Chapter 1. Introduction

Large scale analysis of genomes remains a very new field of study. The publication at the end of 1998 of the 102 megabase *C. elegans* genome provided our first glimpse at a near-complete sequence for a multicellular organism [*C. elegans* Sequencing Consortium 1998]. This was followed in 2001 and 2002 by draft sequences for the human [IHGSC 2001] and mouse [MGSC 2002] genomes respectively. Both of these are – at least in terms of number of nucleotides – around 30 times larger than *C. elegans*. Since then, the rate of genome sequencing has continued to accelerate and many more sequences have been published, including a number of additional vertebrates. This genomic revolution has promised great developments, both in terms of pure science and in practical developments in the fields of medicine and commercial biotechnology. However, unlocking this potential requires additional experimental work, and also high-throughput analysis and data-mining methods.

The function of the genetic material in the cell is part of the field known as molecular biology. The core processes of molecular biology can be seen as a pipeline of information flowing from the genomic DNA molecules, via a pool of RNA messengers (the transcriptome) to the set of proteins which mediate most of the cell's biochemical functions (the proteome). This has become known as the central dogma of molecular biology, and a schematic of the information flow is shown in figure 1.1. This model is something of a simplification, since while the primary role of RNA is to provide an intermediate stage in information flow between DNA and protein, many RNA molecules perform important functions – catalytic, structural or regulatory – in their own right [Eddy 2001].

An alternative, sequence-centric, view of this process is given in figure 1.2. A primary transcript is produced from the region between the transcription start and termination sites,

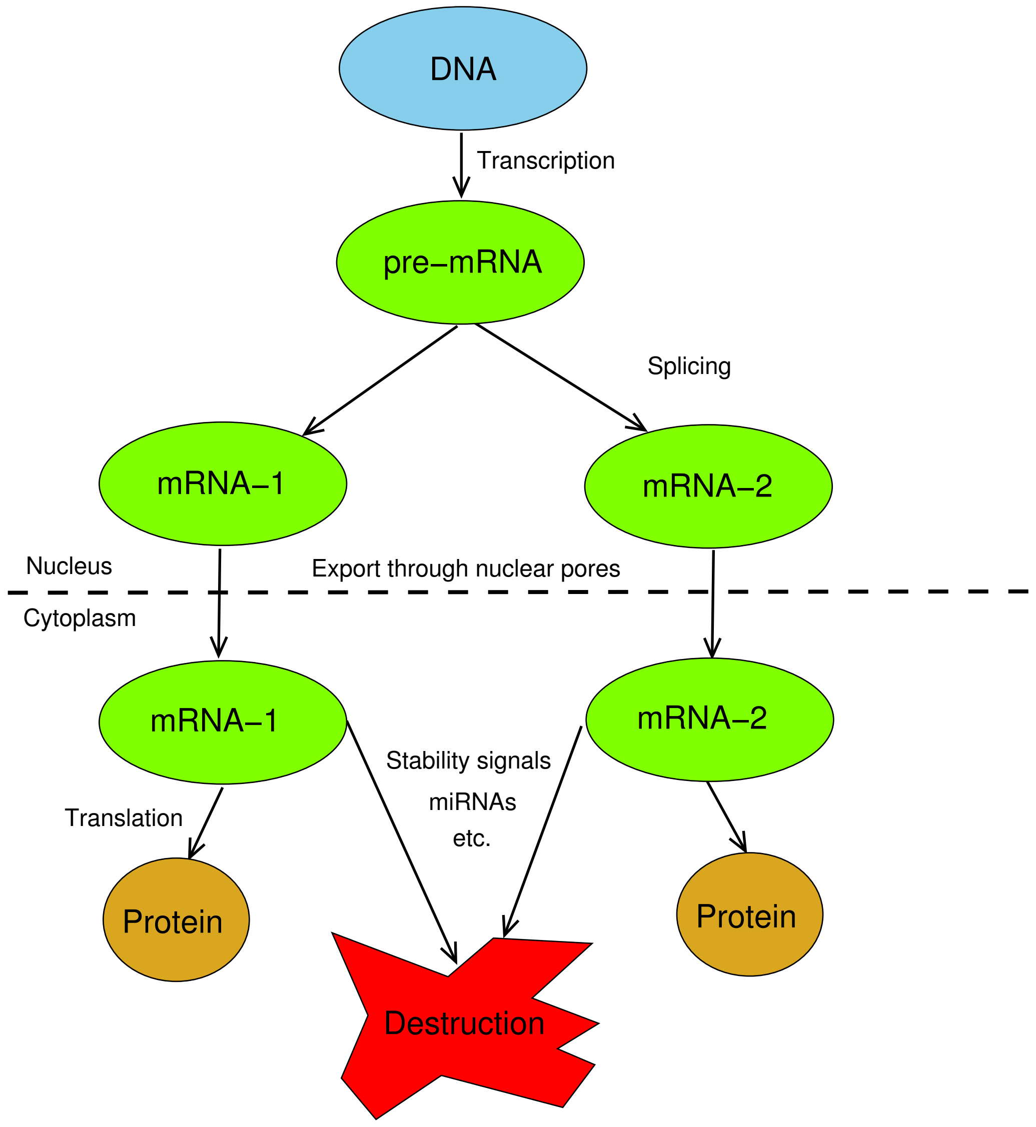

Figure 1.1. Flow of biological information from the genome to the proteome. In this case, the pre-mRNA can be spliced in two possible ways, leading to different protein products.

