Siow, Cheuk Chuen (2011) Comparing yeast genome assemblies. MRes thesis, University of Nottingham.

# COMPARING YEAST GENOME ASSEMBLIES

CHEUK CHUEN SIOW, BSc (Hons).

Thesis submitted to the University of Nottingham
for the degree of Master of Research in Bioinformatics

DECEMBER 2011

*To pa, ma, and beloved*

# ABSTRACT

The recent transition to Next-Generation Sequencing technology has accelerated the growth of genome projects exponentially. This explosion includes a multitude of species with different strains/individuals being sequenced and made available to the scientific community. As time passes, errors in genome assemblies are also being discovered and corrected. Biologists need to update their working assembly to a newer version or to convert between different strains or species for comparisons. The LiftOver utility in the UCSC Genome Browser handles these tasks with ease. Unfortunately, the choice for yeast genome conversions is limited. Here, I extend the capabilities of LiftOver by developing applications that generate the chain files required by LiftOver in an efficient way. These files are then utilised by a website that I built to allow conversion between assemblies, strains, or species of yeast using LiftOver. Also, I used R to produce dot-matrix plots of sequence alignment for rapid comparative analysis of a new genome sequence.

One important aspect of genome biology is the characterisation of the replication start sites, called DNA replication origin. Studies with confirmed and predicted replication origin locations, specifically in budding yeast *Saccharomyces cerevisiae*, are collated in a database (OriDB). However, the structure of OriDB is complex to maintain and currently includes just a description of *S. cerevisiae* replication origins. Here, I revamp the OriDB website and database to be future-proof so that additional studies or species can be added to the database without difficulties and maintenance can be carried out with ease. The database will also include data of *Schizosaccharomyces pombe* replication origins.

# ACKNOWLEDGEMENTS

For the duration of the course since I first admitted in September 2010, most of my work requires collaboration with my supervisor, Dr. Conrad Nieduszynski, and I would like to express my gratitude to him for taking the role of mentoring me throughout the term. At times I was overwhelmed by the workload given in such a short period of time. Even so, I could not possibly achieve what I had done so far without his supervision and demand of quality work by setting high standards. This helps me in becoming a well-rounded person. His attention to detail and wise decisions are greatly appreciated.

Members of Dr. Conrad's laboratory at the time consist of three post-doctoral researchers: Dr. Amy Upton, Dr. Renata Retkute, and Dr. Michelle Hawkins, as well as two Ph.D. students: Carolin Müller and Sri Maddinapudi, for whom they gave valuable feedbacks for my work. I am utterly grateful to them for their support and insightful responses. Also, appreciation to Carolin Müller for making use of the LiftOver utility that I had developed to compare replication timing profiles between yeast genome assemblies.

Nonetheless, the leased MacBook Pro provided by the School of Biology in University of Nottingham is a wonderful platform and my productivity increased tenfold because of it. Mac OS X is ideal for the work I had done as it is Unix-based operating system and therefore compatible with Jim Kent's source from University of California, Santa Cruz as well as Local Alignment Search Tool, blastZ-like from Pennsylvania State University. We utilise these tools for our use and therefore would like to give full credits to the developers of the tools.

# Table of Contents

# List of Figures

# List of Tables

CHAPTER 1

# INTRODUCTION

## 1.1 Genome Sequencing

### 1.1.1 History

The history of genome sequencing dates back to the year 1977 when the genome of bacteriophage $\phi$X174 was sequenced by Frederic Sanger [1]. It was the very first viral genome to be completely sequenced and soon after this, Andre Goffeau established a consortium to sequence the genome of the budding yeast [2]. *Saccharomyces cerevisiae* was the first eukaryotic genome to be sequenced and published in the year 1996. Since then, extensive research on *S. cerevisiae* has been carried out including the study of DNA replication, damage and repair mechanisms [3]. Advances in yeast research have developed *S. cerevisiae* as a powerful model organism in large part due to the advantages of being the first organism with a fully sequenced genome.

All these were made possible when Frederic Sanger introduced a sequencing technique (Sanger sequencing) that used chain-termination methods [4]. Sanger sequencing was more efficient compared with other sequencing methods available during that period of time and was deemed the method of choice. However, this technique of sequencing comes with a cost; it is expensive and time-consuming to sequence a genome using lower throughput technologies such as Sanger sequencing. This led to demand for a cheaper and faster solutions that could yield higher throughput, which spurred the development

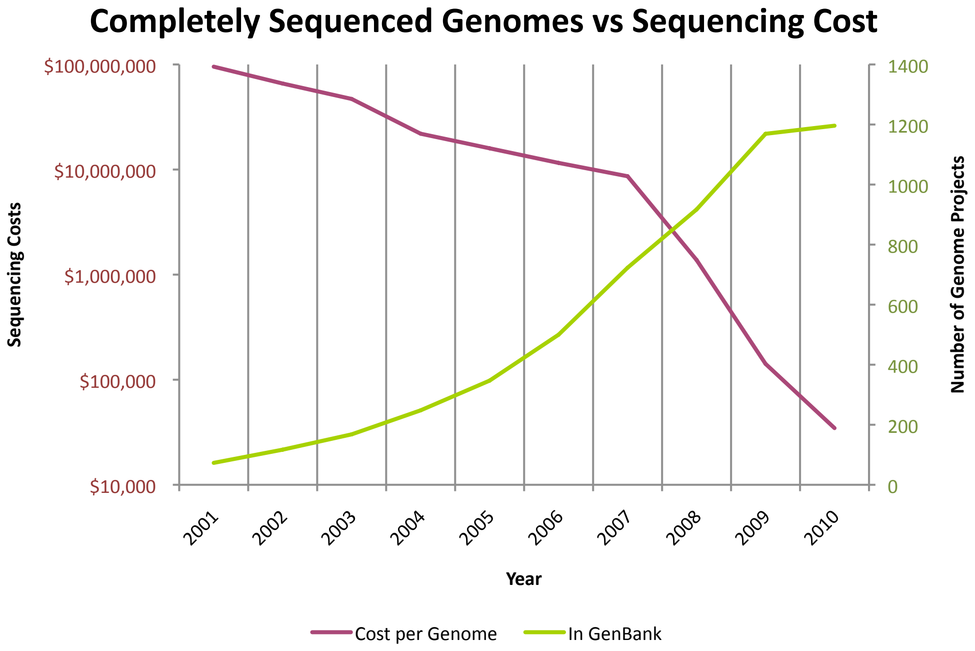


**Figure 1.1** *Graph displaying two data: cost of genome sequencing which declines over time and number of completely sequenced genomes which increases over time. It shows the inverse correlation between the two data and that more and more genome sequences are being published over the years.*

*Sources:* `http://www.genome.gov/pages/der/sequencing_cost.xls` *and* `http://www.genomesonline.org/Gold_Stats.xls`*.*

of high-throughput sequencing technologies — first launched by Lynx Therapeutics [5]. These kind of sequencing methods, also known as Next-Generation Sequencing (NGS), split sequencing processes into parts and execute them simultaneously on different processors of the machine. This enables a great quantity of data to be processed in parallel [6].

With the inception of NGS, genome sequencing is becoming more efficient and cost-effective. Figure 1.1 refers to the data collected by the National Human Genome Research Institute (NHGRI) and the Genomes OnLine Database (GOLD) projects. NHGRI reported that sequencing costs drop each year, but there was a drastic drop following 2007 when Sanger sequencing transitioned to NGS [7]. During this phase, there is a noticeable impact to the exponential increase in the number of genome projects in GenBank as reported by GOLD [8]. The explosion in complete genome projects resulted in various prokaryotic and eukaryotic organimsms being completely sequenced. This includes various yeast species and strains.

### 1.1.2 Motivation

The surge of genome sequence data in recent years has resulted in a number of challenges. The initial genome was sequenced in 1977, but GOLD had recorded over 1,000 complete genome sequences in GenBank by the year 2010. Likewise, the first complete genome of an eukaryotic organism, *S. cerevisiae*, was published in 1996. Since then, scientists have identified multiple errors in the original genome sequence which have resulted in the necessity for corrections to be made. The date when the corrections are made will be used as an identifier for the altered genome assembly once it is made public. Over time, as more errors are discovered and corrected, further genome assemblies are released.

Those assemblies are frequently used by microarray platforms and have also been made popular by their availability at the UCSC genome browser. The updates for genome assemblies are maintained on a regular basis by *Saccharomyces* Genome Database (SGD) curators [9]. Table 1.1 summarises the total number of sequence updates made for each chromosome of the *S. cerevisiae* reference genome. These numbers signify the frequent updates, since 1996 through 2011, to the genome assemblies. Two notable assemblies for *S. cerevisiae*, with significant corrections made over the past years, are frequently used as standards. These are the October 01, 2003 and June 28, 2008 assemblies.

Two issues arise for biologists working with those genome sequences: $(i)$ there are now many genome sequences available for the different strains and species of yeast, and $(ii)$ there are many genome assemblies for each strain, due to the updating and correcting of assemblies (Table 1.1). Therefore there is a demand from biologists for tools to compare different datasets between assemblies, strains, or species. For example, they want to convert their current working genome assembly to a new assembly to maintain an updated version.

| | Sequence Updates | |
|---|---|---|
| Chromosome History | Total Number | Last Update |
| I | 115 | 2011-02-03 |
| II | 192 | 2011-02-03 |
| III | 705 | 2011-02-03 |
| IV | 63 | 2011-02-03 |
| V | 20 | 2011-02-03 |
| VI | 36 | 2011-02-03 |
| VII | 125 | 2011-02-03 |
| VIII | 33 | 2011-02-03 |
| IX | 9 | 2011-02-03 |
| X | 106 | 2011-02-03 |
| XI | 79 | 2011-02-03 |
| XII | 28 | 2011-02-03 |
| XIII | 13 | 2011-02-03 |
| XIV | 43 | 2011-02-03 |
| XV | 63 | 2011-02-03 |
| XVI | 23 | 2011-02-03 |
| Mitochondrial Genome | 0 | N/A |

**Table 1.1** *Summary of chromosome update history. Each of the chromosomes have been updated frequently since 1996, with the exception of the mitochondrial genome.*

*Data extracted from* `http://www.yeastgenome.org/cgi-bin/chromosomeHistory.pl`.

### Genome Coordinate Systems

The raw data from genome sequences consist of nucleotide bases, usually stored in FASTA format. These kind of data are lengthy when dealing with long sequences. To compact data, coordinate systems are used for recording genome annotations (e.g. the location of genes). Sections of sequences can be represented in start-end numbering of genomic coordinates and thereby reduces data space. One example is the BED file shown in Table 1.2. Two coordinate systems exist in genome bioinformatics: "one-based" or "zero-based" [10]. Both conventions are widely used by major genome browsers.

| chrom | chromStart | chromEnd | name | score | strand | thickStart | thickEnd | itemRgb | blockCount | blockSizes | blockStarts |
|---|---|---|---|---|---|---|---|---|---|---|---|
| chr10 | 109957 | 111151 | YJL164C | 0 | - | 109957 | 111151 | 0 | 1 | 1194, | 0, |
| chr10 | 111659 | 113327 | YJL163C | 0 | - | 111659 | 113327 | 0 | 1 | 1668, | 0, |
| chr10 | 114174 | 115623 | YJL162C | 0 | - | 114174 | 115623 | 0 | 1 | 1449, | 0, |
| chr10 | 117238 | 117781 | YJL161W | 0 | + | 117238 | 117781 | 0 | 1 | 543, | 0, |
| chr10 | 118277 | 118820 | YJL160C | 0 | - | 118277 | 118820 | 0 | 1 | 543, | 0, |
| chr10 | 120443 | 121376 | YJL159W | 0 | + | 120443 | 121376 | 0 | 1 | 933, | 0, |
| chr10 | 121961 | 122645 | YJL158C | 0 | - | 121961 | 122645 | 0 | 1 | 684, | 0, |
| chr10 | 123532 | 126025 | YJL157C | 0 | - | 123532 | 126025 | 0 | 1 | 2493, | 0, |
| chr10 | 126586 | 128650 | YJL156C | 0 | - | 126586 | 128650 | 0 | 1 | 2064, | 0, |
| chr10 | 128982 | 130341 | YJL155C | 0 | - | 128982 | 130341 | 0 | 1 | 1359, | 0, |

**Table 1.2** *An example of BED format which stores a list of annotations in chromosome 10 with their start and end locations using coordinate system, as well as other relevant details. The inclusion of heading is optional in BED file.*

In this context, corrections in genome assemblies bring forth an issue. For example, SGD had reported two major corrections in an open reading frame (ORF) of chromosome 10 in *S. cerevisiae*. The affected feature (YJL159W in Table 1.2) went through the first major correction in February 18, 2004 where 220 bases were inserted in the position 121,258 and another major correction in October 4, 2006 where 104 bases between the positions 120,806 and 120,909 were substituted with a unique sequence of 182 bases.* That being said, whenever there is a new version of genome assembly being published, the information of the chromosomes will be stored in FASTA files which only contains data of nucleotide bases; it does not include information where changes occurred in specific locations after the corrections.

This proves to be a challenge as genomic coordinates do not automatically follow the modifications of genome assemblies. These errors on a single chromosome may accumulate and become more significant towards the end of the chromosome sequence. When biologists need to query a specific genome annotation, it may report the wrong sequence if they are working with different assemblies. To overcome the problem, it is necessary to be able to convert the genomic coordinates of one assembly to other available assemblies; correct genomic coordinates will then yield correct genome annotations. There are several ways of converting genome coordinates and these are described in the next section.

## 1.2 Approaches to Convert Genome Coordinates

### 1.2.1 Sequence Alignment

Sequence alignment tools were developed to align regions with similar nucleotide sequences. Various tools such as Basic Local Alignment Search Tool (BLAST), BLAST-Like Alignment Tool (BLAT), Sequence Search and Alignment by Hashing Algorithm (SSAHA), or MegaBLAST have their own algorithms for

*Data observed in http://www.yeastgenome.org/cgi-bin/chromosomeHistory.pl?chr=10.

searching alignments. Each of them are optimised for specific tasks based on their strengths and weaknesses. BLAT, for example, is particularly useful at aligning short-read sequences. Nevertheless, these tools can also be used to directly search for a sequence [10]. This is achieved by extracting the DNA sequence from the “old” assembly (based upon the coordinates) and using this to search (e.g. by BLAST) the new assembly. The result will include the coordinates of the sequence in the new assembly. This approach of conversion is usually not difficult, but with a low throughput because everything has to be done manually. In fact, these tools cannot cope with a single base coordinate or short sequences because such sequences might be repeated throughout the genome. This is the approach generally employed by the yeast genome database.

### 1.2.2 Rule-based Conversion

Alternatively, it is possible to write a script that includes the rules for conversion. These rules are set at specific locations where the regions after the specified locations will be manipulated according to the rules. To obtain the rules, changes need to be tracked for each update. Unfortunately, this approach is not flexible because it only allows direct conversion from one assembly to another as specified in the script. If there are numerous changes between the assemblies, it will be exhaustive to include every conversion rule as the script has to be written manually.

### 1.2.3 LiftOver

The aforementioned methods to convert genome coordinates are by themselves laborious and time-consuming. It is unwise to use those methods to convert large datasets. Realising the problem at hand, Jim Kent came up with a tool called LiftOver [11]. Basically this programme relies on a sequence alignment tool to search for similarities between two sequences. These standard alignments are linked together to form a sequence of gapless aligned blocks, known as chains [12]. Chains are then grouped with the highest-scoring non-overlapping chains on top of the hierarchy. This hierarchical collection of chains is known as nets [12]. Sets of rules can then be extracted from these nets.

As such, LiftOver is the method of choice over the previous two because of the completeness in converting genome coordinates; it is sophisticated. This LiftOver utility is embedded in the UCSC Genome Browser and is available for the public. It has a multitude of species that the users can select for conversion. Despite that, the choice for yeast genomes is restricted to *S. cerevisiae*. Only the October 2003 and June 2008 assemblies are provided where conversion is allowed just between the two, manipulating the chromosome naming during conversion between roman and arabic numerals (chr5 of October 2003 assembly ⇔ chrV of June 2008 assembly). In this regard, we aim to expand the capabilities of LiftOver further so that conversions between a variety of strains and species of yeast genome assemblies can be achieved while setting a standard in chromosome naming convention.

Originally, LiftOver makes use of two alignment tools: $(i)$ BLAT for same-species conversion where it is optimised for speed [13], and $(ii)$ BLASTZ which is more compatible for cross-species conversion [14]. However, BLASTZ has many parametric choices and the settings can be cumbersome when dealing with different sets of species. Later on, BLASTZ became obsolete when an improved version of the programme called Local Alignment Search Tool, blastZ-like (LASTZ) replaces it. LASTZ has two clear advantages over BLASTZ: $(i)$ it infers appropriate scoring parameters automatically, and $(ii)$ it requires much less memory to process larger sequences [15]. Because of the speed it can achieve during the aligning process and with easier implementation for cross-species conversion, LASTZ will be used here to substitute both BLASTZ and BLAT.

## 1.3 Origin of Replication

### 1.3.1 Background Study

DNA undergoes replication during cell division. The entire double-stranded DNA is duplicated in all living organisms and undergoes mitosis to form two diploid cells or meiosis to form four haploid cells. Replication initiates at specific sites in the DNA, known as replication origins. When initiated, certain proteins target these replication origins to separate the two strands and perform DNA synthesis [16].

Figure 1.2 shows that the DNA replication process starts at origin sites in a given example of circular viral chromosome. Viral chromosomes have a single replication origin, but eukaryotes with large chromosomes are replicated

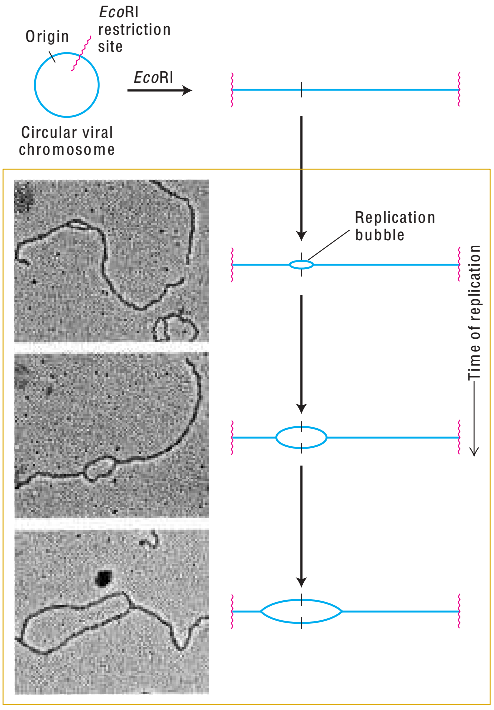


**Figure 1.2** *Illustration of DNA replication process starts at origin sites in a circular viral chromosome, forming replication bubble which gets larger in size for the duration of replication. The enzyme EcoRI linearises the circular chromosome. Image courtesy of Lodish et al., [16].*

from multiple origins of replication on each chromosome to help replicate the whole genome rapidly [16, 17].

The importance of DNA replication requires mechanisms of tight regulation to avoid genomic instability. Errors in DNA replication may occur nonetheless, and these can lead to diseases. Hence understanding where replication origins are located is key to truly understanding genome integrity [18].

### 1.3.2 Database

*S. cerevisiae* was first sequenced in 1996 and since then, it is well studied and has become a powerful model organism. *S. cerevisiae* is highly suited for the study of replication origins because of its characteristics. A number of studies have mapped the locations of replication origins and these different studies provide complementary information. These datasets were collated to produce a single list of replication origin locations. To allow viewing of these collated datasets, OriDB was built and made publicly available in 2006.

OriDB acts as a repository that stores confirmed and predicted *S. cerevisiae* DNA replication origin locations [19]. Presented information about replication origins include genomic location and chromosome state of origin sites, origin replication time, DNA sequence of origin elements, free energy required for stress-induced DNA duplex destabilisation (SIDD), and phylogenetic conservation of sequence elements [19]. All these can be viewed through the OriDB website (`http://www.oridb.org/`) in text or graphical formats where relevant.

With the recent explosion in genome projects due to NGS, many different yeast species are being sequenced and extensively studied to gain valuable insights into particular biological phenomena. Another well studied example is the fission yeast *Schizosaccharomyces pombe* which is also widely used as a model organism. Several studies have mapped the locations of replication origins for *S. pombe* [20]. At present OriDB only provides origin data for *S. cerevisiae*. There is a need for OriDB to include origin data for *S. pombe* and perhaps some other species in the future. As the number of studies increases, OriDB will also need an efficient way to incorporate new studies so that new information can be displayed in the OriDB website.

