

Conceptual Modelling of Genomic Information

Norman W. Paton¹, Shakeel A. Khan², Andrew Hayes², Fouzia Moussouni¹, Andy Brass², Karen Eilbeck², Carole A. Goble¹, Simon J. Hubbard³ and Stephen G. Oliver²

¹Department of Computer Science, University of Manchester,
Oxford Road, Manchester M13 9PL, UK
(norm,fouzia,carole)@cs.man.ac.uk

²School of Biological Sciences, University of Manchester,
Oxford Road, Manchester M13 9PL, UK
(abrass,steve.oliver)@man.ac.uk

³Department of Biomolecular Sciences, UMIST,
PO Box 88, Manchester 1QD, UK
sjh@sjh.bi.umist.ac.uk

Abstract

Motivation: Genome sequencing projects are making available complete records of the genetic make-up of organisms. These core data sets are themselves complex, and present challenges to those who seek to store, analyse and present the information. However, in addition to the sequence data, high throughput experiments are making available distinctive new data sets on protein interactions, the phenotypic consequences of gene deletions, and on the transcriptome, proteome, and metabolome. The effective description and management of such data is of considerable importance to bioinformatics in the post-genomic era. The provision of clear and intuitive models of complex information is surprisingly challenging, and this paper presents conceptual models for a range of important emerging information resources in bioinformatics. It is hoped that these can be of benefit to bioinformaticians as they attempt to integrate genetic and phenotypic data with that from genomic sequences, in order to both assign gene functions and elucidate the different pathways of gene action and interaction.

Results: This paper presents a collection of conceptual (i.e., implementation-independent) data models for genomic data. These conceptual models are amenable to (more or less direct) implementation on different computing platforms.

Availability: Most of the information models presented here have been implemented by the authors using an ob-

ject database. The implementation of a public interface to this database is work in progress. We hope to have a public release in the Autumn of 2000, available from <http://img.cs.man.ac.uk/gims>.

Contact: norm@cs.man.ac.uk

Introduction

The recent availability of complete genome sequences provides biologists with new opportunities for identifying and understanding properties of the genome that have hitherto been out of reach, and for conducting comparisons of genomes. The exploitation of this new information resource is, however, dependent upon the provision of effective tools for the management, integration and presentation of genome sequences and related information.

The storage, sharing, and analysis of genomic data sets is made more challenging still by the emergence of new information resources for which there are few established bioinformatics techniques, such as transcriptome data from hybridisation-array analyses. The fact that many insights are likely to emerge from the combined use of core genome sequence data with the data on functional analyses in turn implies that consistent and integrated representations of genomic data are likely to be important to post-genomic bioinformatics.

This paper provides conceptual models that describe

eukaryotic genome sequence data and genome organisation, plus a number of important functional data sets, namely protein interaction data, transcription data, and results from gene deletions. The information models presented have been developed in the context of a project that is focusing initially on the management of *Saccharomyces cerevisiae* data, but where the models are biased towards *S. cerevisiae*, we seek to make this explicit. The models are described using UML (Booch *et al.*, 1999), the emerging standard object-modelling language.

The work described in this paper is by no means unique in providing object-oriented models of biological data. One of the authors was involved in an early project on the use of object databases with protein structure data (Gray *et al.*, 1990), but there is more recent work that presents conceptual or object-based models for biological data. For example, (Okayama *et al.*, 1998) describes the conceptual schema of a DNA database using an extended entity-relationship model, (Chen & Markowitz, 1995) has indicated how an extended object data model can be used to capture the properties of scientific experiments, and (Medigue *et al.*, 1999) includes models for representing genomic sequence data.

A number of researchers have also explored the use of object-oriented implementation technologies for biological data. For example, (Ellis *et al.*, 1998) describes how a metabolic pathway database has been developed using Java and a commercial secondary storage manager. The ooTFD transcription factors database is outlined in (Ghosh, 1999). In the genomic setting, (Hu *et al.*, 1998) indicates how the object-oriented middleware CORBA has been used to provide distributed access to a genome mapping database, and (Jungfer & Rodriguez-Tome, 1998) describes how CORBA was employed in the construction of a map viewer. Most closely related to the current work is that described in (Eilbeck *et al.*, 1999), which describes the implementation of a protein interaction database that uses one of the conceptual models presented in this paper. In (Maltchenko, 1998), a proposal is made for some generic object classes for use in a distributed computing environment; this seems less suitable to the authors than a carefully defined collection of less generic classes.

It is probably also important to differentiate this work from the earlier work of some of the authors on an ontology for biological data in the TAMBIS project (Baker *et al.*, 1999). The purpose of an ontology is not so much to provide a conceptual representation of structures for storing data, but rather to provide a description of the terminology used in a domain. As a result, for example, it would not be obvious how to derive a database schema from the TAMBIS ontology. In addition, a fur-

ther distinction between the TAMBIS ontology and the models presented in this paper is that the latter focus on genomic data sets, which are outside the scope of the TAMBIS ontology.

The purpose of this paper is to provide clear conceptual models for genomic data that can be used to direct subsequent implementation activities. By separating the information models from a description of a system, it is hoped that important issues relating to the way in which data can be described will be made explicit, to the benefit of developers of future information systems for genomic data.

Systems and Methods

This section presents the information models using the class diagram notation of the Unified Modelling Language (UML) (Booch *et al.*, 1999).

In UML class diagrams, such as the one in Figure 1, classes are drawn in rectangles, with the name of the class at the top, and optionally with the attributes and operations of the class shown below. In this paper, we list the attributes of the class only when these are felt to be important to the understanding of the model as a whole – listing all the reasonable attributes of all the classes would consume a prohibitive amount of space. Generalization relationships (e.g. *Terminator* and *Promoter* are both kinds of *Regulatory Sequence*) are drawn using a line with an arrowhead at the most general class. Relationships between classes are represented by lines connecting the classes, with the name of the role that the class plays in the relationship written beside the line, along with the multiplicity, which indicates the number of objects that may participate in the relationship. Where the relationship represents a part/whole relationship, the line depicting the relationship is adorned with an open diamond at the whole end of the relationship. For example, a *Genome* contains *many* (denoted by *) *Chromosomes*, each of which is part of *one* *Genome*.

Genome Sequence Model

This subsection describes the basic information that must be stored to describe a fully sequenced genome. The focus is on the fully sequenced genome, rather than on the information from which the final sequence is derived. The schema diagram for the core data set is presented in Figure 1.