then introns are spliced out. Note that this diagram is somewhat schematic: in real mammalian genes, introns are usually much longer than exons, and most genes have a number of introns. In many cases, different sets of introns can be spliced to give alternative products from the same primary transcript [Ladd and Cooper, 2002]. As well as interrupting the coding regions, introns often occur in the 5' untranslated region, but only very rarely in the 3' UTR [Pesole *et al.* 2001]. Finally, at least in some cases, a single gene has more than one possible transcription start site (for example, [Laurinn *et al* 2000]), although it is not certain just what fraction of genes possess this extra dimension of complexity.

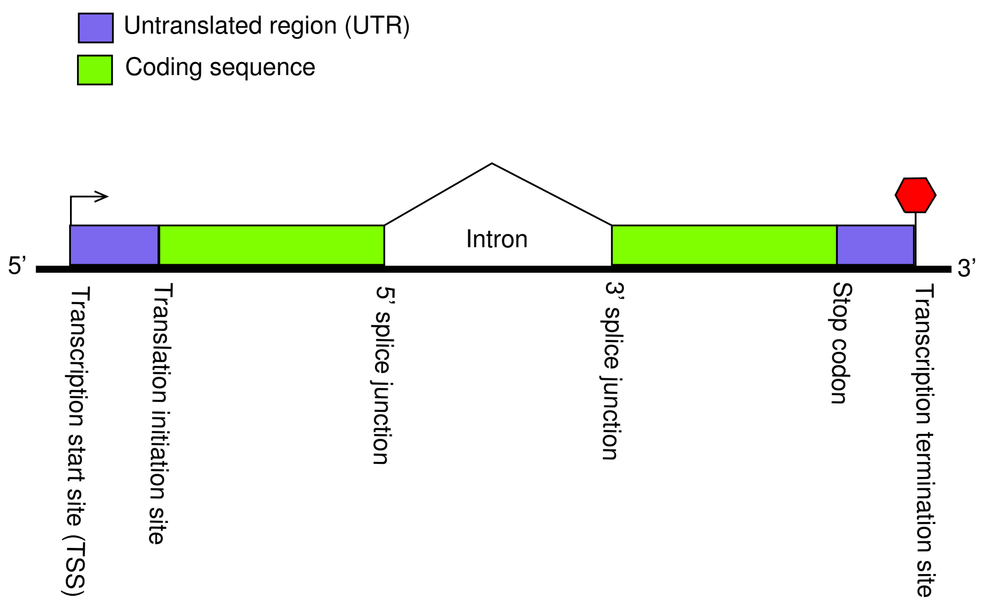
Untranslated region (UTR)
Coding sequence
Intron
5’
3’
Transcription start site (TSS)
Translation initiation site
5’ splice junction
3’ splice junction
Stop codon
Transcription termination site

Figure 1.2. Illustrative diagram showing the basic structure of a eukaryotic protein-coding gene.

In bacteria, the basic flow of information is similar, but the details are a little simpler: since there is no distinct nucleus, there is no specific export step. Bacterial messenger RNAs generally have no introns, removing the need for the splicing step, but eliminating the possibility for multiple splice variants from the same primary transcript. The rest of this thesis, however, concentrates exclusively on the molecular biology of eukaryotic cells, and in particular on the recently sequenced mammalian genomes.

Having identified the basic form of the molecular biology pipeline, the challenge to biologists today is to gain a full understanding of how pure, digital, information encoded as a string over the four-letter DNA alphabet leads can define the biochemistry (and larger-scale properties) of a living organisms. Clearly, this knowledge will play a major part in understanding the action of genetic diseases, particularly the polygenic diseases which cannot be traced back to a single, obvious, defect in a single gene [Wright *et al.* 2003]. As our understanding of molecular biology grows, it also seems likely that an increasing range of manipulations and interventions will be possible. For instance, detailed understanding of the pathways which lead cells in higher organisms to form tissues and organs could ultimately lead to *in vitro* production of replacement organs [Risbud 2001]. A good test of the state of biological knowledge would be to build a computational model the processes of life in sufficient detail that it is possible to predict the effects of any perturbation in the genome.