## 1.4 Aims and Objectives

Yeast is an important organism for in-depth study in the field of molecular biology. Biologists will need sophisticated techniques to allow conversion of genome coordinates between different assemblies for update purposes, or between different strains or species for comparison purposes. LiftOver from UCSC browser does this job, but without much support for yeast genomes. That gives us the opportunity to expand LiftOver's capability further with modest modification specifically for yeast genomes. The objective is to build a website which brings the customised LiftOver utility online.

Secondly, we intend to redesign the OriDB website so that more species can be implemented in the future. The design has to be as generalised as possible because this allows new studies or species to be added easily without jeopardising the database. Besides aiming for future-proof, redesign of the web interface to improve usability is also beneficial to cater a wider user base. Ultimately, the codes are rewritten from scratch instead of reusing the previous codes to impose a clean coding structure. A better structure means that maintenance or future enhancements will be easier to perform.

The next chapter describes the necessary steps needed to execute the two objectives as mentioned. The resulting LiftOver implementation is demonstrated in Chapter 3 whereas Chapter 4 describes the outcome of the revamped OriDB.

CHAPTER 2

# MATERIALS AND METHODS

## 2.1 Tools for Genome Coordinates Conversion

### 2.1.1 Platform

The platform used is a local Mac OS X Server. Only those who are in the university network have the privilege to use the LiftOver tool implemented on the server. With Macintosh being a Unix-based operating system, Unix shell commands are being used extensively in our programme. Common Gateway Interface (CGI) is used for input and output of data to a web browser, in which the web server software delegates the generation of web pages for data output to CGI scripts.

### 2.1.2 Dataset

FASTA files, which contain sequence as nucleotide bases, are used as data input for the creation of chain files. Chain format describes a pairwise alignment that permit gaps in both sequences concurrently. The purpose of a chain file is to map from one assembly to another assembly with the input files of one of the given format: BED, GFF/GTF, GenePred, or Genomic Coordinate Position.*

*Details for each of the file formats are described in http://genome.ucsc.edu/FAQ/FAQformat.html.

### 2.1.3 External Tools

I utilised Jim Kent's source from UCSC Genome Browser which contains a whole set of biological analysis and web display programmes [11]. Only a minor fraction of the entire source tree was required. The following subsection describe the installation stage and procedures to operate the necessary tools.

#### Installation

Several tools are required to generate LiftOver chain files and each of them is a separate application which needs to be installed properly. The following steps serve as a guideline for the installation process which adhere to the comprehensive manual available in the source itself.[†]

i. The source is downloaded from the UCSC Genome Browser available at http://hgdownload.cse.ucsc.edu/admin/jksrc.zip and it requires a GNU gcc compiler to compile C codes.

ii. The environment variable MACHTYPE should exists on Unix systems. To obtain the MACHTYPE value, type the following command in the terminal:

```
uname -p
```

If needed, assign the value to this variable in the shell environment:

```
MACHTYPE=i386
```

The value should be a short non-hyphenated name of the machine type (e.g. i386, i686, x86_64, alpha, or sparc).[‡]

iii. A subdirectory is created in bin of the home directory with the name depending on the MACHTYPE value:

[†] Also available at http://hgwdev.cse.ucsc.edu/~kent/src/unzipped/product/README.building.source.

[‡] Note that the current system gives the uname value of i386, this will be used throughout.

```
mkdir -p ∼/bin/i386
```

Throughout the build, binaries will be moved to this subdirectory.

iv. `MYSQLINC` and `MYSQLLIBS` environment variables are assigned depending where MySQL is installed in the system:

```
MYSQLINC=/usr/local/mysql/include
```

which directs to the MySQL include files, and

```
MYSQLLIBS='/usr/local/mysql/lib/libmysqlclient.a -lz'
```

which directs to the `libmysqlclient.a` library and any other libraries needed to connect network applications.[§]

v. A directory named `jksrc` is created and the source file is uncompressed in this directory. This creates the source hierarchy `./kent` and the `src` directory lies within. The following command is entered in the `src` directory which builds the libraries:

```
make libs
```

This results in the libraries being built from the source directories:

- `jkweb.a` compiled from `kent/src/lib`
- `jkOwnLib.a` compiled from `kent/src/jkOwnLib`
- `jkhgap.a` compiled from `kent/src/hg/lib`
- `jkhpap.a` compiled from `kent/src/hg/protein/lib`

[§] Additional options required for `libmysqlclient.a` in different systems: '`-lxnet`' for Alpha, '`-lsocket -lnsl`' for Solaris, and '`-lsocket -lnsl -lresolv`' for SunOS.

Those `lib.a` files are moved to `kent/src/lib/i386`.

vi. For any particular tool that needs to be installed, go to the directory of the required tool and run:

```
make
```

The resulting executables will be placed into ∼`/bin/i386`. Associated directories for the required tools can be found in Appendix A.1.

### 2.1.4 Development

Most of the programmes are written in Practical Extraction and Report Language (Perl 5) due to its powerful text processing facilities and strong integration with the BioPerl modules. Furthermore, Perl can make use of Unix utilities allowing the usage of UCSC tools. The programme written specifically for `liftOver` function was placed in `/Library/WebServer/CGI-Executables` of the server which takes up the role of a stand-alone CGI script.

A website was built to bring the custom liftOver facility online. For web development, scripting languages are incorporated in HyperText Markup Language (HTML5) web pages. PHP: Hypertext Preprocessor (PHP 5.3) was used for server-side scripting to produce dynamic web pages, whereas JavaScript was used for client-side scripting to add interactivity in HTML pages. jQuery, a fast and concise JavaScript Library, handles and simplifies event handling as well as Ajax interactions.

### 2.1.5 Analysis

To perform comparisons between genome sequences, graphical techniques can be used to visualise quantitative data. R64 version 2.13.0 is used to plot graphs of the sequence comparisons by aligning two genome assemblies. Perl programmes are written to generate LASTZ output with the `rdotplot` setting. This setting outputs files compatible with R and this allows the assignment of data generated by LASTZ into variables in R. The graphs plotted by R can be in Portable Document Format (PDF), Portable Network Graphics (PNG) or Scalable Vector Graphics (SVG) format. The plot diagrams produced are shown in Section 3.7.

## 2.2 Database for DNA Replication Origin

### 2.2.1 Platform

A prototype of the OriDB web interface is being developed through the University of Nottingham's Granby server with SunOS 5.9 operating system. We were allocated a workspace on the server to test the prototype websites for *S. cerevisiae* and *S. pombe*.[¶] After completion, the website was then ported and deployed externally to the actual server (`http://www.oridb.org/`) to bring it online for public access.

### 2.2.2 Dataset

The materials involved consist of FASTA files of the associated genomes: *S. cerevisiae* and *S. pombe* plus lists of origin locations from various studies. These files contain data of nucleotide bases which allow the calculation of whole genome length, length for certain sequences, and to extract sections of nucleotides to view.

### 2.2.3 Development

Although there is an existing OriDB site established five years ago which is still functioning to date, unfortunately the current website is perplexed in code structure and difficult to maintain. The codes do not comply with the programming standards and therefore add complications when new functions or

[¶] Workspace allocated is available at `http://www.nottingham.ac.uk/plzcnlab/`.

genomes are to be incorporated.

As such, the current website is to be revamped with good coding practice in mind such that the codes adhere to professional programming standards. PHP is used for server-side scripting and jQuery is used for client-side scripting. In addition, HighCharts JS is utilised to generate graphical view of chromosome data, replacing the old viewer which uses Adobe Flash.

### 2.2.4 Database

Many of the interactions come from data retrieval of the database. The existing database consists only a single table with all the data in it. In consequence, it is extremely hard to maintain when new data is to be added, given that the single table contains numerous fields. An overly complex data table is prone to errors when all the information is cluttered in one table.

To improve the database, normalisation is performed to partition the single table into several tables, hence minimising redundancy. Structured Query Language (MySQL) is used as a relational database management system. Database design is done with MySQL Workbench and the entity-relationship diagrams (ERDs) are illustrated in Section 4.3.1.

Chapter 3

# Conversion and Collation Between Genome Assemblies

## 3.1 Brief Description

Frequently, biologists need to map from one assembly to another and they require a way to do this task. Three approaches are known to convert genome coordinates: by using sequence alignment tools, to write Perl scripts for rule-based conversions, and by using the LiftOver utility provided by the UCSC Genome Browser (see Section 1.2). LiftOver turns out to be superior over the other two methods, thus it became the norm in terms of assembly-assembly mapping. In some cases, the scientific community might notice some limitations in the LiftOver online utility in that it lacks the required assembly to map for a particular genome, or some organisations just want to incorporate the LiftOver utility to their projects or browsers but do not need the complete package. As a result, tools similar to LiftOver have been spawned, specialising in certain tasks.

## 3.2 Related Work

The UCSC Genome Browser LiftOver utility pioneered the translation tool for genomic coordinates conversion between assemblies [11]. They provide online service as well as downloadable LiftOver executable and Perl scripts that automate the generation of chain files using the chains and nets algorithm. These chain files act as a dictionary that contains all the necessary rules to map between assemblies. Other existing tools that also provide similar services are described in Table 3.1.

For instance, Galaxy is one of the website that fully incorporates UCSC's LiftOver utility. It acts as a platform that merges existing genome annotations databases into a single portal [21]; one of them is the UCSC Genome Browser. Likewise, Galaxy's Lift-Over utility allows the conversion of many genomes but with the exception that it requires users to import UCSC genome data manually from the portal. On the other hand, NCBI Coordinate Remapping Service resembles UCSC's LiftOver whereas Ensembl's Assembly Converter and FlyBase Sequence Coordinates Converter provide a minimalistic web interface for simple usage. These three deal with specific genomes only, unlike the long list of available genomes provided in UCSC Genome Browser.

As an alternative, Perl scripts that allow conversion are available via the internet. Ensembl's `AssemblyMapper.pl` is similar to its online counterpart and suited for larger datasets, but still limited to human and mouse genome assemblies. `convert_yeast_genome_version.pl` by biotoolbox is the only utility available besides LiftOver that supports conversion between yeast genome assemblies. However, it implements the rule-based conversion approach that renders the utility inflexible; every time an additional assembly is required for conversion, the developer has to know how to map from one assembly to that particular assembly precisely and enforce the rules manually in the script. This approach is time-consuming while adding complexities to the code.

UCSC's LiftOver remains the ultimate solution due to its power, speed, and flexibility. For the yeast genome however, LiftOver is only able to convert from October 2003 to June 2008 assembly of *S. cerevisiae* or vice versa. This is because the yeast genome is not part of the focus of the UCSC Genome Browser and therefore there is a lack of datasets. To expand the number of assemblies available for yeast genome assemblies, it is better to have an in-house LiftOver

| Site | Genomes | Online Service | URL | Citation |
|---|---|---|---|---|
| UCSC | 53 available genomes | LiftOver | http://genome.ucsc.edu/cgi-bin/hgLiftOver | [11] |
| Galaxy | Follows that of UCSC. | Lift-Over | http://main.g2.bx.psu.edu/ | [21] |
| NCBI | *Mus musculus, Bos taurus, Homo sapiens* | NCBI Genome Remapping Service | http://www.ncbi.nlm.nih.gov/genome/tools/remap | [22] |
| Ensembl | *Homo sapiens, Mus musculus* | Assembly Converter | http://www.ensembl.org/Homo_sapiens/UserData/SelectFeatures | [23] |
| FlyBase | *Drosophila melanogaster* | FlyBase Sequence Coordinates Converter | http://flybase.org/static_pages/downloads/COORD.html | [24] |

| Source | Perl Script | Description |
|---|---|---|
| UCSC | http://hgwdev.cse.ucsc.edu/~kent/src/unzipped/hg/utils/automation/doSameSpeciesLiftOver.pl | Automates same-species chain file creation using BLAT. Optimised for speed but incompatible with cross-species conversion. |
| UCSC | http://hgwdev.cse.ucsc.edu/~kent/src/unzipped/hg/utils/automation/doBlastzChainNet.pl | Automates cross-species chain file creation using BLASTZ. Requires appropriate settings depending on the distance between species. |
| Ensembl | ftp://ftp.ensembl.org/pub/misc-scripts/Assembly_mapper_1.0/AssemblyMapper.pl | Map slices from old assemblies to the latest assembly. Deals with larger datasets as compared with its web service (Assembly Converter). |
| biotoolbox | http://code.google.com/p/biotoolbox/source/browse/trunk/scripts/convert_yeast_genome_version.pl | Rule-based conversion with the conversion rules being hard-coded in the script with limited set of assemblies. |

**Table 3.1** *An overview of LiftOver and its related tools. The upper portion of the table lists the available online services for conversion between assemblies of given genomes, whereas the lower portion of the table describes the available Perl scripts for ready-made in-house conversion.*

utility; the LiftOver executable can be downloaded or compiled manually, and necessary chain files can be generated using a series of tools from Jim Kent's source.

`doSameSpeciesLiftOver.pl` and `doBlastzChainNet.pl` automate chain file creation and are downloadable from UCSC Genome Browser. In spite of that, these scripts come with a multitude of module dependencies with the usage of the legacy BLASTZ or the less flexible BLAT. Due to the complexity of the codes, it is unwise to customise those scripts to our requirements. As such, we attempt to build our personalised LiftOver online utility that deals primarily with yeast genome, encompassing those further away from the phylogenetic tree. The following sections show the stages involved to implement the generation of LiftOver chain files, the web interface for the utility, and additionally the analysis for genome comparisons.

## 3.3 Procedure

A series of operations has to be conducted in order to generate a single chain file.[*] The actions listed below are fundamentally Unix shell commands. In essence, target denotes the original genome build, whereas query denotes the new genome build.[†] Refer to http://genome.ucsc.edu/FAQ/FAQformat.html for further descriptions of the data file formats involved.

i. FASTA files from the query genome build are partitioned into several chunks, each with 3000 bases which will be stored in one file. Output is specified in .lft file format which stores information on how to reconstruct the genome sequence from these fragments.

[*] Adapted from: http://genomewiki.ucsc.edu/index.php/Minimal_Steps_For_LiftOver.

[†] T and Q after input or output indicate that target or query side is to be placed in particular. @ before input or output symbolises an array.

```
faSplit -lift=outputQ.lft size inputQ.fa -oneFile 3000 outputQ
```

ii. A list of sequences from FASTA files is gathered and all the information of these sequences are stored in .2bit file. This single file acts as a database for multiple DNA sequences.

```
faToTwoBit @input.fa output.2bit
```

iii. Information about sequence lengths in a .2bit file is retrieved and stored in chrom.sizes file which is required in step viii.

```
twoBitInfo input.2bit output.chrom.sizes
```

iv. Query sequence is aligned with the whole target sequences using LASTZ. `-notransition` and `-step=20` lower the sensitivity level, hence reducing runtime and memory consumption. `-nogapped` removes the computation of gapped alignments. These settings speed up the alignment process. Output is the pairwise alignment format (.axt).

```
lastz inputT.2bit[multiple] inputQ.fa -notransition -step=20
-nogapped -format=axt > output.axt
```

v. Coordinates of an axt alignment file are converted to the parent coordinate system where the query side is lifted rather than the target side. The lift specification for query build is used to generate merged and lifted .axt files.

```
liftUp -axtQ output.axt inputQ.lft warn inputT.axt
```

vi. Alignments are chained together at this stage. In the chaining process, two neighbouring alignments which correspond are merged into a single fragment if they are evenly matched [12]. The minimum score for chain is set to

1000 by default where the higher-scoring chain tends to be syntenic. As for the linear gap cost, 'medium' is used in the case of same species conversion, while 'loose' is intended for cross-species conversion.[‡]

```
axtChain -minScore=1000 -linearGap=<medium|loose> input.axt
inputT.2bit inputQ.2bit output.chain
```

vii. Sorted chain files are combined into a larger sorted file, piped into another function which takes as input the larger sorted file, and then chains are partitioned by target or query sequences to the output folder.

```
chainMergeSort *.chain | chainSplit outputFolder stdin
```

viii. In the netting process, alignment nets are formed by grouping blocks of chained alignments into a hierarchy, sorted to begin with the highest-scoring non-overlapping chains [12]. The output for a target-centric net in query coordinates is ignored since the reciprocal nets are symmetrical.

```
chainNet input.chain inputT.chrom.sizes inputQ.chrom.sizes
outputT.net /dev/null
```

ix. Ultimately, a chain file is created which derived from subset of chains in net.

```
netChainSubset inputT.net input.chain output.chain
```

x. The LiftOver chain file will be usable with a BED file as the default format.

[‡] `<medium|loose>` denotes an option to select between `medium` or `loose`.

```
    liftOver input.bed input.chain output.bed unmapped
```

## 3.4 Chain File Creation

To automate this overly complex process, I wrote a Perl script to manage the procedure. Subroutines are defined for each operation to allow sets of instructions to be performed in order.[§] Figure 3.1 illustrates the inputs required for the Perl script `liftOverChainCreation.pl`, intermediate processes involved during runtime, and the resulting output which is the chain file. Those processes are performed sequentially with a certain loop count depending on the number of FASTA files, where each FASTA file represents each individual chromosome sequence of a genome.

Likewise, `liftOverChainCreation.pl` automates the working units precisely and therefore encapsulates the intermediate processes from the user. It directly generates the chain output for the user, requiring merely two genome builds of FASTA format for input and the necessary tools to perform the processes; no extraneous dependencies are involved. For convenience, the script can be used as follows:

```
      liftOverChainCreation.pl
```

or

```
      liftOverChainCreation.pl -i inputT outputQ
```

where the first method allows users to input the required information when prompted. The `-i` flag in second method opens up an option to directly input

[§] Instructions are based on http://hgwdev.cse.ucsc.edu/~kent/src/unzipped/hg/doc/liftOver.txt, but with modest modification.

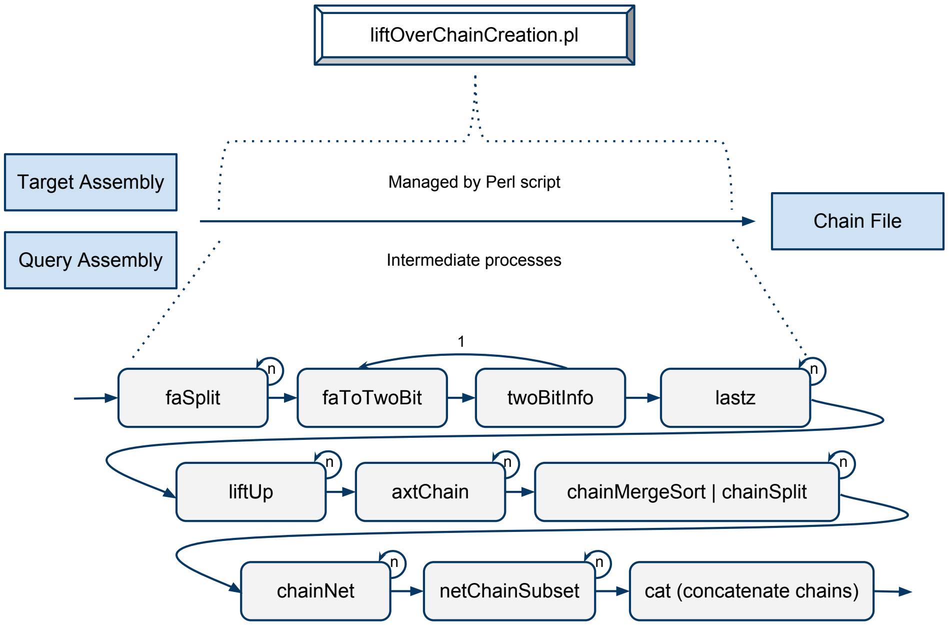


**Figure 3.1** *Subroutines for chain file creation. n represents the execution count for each subroutine depending on the number of FASTA files of a genome build (i.e. the chromosome number).* `faToTwobit` *and* `twoBitInfo` *are performed twice in total, once for each genome.*

the information for both target and query assemblies into the arguments, allowing the script to be embedded in another script. These assemblies should be stored in the directory ~/`bin/fastaFiles` because the script is programmed to search in that specific location.[¶]

An assembly consists of a directory containing the associated FASTA files of a genome, with the name of the directory corresponding to the name of the assembly. In this case, we established a nomenclature in which the name is divided into three sections; first is the species, followed by strain, and ending with the assembly date. For example:

[¶] The script sets the directory ~/`bin/fastaFiles` to locate genome assemblies with FASTA format as well as ~/`bin/chainFiles` to store chain files for convenience. Users have to modify the codes if they want those files to be located elsewhere.

```
sacCer_REF_20080628
```

refers to *S. cerevisiae* of the reference genome (strain S288c) with 28 June 2008 assembly build. The name for the chain files depend on the names of the assemblies:

```
sacCer_REF_20031001_-_sacCer_REF_20080628.over.chain
```

refers to the chain file that maps *S. cerevisiae* of the reference genome with 01 October 2003 assembly build to *S. cerevisiae* of the reference genome with 28 June 2008 assembly build. The generated chain files will be stored in the directory ~/bin/chainFiles.

