Letter from the President

Managers and engineers enjoy learning about statistical thinking. But when they try to put what they have learned into practice, they face a lot of problems.

In one case I know of, an experiment was planned, but at the last minute production time was refused.

In another case, the product was prepared according to a designed experiment, but it was shipped before it could be evaluated.

Such problems suggest a lack of buy-in from higher management, but even though upper management were supportive, they had to put customer requirements first.

On the positive side, the same team were alerted to the effect of an interaction between factors in the assembly of a complex component and proposed minor changes to the assembly process. The result was that the reject rate was halved.

Like succeeding in all walks of life, it is not enough to have knowledge and skill; you also need to have determination, perseverance and luck.

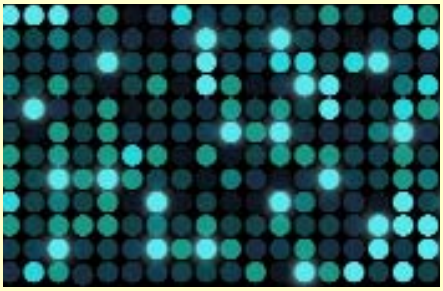
Shirley Coleman
President of ENBIS
Shirley.Coleman@ncl.ac.uk

researchers to predict the type of protein the (new) sequence encodes. This not only yields 'targets' early in drug development, but also weeds out many that would have turned out to be dead-ends. The most popular bioinformatics tool for this task is BLAST, whose core part is a beautiful and powerful example of the application of probability theory and statistics (comprising aspects of random walk theory, renewal theory, and asymptotic distribution theory) within bioinformatics.

However, even with such pair-wise alignments, interpretation is a bit of an art. On the other hand, for 'multiple sequence alignments' the scores that say how reliable the database search is do not yet exist. In addition, multiple sequence alignment methods are not perfect, as exact algorithms exhaust current computational resources. Hence imperfect heuristic algorithms, such as the ClustalW algorithm, are used. The people who make them know this, and the users who apply them should also consider this fact.

Multiple alignments are the basis for the construction of phylogenetic trees. On the other hand, multiple alignment aims at aligning a whole set of sequences to determine which sub-sequences are conserved – and this works best when a phylogenetic tree of related proteins is available! The resources available for making multiple sequence alignments online are almost overwhelming, using, for example, the Gibbs sampler, genetic algorithms or simulated annealing.

Sequence analysis seeks to tease out information based on a sequence itself, or on the similarity of one sequence to another ('pair-wise alignment'), or on patterns among groups of sequences ('clustering'). Another example of unsupervised learning is affinity grouping ('categorical sequence mining'), where one wants to discover sequences that commonly occur together, such as in a set of DNA sequences ACGTC is followed by GTCA after a gap of nine, with a probability of 30 per cent. On the other hand, supervised data mining techniques can be applied as well. For example, the goal of DNA sequence classification is to distinguish junk segments from coding segments, and this can be done using supervised learning.



Microarrays can analyse tens of thousands of samples simultaneously.

U.S. Department of Energy Human Genome Program

Production sequencing facilities like this one generate vast amounts of data for analysis.

Sequence data are not the only digital biological information available to researchers. Another data-type beginning to fill countless disk drives is that resulting from gene-expression analysis. Genomic sequence itself reveals only the possibilities of genetic manifestation. Within any given cell, only a small fraction of genes are 'expressed', that is, actively translated into proteins through intermediate RNA molecules. In the past few years, a new technology, called DNA micro-arrays (or gene chips), has attracted tremendous interests among biologists. This technology promises to monitor gene expression on a single chip, so that researchers can have a better picture of the interactions among hundreds to thousands of genes simultaneously.

Micro-array experiments produce two-dimensional gene expression images; images that must be converted to numbers before analysis can proceed. These image files require significantly more storage than one-dimensional sequence data. Complex image analysis techniques are needed to extract quantitative cleaned-up expression data from the images.

Once the data is derived from the images, the computational problem can become one of unsupervised statistical data mining: looking for patterns of expression across thousands of genes (high dimensional data) from any number of samples (normally only containing a very limited number). Unsupervised techniques used include hierarchical clustering, k-means, or self-organising maps, in order to identify new subgroups or classes, and association analysis.

There are two ways in which clustering might occur. First, groups of genes may have a similar expression pattern across different samples. Because genes involved in the same functional pathway tend to have similar expression patterns, such clusters might provide insights into novel genes. The second type of clustering is if there are classes, such as tumour, disease or tissue type, in the samples. Once such groups are known, supervised data mining techniques can be applied. For example, to classify entities into known classes, such as tumours, diseases or therapies, one could apply discriminant analysis, nearest-neighbour methods, artificial neural networks, Bayesian networks, support-vector machines, decisions trees, boosting, bagging, random forests, or independent component analysis.