To date, most analysis of the genome sequences has focused on the protein-coding regions – those portions of the sequence which are expressed as messenger RNAs and act directly as templates for protein synthesis. At a basic level, at least, these pose rather easy targets for study: there is a simple, well-defined code which relates groups of three nucleotide "symbols" in the DNA genome (or the RNA messenger) to one amino acid "symbol" in the protein product. This is often called the universal genetic code, since it is conserved with only a few minor variations throughout all known branches of terrestrial life. Detecting coding sequences is a fairly well-understood problem. Typical methods combine a model of coding sequence (often expressed as a table of hexamer frequencies) with additional logic to detect likely splice sites [reviewed in Mathé *et al.* 2002]. A rather different approach, which emphasizes just how distinctive coding genes can be from the genomic background, is described in [Pocock 2001]. Here, an unsupervised machine learning approach is used to partition the genome of *Plasmodium falciparum* (a malarial parasite) into several classes, with no *a priori* definition of what these classes should actually represent. This approach consistently identified the parasite's coding regions as a single, distinct, class.

Detecting vertebrate genes – which tend to be large entities with many big introns separating relatively small (often less than 100 nucleotides) fragments of actual coding sequence – is arguably the greatest challenge in gene prediction. However, a combination of purely computational methods such as the widely used Genscan algorithm [Burge and Karlin, 1997] with experimental evidence such as Expressed Sequence Tag (EST) sequences [Adams *et al.* 1991] can give good quality gene annotation. For the human genome, a manual curation process is currently underway to create a definitive annotation set, but in the mean time there are also fully-automated methods, notably the Ensembl [Hubbard *et al.* 2002] annotation pipeline, which uses a range of experimental data (protein, cDNA, and EST sequences) plus some computational input to provide a good quality first-pass annotation set. While Ensembl was created primarily to analyze human sequences, the annotation pipeline is now routinely applied to a number of other metazoans.

Before the sequencing of the human genome, there was a great deal of speculation about the number of human genes – especially once the *C. elegans* gene was published, confirming that this simple nematode worm has around 19,000 genes. Estimates from 30,000 to 120,000 human genes have been widely circulated, with many commentators favouring the higher end of this range (see, for example, [Fields *et al.* 1994]). A final figure will have to wait for more comprehensive curated annotation of the genome, but at the time of writing, the current Ensembl human genome release (version 13.31) contains predictions for 24,847 protein-coding genes. Ensembl annotation methods are, by design, relatively conservative so this is probably an underestimate, but many researchers today expect a final count of around 30,000 protein-coding genes. Relatively speaking, this is only a small increase compared to the worm. This number does not seem to reflect the apparent difference in complexity between the human body (with around 10^{13} cells and a complex nervous system) and the worm (with a fixed developmental pathway culminating in an organism with just over 10^{3} cells). The conclusion here has to be that while the set of building blocks for a human may not be substantially larger than that for some much lower organisms, the networks of regulatory molecules which process information during the developmental process and mark out the parts of the body are substantially more complex. To understand biology, and especially developmental processes, it will be important to look at complete networks as well as their individual building blocks.

All the processes shown in figure 1.1 are regulated to a greater or lesser extent, either by proteins, small molecules (usually interacting with DNA via a protein), or by regulatory RNA molecules such as the micro-RNAs (miRNAs) [Lewin 2000, Grosshans and Slack, 2002]. So the interplay of the cell's current population of proteins and RNA influences the population at some point in the future, providing the basis for regulatory networks. The basis for most of the known regulated processes in molecular biology – transcription, alternate splicing, and RNA stability – involves interactions between the regulator model and specific regulatory regions of a nucleic acid molecule, which might be either the genomic DNA or an RNA transcript. In either case, sequences which function as regulatory targets share the genome with (and in some cases, such as exonic splice enhancers, actually overlap) the protein-coding regions [Blencowe 2000]. Therefore, they can be seen as a second genetic code which conveys information about when, where, and how much of a protein should be produced. Today, our techniques of genome analysis make it possible, given some raw sequence data, to identify protein-coding genes and to predict the sequence of the protein they code for. This can then be compared with known protein sequences to give at least some indication of the protein's likely structure and function. An important step towards complete understanding of the genome will be to perform an analogous process on the regulatory regions: first to identify these regions in bulk genomic sequence, and then to "read" the regulatory codes, gaining some information about the set of regulator molecules which interact with a particular region, and thus where it fits into the cell's regulatory network.

1.1. Objectives of this project

An important direction for computational biology over the coming years is to develop methods for identifying and decoding regulatory regions in genomic sequences – especially large and complex genomes, such as those of vertebrates. The order here is significant: it is necessary to identify the regions of interest before detailed study and decoding are possible. For this reason, in this project I concentrated primarily on this first identification step. I also decided to focus on one particular class of regulatory signals: the promoters, which are located near (primarily upstream of) the transcription start site and regulate transcription initiation. While these are certainly not the only significant type of regulatory region in the genome, they are clearly vitally important since every gene has one. This property also means that the set of examples available for study is large, and that it is reasonably to assume that every point in the genome which can be shown to function as a transcription start site (TSS) must have some kind of associated promoter region.