## 3.5 Batch Chain File Creation

Despite liftOverChainCreation.pl being able to create chain files, it produces only a single chain file at a time. It will be time-consuming to use this script to generate large number of chain files. For $n$ genome assemblies, it must be run $n \times (n - 1)$ times to cover each and every available genome assemblies along with reverse conversions. Therefore the number of runs needed will grow exponentially as the number of genome assemblies increases linearly. As shown in the left of Figure 3.2, liftOverChainCreation.pl alone cannot cope with the vast number of genome assemblies.

To harness the capability of liftOverChainCreation.pl, another Perl script was written to utilise liftOverChainCreation.pl and augment the ability to generate chain files in batch. liftOverMultiChain.pl automates the process and does the job thoroughly; it searches all the genome assemblies in the directory ~/bin/fastaFiles and checks whether the chain files of any two genome assemblies for forward and reverse conversions exist in the directory ~/bin/chainFiles. If a chain file exists for two given genome assemblies in ~/bin/fastaFiles, it skips the process and checks the subsequent assemblies, otherwise liftOverMultiChain.pl will call liftOverChainCreation.pl which

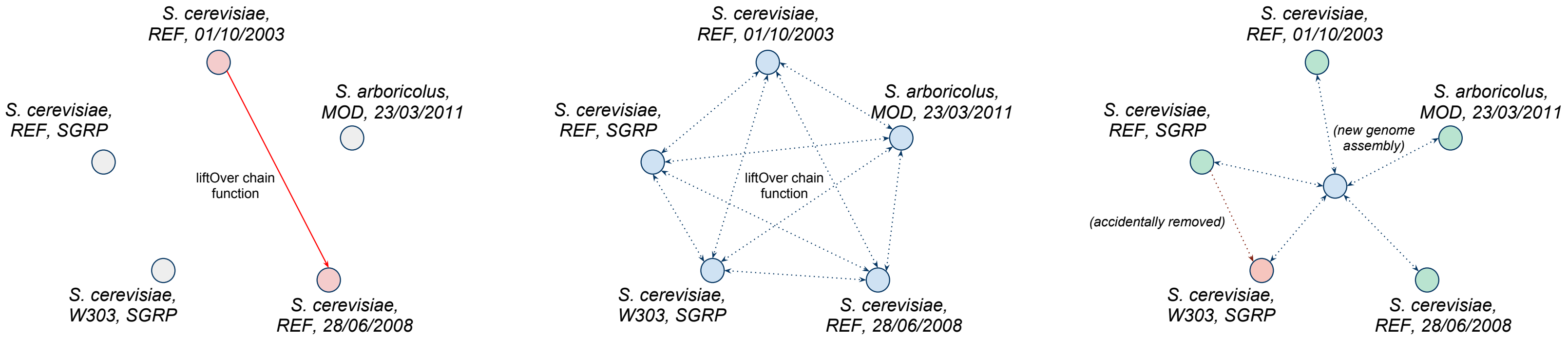

S. cerevisiae, REF, 01/10/2003
S. arboricolus, MOD, 23/03/2011
S. cerevisiae, REF, SGRP
liftOver chain function
S. cerevisiae, W303, SGRP
S. cerevisiae, REF, 28/06/2008
S. cerevisiae, REF, 01/10/2003
S. arboricolus, MOD, 23/03/2011
S. cerevisiae, REF, SGRP
liftOver chain function
S. cerevisiae, W303, SGRP
S. cerevisiae, REF, 28/06/2008
S. cerevisiae, REF, 01/10/2003
S. arboricolus, MOD, 23/03/2011
S. cerevisiae, REF, SGRP
(new genome assembly)
(accidentally removed)
S. cerevisiae, W303, SGRP
S. cerevisiae, REF, 28/06/2008

**Figure 3.2** *Schematic representation of applying LiftOver to genome assemblies.*

*Left:* `liftOverChainCreation.pl` *is capable of generating a chain file for any given assembly comparison. A set of genomes with different assemblies, strains, and species exist in the directory.*

*Middle: Batch output of chain files with the help of* `liftOverMultiChain.pl` *utilising* `liftOverChainCreation.pl`. *Every genome assembly are linked to one another in both directions after running the programme.*

*Right: Ability of* `liftOverMultiChain.pl` *to process recently added genome assemblies or to recreate chain files that have been removed, whilst skipping the process when a particular chain file exists. This potentially reduces the computational time.*

in turn performs all the subroutines to generate that chain file.

The automation of liftOverMultiChain.pl is demonstrated by the middle and right of Figure 3.2 where every genome assemblies will be accounted for in the generation of chain files. Since existing chain files do not need to be created again, this greatly reduces computational time. Whenever new genome assemblies are added to the folder or if one of the chain files is removed, running the programme once will automatically produce those chain files. On top of that, it is fairly simple to run the programme, require only the script in the working directory and the user to enter the following to the command prompt:

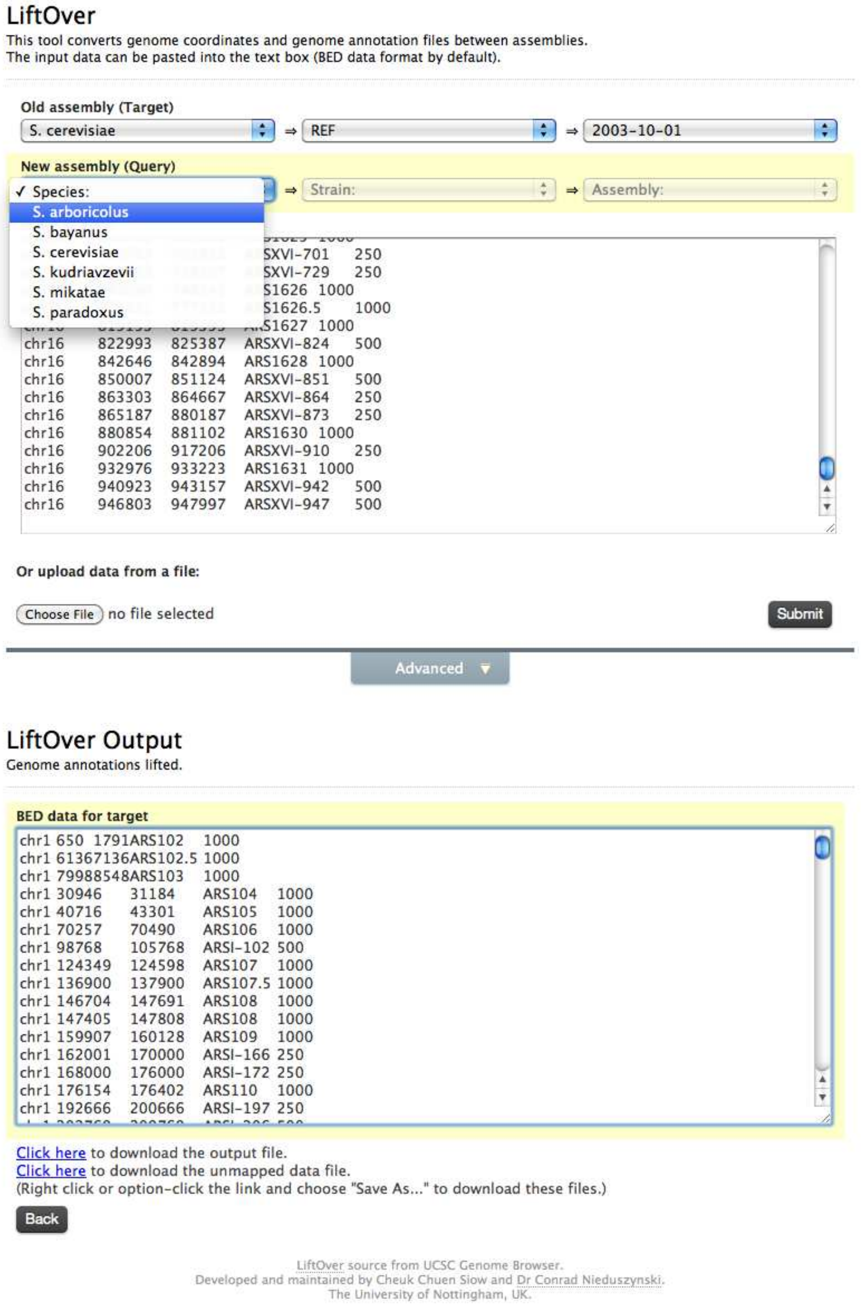
```
liftOverMultiChain.pl
```

## 3.6 Web Interface

To provide the LiftOver utility where users can access the service easily, a user-friendly website was developed (University of Nottingham access only: http://128.243.182.148/~siow/liftover/). The website proves to be simple yet effective in delivering the service. Figure 3.3 demonstrates two web pages for the site; the top illustrates the page for input which redirects to the page for output represented at the bottom.

For the input page, users are given two rows of options; target and query assemblies, each with three fields to choose from. The first field denotes the available species to choose from, followed by the available strains for the selected species, and finally the assemblies for that particular strain. The options are automatically generated based upon the chain files available. Users paste the data for target assemblies into the text box provided or upload the file if they so choose. The advanced tab permits the users to further configure how the LiftOver works to suit their preference. The configurations imitate that which UCSC Genome Browser has to offer [11].

Instead of giving a whole list of genome assemblies in one field, three



LiftOver

This tool converts genome coordinates and genome annotation files between assemblies.
The input data can be pasted into the text box (BED data format by default).

Old assembly (Target)
S. cerevisiae ⇒ REF ⇒ 2003-10-01

New assembly (Query)
✓ Species: ⇒ Strain: ⇒ Assembly:

```
                          SXVI-701    250
                          SXVI-729    250
                          S1626  1000
                          S1626.5     1000
                          S1627  1000
chr16    822993   825387  ARSXVI-824  500
chr16    842646   842894  ARS1628  1000
chr16    850007   851124  ARSXVI-851  500
chr16    863303   864667  ARSXVI-864  250
chr16    865187   880187  ARSXVI-873  250
chr16    880854   881102  ARS1630  1000
chr16    902206   917206  ARSXVI-910  250
chr16    932976   933223  ARS1631  1000
chr16    940923   943157  ARSXVI-942  500
chr16    946803   947997  ARSXVI-947  500
```

Or upload data from a file:

LiftOver Output

Genome annotations lifted.

BED data for target

```
chr1 650  1791ARS102   1000
chr1 61367136ARS102.5 1000
chr1 79988548ARS103   1000
chr1 30946    31184    ARS104   1000
chr1 40716    43301    ARS105   1000
chr1 70257    70490    ARS106   1000
chr1 98768    105768   ARSI-102 500
chr1 124349   124598   ARS107   1000
chr1 136900   137900   ARS107.5 1000
chr1 146704   147691   ARS108   1000
chr1 147405   147808   ARS108   1000
chr1 159907   160128   ARS109   1000
chr1 162001   170000   ARSI-166 250
chr1 168000   176000   ARSI-172 250
chr1 176154   176402   ARS110   1000
chr1 192666   200666   ARSI-197 250
```

Click here to download the output file.
Click here to download the unmapped data file.
(Right click or option-click the link and choose "Save As..." to download these files.)

LiftOver source from UCSC Genome Browser.
Developed and maintained by Cheuk Chuen Siow and Dr Conrad Nieduszynski.
The University of Nottingham, UK.

**Figure 3.3** *Screenshots of liftOver website.*

*Top: Home page where the user selects the target and query genome assemblies as well as to paste or upload data input for the assembly to be converted. The advanced tab allows the user to configure the conversion settings.*

*Bottom: Results page displaying the converted coordinates of a given genome assembly. User is able to download the output as .txt file or to view the unmapped regions for the conversion.*

fields of categories are designed to minimise the selection from each of the lists. The following fields are disabled unless the option for the current field is selected and the choice for each subsequent field depends on the options chosen formerly. These greatly enhance the usability of the selection whereby each category is easily distinguished from one another and the option lists remain concise depending on the chain files available.

Once inputs are set and submitted, the web server redirects the data to a specified CGI script for processing. The script then processes the incoming data and perform LiftOver. The whole operation is hidden from the users and they will be redirected to the results page. On the results page, the converted genome coordinates are displayed in the uneditable text box for viewing. Users are able to download a coordinates file with the format being decided during input. Furthermore, the unmapped regions which the LiftOver might produce can also be downloaded as per users choice.

## 3.7 Visualising Comparisons

As a sequence alignment tool, LASTZ has the potential to produce data for analytical purposes besides utilising LASTZ's output to generate chain files. Two genome sequences were aligned using LASTZ and the alignment data was passed to R for processing. R is capable of plotting graphical displays of data. For that reason, I used R to generate dot-matrix plots of the alignments for further sequence analysis.[‖]

Dot-matrix plots are useful for determining shared regions between two large sequences to identify certain features. To construct a dot-matrix plot, the *S. cerevisiae* and *S. arboricolus* genome sequences are placed in the $x$ and $y$ axis of a two-dimensional matrix, respectively. A dot is plotted for any region with two identical nucleotides. LASTZ's output from aligning *S. cerevisiae* and *S.*

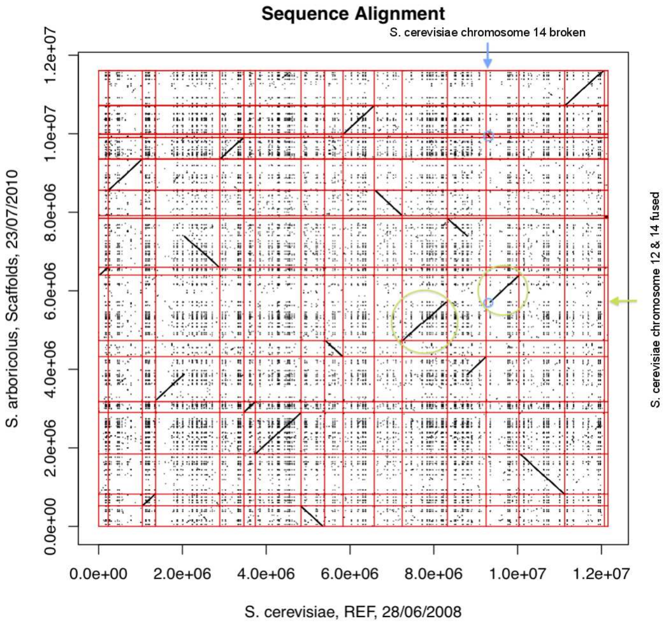
[‖] Another existing programme (`Dotlet`) also draws dot plots using different matrices [25]. `Dotlet` is available at `http://myhits.isb-sib.ch/cgi-bin/dotlet` and requires Java plug-in to run, but I wrote our own programme instead to produce graphs that suit our requirement.

*arboricolus* genome sequences will have this data for plotting. The dots plotted as a single diagonal line along a region shows that two sequences are closely related and a backward slash means that the sequence from $y$ axis is on the reverse strand relative to the sequence on the $x$ axis. In addition, the red grid lines in the plot represent the chromosomes in ascending order.

*S. arboricolus* is a close relative of *S. cerevisiae* and belongs to the *Saccharomyces sensu stricto* group of yeasts [26]. The genome of *S. arboricolus* was sequenced at the Next-Generation Sequencing Facility (DeepSeq) at the University of Nottingham using Roche 454 pyrosequencing. To assess the quality of the various assemblies of the sequenced *S. arboricolus* genome, we compared each assembly to the *S. cerevisiae* genome sequence by analysing the dot-matrix plots produced by R.

Figure 3.4 illustrates the graphs plotted for genome sequences of *S. cerevisiae* aligned to *S. arboricolus*. The reference genome of *S. cerevisiae* (the assembly from 28 June 2008) was used as a base to compare with *S. arboricolus* genome. For the top of the figure, the *S. arboricolus* genome with 23 July 2010 assembly was used for comparison as a control. By analysing the dot-matrix plot, anomalies were detected in that assembly; chromosome 14 of *S. cerevisiae* is broken into two scaffolds in *S. arboricolus*. Moreover, chromosomes 12 and 14 of *S. cerevisiae* are fused into a single large scaffold in *S. arboricolus*. These anomalies mean that errors were discovered in that genome assembly.

The DeepSeq facility has generated ten different *S. arboricolus* genome assemblies by combining different sequencing runs and setting different parameters for the assembly software (using Newbler 2.3). After comparing all of them to *S. cerevisiae* reference genome, one of the ten genome assemblies was found to exhibit no anomalies. This particular *S. arboricolus* genome assembly “combined_R_2009_10_06_R_2010_03_ 29_R_2010_05_28.454Scaffolds” was set as a new assembly (23 March 2011). The bottom of Figure 3.4 represents the dot-matrix plot for this build. Refer to Appendix A.2 for a complete set of the ten comparisons.



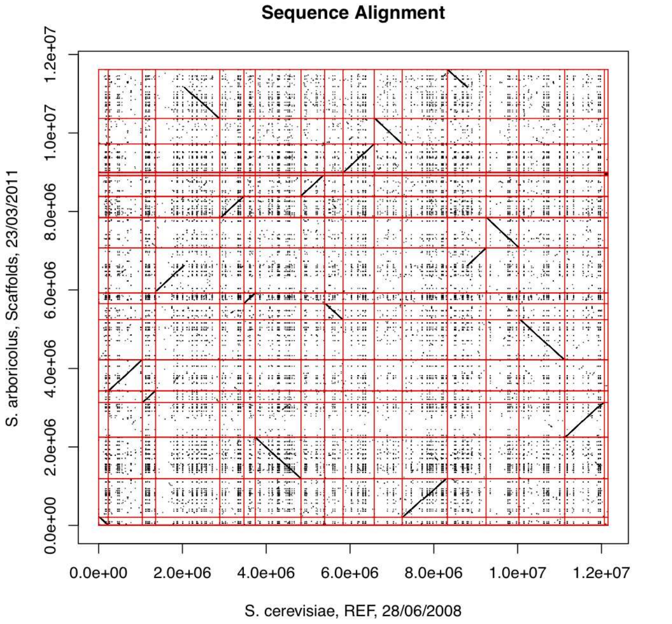


**Figure 3.4** *Dot-matrix plots for sequence alignment between S. cerevisiae and S. arboricolus. The dots other than the aligned sequences are the recurring noises and can be ignored.*

*Top: The older assembly of S. arboricolus shows the fusion of chromosomes 12 and 14 of S. cerevisiae into one large chromosome in S. arboricolus. Also, chromosome 14 in S. cerevisiae is broken into two in S. arboricolus.*

*Bottom: The new assembly of S. arboricolus shows no signs of fusions or broken chromosomes.*

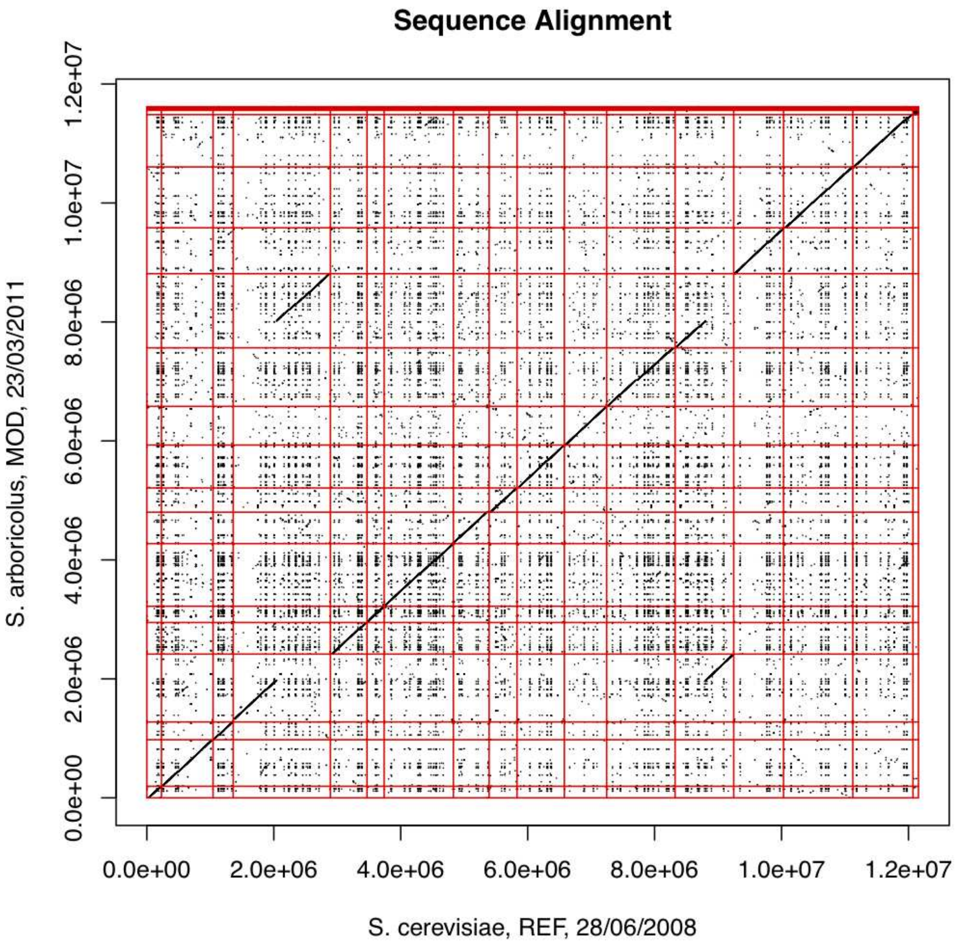


**Figure 3.5** *Dot-matrix plot for sequence alignment between S. cerevisiae and the modified S. arboricolus genome sequence assembly. The matrix shows a fairly straight line along the main diagonal indicating that the two sequences are closely related, with the exception of the reciprocal translocation in chromosomes 4 and 13. Any anomalies can be detected easily when a line does not follow the main diagonal and is plotted in other regions.*

We renamed the scaffolds of *S. arboricolus* to follow the chromosomes of *S. cerevisiae* that shares the homologous sequence. For example, scaffold 3 of *S. arboricolus* was aligned with chromosome 1 of *S. cerevisiae*, thus scaffold 3 was reverse complimented (because it was a reverse strand), and then renamed to chr1. The same process was repeated for each of the *S. arboricolus* scaffolds that linked with *S. cerevisiae* chromosomes and the rest of the scaffolds with sequences small in length were numbered (all <10kb and representing repeat sequences). This modification aids in species comparisons, hence the 'MOD' under the strain category to follow the nomenclature that we had set. The result of modification is shown in Figure 3.5.