The model in Figure 1 describes the basic components of a genome. The complete *Genome* consists of a collection of chromosomes. Each *Chromosome* can be considered to be a long sequence of DNA, which in turn

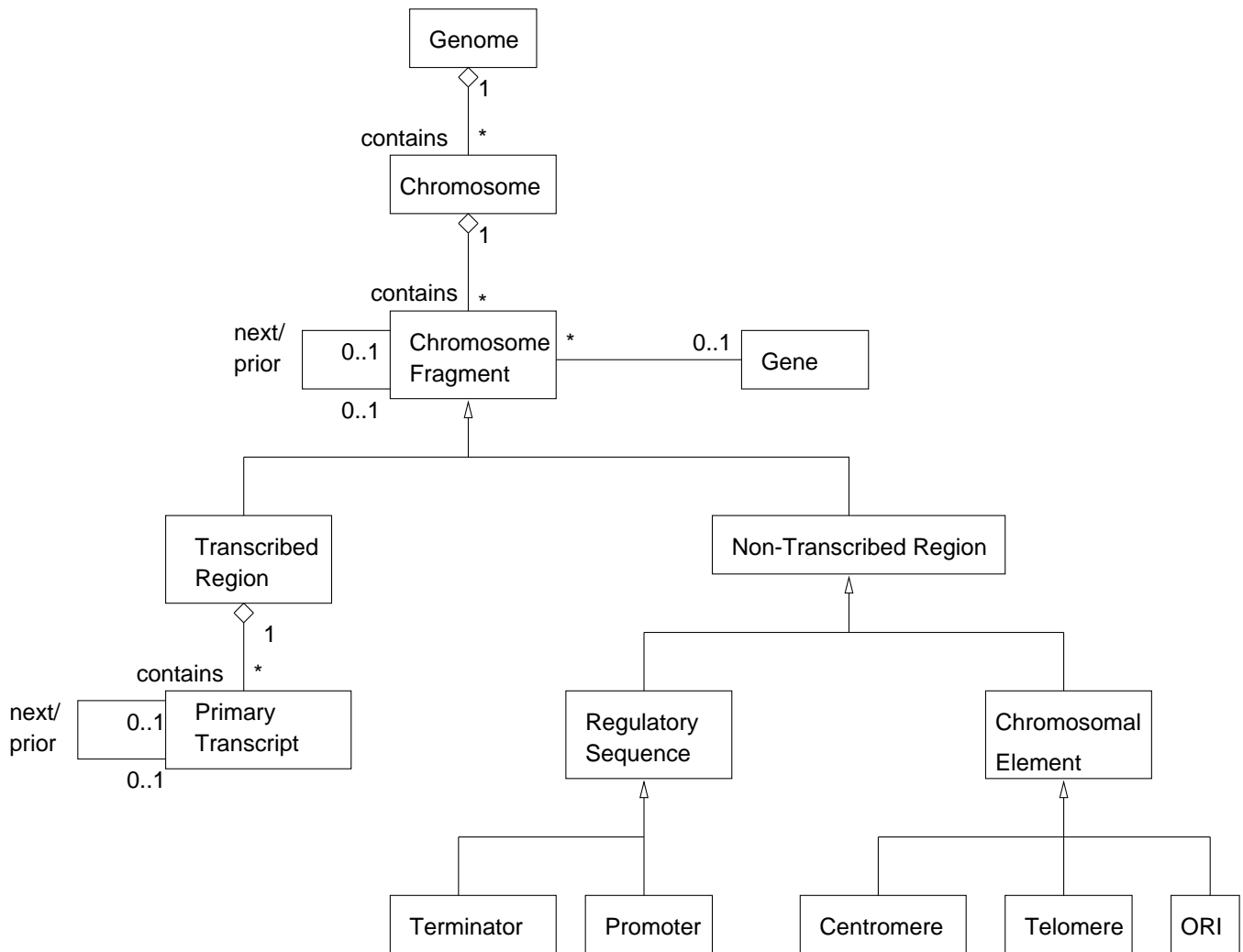


Fig. 1. Basic schema diagram for genomic data.

consists of a sequence of (potentially overlapping) *Chromosome Fragments*. These fragments are either *Transcribed* regions of DNA or *Non-Transcribed* regions (i.e., both *Transcribed* and *Non-Transcribed* regions are kinds of *Chromosome Fragment*).

Different approaches to the modelling of genomic information may place different interpretations on some biological terms, even for familiar notions such as *Gene* and *ORF*. In the model in Figure 1, a *Gene* is a segment of the chromosome that is transcribed into RNA. Its flanking non-transcribed sequences/regions (the *Regulators*) are included as parts of the *Gene*. There is a possibility that two adjacent genes may overlap, where a part of the coding sequence, of the preceding one, acts as the promoter for the following one. This can be handled in the model because *Chromosome Fragments* may over-

lap. Thus, within the model (as in genetics), “gene” is defined in functional terms and no artificial constraints are placed on that portion of the DNA sequence that represents a “gene”.

The *Non-Transcribed* regions are either *Regulators*, which control the expression of genes, or *Chromosomal Elements*, which include the *Centromere*, the *Telomeres* and the Origins of Replication (*ORI*).

The *Transcribed* chromosome fragments are illustrated more fully in Figure 2. Each *Transcribed* region contains a collection of *Primary Transcripts*. Each of the primary transcripts is either an *Intron* or a *Spliced Transcript Component*. In the model, there is no direct relationship between *Intron* and *Regulator*, and information on the regulatory function of an *Intron* would be captured using attributes of *Intron*. In addition, the model in its