Having chosen this objective, chapter 3 of this thesis describes a novel method for predicting transcription start sites: the key points around which promoters are organized. A second predictive method is described in chapter 4. Chapter 5 discusses how sequence from a second species – in this case, mouse – can be used to highlight regions of extremely high evolutionary conservation which are believed to indicate the cores of the functional promoter regions.

I also hoped that, in achieving my primary objective, I would develop technologies that will prove useful in the coming years for dissecting regulatory regions, identifying the sequences which make these regions targets for particular regulatory pathways, and therefore providing the knowledge to start actively decoding novel regulatory regions. Chapter 2 describes a recent, highly versatile, approach to machine learning which can be shown, on the basis of results in chapters 3 and 4, to be effective for sequence analysis problems. This approach may well be applicable to further analysis of regulatory regions.

The remainder of this chapter gives a brief background on the current state of knowledge about promoter regions, and discusses some tools and resources which were helpful for studying them.

1.2. What is a promoter

Briefly, a promoter is a region of DNA sequence close to a gene's transcription start site, at which molecular events occur leading to the initiation of transcription. They are sometimes described as *cis*-regulatory regions, as opposed to *trans*-regulation which implies regulators situated some distance from the gene itself. Needless to say, few if any promoter regions, even in the simplest organisms, blindly trigger the production of new transcripts. Even single cells will have many genes which must be turned on and off under specific circumstances, for example enzymes forming the pathways for metabolizing various different energy-source compounds. And no protein is required in an unlimited amount, so feedback mechanisms which turn off transcription once an adequate concentration has been achieved will make the cell function more efficiently Multicellular organisms require proportionately more complex regulation – at a minimum, each cell type will have its own pattern of gene expression, and there is usually also extra complexity in the form of signaling mechanisms exchanging information between cells and tissues. A number of enhancer sequences, which control the rate of transcription from locations some distance from the transcription start site (and thus represent a form of *trans*-regulation) have also been identified. While these are clearly an important part of the transcriptional regulation machinery, their location further from the transcription start site and the fact that many genes may not have any associated enhancer region at all mean that they are currently rather difficult to study, and they are not considered further in this thesis.

Promoters have been studied in the laboratory for many years, first in bacteria [Pardee *et al.* 1959] and then in eukaryotic cells. A wide variety of experimental techniques have been used, including directly mapping the interactions of proteins to DNA [Galas and Schmitz, 1978], and attempting to carry out transcription *in vitro* [Hayashi 1965]. In some senses, promoter biology is advancing reasonably well. Given a gene, and some laboratory time, it is often possible to clone the promoter region then couple it to a reporter gene – typically either a fluorescent protein or an enzyme which produces a coloured product – to identify the expression pattern. It is then possible to delineate the actually boundaries of the promoter experimentally, for example by performing targeting deletions. A number of strong (in the sense of causing high levels of transcription) promoters with useful and well-characterized expression patterns are available “off the shelf”, and are used to drive the expression of transgenes for experimental purposes, or in practical biotechnology applications. This is especially relevant in plants, where a small number of promoters, such as the Cauliflower Mosaic Virus promoter, are used for many purposes. However, for this kind of application it is usually sufficient to treat the promoter as a black box, assuming that if it gives a particular expression level and pattern for one gene, it will do the same for a second gene in the same (or even a related) organism. This does not directly improve our ability to decode arbitrary promoters, or to engineer new promoters from scratch to achieve a specific expression pattern. In addition, promoter cloning and delineation remains a fairly complex and labour-intensive process. And methods for determining the expression pattern, which are generally most effective when dealing with reasonably high levels of expression, may fail for genes which are only expressed at the level of a few copies per cell, or which are only active under a very small range of circumstances. It seems unlikely that the full set of promoters from any complex organism will be identified in this manner in the foreseeable future.

Some promoters have been studied in much more detail, to the level of understanding many of the interactions between DNA and protein. Simple model organisms, notably yeast, are the convenient and popular choice for this kind of research, although more complex model organisms, and even human cell lines, have been studied in some cases. To date, this kind of work has been our most important source of understanding in how promoters and the transcriptional machinery actually function. While it is unlikely that the full repertoire of promoters will be studies in depth by individual experiments, there is now an approach to mapping protein-DNA interactions, at least at a moderate resolution, which appears amenable to high-throughput application. Briefly, preparations of chromatin are fragmented, then the fragments associated with a protein of interest are precipitated using appropriate antibodies. DNA fragments extracted from the precipitate can then be mapped back to the genome by hybridization onto a DNA microarray [Ren *et al.* 2000]. The future significance of this technique, known as ChIP-on-chip, is discussed in the concluding chapter of this thesis.