The alignment between *S. cerevisiae* and the modified *S. arboricolus* shows that the two genomes are mostly syntenic represented by a fairly straight diagonal line across the dot-matrix plot, with the exception of the reciprocal translocation between chromosomes 4 and 13 that has been confirmed experimentally through laboratory tests. The second-half portion of chromosome 4 is shifted to chromosome 13 and the same goes for the second-half portion of chromosome 13. No anomalies were detected in the new assembly and this validates the result. This technique makes use of the sequence alignment tool to generate data for the dot-matrix plots produced by R and allows for rapid analysis of a new genome sequence.

CHAPTER 4

# DNA REPLICATION ORIGIN DATABASE

## 4.1 Brief Description

Many studies have mapped the replication origin sites of *S. cerevisiae* which provide complementary information. It is advantageous to be able to view all of these datasets at a single location. OriDB was developed to collate all the relevant studies of replication origins, curate and store the datasets in the database, and ultimately to display meaningful information through a website. However, the website was developed five years ago, and is hard to maintain and expand. There is a need to update both the website and database to accommodate new studies and to support more species. As a result, I attempt to enhance OriDB. Prototype websites were developed through a test server before being ported to the actual server with the URLs as shown in Table 4.1.

| Species | Test Server | Actual Server |
|---|---|---|
| *S. cerevisiae* | `http://www.nottingham.ac.uk/plzcnlab/oridb/cerevisiae/` | `http://cerevisiae.oridb.org/` |
| *S. pombe* | `http://www.nottingham.ac.uk/plzcnlab/oridb/pombe/` | `http://pombe.oridb.org/` |

**Table 4.1** *URLs for development and deployment of OriDB. Websites for both S. cerevisiae and S. pombe were developed in the provided test server. The completed websites were then deployed in the actual OriDB server.*

## 4.2 Literature Review

Various biological databases were established to provide a set of services. They are the libraries that store life sciences information collated from published literature, and provide information retrieval through the website for analyses. With the rapid technological advancements in this modern era, there is a paradigm shift in the domain of biological databases [10]. New technologies including NGS are the cause of overwhelming biological data and there is a need to cater for all the varying data types generated by biological research in different fields.

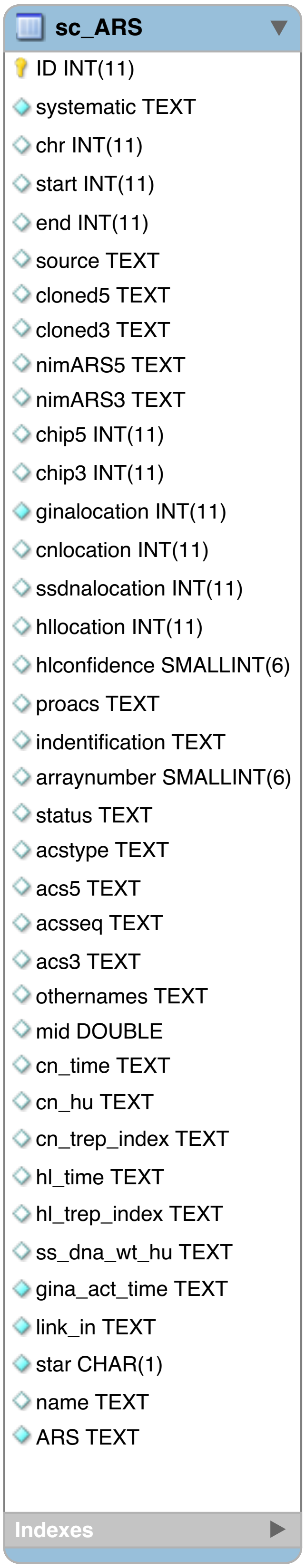
This explains the availability of 1330 online databases in total that are featured in the current *Nucleic Acids Research* (NAR) annual database issue along with 96 new databases and 83 database updates, as well as a new forum for biological databases and curation being established (DATABASE) [27, 28]. The recent redesign of RCSB Protein Data Bank (PDB) website and web services serve as an exemplar of a database update with several enhancements, new features, and improved usability to accommodate a wider community [29].

## 4.3 Database Design

### 4.3.1 Database Model

A poorly structured database can be error-prone. The current database for OriDB comprises only a single table containing a large number of data columns (Figure 4.1). When adding data into a particular field of the table, corresponding data must be added for all the other fields. These data can be represented by null values depending on the criteria set upon each field, but the entire row must be populated nonetheless. This kind of table introduces redundancy and is expressed in terms of functional dependencies. Redundant data leads to various subtle, but significant problems: insert, update, and delete anomalies [30]. Such organisation of data is in fact prone to error as further data is added.

These data could potentially constitute valuable information for knowl-



sc_ARS

ID INT(11)
systematic TEXT
chr INT(11)
start INT(11)
end INT(11)
source TEXT
cloned5 TEXT
cloned3 TEXT
nimARS5 TEXT
nimARS3 TEXT
chip5 INT(11)
chip3 INT(11)
ginalocation INT(11)
cnlocation INT(11)
ssdnalocation INT(11)
hllocation INT(11)
hlconfidence SMALLINT(6)
proacs TEXT
indentification TEXT
arraynumber SMALLINT(6)
status TEXT
acstype TEXT
acs5 TEXT
acsseq TEXT
acs3 TEXT
othernames TEXT
mid DOUBLE
cn_time TEXT
cn_hu TEXT
cn_trep_index TEXT
hl_time TEXT
hl_trep_index TEXT
ss_dna_wt_hu TEXT
gina_act_time TEXT
link_in TEXT
star CHAR(1)
name TEXT
ARS TEXT

**Figure 4.1** *ERDs of the existing OriDB database system. It comprises of only a single table with all the data cluttered in it. This approach is error-prone when updating the table.*

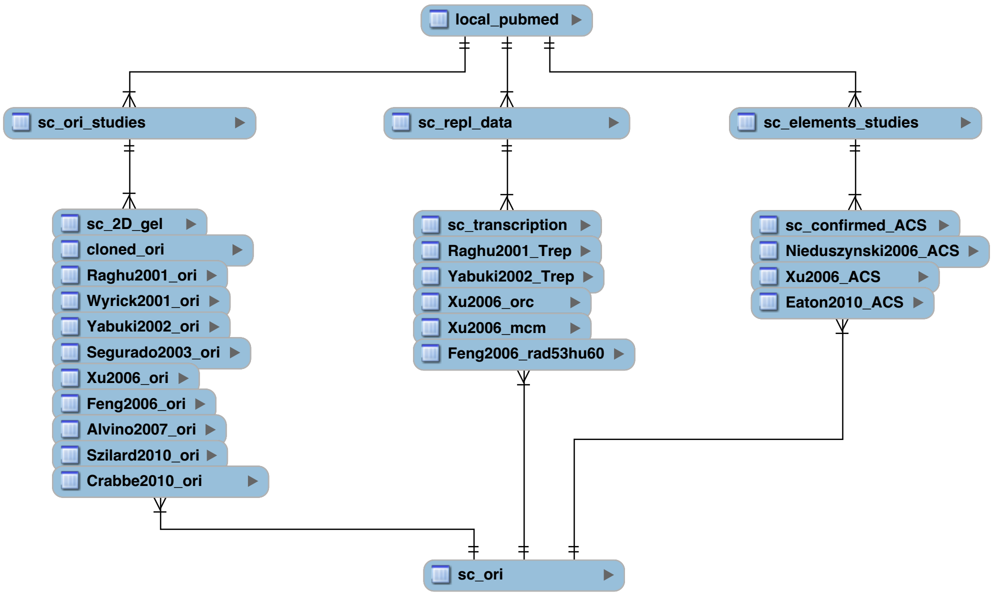


**Figure 4.2** *ERDs of the new S. cerevisiae OriDB database system (simplified for compactness). Data are categorised in tables to have an organised structure of the database. Data from 'local_pubmed' will be shared whenever new species has been added.*

edge extraction and a database with an organised storage of data is mandatory in data mining [28]. To achieve a well-structured database, instead of using just a table to catalogue all of the data, the solution is to normalise the database. Database normalisation involves partitioning large tables into smaller less redundant tables while imposing relations between them. These relations set the criteria to link data from different tables. Here we adopt the relational approach of modelling (entity-relationship modelling or ERM) to build a relational database management system (RDBMS).

During the modelling process, entity-relationship diagrams (ERDs) were generated as a blueprint for the design. Figure 4.2 shows the relationships between tables using Crow's Foot notation. Relationships are associated among these tables so that data are interconnected between linked tables. The purpose is to allow the propagation of data to all the tables via the defined relationships when data is added, deleted, or updated in one of the tables [30]. Maintenance on OriDB can then be performed without difficulties using the ERM approach. The produced ERDs (Figure 4.2) represent the current version of the database; in the future, further tables can be added as more studies published and curated. Refer to Appendix B.1 for a comprehensive view of the entities with their attributes.

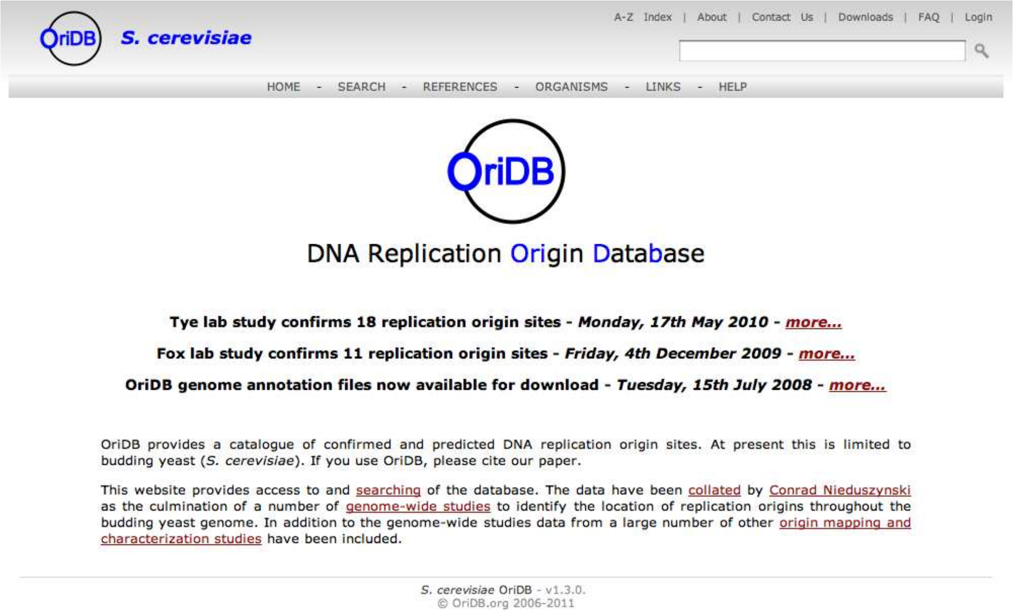
### 4.3.2 Information Retrieval

The 'local_pubmed' table (in Figure 4.2) stores information about publications relevant to the studies curated in the OriDB website. New references are needed when adding new studies to the database and these references are automatically retrieved from PubMed. As an added function, OriDB stores these reference data in the database. Whenever a user visits a reference, OriDB retrieves the data from 'local_pubmed' via PubMed ID associated to that reference. If the PubMed ID data is not available locally, it is extracted directly from PubMed and stored locally for future use. References need to be downloaded only once and any future visits will retrieve the references locally. This helps minimise the downtime when loading the references in OriDB; it is faster to load data locally rather than to download from PubMed each time for display.

## 4.4 Website Design

A website with strong foundation relies on the initial design. To totally revamp the website, code from the previous OriDB was not be reused; my intention was to develop OriDB from scratch. Most of the functionality remains however so that users of the existing OriDB will have a familiar experience but with the increased quality of the new OriDB. Even so, OriDB will continue to evolve step-by-step through feedback from the users.

Future-proofing is desired for OriDB in the sense that new studies can be displayed swiftly when added to the database, or new species can be incorporated to the website with ease. This is the long-term vision of OriDB to allow



OriDB
S. cerevisiae
A-Z Index | About | Contact Us | Downloads | FAQ | Login
HOME - SEARCH - REFERENCES - ORGANISMS - LINKS - HELP
OriDB

DNA Replication Origin Database

**Tye lab study confirms 18 replication origin sites** - ***Monday, 17th May 2010*** - ***more...***

**Fox lab study confirms 11 replication origin sites** - ***Friday, 4th December 2009*** - ***more...***

**OriDB genome annotation files now available for download** - ***Tuesday, 15th July 2008*** - ***more...***

OriDB provides a catalogue of confirmed and predicted DNA replication origin sites. At present this is limited to budding yeast (*S. cerevisiae*). If you use OriDB, please cite our paper.

This website provides access to and searching of the database. The data have been collated by Conrad Nieduszynski as the culmination of a number of genome-wide studies to identify the location of replication origins throughout the budding yeast genome. In addition to the genome-wide studies data from a large number of other origin mapping and characterization studies have been included.

*S. cerevisiae* OriDB - v1.3.0.
© OriDB.org 2006-2011

**Figure 4.3** *Screenshot of OriDB's new website displaying its main page. It is visually easy on the eyes and simple to navigate through the web pages. The ORGANISMS link will lead the users to a list of available species and the selected species will show its name on the top left corner of every page.*

the website to accommodate such expansion with the advancement of replication origin research. To benefit from the idea of future expansion, OriDB needs to be designed in such a way that the website can manage new data without trouble. Whenever new data is added to the database, the revamped OriDB will automatically include the new data and thus display that information in the appropriate web pages.

Furthermore, the layout of the website has been redesigned to increase readability and usability. Usability is vital in navigating the web pages and having a user-friendly interface is therefore crucial. Web links are planned accordingly to lead the users to the most relevant information. Figure 4.3 displays the main page of OriDB where users have access to most of the links on both the top bar and the navigation bar as well as to search for the names of replication origins in the search text box.

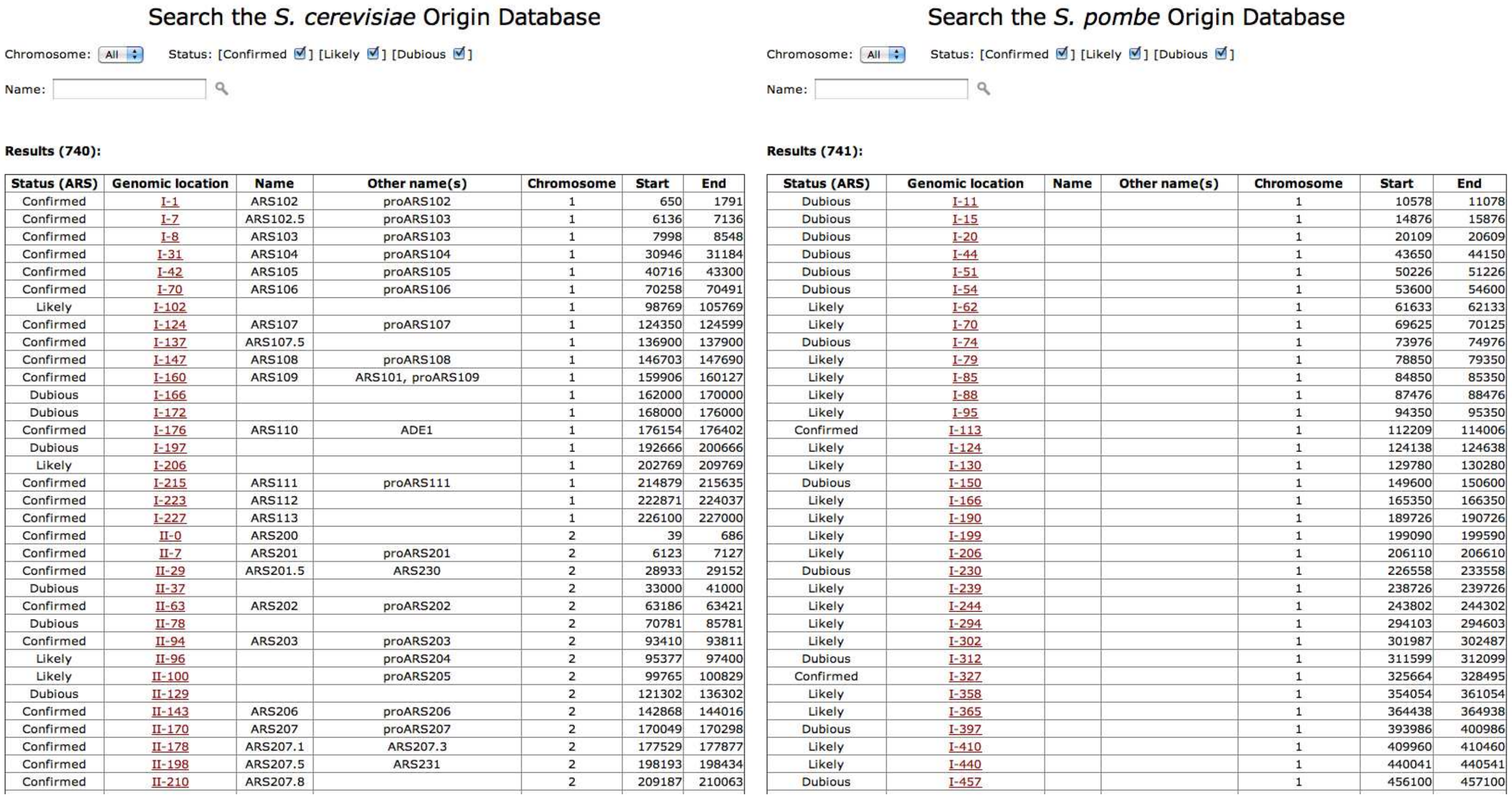
### 4.4.1 Web Structure

One of the major enhancement is that *S. pombe* is incorporated in OriDB. Users are able to access the site for *S. cerevisiae* and *S. pombe* replication origins through “ORGANISMS” located in the navigation bar. A list of replication origins with their assigned status, genomic location, the names allocated to the origins, and the chromosomes with start and end coordinates associated to the origins can be extracted from the database through the “SEARCH” facility or the search text box provided. Figure 4.4 shows part of the search pages for both *S. cerevisiae* and *S. pombe*, each retrieved a total of 740 and 741 results respectively from the database. Since the incorporation of *S. pombe* to OriDB is fairly recent, it currently does not have the names of origins recorded but this will be included in the future. For the time being, examples from *S. cerevisiae* will be used throughout.

The search criteria includes the chromosome number, the assigned status of the origin, and the names of the origins. For instance, entering ‘606’ with all the status and chromosomes included in the criteria will return two origins, one with the name ‘ARS606’ (or proARS606) and the other origin with an alternative name associated to it (proARS1606). Links are provided under the “Genomic location” column which redirect users to the details page.*

The details page will be similar with the previous OriDB to some extent, displaying a panel with seven components: Origin Summary Information, Origin Summary Graphics, Origin Location Assignments, Origin Sequence Elements, Phylogenetic Sequence Conservation, User Notes, and References for this Origin [19]. The details page for *S. pombe* has four tabs instead to hide information irrelevant to the species. Changes made to the tabs include the use of jQuery to implement asynchronous event loading. Tabs can be navigated rapidly without the need to reload the pages each time a tab is clicked. More-

*Genomic location of an origin is represented in the following format: VI-168 where VI is the chromosome number in roman numerals and 168 is the location calculated by adding the start and end coordinates of the origin, then divide by 2000, and finally to round the result.