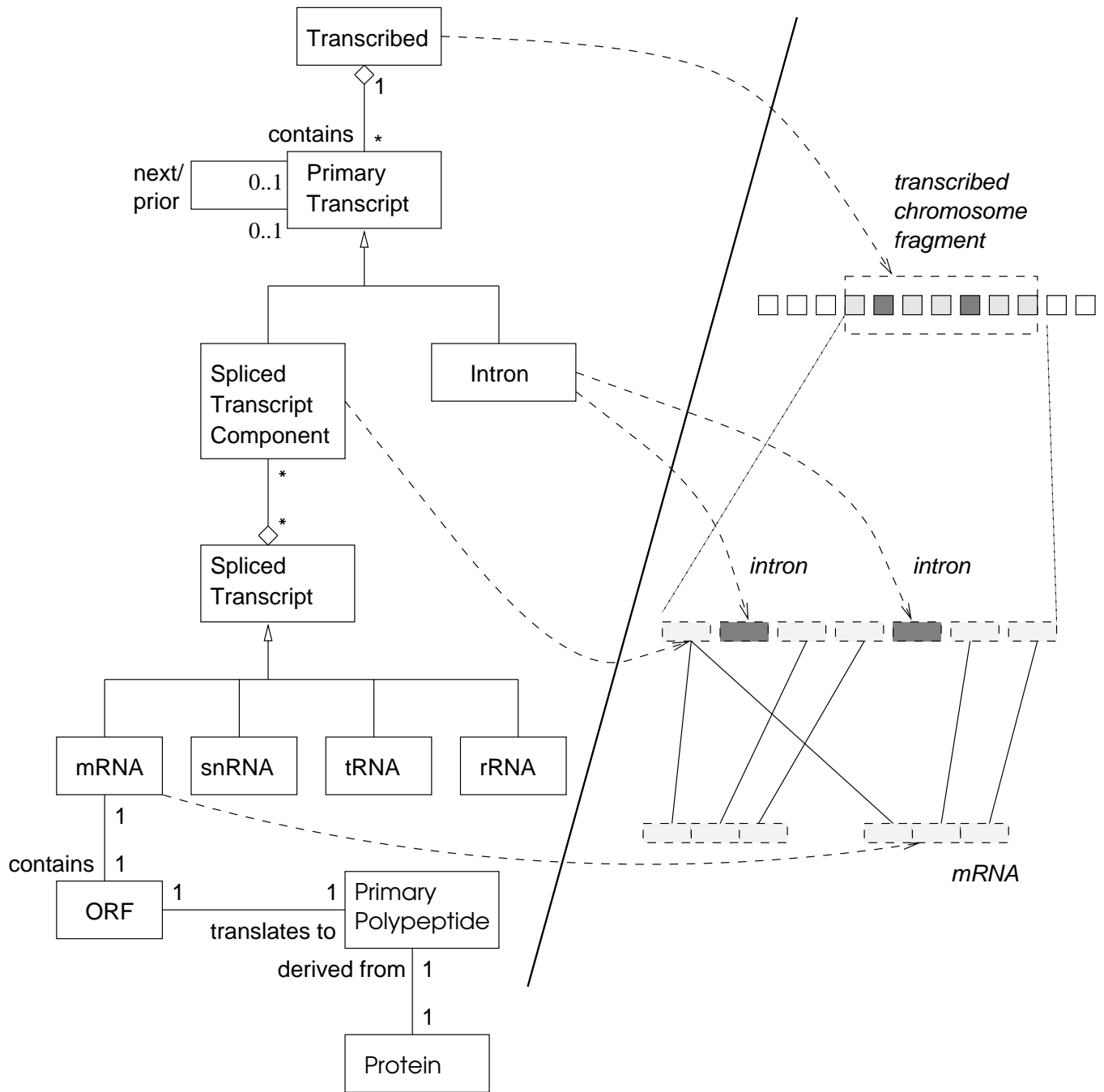


Fig. 2. Fragment of core schema diagram with illustration of associated data.

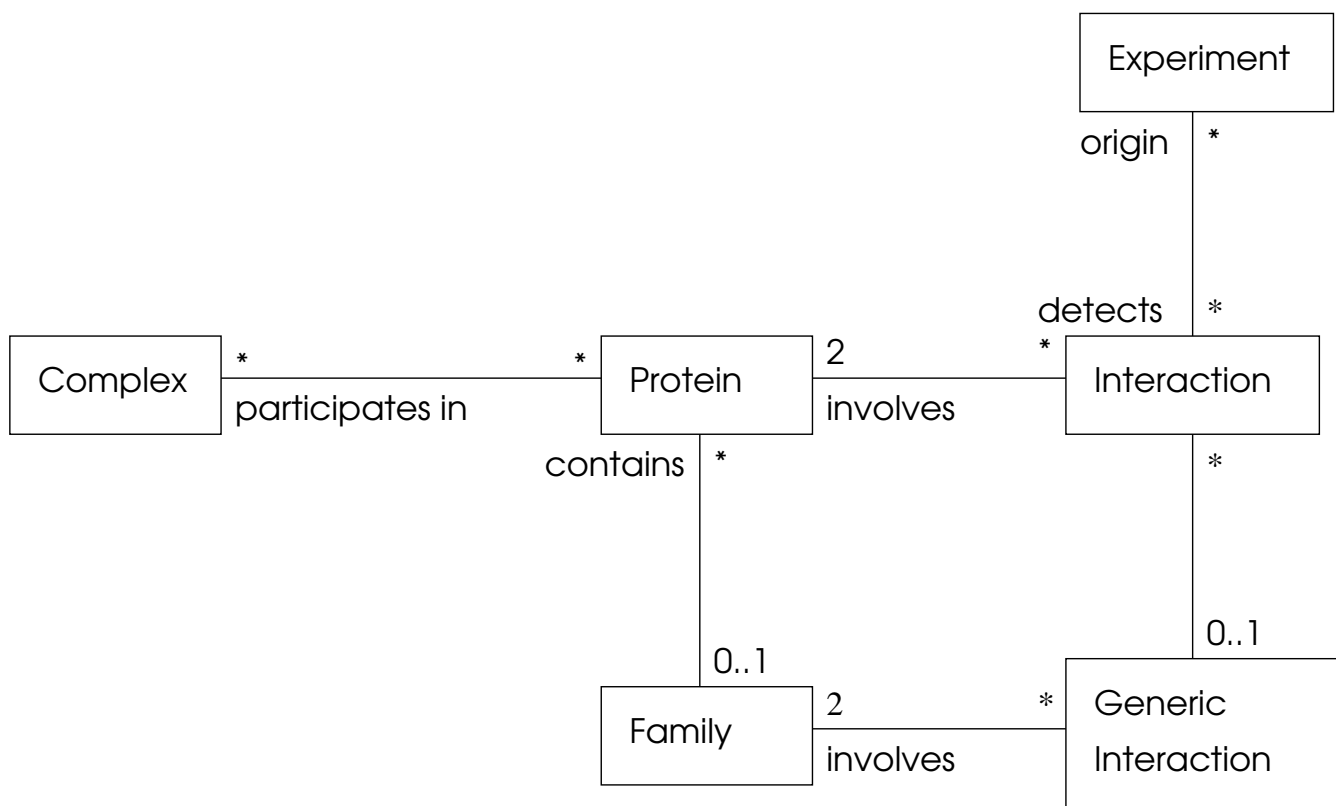


Fig. 3. Class diagram for protein interaction data

present form deals only with the simplest form of alternate splicing. While adequate for *S. cerevisiae*, this part of the model will have to be extended if it is to be applied to more complex eukaryotes.

The *Spliced Transcript Components* are assembled (with the *Introns* removed) to form *Spliced Transcripts*. Each *Spliced Transcript*, is categorised as being some kind of RNA. An additional complicating factor is that a single *Spliced Transcript Component* can be used in the synthesis of more than one *Spliced Transcript*. For example, Figure 2 illustrates two different *Spliced Transcripts* that share common *Spliced Transcript Components*.

Every *mRNA* contains an open reading frame (*ORF*), which consists of a series of triplets (codons) that specify the amino-acid sequence of the *Primary Polypeptide* that a gene encodes. The *ORFs* begin with an initiation (start) codon, usually ATG, and end with a termination codon, either TAA, TAG or TGA. The *Primary Polypeptide* undergoes certain post-translational modifications to become a functional *Protein*.