Eukaryotic cells actually have three different forms of RNA polymerase, the enzyme responsible for transcribing RNA from DNA. Two of these, *polI* and *polIII* have only very specialized roles relating to transcription of certain noncoding RNA genes. All protein-coding RNAs, and many others, are transcribed by *polII* ,making this the most interesting (and best studied) of the three. Studying the complexes which the *polII* core forms with DNA, it is found to be associated with a wide range of proteins. The bulk of these are characterized as *transcription factors* (TFs) [Lewin 2000]. There proteins containing DNA-binding domains from one of several large families such as the zinc finger and leucine zipper families, and recognize particular elements in promoter regions. The structure of one transcription factor (PU.1, a form of helix-turn-helix factor), bound to a short synthetic DNA molecule, is shown in figure 1.3. This was drawn from the PDB entry 1pue [Kodandapani *et al.* 1996].

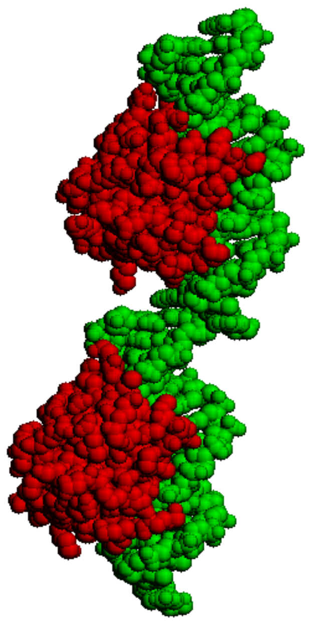

Figure 1.3. X-ray structure of transcription factor Pu.1 (shown in red) bound to a synthetic DNA substrate (green).

Transcription factors are very common: the GO annotation of Ensembl human gene predictions (see page 28) lists 1028 transcription factors, and since not every gene is annotated in this way, this is almost certain to be an underestimate. Also involved in *polII* complexes are the general transcription factors (GTFs) and adapters, which do not necessarily recognize specific DNA sequences, but are required parts of the functional transcription complex. A naive view of transcription initiation is shown in figure 1.4.

In fact, we know that figure 1.4 is a significant over-simplification in several respects. Genomic DNA in its natural context is certainly not a simple linear molecule, and the

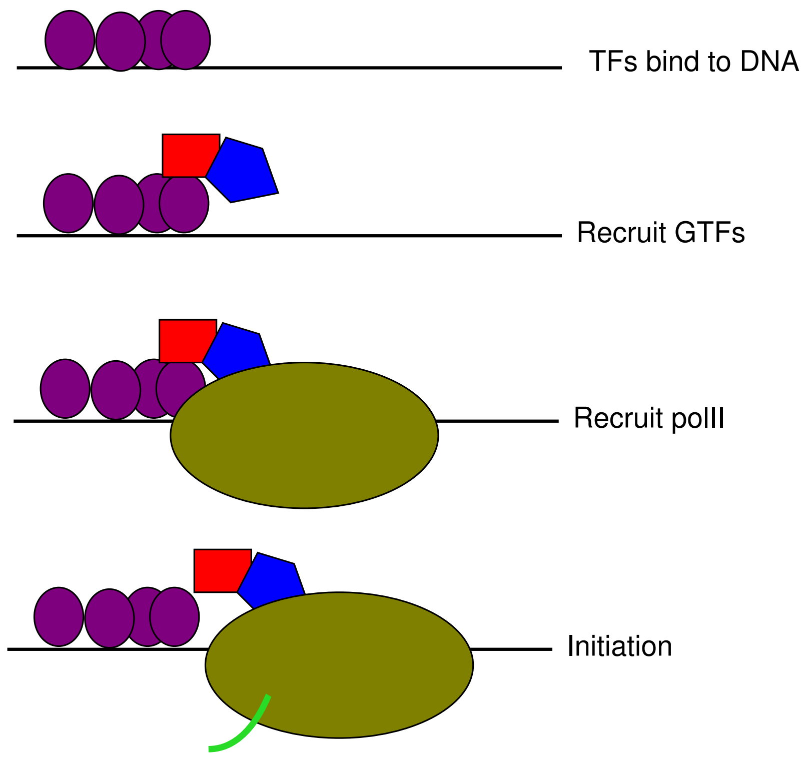

Figure 1.4. A (simplistic) overview of keys steps in transcription initiation at a *polII* promoter.

transcription factors discussed here are unlikely to have unimpeded access to their binding sites. The DNA double-helix is packaged at several levels. Firstly, units of around 150 nucleotides are coiled around octamers of the core histone proteins to form nucleosomes, which can be seen as "beads" on a string when viewed under the electron microscope. The structure of a nucleosome has been determined by X-ray crystalography [Luger *et al.* 1997] and is shown in figure 1.5. The DNA is then packaged further by weaker associations with histone H1 to form the chromatin fibre [Kornberg 1999]. While this system may have evolved primarily to protect the DNA and make it easier to move the chromosomes around during cell division, it must clearly impact on all other processes which involve interaction with the DNA. A number of systems have been identified which alter DNA packaging, including enzymes which add and remove acetyl groups from the histone cores, and remodeling complexes which appear to physically move nucleosomes along the DNA molecule. Many of these enzymes have been found in complexes with *polII* and transcription factors [Brownell *et al.* 1996]. One conclusion to draw from this is that it is generally wrong to think of transcription initiation as an isolated event.