Search the *S. cerevisiae* Origin Database

Chromosome: All

Status: [Confirmed ☑] [Likely ☑] [Dubious ☑]

Name:

**Results (740):**

| Status (ARS) | Genomic location | Name | Other name(s) | Chromosome | Start | End |
|---|---|---|---|---|---|---|
| Confirmed | I-1 | ARS102 | proARS102 | 1 | 650 | 1791 |
| Confirmed | I-7 | ARS102.5 | proARS103 | 1 | 6136 | 7136 |
| Confirmed | I-8 | ARS103 | proARS103 | 1 | 7998 | 8548 |
| Confirmed | I-31 | ARS104 | proARS104 | 1 | 30946 | 31184 |
| Confirmed | I-42 | ARS105 | proARS105 | 1 | 40716 | 43300 |
| Confirmed | I-70 | ARS106 | proARS106 | 1 | 70258 | 70491 |
| Likely | I-102 | | | 1 | 98769 | 105769 |
| Confirmed | I-124 | ARS107 | proARS107 | 1 | 124350 | 124599 |
| Confirmed | I-137 | ARS107.5 | | 1 | 136900 | 137900 |
| Confirmed | I-147 | ARS108 | proARS108 | 1 | 146703 | 147690 |
| Confirmed | I-160 | ARS109 | ARS101, proARS109 | 1 | 159906 | 160127 |
| Dubious | I-166 | | | 1 | 162000 | 170000 |
| Dubious | I-172 | | | 1 | 168000 | 176000 |
| Confirmed | I-176 | ARS110 | ADE1 | 1 | 176154 | 176402 |
| Dubious | I-197 | | | 1 | 192666 | 200666 |
| Likely | I-206 | | | 1 | 202769 | 209769 |
| Confirmed | I-215 | ARS111 | proARS111 | 1 | 214879 | 215635 |
| Confirmed | I-223 | ARS112 | | 1 | 222871 | 224037 |
| Confirmed | I-227 | ARS113 | | 1 | 226100 | 227000 |
| Confirmed | II-0 | ARS200 | | 2 | 39 | 686 |
| Confirmed | II-7 | ARS201 | proARS201 | 2 | 6123 | 7127 |
| Confirmed | II-29 | ARS201.5 | ARS230 | 2 | 28933 | 29152 |
| Dubious | II-37 | | | 2 | 33000 | 41000 |
| Confirmed | II-63 | ARS202 | proARS202 | 2 | 63186 | 63421 |
| Dubious | II-78 | | | 2 | 70781 | 85781 |
| Confirmed | II-94 | ARS203 | proARS203 | 2 | 93410 | 93811 |
| Likely | II-96 | | proARS204 | 2 | 95377 | 97400 |
| Likely | II-100 | | proARS205 | 2 | 99765 | 100829 |
| Dubious | II-129 | | | 2 | 121302 | 136302 |
| Confirmed | II-143 | ARS206 | proARS206 | 2 | 142868 | 144016 |
| Confirmed | II-170 | ARS207 | proARS207 | 2 | 170049 | 170298 |
| Confirmed | II-178 | ARS207.1 | ARS207.3 | 2 | 177529 | 177877 |
| Confirmed | II-198 | ARS207.5 | ARS231 | 2 | 198193 | 198434 |
| Confirmed | II-210 | ARS207.8 | | 2 | 209187 | 210063 |

Search the *S. pombe* Origin Database

Chromosome: All

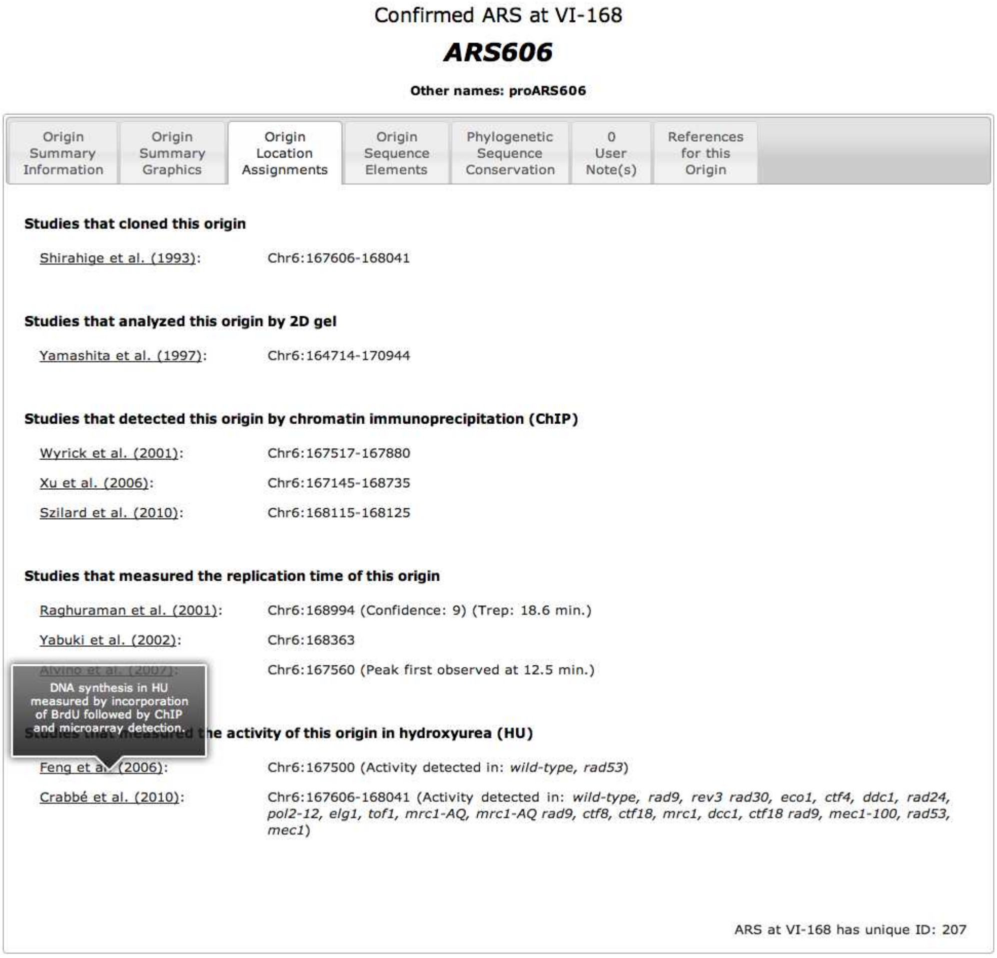
Status: [Confirmed ☑] [Likely ☑] [Dubious ☑]

Name:

**Results (741):**

| Status (ARS) | Genomic location | Name | Other name(s) | Chromosome | Start | End |
|---|---|---|---|---|---|---|
| Dubious | I-11 | | | 1 | 10578 | 11078 |
| Dubious | I-15 | | | 1 | 14876 | 15876 |
| Dubious | I-20 | | | 1 | 20109 | 20609 |
| Dubious | I-44 | | | 1 | 43650 | 44150 |
| Dubious | I-51 | | | 1 | 50226 | 51226 |
| Dubious | I-54 | | | 1 | 53600 | 54600 |
| Likely | I-62 | | | 1 | 61633 | 62133 |
| Likely | I-70 | | | 1 | 69625 | 70125 |
| Dubious | I-74 | | | 1 | 73976 | 74976 |
| Likely | I-79 | | | 1 | 78850 | 79350 |
| Likely | I-85 | | | 1 | 84850 | 85350 |
| Likely | I-88 | | | 1 | 87476 | 88476 |
| Likely | I-95 | | | 1 | 94350 | 95350 |
| Confirmed | I-113 | | | 1 | 112209 | 114006 |
| Likely | I-124 | | | 1 | 124138 | 124638 |
| Likely | I-130 | | | 1 | 129780 | 130280 |
| Dubious | I-150 | | | 1 | 149600 | 150600 |
| Likely | I-166 | | | 1 | 165350 | 166350 |
| Likely | I-190 | | | 1 | 189726 | 190726 |
| Likely | I-199 | | | 1 | 199090 | 199590 |
| Likely | I-206 | | | 1 | 206110 | 206610 |
| Dubious | I-230 | | | 1 | 226558 | 233558 |
| Likely | I-239 | | | 1 | 238726 | 239726 |
| Likely | I-244 | | | 1 | 243802 | 244302 |
| Likely | I-294 | | | 1 | 294103 | 294603 |
| Likely | I-302 | | | 1 | 301987 | 302487 |
| Dubious | I-312 | | | 1 | 311599 | 312099 |
| Confirmed | I-327 | | | 1 | 325664 | 328495 |
| Likely | I-358 | | | 1 | 354054 | 361054 |
| Likely | I-365 | | | 1 | 364438 | 364938 |
| Dubious | I-397 | | | 1 | 393986 | 400986 |
| Likely | I-410 | | | 1 | 409960 | 410460 |
| Likely | I-440 | | | 1 | 440041 | 440541 |
| Dubious | I-457 | | | 1 | 456100 | 457100 |

**Figure 4.4** *Screenshot of the search page for both S. cerevisiae and S. pombe, each displaying all the origins available in the database. Users can set the search criteria to narrow the returned search results. S. pombe has only recently been added to the database and so without the names filled at the moment. This is just to prove that new species can be incorporated in the new OriDB.*


Origin Summary Information
Origin Summary Graphics
Origin Location Assignments
Origin Sequence Elements
Phylogenetic Sequence Conservation
0 User Note(s)
References for this Origin

Confirmed ARS at VI-168

***ARS606***

**Other names: proARS606**

**Studies that cloned this origin**

| | |
|---|---|
| Shirahige et al. (1993): | Chr6:167606-168041 |

**Studies that analyzed this origin by 2D gel**

| | |
|---|---|
| Yamashita et al. (1997): | Chr6:164714-170944 |

**Studies that detected this origin by chromatin immunoprecipitation (ChIP)**

| | |
|---|---|
| Wyrick et al. (2001): | Chr6:167517-167880 |
| Xu et al. (2006): | Chr6:167145-168735 |
| Szilard et al. (2010): | Chr6:168115-168125 |

**Studies that measured the replication time of this origin**

| | |
|---|---|
| Raghuraman et al. (2001): | Chr6:168994 (Confidence: 9) (Trep: 18.6 min.) |
| Yabuki et al. (2002): | Chr6:168363 |
| Alvino et al. (2007): | Chr6:167560 (Peak first observed at 12.5 min.) |

DNA synthesis in HU measured by incorporation of BrdU followed by ChIP and microarray detection.

**Studies that measured the activity of this origin in hydroxyurea (HU)**

| | |
|---|---|
| Feng et al. (2006): | Chr6:167500 (Activity detected in: *wild-type, rad53*) |
| Crabbé et al. (2010): | Chr6:167606-168041 (Activity detected in: *wild-type, rad9, rev3 rad30, eco1, ctf4, ddc1, rad24, pol2-12, elg1, tof1, mrc1-AQ, mrc1-AQ rad9, ctf8, ctf18, mrc1, dcc1, ctf18 rad9, mec1-100, rad53, mec1*) |

ARS at VI-168 has unique ID: 207

**Figure 4.5** *Screenshot of the details page for S. cerevisiae containing seven tabs on the panel (four tabs for S. pombe). These tabs were implemented using jQuery to allow swift shifting between tabs without reloading the whole page. Some of the contents provide jQuery tooltips to display detailed information.*

over, some of the elements, in Origin Location Assignments tab for example, provide extra information in the tooltip when users hover their cursor over the authors as shown in Figure 4.5. These improvements enhance the users experience by giving them responsive and interactive website, whilst providing more useful information.

### 4.4.2 Graphical Data Presentation

The ability to visualise data allows users to view relevant experimental data in graphical format. The former Chromosome Viewer uses Adobe Flash technology to visualise data. However, it takes time when loading the viewer as it is cumbersome. The complexity of the viewer makes it difficult to add new studies into the graph. It is therefore obsolete when it comes to future expansion. By utilising technology that specifically deals with generating charts, Highcharts JS does the job perfectly. Highcharts is simple yet powerful in that it is a charting library which utilises JavaScript and is compatible to most of the modern browsers. The graphs produced by Highcharts is in SVG format as default, with the exception of Internet Explorer which Highcharts uses Vector Markup Language (VML) as an alternative for browser compatibility.

The new Data Viewer utilises Highcharts for graphical data presentation. The new viewer adds interactivity where users can hover across the data of a study and view the value on a particular location, or to view extra information by hovering on the studies in the legend. Figure 4.6 illustrates the three different views of Data Viewer in the Origin Summary Graphics tab of the details page: Default, Detailed, and Zoomed Out view. The information presented in Data Viewer does not change much compared to the previous viewer, except for the data displayed in the Detailed view. Instead of displaying data of superhelically induced duplex destabilisation (SIDD), DNA helical stability data is used in place of SIDD. It follows that of WEB-THERMODYN where the free energy required to unwind the double-stranded helix for a particular sequence is calculated and displayed in the viewer [31].

The charts shown in Figure 4.6 were used as an example where images can be downloaded (in PNG, JPEG, PDF, or SVG format) using Highcharts additional module for print and export of the generated charts. This is convenient for users who want charts with high-resolution to be embedded in their reports instead of taking screenshots of the browser and cropping it. Users are also given a choice to print the charts on the spot without including the whole page.

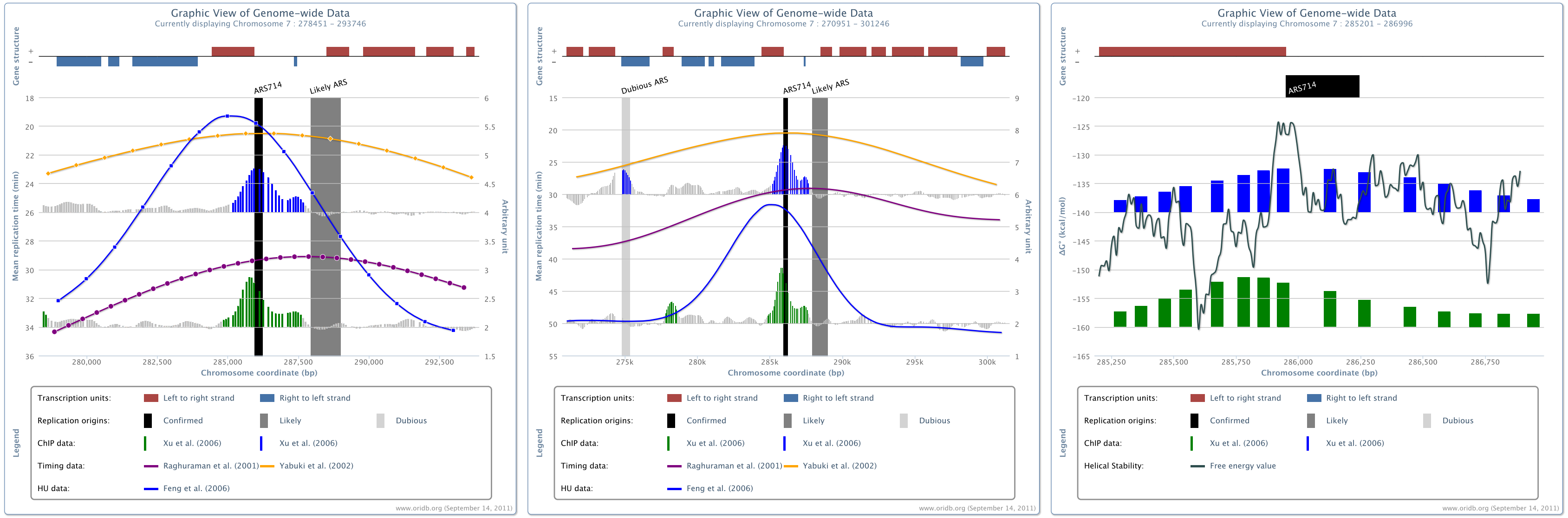
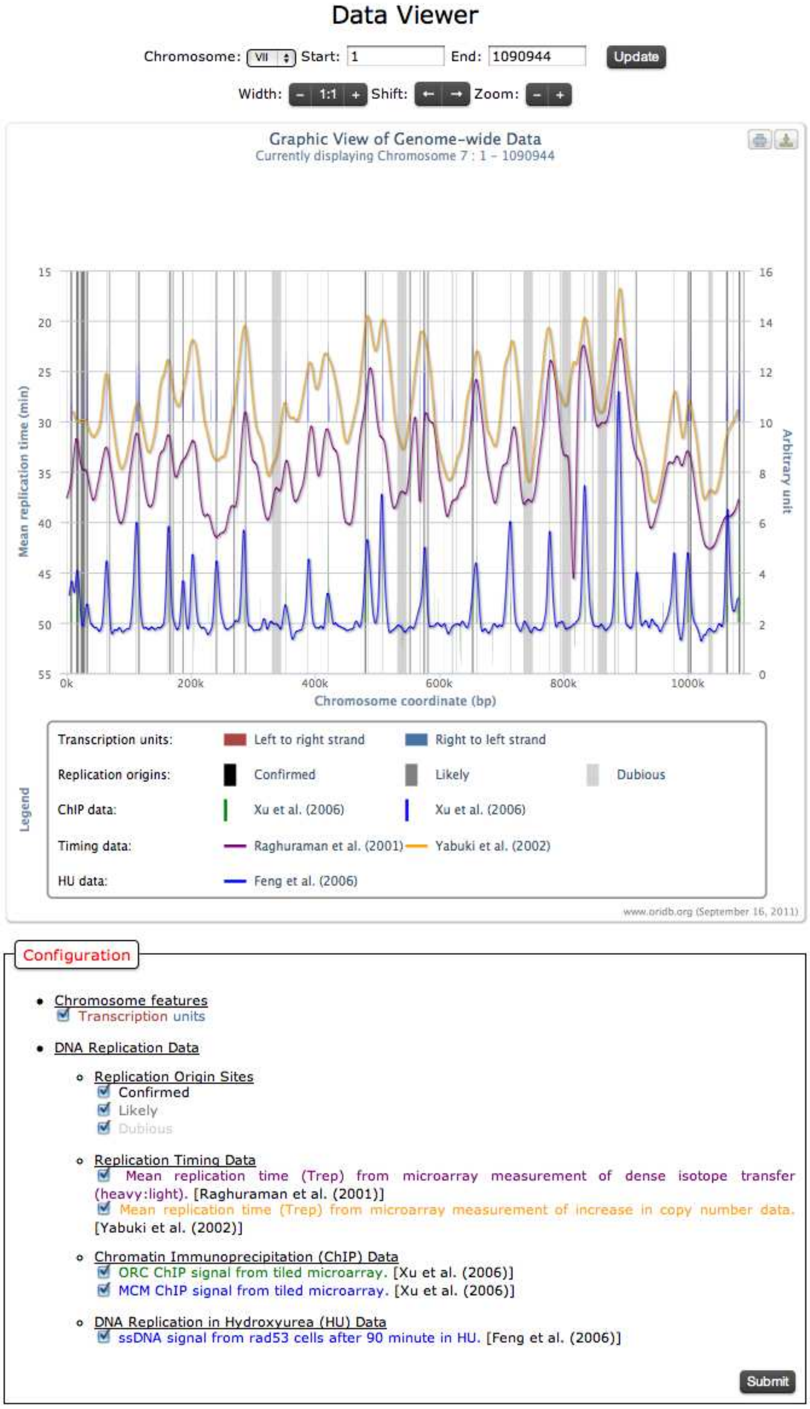

Graphic View of Genome-wide Data
Currently displaying Chromosome 7 : 278451 - 293746
Gene structure
ARS714
Likely ARS
Mean replication time (min)
Arbitrary unit
Chromosome coordinate (bp)
280,000
282,500
285,000
287,500
290,000
292,500
Legend
Transcription units: Left to right strand Right to left strand
Replication origins: Confirmed Likely Dubious
ChIP data: Xu et al. (2006) Xu et al. (2006)
Timing data: Raghuraman et al. (2001) Yabuki et al. (2002)
HU data: Feng et al. (2006)
www.oridb.org (September 14, 2011)
Graphic View of Genome-wide Data
Currently displaying Chromosome 7 : 270951 - 301246
Dubious ARS
ARS714
Likely ARS
275k
280k
285k
290k
295k
300k
Graphic View of Genome-wide Data
Currently displaying Chromosome 7 : 285201 - 286996
ARS714
ΔG° (kcal/mol)
285,250
285,500
285,750
286,000
286,250
286,500
286,750
Helical Stability: Free energy value

**Figure 4.6** *Images of default, expanded, and reduced view of chromosome data respectively. When the length of chromosome is lower than a set threshold, helical stability data is displayed in place of timing and HU data (reduced view).*

*These high-resolution images were downloaded through the Highcharts print and export module. Date of download is embedded at the bottom right corner.*



Data Viewer

Chromosome: VII Start: 1 End: 1090944 Update

Width: − 1:1 + Shift: ← → Zoom: − +

Graphic View of Genome-wide Data

Currently displaying Chromosome 7 : 1 – 1090944

Legend

Transcription units: 🟥 Left to right strand 🟦 Right to left strand

Replication origins: ■ Confirmed ■ Likely ■ Dubious

ChIP data: | Xu et al. (2006) | Xu et al. (2006)

Timing data: — Raghuraman et al. (2001) — Yabuki et al. (2002)

HU data: — Feng et al. (2006)

www.oridb.org (September 16, 2011)

Configuration

- Chromosome features
  - ☑ Transcription units
- DNA Replication Data
  - Replication Origin Sites
    - ☑ Confirmed
    - ☑ Likely
    - ☑ Dubious
  - Replication Timing Data
    - ☑ Mean replication time (Trep) from microarray measurement of dense isotope transfer (heavy:light). [Raghuraman et al. (2001)]
    - ☑ Mean replication time (Trep) from microarray measurement of increase in copy number data. [Yabuki et al. (2002)]
  - Chromatin Immunoprecipitation (ChIP) Data
    - ☑ ORC ChIP signal from tiled microarray. [Xu et al. (2006)]
    - ☑ MCM ChIP signal from tiled microarray. [Xu et al. (2006)]
  - DNA Replication in Hydroxyurea (HU) Data
    - ☑ ssDNA signal from rad53 cells after 90 minute in HU. [Feng et al. (2006)]

Submit

**Figure 4.7** *Screenshot of the chromosome Data Viewer displaying data of the whole range of chromosome 7. The transcription unit will be hidden when the chromosome length viewed exceeds a set threshold. Control and navigation of the Data Viewer are located at the top, and the collapsible configuration panel below offers the users to view or hide some of the studies. The top right corner of the viewer provides the print and export module for the chart.*

To provide more control over the viewer, users can access a more advance viewer through the link provided below in Origin Summary Graphics. This brings up a separate page which provides flexibility in handling the viewer (Figure 4.7). Zooming and shifting of chromosome details in the viewer does not rely on Adobe Flash anymore, event handling with jQuery was implemented on the navigational buttons instead. This method offers speed and control where users can manipulate the structure of the viewer easily, including the width of the viewer to suit users with wider screens. On top of that, the collapsible panel below the viewer contains more options for the users to select which type of data they want the viewer to display.