As the emphasis in the paper is on fully sequenced genomes, no models are provided for mapping data. An example of a model for mapping data is provided by

(Barrilot *et al.*, 1999).

There is a question as to where information on experiments or on the literature should be provided in class diagrams such as those provided in figures 1 and 2. As there could be interest in providing such information at many different points in the diagram, it might be appropriate to provide an abstract superclass that allows literature or experimental details to be provided for all classes in these diagrams.

Protein-Protein Interaction Data

This and subsequent subsections describe information models that are related to the core genome sequence. The models provided have been selected because they can be used to describe experimental results that are becoming available as a result of systematic functional analyses.

Protein-protein interactions are essential to most cellular processes such as signal transduction, DNA replication, and metabolism. Experimental techniques, particularly the yeast two-hybrid system, are being used to provide comprehensive analyses of interactions. For example, the protein-protein interactions of bacteriophage

T7 were mapped using yeast 2-hybrid to reveal 22 interactions between 53 proteins (Bartels *et al.*, 1996). Since then, attempts have been made to systematically map the interactions between the 6000 or so expressed proteins in yeast (Fromont-Racine *et al.*, 1997).

A class diagram for protein-protein interaction data is provided in Figure 3. This intersects with Figure 2 at the class *Protein*. In this model, every *Interaction* involves two proteins, which allows the model to capture the details about each interface. Each *Interaction* can in turn be validated by many *Experiments*. As the different experimental methods for protein interaction detection can produce results of differing quality, information on the experimental techniques used gives an indication of the confidence that can be held in the interaction. For example, an interaction detected by yeast two-hybrid and affinity chromatography would be considered to be more likely than one detected by yeast two-hybrid alone.

The class *Complex* contains information about proteins interacting in groups of more than 2. One source of complexes used to populate this section of the database is MIPS (Mewes *et al.*, 1998), in which there are 230 multi-protein complexes at time of writing.

The class *Generic Interaction* is used to describe categories of interaction that are apparent in many species. This generality is captured by relating individual *Interactions* with the *Generic Interaction* of which they are an example. Generic interactions are in turn described in terms of the gene families that participate in them. An example of a *Generic Interaction* is that the families *MAPKKK* and *MAPKK* interact in the pheromone signalling pathway. A specific example of this is that in *S. cerevisiae* the protein *Ste11* (a member of the family *MAPKKK*) interacts with *Ste7* (a member of the family *MAPKK*).

Transcriptome Data

Gene expression data provide important clues as to the function of novel genes identified during genome sequencing (Planta *et al.*, 1999), and also provide insights on the behaviour of cells in different conditions (Spellman *et al.*, 1998). The recent development of experimental techniques that allow transcript levels to be measured for every gene in an organism simultaneously (the transcriptome) will make available a substantial information resource that will, in turn, require novel bioinformatic techniques in the area of functional analysis. The focus of this section is on the modelling of transcriptome data.

Transcriptome data can be generated using different experimental techniques applied to DNA arrays that yield different kinds of data. A DNA array is generally prepared (a) by crosslinking a PCR product on a solid

support like a glass slide or a nylon membrane; or (b) by synthesising oligonucleotides on a gene-chip.

There are now a number of transcriptome studies on *S. cerevisiae* for which the data are publicly available. For instance, protein-encoding genes whose expression is phased in the yeast cell cycle have been identified by two groups (Spellman *et al.*, 1998). In this work, the relative abundance of each transcript in the test sample (from each time point of a yeast culture where the cells undergo division in a synchronous manner) is compared with the reference (time-zero) sample, by measuring as the ratio of red to green fluorescence. The greater relative abundance of a particular transcript (mRNA) in the test sample results in a higher ratio of red-labelled to green-labelled copies of the corresponding cDNA. From a modelling perspective, the key feature of the results is that a *relative* expression level, at different times in the cell cycle, is obtained on a *per-gene* basis.

In an alternative approach, by EUROFAN, the regulation of expression of all protein-encoding genes in yeast was investigated under carbon and nitrogen limitation conditions. Following normalisation, the abundance of each transcript is expressed as its fractional contribution to the total level of transcription. To identify genes under carbon or nitrogen control, levels of expression of individual ORFs are compared between the two conditions. From a modelling perspective, the key feature of the results of this experiment is that each data item is a normalised *absolute* expression level for each gene.

A fragment of data from an absolute-value expression experiment is provided in Figure 5. This Figure shows two filters that have been hybridised with radiolabelled cDNAs prepared from mRNA extracts taken from steady-state carbon-limited and nitrogen-limited cultures. Each “spot” represents an ORF that is expressed under each condition. The intensity (or darkness) of the spot is proportional to the level of expression of the particular gene. The spots that have been highlighted with arrows show major differences between the two conditions. The context (e.g. growth condition) in which a particular gene is expressed provides clues as to its function.

The class diagram for transcriptome data is given in Figure 4. The *mRNA* class is common to this schema diagram and that in Figure 2. Arrays employed in yeast transcriptome analysis currently comprise oligonucleotides or PCR products containing sequences exclusively derived from ORFs. The current model does not deal with the fact that, because of genetic redundancy, more than one spot in the array may be hybridised by a single mRNA class (Delneri *et al.*, 1999).