When a transcription complex is recruited to a promoter, as well as triggering the production of a single RNA copy, it also makes more permanent changes to the chromatin around the promoter, facilitating initiation of subsequent transcripts [Kadonaga 1998].

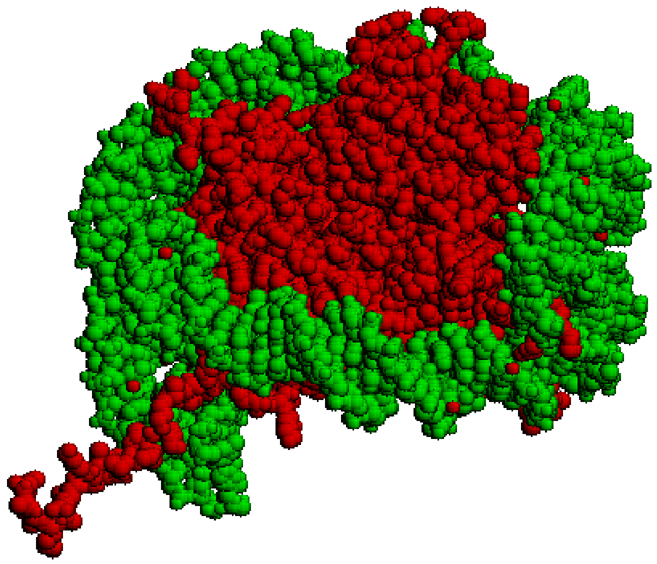

Figure 1.5. X-ray structure of DNA (shown in green) coiled around an octamer of core histone proteins (red).

Another issue with the model of figure 1.4 is that many of the components of the transcription complex appear to exist in complexes – called holoenzymes by analogy to complexes involved in bacterial transcription mechanism – even when not associated with DNA [Carey 1995]. Holoenzymes were first observed in budding yeast, but have since been found in other species as well [Myer and Young, 1998]. Instead of a model where proteins are recruiting one at a time to a transcription complex, it is now thought that large parts of the complex arrive in a single step [Barberis and Gaudreau, 1998]. Presumably there is still some initial binding of individual TFs to the DNA which starts the process and causes recruitment of the holoenzyme.

One final complication, especially in higher eukaryotes, is that covalent modifications can occur to the DNA nucleotide residues themselves, so in effect the alphabet of genomic DNA is actually larger than the four letters normally considered. In mammals, almost all occurrences of the dinucleotide 5'-cytosine-guanine-3' (usually written CpG) are modified to use methylcytosine in place of cytosine. Demethylation of CpG sites is associated with promoter activation [Razin 1998]. When transcription begins at a previously inactive promoter, demethylation seems to accompany chromatin remodeling. In addition to the permanent regulatory signals encoded in the DNA sequence, promoters have a state, and can be switched between inactive (methylated CpG, tightly packaged chromatin) and active (demethylated, with more "open" chromatin structure) states. Information stored as patterns of DNA or histone modifications, rather than directly in the DNA sequence, is often described as epigenetic information [Holliday 1987].

Nevertheless, the naive model of transcription initiation remains a useful view to bear in mind when working with promoters. It suggests that, at a sequence level, a promoter will consist of a cluster of discrete sites at which the transcription factors bind. In the most naive view, sequence between TF binding sites is irrelevant, but in practice there might also be constraints on this: obvious possibilities are signals which control the positioning of nucleosomes to maximize accessibility of the TF binding sites, and compositional biases around the TSS which make it easier for the transcription complex to "open" the DNA helix. Moreover, individual transcription factors must make contact with other proteins in the transcriptional complex. This means that some TFs might have preferred positions in the complex, and their corresponding binding sites in the promoter region will function more effectively if they are in particular positions (relative either to the transcription start site, or to one another).

1.3. Resources for studying promoters

Since I was addressing this project from a computational point of view, I have been reliant on published experimental work to provide useful sets of data which can be analyzed in order to learn more about promoters. Particularly valuable are large collections of data, either compendiums of individual experimental results, or the output from single high-throughput data collection exercises. The following resources are highly relevant to the study of eukaryotic promoters, and were considered throughout this project.