Another important type of data in bioinformatics is the three-dimensional structural description of proteins and other biologically important molecules. Although there are computer-based efforts to determine protein structure from basic sequence information, the full 'protein-folding' problem is considered a grand challenge in the domain of advanced supercomputing research. Three-dimensional protein structure carries much more biologically relevant

information than the one-dimensional sequence of amino acids, and there are several efforts dedicated to increasing the throughput of structure determination.

Since it is data-rich, but lacks a comprehensive theory of life's organisation at the molecular level, bioinformatics seems ideally suited for statistical data mining approaches, However, data mining is hampered by many facets of biological databases, including their size, their number, their diversity and the lack of a standard ontology to aid the querying of them, as well as the heterogeneous data of the quality and provenance information they contain. Another problem is the range of levels and domains of expertise present amongst potential users, so it can be difficult for the database curators to provide access mechanisms appropriate to all. The integration of biological databases is also lacking, so it can be very difficult to query more than one database at a time. For example, the massive amount of micro-array data collected so far has been generated on multiple platforms and is stored in different formats, levels of detail and locations. This makes it difficult for any research group to re-analyse or verify the data, or compare the results with their own. Finally, the possible financial value of, and the ethical considerations connected with, some biological data means that the data mining of biological databases is not always as easy to perform.

Data mining and bioinformatics are fast expanding research frontiers, and will inevitably grow toward each other – because bioinformatics will not become knowledge discovery without statistical data mining and thinking. A maturity challenge for statistical data miners and bioinformaticians is to widen their focus until true collaboration and the unlocking of the secrets of the cell become reality.

Diego Kuonen is CEO of Statoo Consulting, Lausanne, which provides statistical consulting, data analysis and data mining services. Email: kuonen@statoo.com

Complex systems: tested with confidence?

Bayesian belief networks hold the key to testing complex systems systematically, David Wooff believes

When it comes to a small system, such as the software used to timetable classes in a small school, a human being – perhaps one of the teachers – should be able to test the software effectively. There are only a few inputs and outputs. The teacher probably has good experience of previous years' timetables, and so they can use this information to guide the testing process. The teacher can focus testing so as to spot obvious problems or to highlight areas where the teacher has least confidence – for example a software upgrade allowing new functionality. The teacher might make some mistakes, but would otherwise be expected to do a pretty good job. More or less everything is contained 'in the tester's head' – the tester has sufficient personal expertise, and the problem is sufficiently small-scale, to allow good testing.

For a much larger system, such as the software used to timetable classes across a large multi-faculty, multi-site university, there may still be a single person with overall responsibility for testing the software. There is usually still a lot of expertise and historic information available to focus testing, if only the tester knew how. Generally, it is possible to test only a small fraction of the many combinations of inputs, and to check the resulting outputs. Where does the tester begin? The problem has outgrown, by orders of magnitude, what the tester might hope to accomplish in their head.

There are three principles as to how to tackle the testing of large-scale, complex systems. First, the testing problem is statistical, in that it involves handling and managing uncertainties. For example:

- How much confidence is there in the quality of the system and the development process?
- How much testing is needed? How much time and resource and how many people are needed for a particular level of reliability?
- When there is an existing test set, how

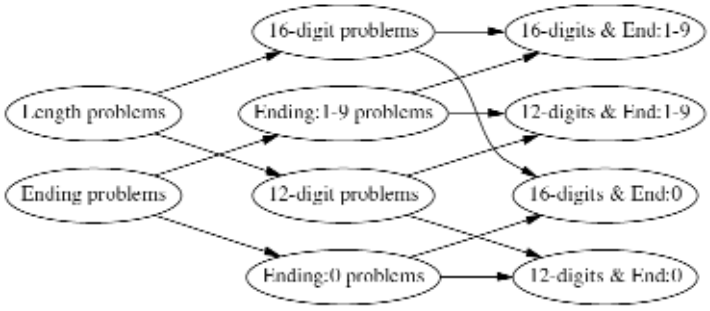


Figure 1: Processing a credit card number with two, possibly related, failure modes: faults in processing card numbers of different lengths; and faults in processing the ending digit. Nodes in the right-hand column represent combinations which can be observed through tests. The judgments of testers are expressed: (1) via a structure which relates potential failure modes to the observables; (2) by providing order-of magnitude probabilities for the different failure modes in the left-hand column (root node probabilities), and the likelihoods that the failures are transmitted downstream (arc probabilities). The effect of tests is to update probabilities in the right-hand column, and these effects then propagate back through the rest of the model to update the remaining probabilities. Utility nodes (for consequences) and implicit nodes have been omitted from this simplified diagram.

effective and efficient is it?
- Given a vast number of possible combinations of inputs, which combinations are actually tested? Indeed, how can tests be designed automatically?
- What are the risks of not running certain groups of tests and so leaving some functionality untested?
- How to take into account that some consequences of failure are more important than others?
- How to optimise the testing process to minimise time and cost?
- How can the tester tell if the system is fit to be released? and
- When faults have been found and fixed, how much re-testing will need to be done?