There are two kinds of *Experiment*, *Absolute* exper-

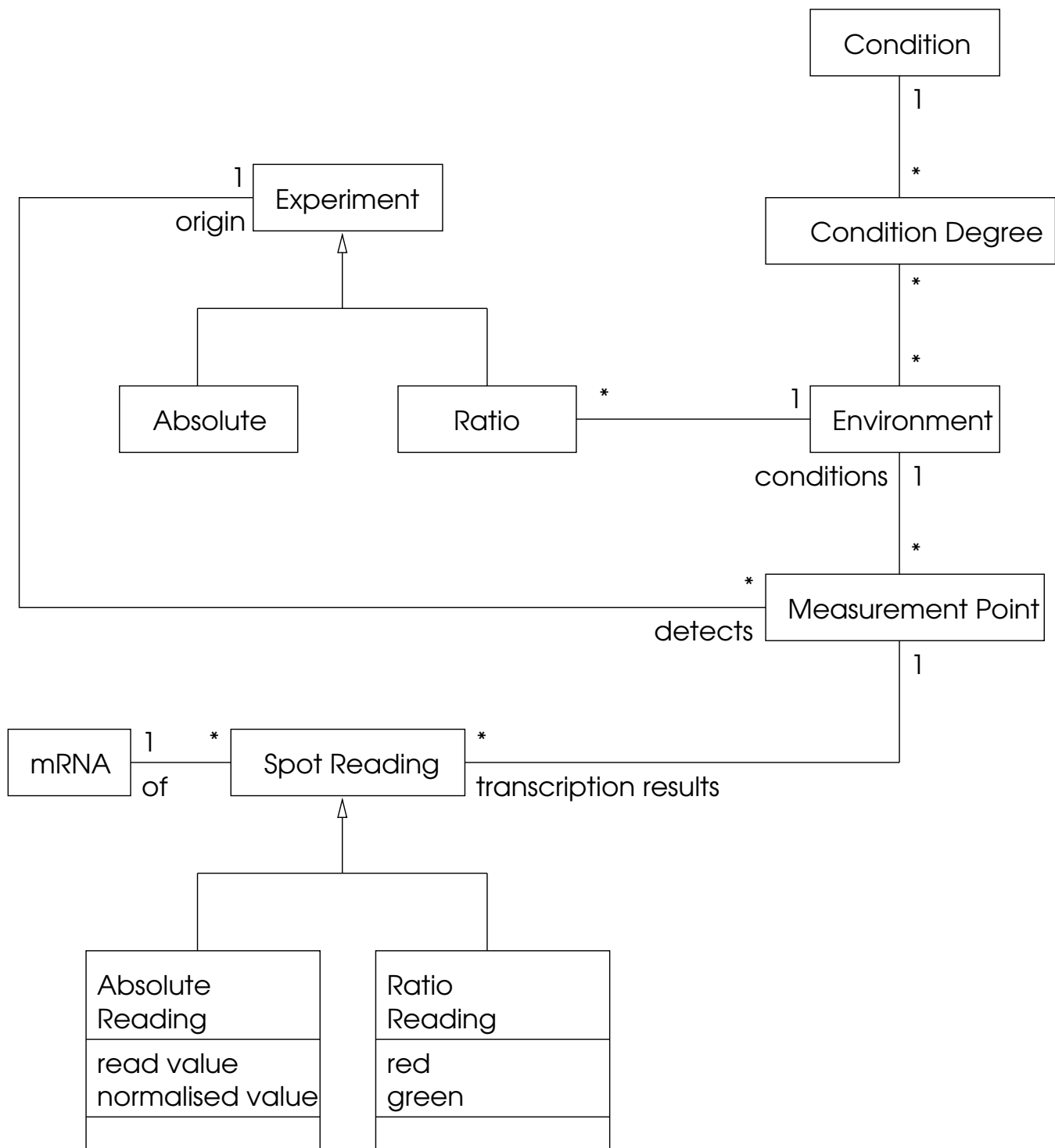


Fig. 4. Schema diagram for transcriptome data.

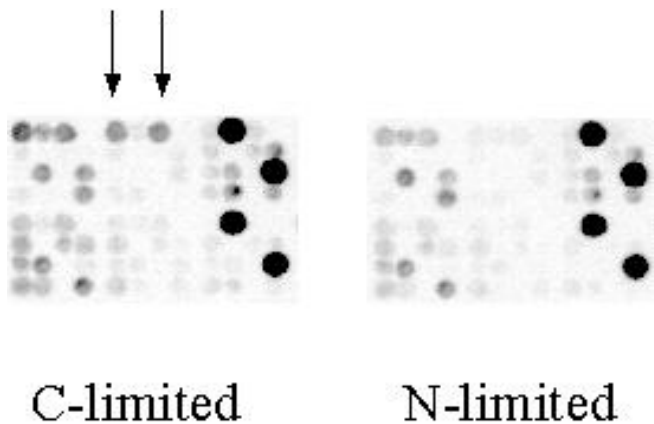


Fig. 5. Example expression data results.

iments and *Ratio* experiments¹ reflecting whether the technique used in the experiment measures the absolute quantity of mRNA expressed or the amount of mRNA expressed relative to some baseline (in practice, whether the two-colour fluorescence technique or radioisotopic labelling was used). Each *Experiment* has a collection of *Measurement Points*, each of which is at some time from the start of the experiment. The *Environment* of a *Measurement Point* is a description of the conditions that prevail at the time of the measurement. In particular, an *Environment* records the extent (i.e., *Condition Degree*) to which some property (i.e., *Condition*) holds when a measurement is taken.

The relationship between *Ratio* and *Environment* captures the environment of the baseline. Thus in a *Ratio* experiment in which there are n *Measurement Points*, there will be $n+1$ *Environments*, one for each *Measurement Point* and one for the baseline.

Each *Measurement Point* associates the description of the *Environment* with a collection of *Spot Readings*. Each *Spot Reading* is either an *Absolute Reading* or a *Ratio Reading*, depending on the type of *Experiment* being carried out. An *Absolute Reading* captures the quantity of a particular gene expressed at the time of the measurement, plus some normalised value (such as the % of the total mRNA present in the cell that this reading represents).

To relate the model with the experimental values in Figure 5: each *mRNA* species represents a position in the array; the *read value* of each *Absolute Reading* represents the intensity of the “spot” at a specific position; there is a separate image of the array for each *Measurement*

¹Although in ratio experiments the ratios are computed from absolute expression levels, the absolute values are not normally made public.

Point, and the *Condition* and *Condition Degree* classes represent the labels on the images of the arrays, such as *C-limited* and *N-limited*. In the case of *C-limited*, the *Condition* is carbon limitation, and the degree of the condition is the dilution rate (e.g., $0.1h^{-1}$).

The rapidly increasing production of transcriptome data sets has led to a number of proposals for modelling such data. For example, (Brazma *et al.*, 1999) presents a design for a repository of array data. The models for transcriptome data presented in this paper focus principally on the information that is derived from the array experiments, whereas the models in (Brazma *et al.*, 1999) seek to capture as much detail as possible on the techniques used and the images obtained. We consider that the wider scope of the models in (Brazma *et al.*, 1999) is appropriate in the context of a repository, and that the emphasis in the models presented in this paper is appropriate for an information resource that focuses on the integration of different kinds of information for analysis purposes.

Genome Modifications

Genome sequencing projects reveal the presence of many genes with unknown functions. Knowing the function of all the genes in the genome is important to the long-term value of the genome data, and systematic approaches can be taken that seek to assign functions to novel genes. For example, upon completion of the yeast genome sequence, an international, integrated effort was initiated to determine the possible functions of as many novel genes as possible (Oliver, 1996). In order to achieve this, ORFs with unknown functions were deleted by a PCR-mediated gene replacement method (Baudin *et al.*, 1993; Wach *et al.*, 1994).

The modelling of experiments relating to naturally occurring or induced modifications to a genome involves describing the modification that has taken place, the ploidy of the strain, and the consequences of the modification for the organism.

The schema for describing genome modifications is given in Figure 6. A *Strain* is described as a collection of modifications to *Genes* in a *Genome*. A *Strain* is associated with a collection of variations in the phenotype of the organism. In experiments involving yeast, the *effects* of a *Variation* that are recorded include the viability and rate of growth of the strain. This is straightforward to describe, but modelling phenotypic behaviour in other organisms is likely to present a significant challenge for controlled vocabularies or ontologies, and is beyond the scope of this paper.