1.3.1. EPD: the eukaryotic promoter database

The Eukaryotic Promoter Database [Périer *et al.* 2000, EPD, http://www.epd.isb-sib.ch/] is, as the name suggests, a collection of promoter sequences from a range of organisms in the eukaryotic domain. Sequences are accepted according to a set of fairly demanding criteria, which are set out in the user manual [http://www.epd.isb-sib.ch/current/usrman.html]. Notably, transcription start sites must be mapped experimentally, and researchers submitting entries are expected to identify the main TSS with an accuracy of ±5 bases. The one exception to this is for promoters which are identified purely on the basis of a strong homology to a previously characterized sequence. However, when using databases such as EPD for training or testing sequence analysis methods, it is normal to work with a non-redundant subset of the sequences, so all sequences which were accepted into the database on the basis of homology should be discarded anyway.

Although EPD is named as a promoter database, it does not necessarily give information about what biologists would consider to be promoter regions. A basic EPD entry just specifies a transcription start site (which is presumed to corresponds roughly to the downstream ends of the promoter regions). This is not necessarily a defect:the concept of a transcription start site can be defined fairly rigorously, and experimental techniques for TSS mapping, such as primer extension [Green and Roeder, 1980], are well known and accepted. On the other hand, it is somewhat harder to say where the exact boundaries of an active "promoter region" lie, and even the previously described strategy of making a series of targeting deletions is unlikely to give an absolutely precise definition of the required region.

One point to note about EPD is that its contents are the results of fairly laborious experimentation, and it is therefore quite small. Recently, it has been growing extremely slowly. Release 62, dating from 2000, contained 389 mammalian promoters (after discarding near-duplicates). Two years later, in release 71 of the database, the number of distinct mammalian promoters had only grown to 400. Very recently, the database *has* grown much more rapidly, with over 2000 promoters in release 73, but the balance consists almost entirely of TSS predictions made on the basis of full-length cDNA sequencing projects, as described below. These methods have clear advantages in that they are comparatively easy to implement in high-throughput environments, but it is not yet clear that they offer TSS mapping of the accuracy offered by "traditional" EPD entries. Therefore, researchers may still wish to work with the traditional entries. I made significant use of EPD in this project, but all work on this database pre-dated the addition of the cDNA-based entries.

1.3.2. Full-length cDNA sequences

Complementary DNA (cDNA) is a general term for any DNA molecule produced from an RNA template by the enzyme reverse transcriptase – an enzyme which is required in the life-cycle of the retroviruses, but which has also proved to be a very valuable molecular biology tool. In practice, cDNAs of interest are generally DNA copies of messenger RNAs, produced in the laboratory as a method of studying the transcriptome. Sequencing of cDNAs has been taking place for many years. Initially, the emphasis was on sequencing just representative portions of the mRNA molecules, giving expressed sequence tag (EST) sequences [Adams *et al.* 1991]. But several recent projects have attempted to clone and sequence the complete length of messenger RNAs. The issue here is to distinguish between cases where the cDNA represents the full length of mRNA found in the cell, and those where it is truncated for some reason. Likely causes for truncation are the reverse transcriptase dissociating before reaching the end of the mRNA, or simple degradation of the RNA (which is chemically rather unstable *in vitro*). Recent projects such as the FANTOM collection of mouse cDNAs [Kawai *et al.* 2001] and DBTSS, a similar human project [Suzuki *et al.* 2002], have used an approach called cap-trapping. This takes advantage of the fact that all eukaryotic mRNAs have a specially modified nucleotide residue (the cap) present at their 5' ends. This is added by specialized capping enzymes which are thought to function cotranscriptionally [Proudfoot *et al.* 2002]. A technique exists in molecular biology to attach oligonucleotide markers specifically to RNAs which possess this cap nucleotide, as illustrated in figure 1.6. Once this step has been performed, cDNAs are produced as normal, then cloned into plasmids using a protocol which requires the presence of a particular tag sequence – matching that of the oligonucleotide which was added in the cap-trap stage. In this way, it is possible to build a library consisting primarily of 5'-complete cDNA clones [Maruyama and Sugano, 1994].

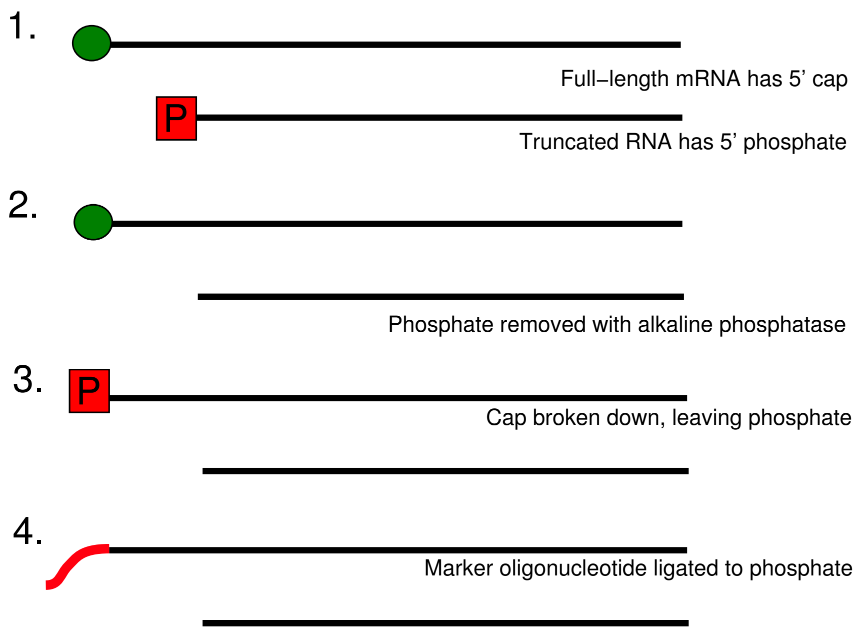
1.
Full–length mRNA has 5' cap
P
Truncated RNA has 5' phosphate
2.
Phosphate removed with alkaline phosphatase
3.
P
Cap broken down, leaving phosphate
4.
Marker oligonucleotide ligated to phosphate