The second principle is that good testing should exploit existing expertise. For example:
- Testers often have a good idea as to which parts of the system are likely to contain the most faults; and
- Testers may be able to relate a testing problem to a previous one, and thereby make judgements about the resources required to test the new system.

The third principle is that the testing process needs sound organisational structures. As the scale of a problem increases, the tester's ability to manage and organise the testing of a system's highly interdependent components and their various combinations of inputs becomes impossible without recourse to organising tools.

Many testing approaches pay at least some attention to the principle that testing be statistical, and all offer some kind of organising principle. However, the statistical element is often either crude or limited. For example, experimental design techniques have been used to choose combinations of inputs for testing – this can be useful insofar as efficient choice of tests is concerned, but can't answer any of the uncertainties posed above, because there is no meaningful underlying metric allowing them to do so.

What then does it mean to ask whether we can test with confidence? Suppose we trust our tester. The actual testing process begins with the tester having some degree of belief in the reliability of the system, followed by tests, and the fixing of any faults revealed, following which the tester's confidence in the reliability of the system should increase, to the extent to which they are happy to pass it as tested, perhaps subject to stated constraints. The tester's 'confidence' may be difficult to define and measure, but it is inherently the only natural and meaningful metric available.

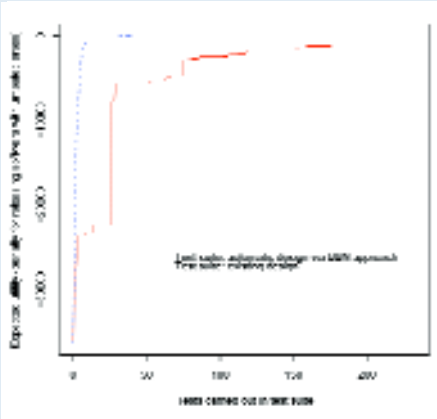


Figure 2: The BBN approach allows costs of failure, on a scale meaningful to the company, to be attached to each kind of failure. As tests are run and faults fixed, the probability that the system contains faults diminishes and the expected overall penalty for releasing the system still containing undetected faults falls. It is straightforward to assess the implications of competing test suites.

This is where our second principle comes in. There is an existing core methodology that brings together the three principles of employing statistical procedures, exploiting existing expertise, and using appropriate organisational structures: it is Bayesian statistical methodology, using Bayesian belief networks (BBNs) in particular. This methodology has been developing rapidly over the past 15 years or so, and is becoming the dominant form of statistical methodology for complex modelling and decision problems, such as occur in medical science.

BBNs, which are usually portrayed as graphs of interconnected variables, represent uncertainties via probabilities and relationships between variables via conditional probabilities. A system-testing problem can be organised via collections of BBNs, each representing a part of the system that the tester deems independent of other parts as far as reliability is concerned. System functionality is decomposed into nodes on the graph, according to the tester's best knowledge of the implications of a test, to whatever level of detail is deemed reasonable. Informally, if the result of a test on one variable has implications for other variables, those variables should be organised to be within the same BBN.

Existing knowledge about the likely initial faultiness of the system is established using probability as the metric, this being the appropriate mathematical language for uncertainty. Possible tests are mapped directly to the variables in the BBNs. Tests that have been carried out then provide the data that is used to update the BBNs. The effect of the updating from tests is to increase or decrease the probabilities for faultiness of the subsystems represented by the BBNs. Figure 1 shows a simple example of a BBN constructed to test processing of 12- and 16-digit credit card numbers, where the final digit is a special indicator, and where the tester has judged that there are two possible failure modes.

Once the model and the probability specification have been constructed, the full panoply of coherent statistical methodology can be brought to it. All the uncertainties given above can be addressed; diagnostic checks and sensitivity analyses can be handled routinely. Consequences of faults can be measured on a utility scale, and this – together with information on the resource required for undertaking certain tests – can be used to design tests to maximise expected utility, to provide information on the likely length of testing before the system reaches a desired level of reliability, to give a complete approach for controlling risk, and so forth. In short, once these structures are created, all the desired statistical and management tools follow naturally, and indeed almost trivially.

In the next issue, I shall describe how the approach was used for testing the software to manage the credit card operations for a major telecommunications company. The methodology is appropriate to any high-reliability testing problem characterised by complexity, large numbers of combinations of inputs to test, and significant uncertainties.

David Wooff is Director of the Statistics and Mathematics Consultancy Unit, & Senior Lecturer in Statistics, University of Durham. Department of Mathematical Sciences, Science Laboratories, South Road, Durham, DH1 3LE, UK. email: d.a.wooff@dur.ac.uk

This research was developed at Durham University with Frank Coolen and Michael Goldstein, in collaboration with British Telecommunications Plc.