The interpretation of the significance of a variation in phenotype depends on the ploidy of the strain. For ex-

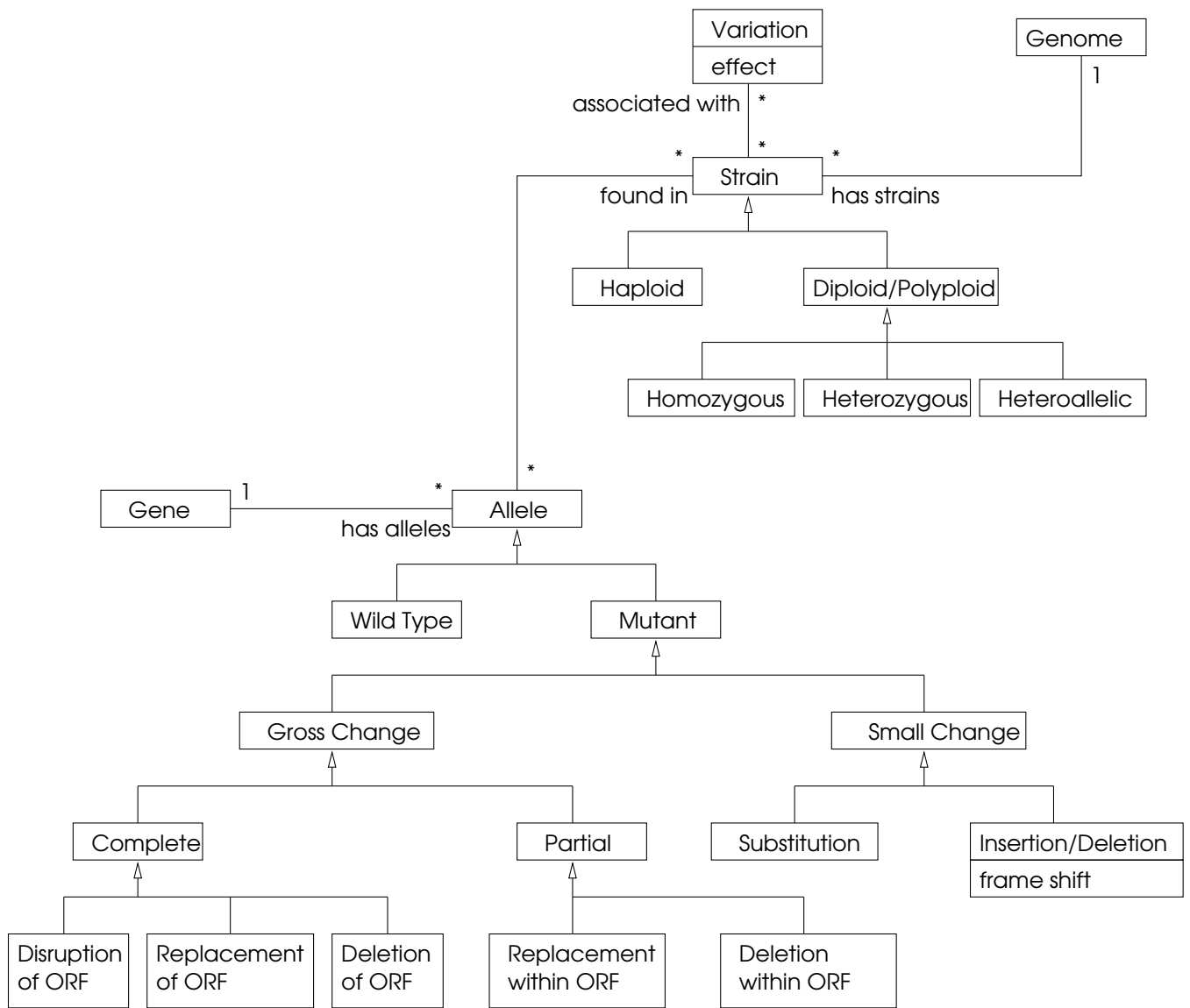


Fig. 6. Schema diagram alleles.

ample, if an essential novel gene is deleted from a diploid strain, the strain may still show viability, whereas a haploid strain is sure to be non-viable. In the model in Figure 6, a *Strain* may be *Haploid* or *Diploid/Polyploid*, reflecting the number of copies of the chromosomes in the organism.

Each of the modifications to a *Gene* in a *Strain* is represented as an *Allele*. *Alleles* are classified as either *Wild Types* or *Mutants*. The model for describing *Mutants* is much richer than that for *Wild Types*, as the nature of the genetic change in any marker genes may not be known.

The modifications within a *Mutant* allele are classified according to their size. This is by no means the only possible dimension that could be used for classification – for example, the hierarchy could have been based on the type of the modification (e.g. insertion, replacement). During modelling, a range of alternative classifications were explored, and that based on the size of the modification seemed the most natural for use with the data sets to which we had access. This, however, is the only model in the paper that has not yet been implemented and used with real data sets, and thus the model has yet to be validated in practice. Whatever dimension is chosen for the classification of the modifications is likely to seem somewhat arbitrary.

A *Small Change* is a modification involving not more than three nucleotides, and may be either a *Substitution*, an *Insertion* or a *Deletion*.

A *Gross Change* is any larger modification to a gene. A *Complete* change involves either the *Disruption* of the gene (i.e., inserting a sequence into the gene that prevents the expression of a functional product), the *Replacement* of the gene with another gene that can be expressed, or the *Deletion* of the entire gene. A *Partial* change in the gene involves either *Replacement* or *Deletion* of a length of sequence within the gene.

An example of an experiment that yields data that can be captured using the model in Figure 6 is described in (El-Moghazy *et al.*, 1999). In that experiment, one of the two copies of YLL035w, that encodes a hypothetical protein in yeast, was replaced with the KanMx marker gene in a heterozygous diploid strain of *S. cerevisiae* (FY1679). This can be represented in the model as a *Diploid* strain, where the strain has two *Alleles* for the *Gene* YLL035w, one of which is a *Wild Type*, and the other a *Mutant*. The *Mutant* is an example of a *Partial Replacement*. The subsequent popping out of the marker gene by homologous recombination resulted in a partial deletion of YLL035w. Sporulation resulted in 4 haploid spores (2 with the same wild-type allele and 2 with the same partially deleted allele). Spores with the wild-type

allele did not show any change in phenotype, while the effect of deletion on the haploid was the loss of its viability. This latter case is represented in the model as two *Haploid* strains. One of these strains has one *Wild Type* allele, and the other has one *Mutant* allele. The *Mutant* allele is a *Partial Deletion*.

A database of human mutations, including a description of a relational data model, is given in (Attimonelli *et al.*, 1999).