## 4.5 Outcome

The revamped OriDB will have better performance in data mining due to the improved database structure. The search space is significantly lower in the new tables, thus decreasing load time while querying the database. The cleaner code structure also allows developers to expand OriDB easily in the future by reusing the code. By introducing new methods such as Highcharts and jQuery implementation to the website, user experience will be enhanced with the added interactivity and ease of navigation through the website. With everything considered, OriDB does indeed improved with faster interface for the users.

CHAPTER 5

# CONCLUSIONS

## 5.1 Future Enhancements

### 5.1.1 LiftOver

A local Mac OS X server was used in place of the Granby server because Granby server comes with SunOS. I had tried to compile LiftOver tool (written in C) in Granby by modifying the codes and changing the settings for installation, to the extent that I contacted the genome bioinformatics group in UCSC for advice, but to no avail. The UCSC's group replied that SunOS 5.9 is a legacy operating system and that we need a skilled technician to port the application. As an alternative, the university holds another server (Caunton) that deals with high-performance computing. It uses Linux operating system which LiftOver is compatible with, but we were not given a workspace in the server. Perhaps in the future when there is an available server capable of running LiftOver, the LiftOver utility that I had developed can be brought online to make the site publicly available for the wider community.

As for the LiftOver tool itself, it was originally designed for converting genome coordinates of closely related species. Conversion between different assemblies of the same strain yields over 99% identity in sequence, whereas conversion between different strains of the same species yields approximately 99% identity in sequence. These results prove the reliability of LiftOver to convert similar strains. By contrast, conversion between different species yields approximately 80% identity in mapped sequence but 20% identity where gaps exist. The percentage varies with more distant species yielding lower identity and therefore more unmapped regions exist. This might be due to the default settings of LASTZ.

To deal with this issue, in the future we could try to lower the threshold of alignments precision. The lower it is, the more regions can be mapped and thus increase the percentage of identity. However, too low a threshold may lead to false positive paradox. If successful, high identity mapping can be done for continuous sequence between different species. Additional configurations can be added to the web interface to implement the settings that control LASTZ. This would equips the LiftOver utility for different kinds of mappings as per users choice.

### 5.1.2 OriDB

At present, OriDB uses the October 2003 assembly for *S. cerevisiae* genome to date. There have been multiple modifications in genome sequence since then. Modifications include insertions, substitutions, or deletions of nucleotides in a particular sequence and these can be drastic to how the data are displayed to the users no matter how small the changes are. In the future, OriDB should employ more recent assemblies for the featured species. Furthermore, OriDB is limited to *S. cerevisiae* and is just in the process of including *S. pombe* in the database. Future addition of different species might be important. To allow the inclusion of new species, OriDB needs to be as general as possible to make expansion as straight forward as possible.

The integration of Web 2.0 and 3.0 for biological databases may become an important feature in the future [28]. Web 2.0 facilitates user-centric information sharing. To implement this feature for example, the User Notes component in the details page can be modified to allow users who are given permission (by logging in to OriDB) to state further manually curated information. Web 3.0 on the other hand facilitates the semantic approach to web usage. This can be thought of having users who logged into OriDB to have a personalised search preference where the search engine gears toward the users instead of keywords. By way of illustration, a person who is interested mostly in graphical view of ARS606 in *S. cerevisiae* for certain features will likely to get the desired results through the search facility in OriDB, rather than having to click through all the way to the Data Viewer with those features that the person needs.

## 5.2 Summary

Presented here is the online LiftOver utility that deals specifically with yeast species. LiftOver chain files can be created when needed for two particular genome or in batch. These chain files are stored in a specified directory for the utility to detect the available chain files and provide lists of three columns for target and query genomes: species, strain, and assembly. Users are able to configure the settings on the LiftOver website and input the required data for genome coordinates conversion. These tools are made to help biologists maximise the value of their data where they can compare yeast genome assemblies with visualised data.

On top of that, OriDB has been revamped for better web structure and ease of maintenance. It is redesigned to be future proof such that new studies or organisms can be added to the database effortlessly. As for the graphical data viewer, Highcharts JS was utilised extensively for interactive chart interface. Layout of the configurations for data viewer are in order and users are able to download or print the graphs from the data viewer output.

# References

[1] Sanger, F., Air, G. M., Barrell, B. G., Brown, N. L., Coulson, A. R., Fiddes, J. C., Hutchison, C. A., Slocombe, P. M., and Smith, M. (1977) Nucleotide sequence of bacteriophage $\phi$X174 DNA. *Nature*, **265**(5596), 687–695.

[2] Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M., Louis, E. J., Mewes, H. W., Murakami, Y., Philippsen, P., Tettelin, H., and Oliver, S. G. (1996) Life with 6000 Genes. *Science*, **274**(5287), 546–567.

[3] Nickoloff, J. A. and Hoekstra, M. F. (1998) *DNA Damage and Repair: DNA repair in prokaryotes and lower eukaryotes*, Vol. 1, New Jersey: Humana Press.

[4] Sanger, F., Nicklen, S., and Coulson, A. R. (1977) DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, **74**(12), 5463–5467.

[5] Brenner, S., Johnson, M., Bridgham, J., Golda, G., Lloyd, D. H., Johnson, D., Luo, S., McCurdy, S., Foy, M., Ewan, M., Roth, R., George, D., Eletr, S., Albrecht, G., Vermaas, E., Williams, S. R., Moon, K., Burcham, T., Pallas, M., DuBridge, R. B., Kirchner, J., Fearon, K., Mao, J.-i., and Corcoran, K. (2000) Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nature Biotechnology*, **18**(6), 630–634.

[6] Schuster, S. C. (2008) Next-generation sequencing transforms today's biology. *Nature Methods*, **5**(1), 16–18.

[7] Wetterstrand, K. A. *DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program.* Available at: `www.genome.gov/sequencingcosts`. Accessed [26.08.2011].

[8] Liolios, K., Mavromatis, K., Tavernarakis, N., and Kyrpides, N. C. (2008) The Genomes On Line Database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research,* **36**(suppl 1), D475–D479.

[9] Cherry, J. M., Adler, C., Ball, C., Chervitz, S. A., Dwight, S. S., Hester, E. T., Jia, Y., Juvik, G., Roe, T., Schroeder, M., Weng, S., and Botstein, D. (1998) SGD: Saccharomyces Genome Database. *Nucleic Acids Research,* **26**(1), 73–79.

[10] Schattner, P. (2008) *Genomes, browsers, and databases: data-mining tools for integrated genomic databases,* New York: Cambridge University Press.

[11] Hinrichs, A. S., Karolchik, D., Baertsch, R., Barber, G. P., Bejerano, G., Clawson, H., Diekhans, M., Furey, T. S., Harte, R. A., Hsu, F., Hillman-Jackson, J., Kuhn, R. M., Pedersen, J. S., Pohl, A., Raney, B. J., Rosenbloom, K. R., Siepel, A., Smith, K. E., Sugnet, C. W., Sultan-Qurraie, A., Thomas, D. J., Trumbower, H., Weber, R. J., Weirauch, M., Zweig, A. S., Haussler, D., and Kent, W. J. (2006) The UCSC Genome Browser Database: update 2006. *Nucleic Acids Research,* **34**(suppl 1), D590–D598.

[12] Kent, W. J., Baertsch, R., Hinrichs, A., Miller, W., and Haussler, D. (2003) Evolution's cauldron: Duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences,* **100**(20), 11484–11489.

[13] Kent, W. J. (2002) BLAT—The BLAST-Like Alignment Tool. *Genome Research,* **12**(4), 656–664.

[14] Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C., Haussler, D., and Miller, W. (2003) Human–Mouse Alignments with BLASTZ. *Genome Research,* **13**(1), 103–107.

[15] Harris, R. S. (2007) *Improved pairwise alignment of genomic DNA.* PhD thesis, The Pennsylvania State University.

[16] Lodish, H., Berk, A., Kaiser, C. A., Krieger, M., Scott, M. P., Bretscher, A., Ploegh, H., and Matsudaira, P. (2008) *Molecular cell biology*, New York: W.H. Freeman. 6th edition.

[17] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002) *Molecular Biology of the Cell*, New York: Garland Science. 4th edition.

[18] Berbenetz, N. M., Nislow, C., and Brown, G. W. (2010) Diversity of Eukaryotic DNA Replication Origins Revealed by Genome-Wide Analysis of Chromatin Structure. *PLoS Genet,* **6**(9), e1001092.

[19] Nieduszynski, C. A., Hiraga, S.-i., Ak, P., Benham, C. J., and Donaldson, A. D. (2007) OriDB: a DNA replication origin database. *Nucleic Acids Research,* **35**(suppl 1), D40–D46.

[20] Dai, J., Chuang, R.-Y., and Kelly, T. J. (2005) DNA replication origins in the Schizosaccharomyces pombe genome. *Proceedings of the National Academy of Sciences of the United States of America,* **102**(2), 337–342.

[21] Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., Miller, W., Kent, W. J., and Nekrutenko, A. (2005) Galaxy: A platform for interactive large-scale genome analysis. *Genome Research,* **15**(10), 1451–1455.

[22] Sayers, E. W., Barrett, T., Benson, D. A., Bolton, E., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., DiCuccio, M., Federhen, S., Feolo, M., Fingerman, I. M., Geer, L. Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D. J., Lu, Z., Madden, T. L., Madej, T., Maglott, D. R., Marchler-Bauer, A., Miller, V., Mizrachi, I., Ostell, J., Panchenko, A., Phan, L., Pruitt, K. D., Schuler, G. D., Sequeira, E., Sherry, S. T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T. A., Wagner, L., Wang, Y., Wilbur, W. J., Yaschenko, E., and Ye, J. (2011) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research,* **39**(suppl 1), D38–D51.

[23] Flicek, P., Amode, M. R., Barrell, D., Beal, K., Brent, S., Chen, Y., Clapham, P., Coates, G., Fairley, S., Fitzgerald, S., Gordon, L., Hendrix, M., Hourlier, T., Johnson, N., Kähäri, A., Keefe, D., Keenan, S., Kinsella, R., Kokocinski, F., Kulesha, E., Larsson, P., Longden, I., McLaren, W., Overduin, B., Pritchard, B., Riat, H. S., Rios, D., Ritchie, G. R. S., Ruffier, M., Schuster, M., Sobral, D., Spudich, G., Tang, Y. A., Trevanion, S., Vandrovcova, J., Vilella, A. J., White, S., Wilder, S. P., Zadissa, A., Zamora, J., Aken, B. L., Birney, E., Cunningham, F., Dunham, I., Durbin, R., Fernández-Suarez, X. M., Herrero, J., Hubbard, T. J. P., Parker, A., Proctor, G., Vogel, J., and Searle, S. M. J. (2011) Ensembl 2011. *Nucleic Acids Research,* **39**(suppl 1), D800–D806.

[24] Tweedie, S., Ashburner, M., Falls, K., Leyland, P., McQuilton, P., Marygold, S., Millburn, G., Osumi-Sutherland, D., Schroeder, A., Seal, R., Zhang, H., and Consortium, T. F. (2009) FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucleic Acids Research,* **37**(suppl 1), D555–D559.

[25] Junier, T. and Pagni, M. (2, 2000) Dotlet: diagonal plots in a Web browser. *Bioinformatics,* **16**(2), 178–179.

[26] Wang, S.-A. and Bai, F.-Y. (2008) Saccharomyces arboricolus sp. nov., a yeast species from tree bark. *International Journal of Systematic and Evolutionary Microbiology,* **58**(2), 510–514.

[27] Galperin, M. Y. and Cochrane, G. R. (2011) The 2011 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic Acids Research,* **39**(suppl 1), D1–D6.

[28] Landsman, D., Gentleman, R., Kelso, J., and Francis Ouellette, B. F. (2009) DATABASE: A new forum for biological databases and curation. *Database,* **2009**.

[29] Rose, P. W., Beran, B., Bi, C., Bluhm, W. F., Dimitropoulos, D., Goodsell, D. S., Prlić, A., Quesada, M., Quinn, G. B., Westbrook, J. D., Young, J., Yukich, B., Zardecki, C., Berman, H. M., and Bourne, P. E. (2011) The RCSB

Protein Data Bank: redesigned web site and web services. *Nucleic Acids Research,* **39**(suppl 1), D392–D401.

[30] Connolly, T. and Begg, C. (2005) *Database systems: a practical approach to design, implementation, and management*, International computer science series: Addison-Wesley.

[31] Huang, Y. and Kowalski, D. (2003) WEB-THERMODYN: sequence analysis software for profiling DNA helical stability. *Nucleic Acids Research,* **31**(13), 3819–3821.

APPENDIX A

# SUPPLEMENTARY DATA - LIFTOVER

## A.1 Utilities and Their Paths

| Utility | Path |
|---:|:---|
| `faSplit` | `kent/src/utils/faSplit` |
| `faToTwoBit` | `kent/src/utils/faToTwoBit` |
| `twoBitInfo` | `kent/src/utils/twoBitInfo` |
| `liftUp` | `kent/src/hg/liftUp` |
| `axtChain` | `kent/src/hg/mouseStuff/axtChain` |
| `chainMergeSort` | `kent/src/hg/mouseStuff/chainMergeSort` |
| `chainSplit` | `kent/src/hg/mouseStuff/chainSplit` |
| `chainNet` | `kent/src/hg/mouseStuff/chainNet` |
| `netChainSubset` | `kent/src/hg/mouseStuff/netChainSubset` |
| `liftOver` | `kent/src/hg/liftOver` |

**Table A.1** *Digest displaying the paths to their associated utilities located in Jim Kent's source. The working directory has to be changed to the specified path when compiling one of the utilities.*

## A.2 *S. cerevisiae* versus *S. arboricolus* variants

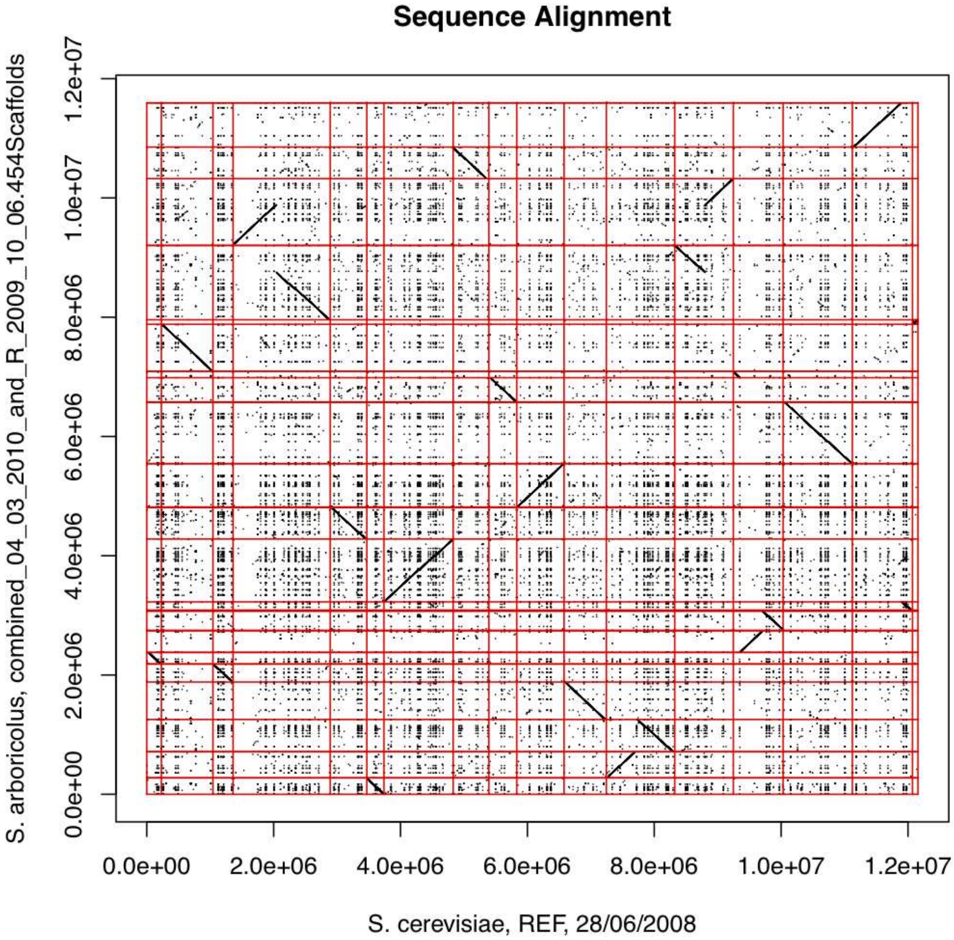


**Figure A.1** *Dot-matrix plot showing the chromosomes 12, 14, and 16 in S. cerevisiae, each broken into two in S. arboricolus.*

*The genome was deemed a flawed assembly.*

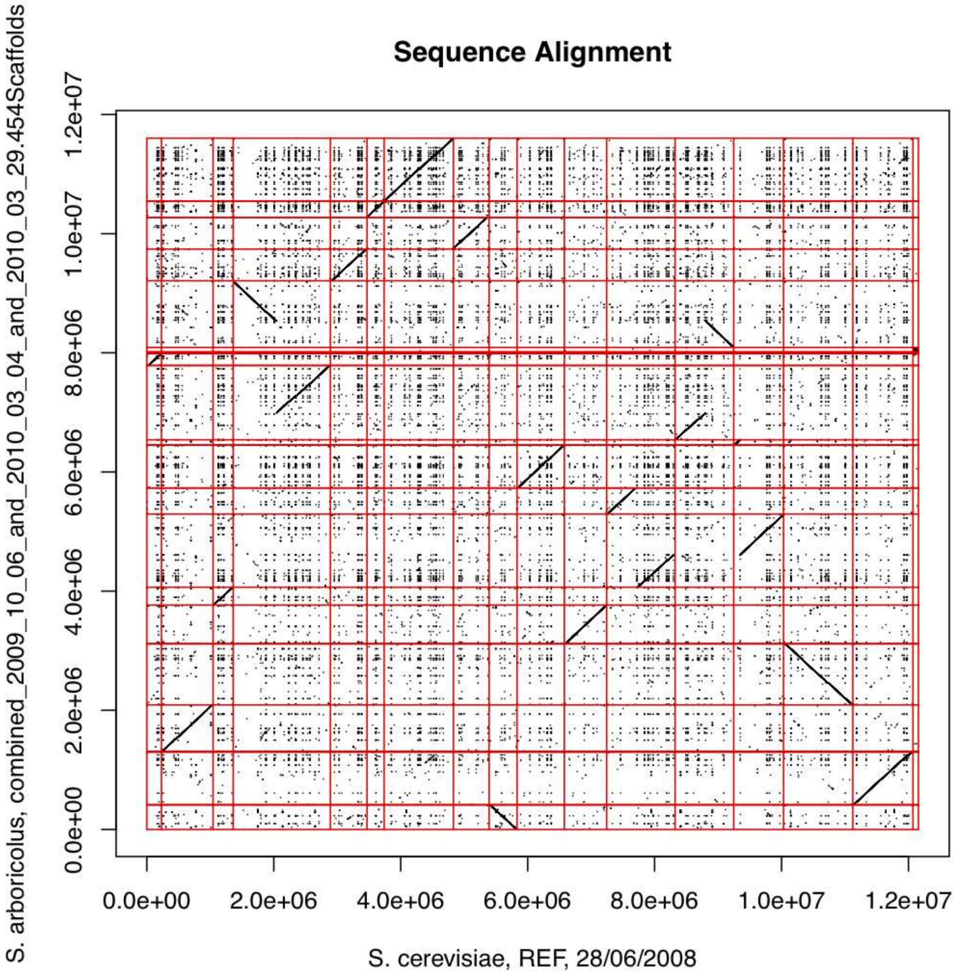


**Figure A.2** *Dot-matrix plot showing the fusion of chromosomes 12 and 14 of* S. cerevisiae *into one large chromosome in S. arboricolus. Also, both the chromosomes 12 and 14 in S. cerevisiae are broken into two in S. arboricolus. The genome was deemed a flawed assembly.*

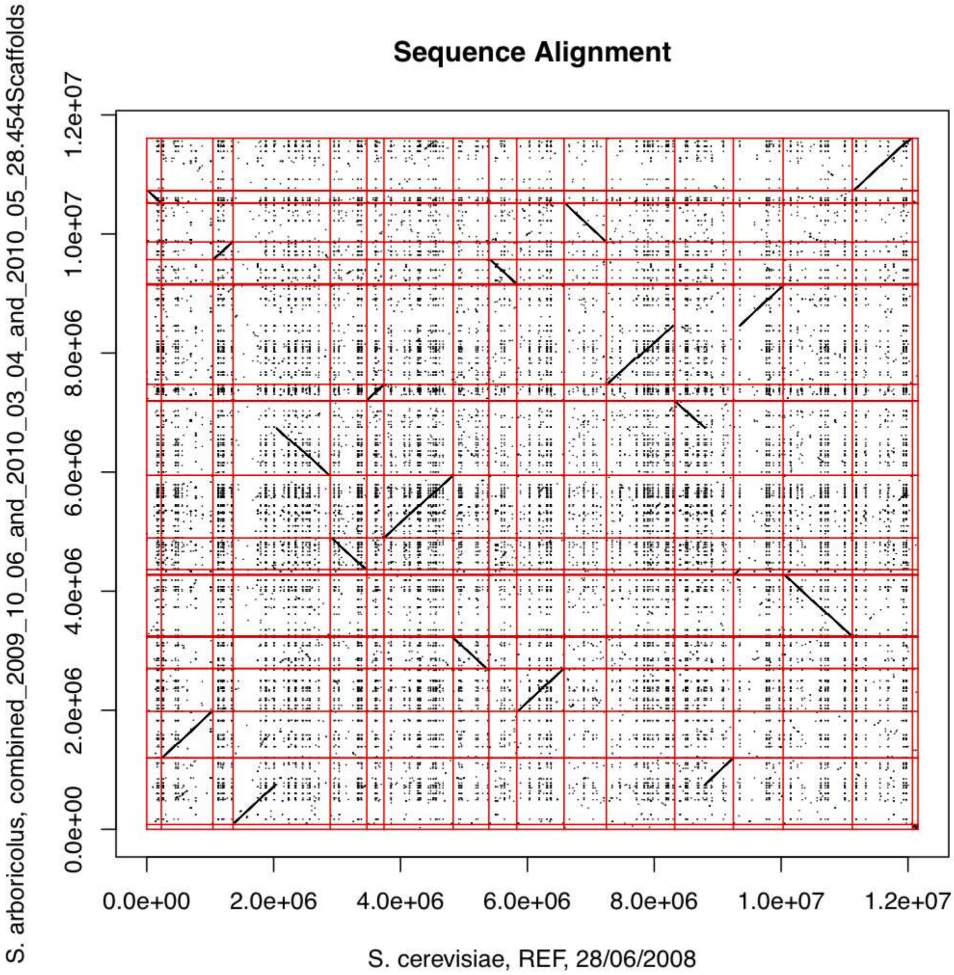


**Figure A.3** *Dot-matrix plot showing the fusion of chromosomes 12 and 14 of* S. cerevisiae *into one large chromosome in S. arboricolus. Also, chromosome 14 in S. cerevisiae is broken into two in S. arboricolus.*

*The genome was deemed a flawed assembly.*

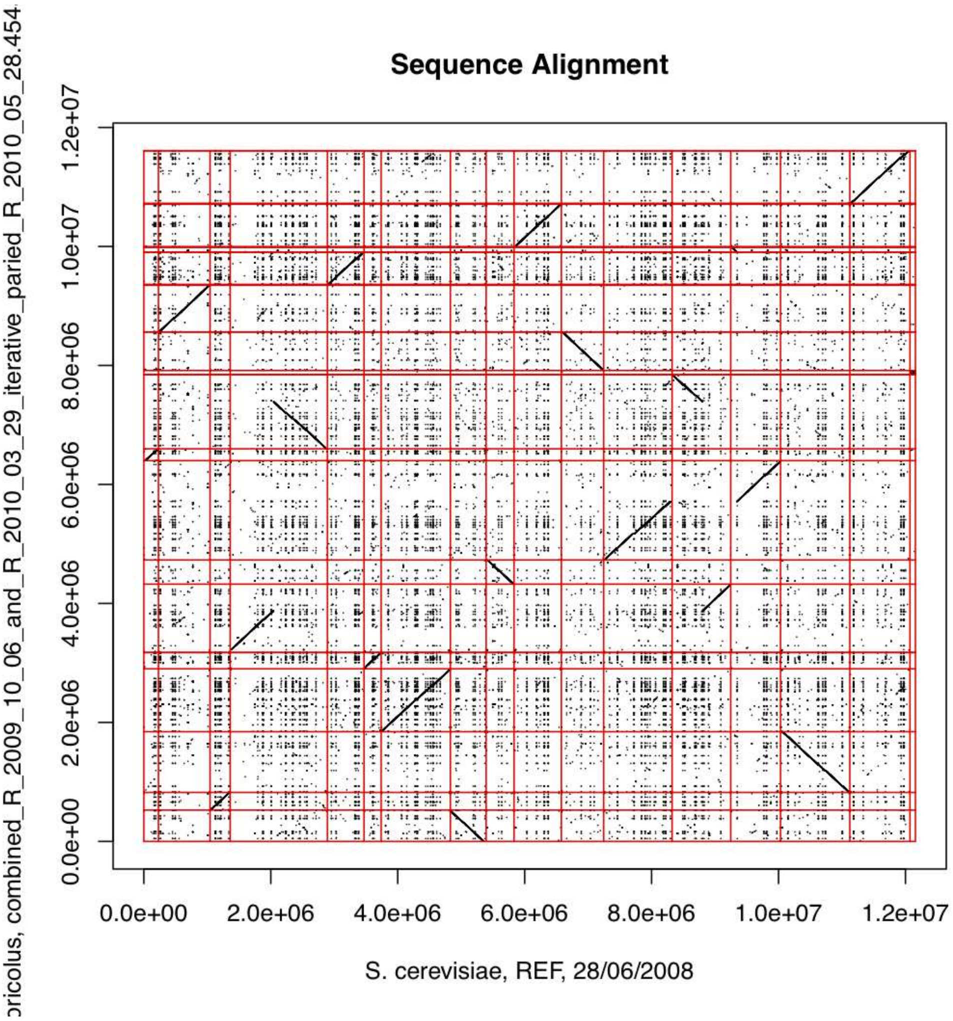


**Figure A.4** *Dot-matrix plot showing the fusion of chromosomes 12 and 14 of* S. cerevisiae *into one large chromosome in S. arboricolus. Also, chromosome 14 in S. cerevisiae is broken into two in S. arboricolus.*

*The genome was deemed a flawed assembly. Apparently this assembly is the same as the 23 July 2010 assembly of S. arboricolus.*

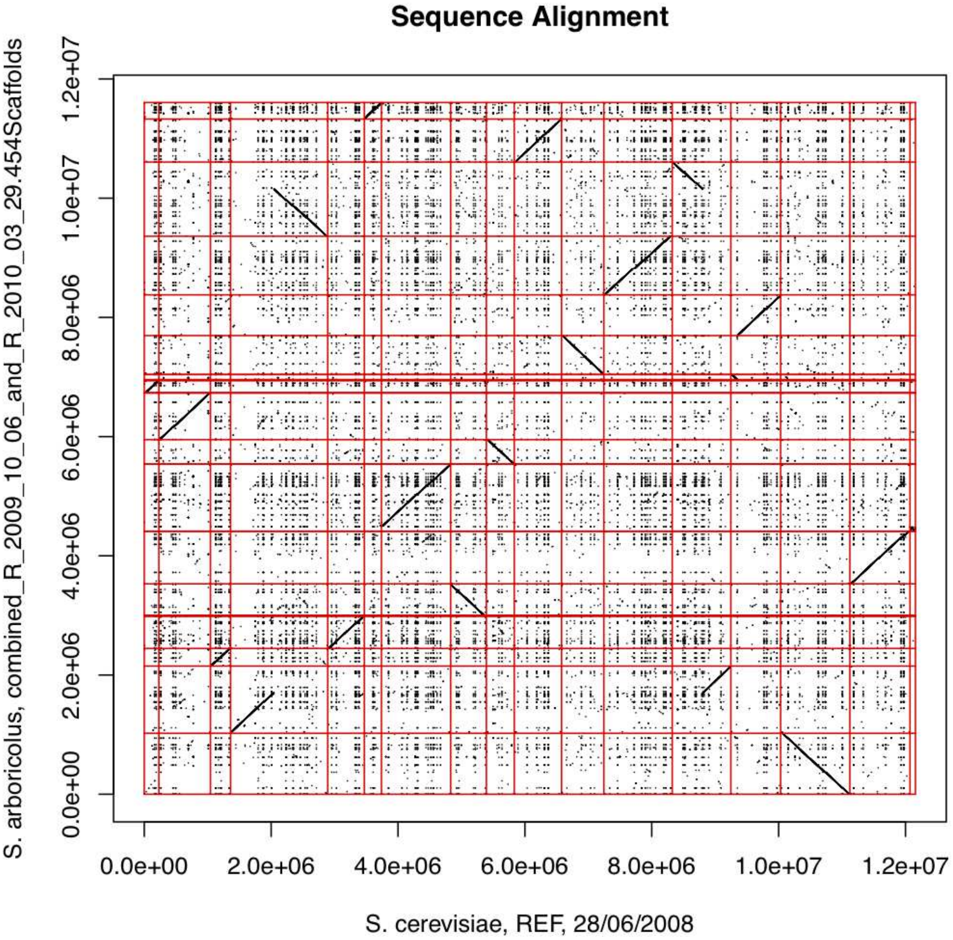


**Figure A.5** *Dot-matrix plot showing the chromosome 14 in S. cerevisiae is broken into two in S. arboricolus.*

*The genome was deemed a flawed assembly.*

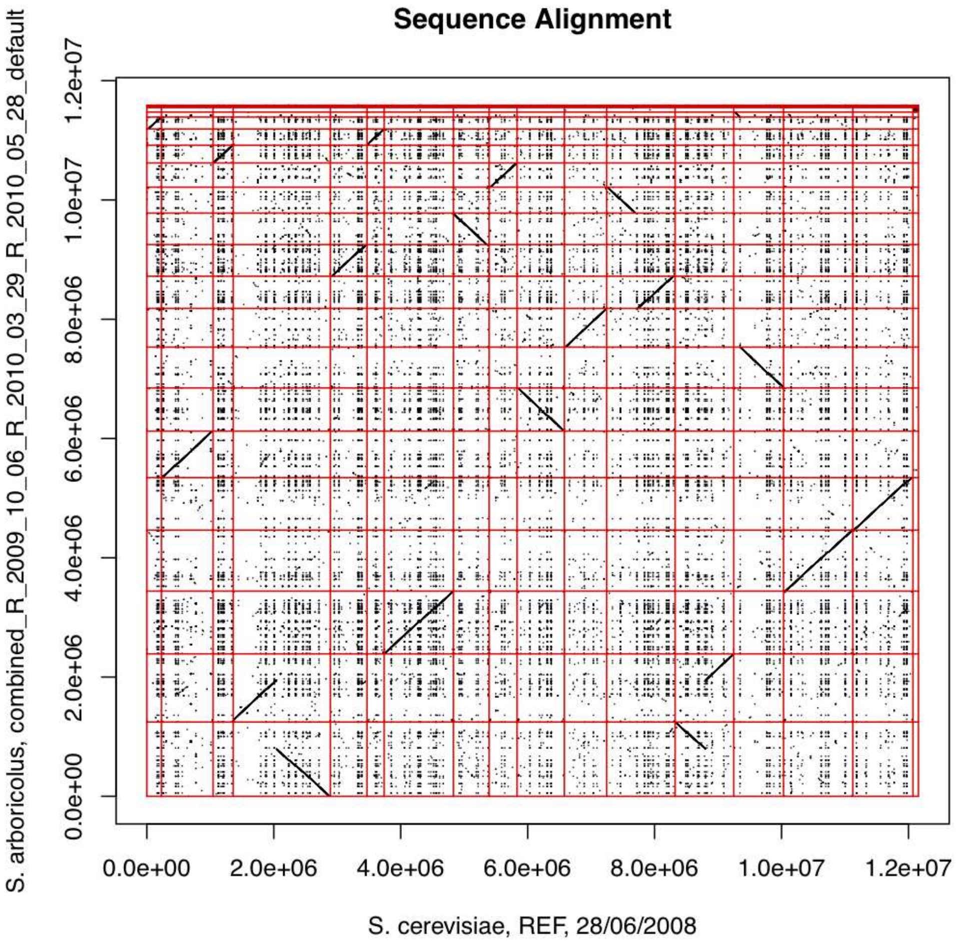


**Figure A.6** *Dot-matrix plot showing the chromosome 12 in S. cerevisiae is broken into two in S. arboricolus.*

*The genome was deemed a flawed assembly.*

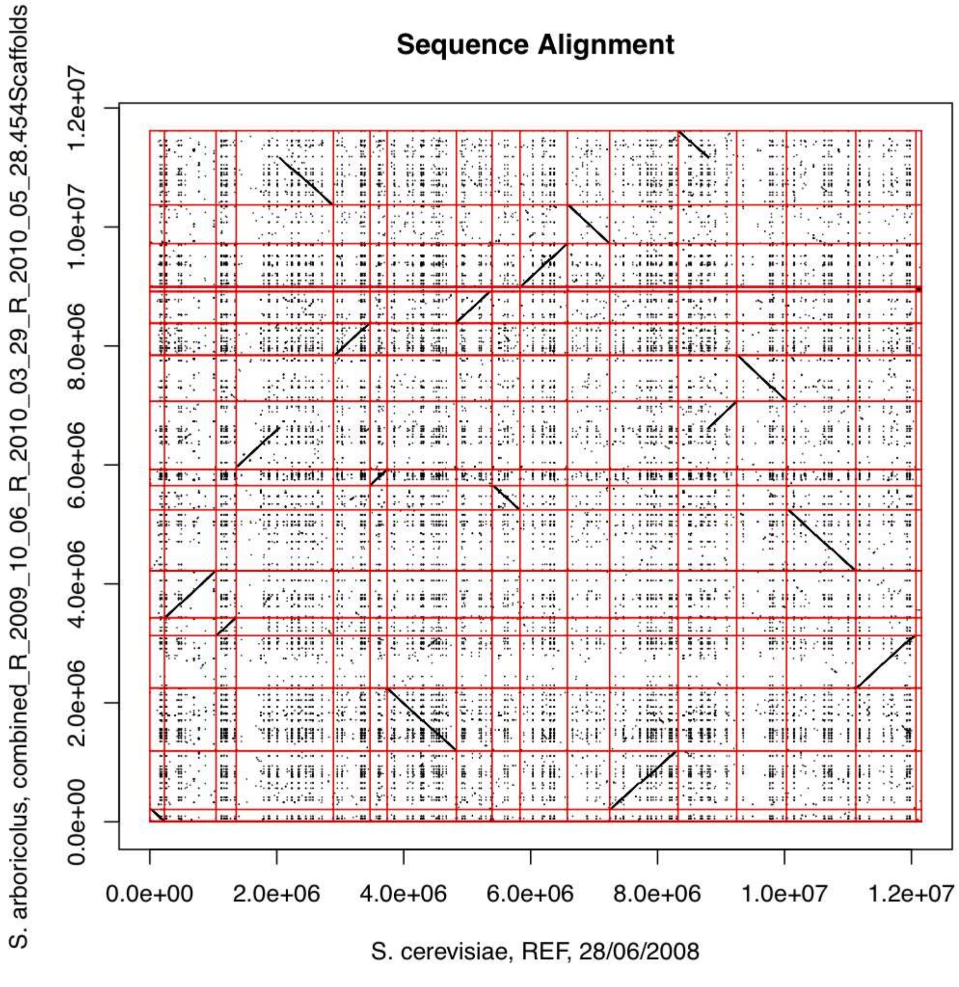


**Figure A.7** *Dot-matrix plot showing no signs of fusion or broken chromosomes. The genome was deemed a flawless assembly, thus was set as an update assembly in 23 March 2011.*

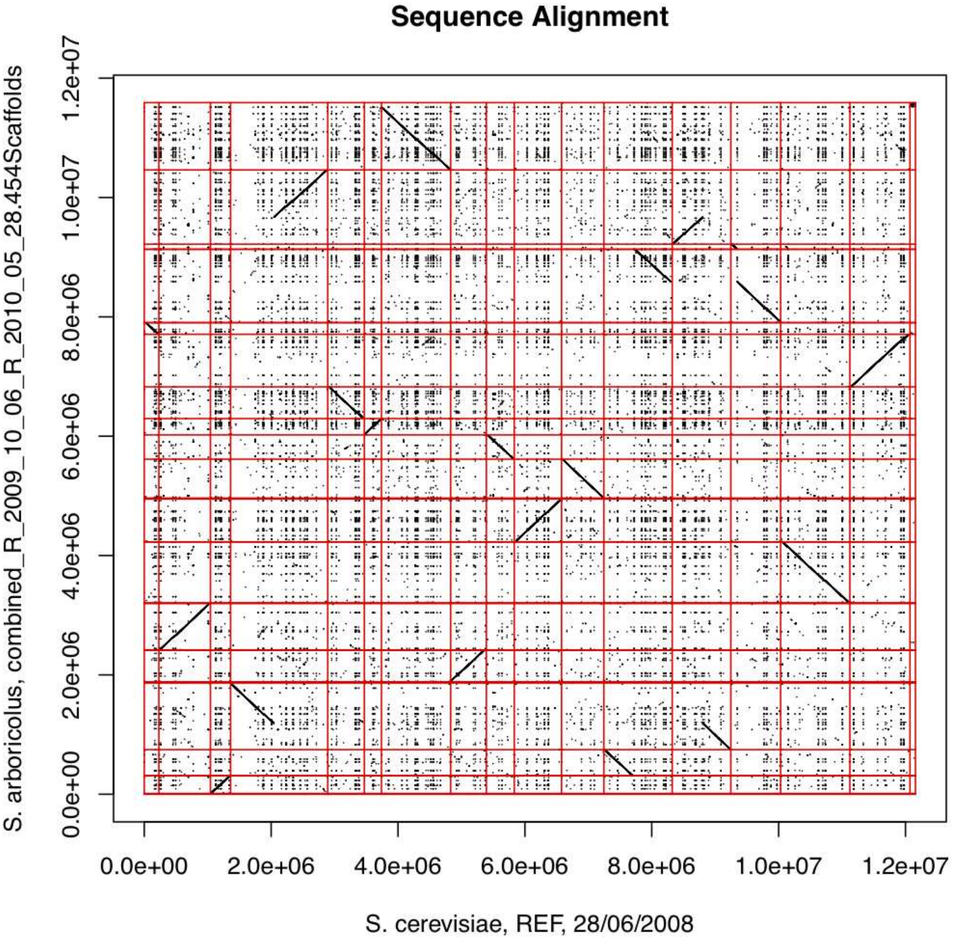


**Figure A.8** *Dot-matrix plot showing the chromosomes 12 and 14 in S. cerevisiae are broken into two in S. arboricolus.*