Implementation

This section presents an overview of the implementation context for the previous models under development in the Genome Information Management System (GIMS) project. GIMS can be seen as a scientific data warehouse. A data warehouse is a repository for data that is also available elsewhere, where the data is replicated to allow complex analyses over the data. In (Widom, 1995), it is stated that the warehousing approach is appropriate in applications in which:

- Clients require specific, predictable portions of the available information.
- Clients require high performance.
- Native applications at the information sources require high performance.
- Clients want access to private copies of the information so that it can be modified, annotated and summarised.

All of these points are applicable to genome information management. However, there are a number of ways in which genome information management differs from classical business data warehousing applications (Anahory & Murray, 1997):

- The core data set is much more complex (i.e., the core data set is that of the genome, rather than, for example, a collection of sales transactions).
- The data sets that “surround” the core data set in *star schemas* are themselves likely to be complex, containing experimental results (e.g. expression data, protein-protein interaction data), rather than more straightforward product or supplier information.
- The role of aggregation in analyses is currently seen as less central to genome information management. Instead, the complexities in the analysis of genomic data often come from the substantial number of data

types that may be visited during a single task, as analyses often involve navigation through a wide range of objects.

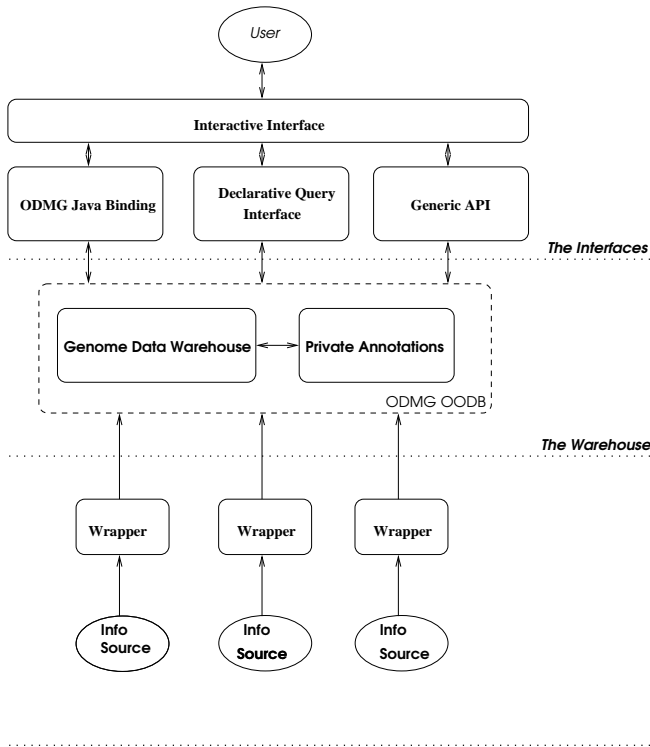


Fig. 7. Architecture of GIMS system.

The architecture of the GIMS system is outlined in Figure 7. The core warehouse is an ODMG compliant object database (Cattell *et al.*, 1997), in our case POET. POET was chosen largely because of its conformance with the ODMG standard, as we were keen not to develop a highly vendor-specific implementation. The database is accessed by the user interface through the standard ODMG Java binding, the declarative query language OQL, and a generic application programming interface. The latter is necessary because some features of the user interface require access to the database in a manner that is data type independent. For example, the classes used to implement forms in the browser are generic in the sense that their implementation contains no domain-specific code – the same user interface classes could be used to browse instances from any database.

At present, the core warehouse is populated principally using data from MIPS (Mewes *et al.*, 1997), and associated information is generally drawn via wrappers from remote sites. The principal way of accessing data within GIMS is expected to exploit *canned queries* – a canned query is a parameterised request for the result

of a query or an analysis which has been anticipated by the developers of the database and coded as part of the interface. An early activity in the GIMS project was a requirements analysis which identified over 50 analysis tasks that potential users indicated they might be interested to ask over a genome information management system. These tasks will form the starting point for a canned query interface in a public version of GIMS in due course.

```
public class Genome
{
    private String    name;
    private String    organism;
    private String    source;
    private double    size;
    private SetOfObject hasChromosomes;
    ...
}

public class Chromosome
{
    private String    number;
    private int       size;
    private Genome    hasGenome;
    private ListOfObject hasFragments;
    ...
}
```

Fig. 8. Java definitions for *Genome* and *Chromosome*.

To give a flavour of the implementation, figure 8 contains parts of the definitions of *Genome* and *Chromosome* in the syntax of the POET Java binding. Classes from figure 1 are represented by database classes, and relationships are represented by attributes of these classes. For example, the fact that a *Chromosome* is associated with a single *Genome* is represented by the attribute *hasGenome* of *Chromosome*, and the fact that a *Chromosome* contains an ordered collection of *Chromosome Fragments* is represented by the *hasFragments* attribute of *Chromosome*. Inverses are stored for all relationships, so that they can be explored in either direction. As well as the information provided by the Java class definitions, POET also provides a configuration file for describing indexes, extents, etc.

Discussion and Conclusions

This paper has presented conceptual models for describing both genome sequences and related functional data sets. Although conceptual models can seem quite

straightforward with hindsight, they are often difficult to develop; construction of the models presented in this paper was a lengthy and iterative process. The models produced have served as the starting point for an implementation activity in which the UML class diagrams are mapped to an object database schema. However, as conceptual representations of the data sets, these models could equally be mapped to different implementation platforms, such as alternative data models, or to object models for handling distributed data (e.g. CORBA) or transient data (e.g. Java).

The biological focus of the paper is, however, the key motivation for this work. Genome sequencing and functional genomics projects are making available new information resources that will motivate the development of a new generation of genome-level bioinformatics. However, it is clear that the information storage and analysis challenges of genomics are extremely great. Effective analysis of genomes will require high performance access to a diverse collection of complex information resources, which are typically developed at a range of sites. These information resources are made available in a wide range of formats, and generally in a form that allows the data to be browsed or downloaded, but not necessarily analysed effectively in conjunction with other information sources.

A number of researchers have investigated tools for distributed querying of biological sequence information sources (e.g. (Buneman *et al.*, 1995; Chen *et al.*, 1997; Baker *et al.*, 1998)). While some of us are involved in one such activity, we believe that the warehousing approach will be at least as relevant in the genomic area. The key difference in genomics is that typical analyses are likely to be much more complex than in non-genomic bioinformatics. In genomics, value is added through the close association of specific data resources, so that genomes can be compared with each other, and information from high throughput experiments, such as those relating to gene expression and protein-protein interactions, can be easily associated and displayed.

This paper seeks to provide one building block that is important to post-genomic bioinformatics – information models for a range of important, genome-level data sets. To be of practical use such models must be implemented, and search, analysis or visualisation techniques developed for use with the models. However, all of these tasks will be made significantly more straightforward if the underlying data sets are described in a consistent and coherent manner. In providing high-level models of important data sets, we hope to provide a worthwhile foundation on which others can build.

Acknowledgements: This work is supported by the BBSRC/EPSRC Bioinformatics Programme, whose sup-

port we are pleased to acknowledge. Our understanding of transcriptome bioinformatics has benefited from interactions with Mike Cornell, Norman Morrison and Magnus Rattray. Karen Eilbeck was supported by a BBSRC CASE award with GlaxoWellcome.

References

- Anahory, S. & Murray, D., eds (1997). *Data Warehousing in the Real World.* : Addison-Wesley.
- Attimonelli, M. *et al.* (1999). Update of the Human MitBASE database. *Nucleic Acids Research*, **27** (1), 143–146.
- Baker, P., Brass, A., Bechhofer, S., Goble, C., Paton, N., & Stevens, R. (1998). TAMBIS - Transparent Access to Multiple Biological Information Sources. In: *Proc. Int. Conf. on Intelligent Systems for Molecular Biology* pp. 25–34, AAAI Press.
- Baker, P., Goble, C., Bechhofer, S., Paton, N., Stevens, R., & Brass, A. (1999). An Ontology for Bioinformatics Applications. *Bioinformatics*, **15** (6), 510–520.
- Barrilot, E. *et al.* (1999). A proposal for a standard CORBA interface for genome maps. *Bioinformatics*, **15** (2), 157–169.
- Bartels, P., Bucher, P., & Hofmann, K. (1996). A protein linkage map of Escherichia coli bacteriophage – T7. *Nature Genetics*, **12**, 72–77.
- Baudin, A., Ozier, K., Dennoul, A., Lacroute, F., & Cullin, C. (1993). A simple and efficient method for direct gene deletion in *S. cerevisiae*. *Nucl. Acid Res*, **21**, 3329–3330.
- Booch, G., Rumbaugh, J., & Jacobson, I., eds (1999). *The Unified Modelling Language User Guide.* : Addison-Wesley.
- Brazma, A., Robinson, A., Vilo, J., Vingron, M., Hoheisel, J., Fellenberg, K., & Muilu, J. (1999). Establishing a Public Repository for DNA Array-Based Gene Expression Data. *EBI Technical Report*, . <http://www.ebi.ac.uk/microarray/>.
- Buneman, P., Davidson, S., Hart, K., Overton, C., & Wong, L. (1995). A Data Transformation System for Biological Data Sources. In: *Proc. 21st VLDB* pp. 158–169, Morgan Kaufmann.
- Cattell, R. *et al.* (1997). *The Object Database Standard: ODMG 2.0.* Morgan Kaufmann.
- Chen, I.-M. A., Kosky, A., Markowitz, V., & Szeto, E. (1997). Constructing and Maintaining Scientific Database Views in the Framework of the Object Protocol Model. In: *Proc. SSDBM*, IEEE Press.
- Chen, I.-M. A. & Markowitz, V. (1995). Modeling Scientific Experiments with an Object Data Model. In: *Proc. Data Engineering* pp. 391–400, IEEE Press.

- Delneri, D., Gardner, D., & Oliver, S. (1999). Analysis of the seven-member AAD gene set demonstrates that genetic redundancy in yeast may be more apparent than real. *Genetics*, pp. 1591–1600.
- Eilbeck, K., Brass, A., Paton, N., & Hodgman, C. (1999). INTERACT: an object oriented protein-protein interaction database. In: *Proc. Intelligent Systems in Molecular Biology (ISMB)* pp. 87–94, AAAI Press.
- El-Moghazy, A., Zhang, N., Ismail, T., Wu, J., Butt, A., Khan, S., Merlotti, C., Woodwark, K., Gardner, D., & Oliver, S. (1999). Functional analysis of six ORFs on the left arm of chromosome XII in *Saccharomyces cerevisiae* reveals two essential genes, one of which is under cell-cycle control. *Yeast*, . in press.
- Ellis, L., Speedie, S., & McLeish, R. (1998). Representing metabolic pathway information: an object-oriented approach. *Bioinformatics*, **14** (9), 803–806.
- Fromont-Racine, M., Rain, J., & Legrain, P. (1997). Toward a functional analysis of the yeast genome through exhaustive two-hybrid screens. *Nature Genetics*, **16** (3), 277–282.
- Ghosh, D. (1999). Object oriented Transcription Factors Database (ooTFD). *Nucleic Acids Research*, **27** (1), 315–317.
- Gray, P., Paton, N., Kemp, G., & Fothergill, J. (1990). An Object-Oriented Database for Protein Structure Analysis. *Protein Engineering*, **4** (3), 235–243.
- Hu, J., Mungall, C., Nicholson, D., & Archibald, A. (1998). Design and implementation of a CORBA-based genome mapping system prototype. *Bioinformatics*, **14** (2), 112–120.
- Jungfer, K. & Rodriguez-Tome, P. (1998). Mapplet: a CORBA-based genome map viewer. *Bioinformatics*, **14** (8), 734–738.
- Maltchenko, S. (1998). The Bio-Objects project. Part 1: The Object Data Model core elements. *Bioinformatics*, **14** (6), 479–485.
- Medigue, C., Rechenmann, F., Danchin, A., & Viari, A. (1999). Imagen: an integrated computer environment for sequence annotation and analysis. *Bioinformatics*, **15** (1), 2–15.
- Mewes, H., Albermann, K., Heumann, K., Liebl, S., & Pfeiffer, F. (1997). MIPS: a database for protein sequences, homology data and yeast genome information. *Nucleic Acids Research*, **25** (1), 28–30.
- Mewes, H., Hani, J., Pfeiffer, F., & Frishman, D. (1998). MIPS: a database for protein sequences and complete genomes. *Nucleic Acids Research*, **26** (1), 33–37.
- Okayama, T., Tamura, T., Gojobori, T., Tateno, Y., Ieko, K., Miyazaki, S., Fukami-Kobayashi, K., & Sugawara, H. (1998). Formal design and implementation of an improved DDBJ DNA database with a new schema and object-oriented library. *Bioinformatics*, **14** (6), 472–478.
- Oliver, S. (1996). From DNA sequence to biological function. *Nature*, **379**, 597–600.
- Planta, R. *et al.* (1999). Transcript Analysis of 250 Novel Yeast Genes from Chromosome XIV. *Yeast*, **15**, 329–350.
- Spellman, P. *et al.* (1998). Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell*, **9**, 3273–3297.
- Wach, A., Bracht, A., Pohlmann, R., & Philippsen, P. (1994). PCR synthesis of marker cassettes with long flanking homology regions for gene disruption in *S. cerevisiae*. *Yeast*, **10**, 1793–1808.
- Widom, J. (1995). Research Problems in Data Warehousing. In: *Proc. 4th Int. Conf. on Information and Knowledge Management* pp. 25–30, .