*The genome was deemed a flawed assembly.*

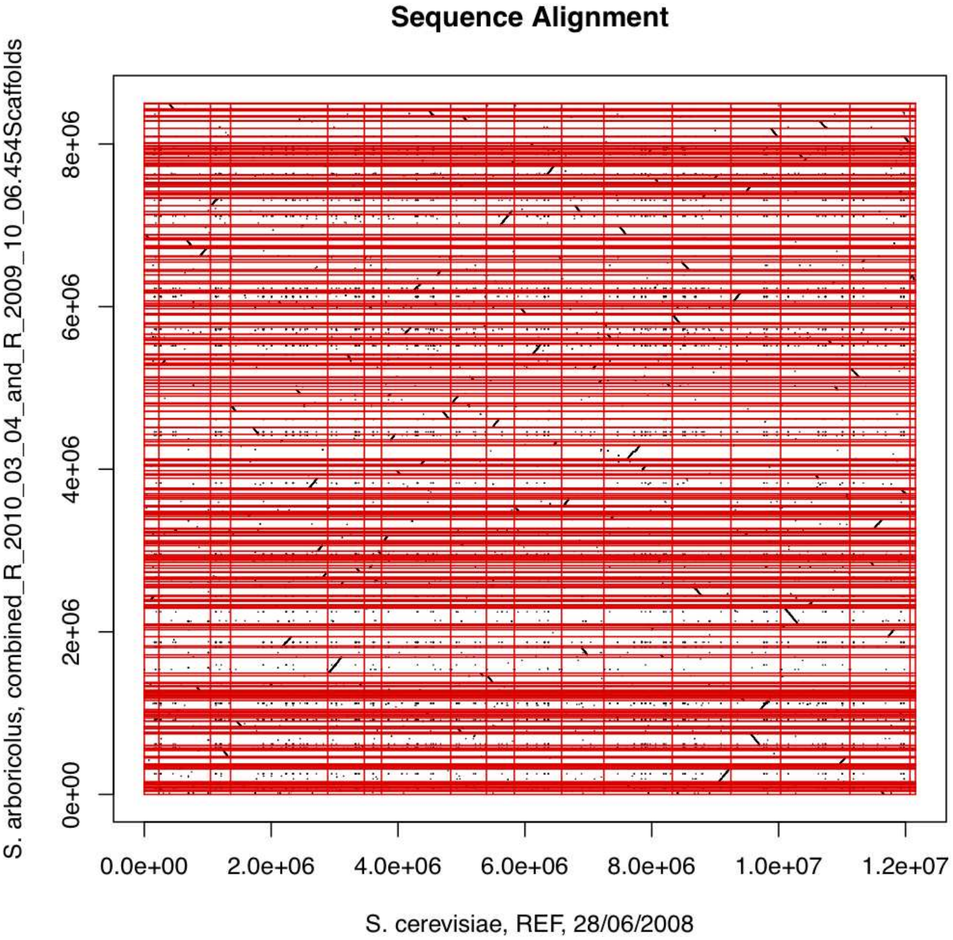


**Figure A.9** *Dot-matrix plot showing massive amount of chromosome fragments in S. arboricolus. Any of the fragments are not large enough to represent valid chromosomes.*

*The genome was deemed a flawed assembly.*

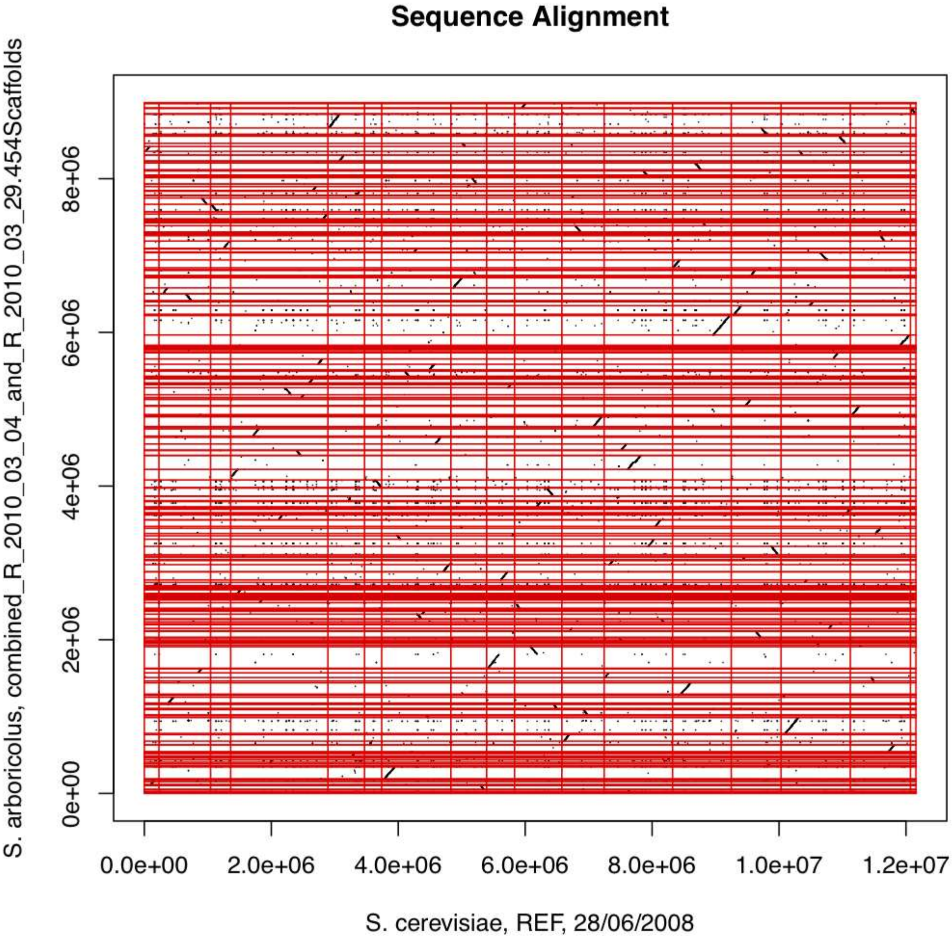


**Figure A.10** *Dot-matrix plot showing massive amount of chromosome fragments in S. arboricolus. Any of the fragments are not large enough to represent valid chromosomes.*

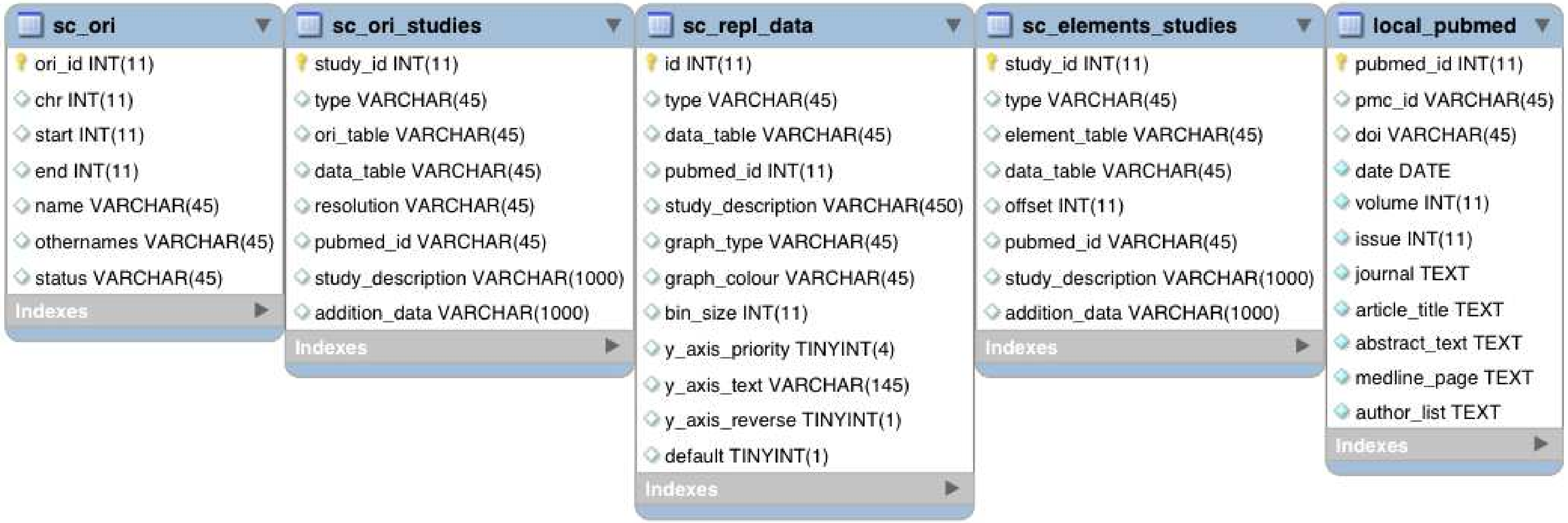
*The genome was deemed a flawed assembly.*

APPENDIX B

# SUPPLEMENTARY DATA - ORIDB

## B.1 Entities and Their Attributes

The following figures show groups of *S. cerevisiae* entities and their attributes. The implementation of *S. pombe* is still in the process and is not included here.



sc_ori
- ori_id INT(11)
- chr INT(11)
- start INT(11)
- end INT(11)
- name VARCHAR(45)
- othernames VARCHAR(45)
- status VARCHAR(45)
- Indexes

sc_ori_studies
- study_id INT(11)
- type VARCHAR(45)
- ori_table VARCHAR(45)
- data_table VARCHAR(45)
- resolution VARCHAR(45)
- pubmed_id VARCHAR(45)
- study_description VARCHAR(1000)
- addition_data VARCHAR(1000)
- Indexes

sc_repl_data
- id INT(11)
- type VARCHAR(45)
- data_table VARCHAR(45)
- pubmed_id INT(11)
- study_description VARCHAR(450)
- graph_type VARCHAR(45)
- graph_colour VARCHAR(45)
- bin_size INT(11)
- y_axis_priority TINYINT(4)
- y_axis_text VARCHAR(145)
- y_axis_reverse TINYINT(1)
- default TINYINT(1)
- Indexes

sc_elements_studies
- study_id INT(11)
- type VARCHAR(45)
- element_table VARCHAR(45)
- data_table VARCHAR(45)
- offset INT(11)
- pubmed_id VARCHAR(45)
- study_description VARCHAR(1000)
- addition_data VARCHAR(1000)
- Indexes

local_pubmed
- pubmed_id INT(11)
- pmc_id VARCHAR(45)
- doi VARCHAR(45)
- date DATE
- volume INT(11)
- issue INT(11)
- journal TEXT
- article_title TEXT
- abstract_text TEXT
- medline_page TEXT
- author_list TEXT
- Indexes

**Figure B.1** *Main entities of OriDB. sc_ori_studies, sc_repl_data, and sc_elements_studies hold data which links to the corresponding studies. local_pubmed holds data for the references whereas sc_ori holds the origins data of S. cerevisiae.*

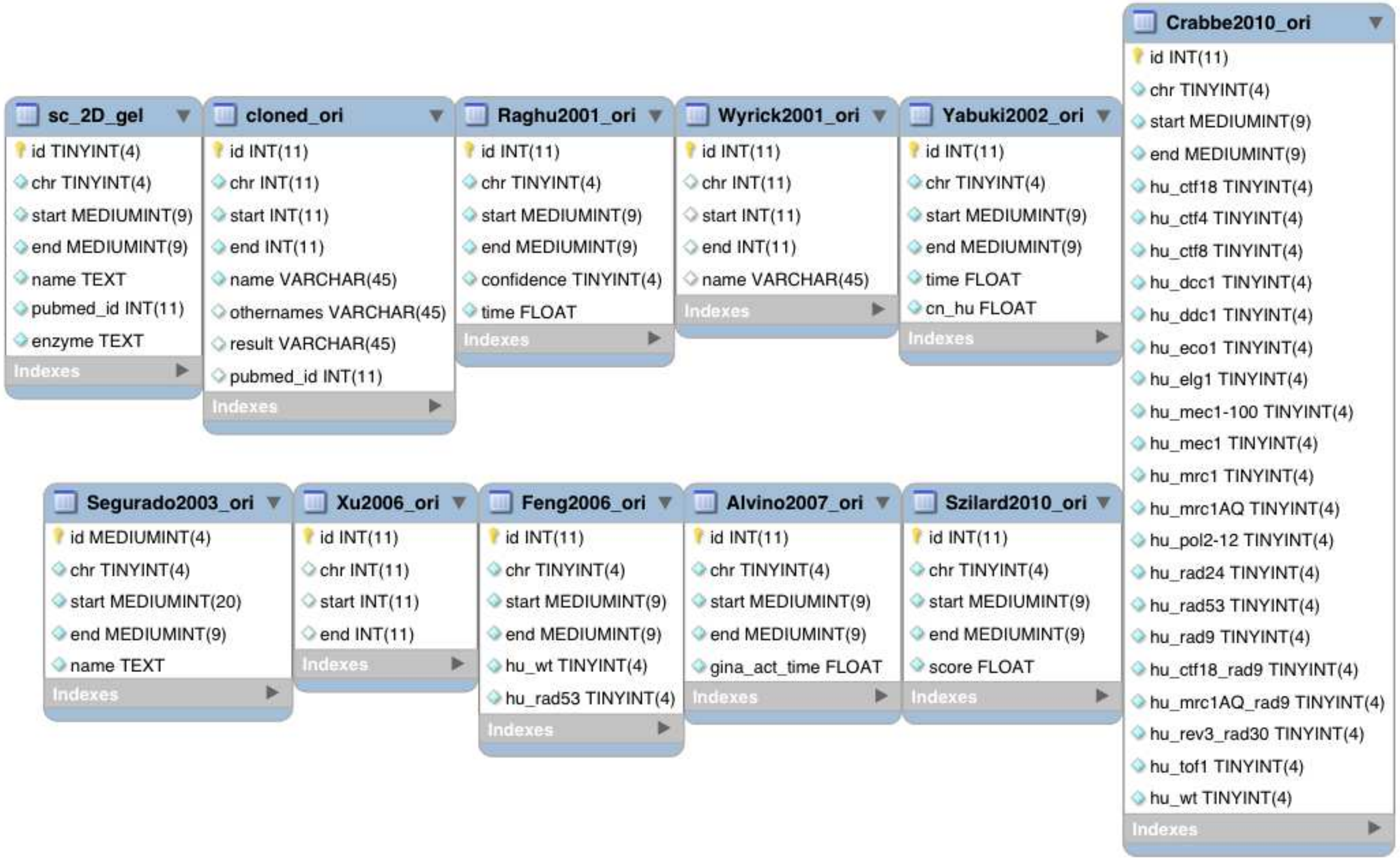
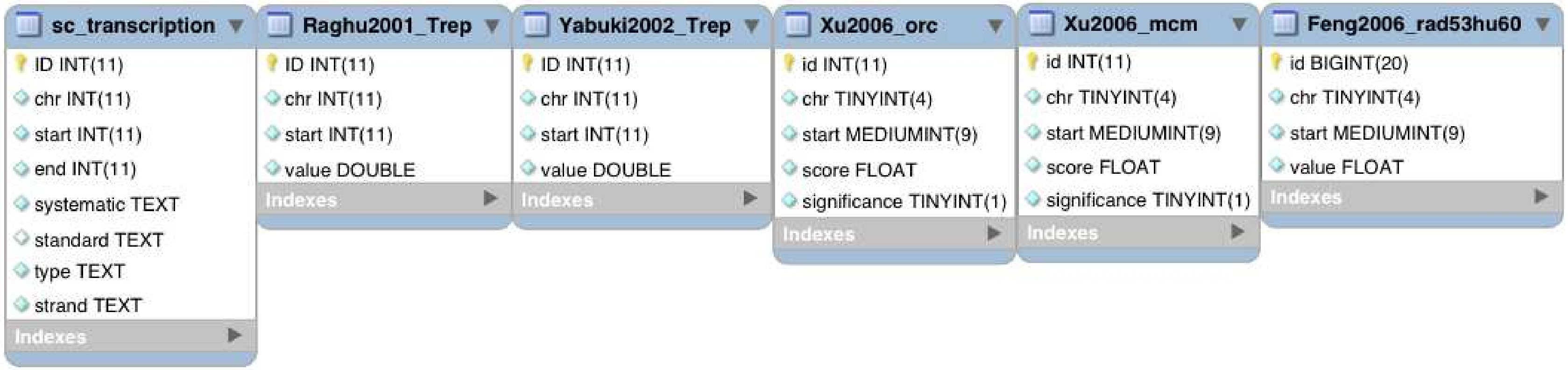


**Figure B.2** *Entities linked from sc_ori_studies.*



sc_transcription
- ID INT(11)
- chr INT(11)
- start INT(11)
- end INT(11)
- systematic TEXT
- standard TEXT
- type TEXT
- strand TEXT

Raghu2001_Trep
- ID INT(11)
- chr INT(11)
- start INT(11)
- value DOUBLE

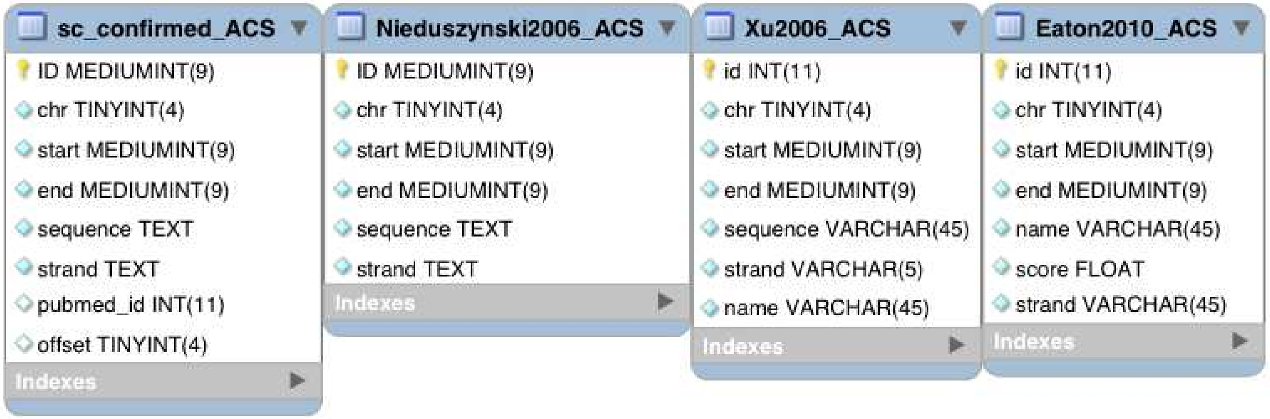
Yabuki2002_Trep
- ID INT(11)
- chr INT(11)
- start INT(11)
- value DOUBLE

Xu2006_orc
- id INT(11)
- chr TINYINT(4)
- start MEDIUMINT(9)
- score FLOAT
- significance TINYINT(1)

Xu2006_mcm
- id INT(11)
- chr TINYINT(4)
- start MEDIUMINT(9)
- score FLOAT
- significance TINYINT(1)

Feng2006_rad53hu60
- id BIGINT(20)
- chr TINYINT(4)
- start MEDIUMINT(9)
- value FLOAT

**Figure B.3** *Entities linked from sc_repl_data.*



sc_confirmed_ACS
- ID MEDIUMINT(9)
- chr TINYINT(4)
- start MEDIUMINT(9)
- end MEDIUMINT(9)
- sequence TEXT
- strand TEXT
- pubmed_id INT(11)
- offset TINYINT(4)

Indexes

Nieduszynski2006_ACS
- ID MEDIUMINT(9)
- chr TINYINT(4)
- start MEDIUMINT(9)
- end MEDIUMINT(9)
- sequence TEXT
- strand TEXT

Indexes

Xu2006_ACS
- id INT(11)
- chr TINYINT(4)
- start MEDIUMINT(9)
- end MEDIUMINT(9)
- sequence VARCHAR(45)
- strand VARCHAR(5)
- name VARCHAR(45)

Indexes

Eaton2010_ACS
- id INT(11)
- chr TINYINT(4)
- start MEDIUMINT(9)
- end MEDIUMINT(9)
- name VARCHAR(45)
- score FLOAT
- strand VARCHAR(45)

Indexes

**Figure B.4** *Entities linked from sc_elements_studies.*

The End