Figure 1.6. A protocol for oligonucleotide-labeling of mRNAs with intact cap structures, used to preferentially clone full-length mRNA sequences.

If it was effective, full-length cDNA sequencing would be an attractive way to determine transcription start sites. However, it is not a complete solution to the problem: firstly, it may be difficult to isolate cDNAs from genes which are expressed at low levels, or from rare variants transcribed from alternatives sites. Second, there remain questions about the effectiveness of the cap-trapping procedure. Cap-trapped sequences were used in this project, but rather than assuming that the 5' end of one of these clones always reflects a true transcription start site, some caution was taken to use only those clones which appear most likely to be complete. Some results which seem to justify this cautious approach are given in chapter 3.

1.3.3. TRANSFAC

TRANSFAC is a database of known transcription factors and their binding sites – both natural sites identified by *in vivo* study of promoters and synthetic oligonucleotides to which the factors have been shown to bind *in vitro* [Matys *et al.* 2003]. The database is made up of several parts:

- A database of known transcription factor proteins (transfac)
- A database of binding sites for many of these factors (tfsite)
- A database of position weight matrices (see section 1.4.1) describing the preferred binding sites of a few factors where a sufficiently large number of distinct binding site sequences are known (tfmatrix)

The database of factors is clearly of interest when studying gene regulatory networks. Using the database of sites is more difficult: this contains a large number of entries (8414 in release 3.4) with binding site sequences varying for 2 to 128 bases in length (strongly biased toward the low end of that range). This is a diverse set of sequences: it includes all 1024 possible 5-letter DNA "words", over 99% of the 6-letter words, and 84% of 7-letter combinations. Even considering 9-mers, almost 13% of them can be found in, so the probability of any sequence of less than 10 bases being present in tfsite is high. While I noted the existence of TRANSFAC, is was not directly useful in the course of this project.

1.4. Existing computational methods for studying promoter

Before this project, a number of attempts had been made to develop methods which could automatically annotate promoters regions or transcription start sites in eukaryotic DNA. Several attempts were made in the pre-genomic era to apply well-known sequence analysis methods such as hidden Markov models [Audic and Béraud-Columb 1997] and neural networks [Knudsen 1999] to the problem of promoter prediction. Work in this era was reviewed in [Fickett and Hatzigeorgiou 1997]. The authors of that review also performed an evaluation of the available methods on a small set of newly-published test sequences. This independent evaluation had the advantage that all the test sequences were obtained after the evaluated methods were published, making that the test entirely impartial since none of the test sequences could have been used when training and developing the prediction methods. However, the test set consisted only of 18 sequences, all of which were rather short (12 were less than 2 kilobases in length), and with some clear problems for any promoter recognition software – in one case, the mapped TSS was only 28 bases from the start of the sequence, giving little information for any method which looks for upstream signals. The evaluated methods all performed better than would be expected for random predictions, but in all cases, specificity was low. In the modern era of sequencing, where researchers often want to look at whole mammalian genomes rather than 2kb regions of interest, specificity is a vital requirement, as discussed in [Scherf *et al.* 2000].

Below, I discuss three specific methods of promoter recognition: one based on a machine learning approach, and two which are more biologically motivated. These methods will appear again in chapter 3 of this report, as benchmarks used when evaluating the EponineTSS method.

1.4.1. Well-known motifs and Position Weight Matrices

There are a number of short sequence motifs which have been associated with transcription initiation. Most of these are believed to represent preferred binding sites for one particular transcription factor. Few of these are 100% specific – in other words, there is not one single sequence which is recognized, but instead there are several choices of allowable nucleotide for at least some positions in the motif. Given a large number of examples, a good way to represent knowledge about a motif is a Position Weight Matrix (PWM). This is a matrix where each element represents the probability of a given nucleotide occurring at a particular position in the sequence. In other words, each column of the matrix is a probability distribution over the DNA alphabet. A PWM can be viewed as a probabilistic model of a fixed-length sequence:

$$\overrightarrow{W}(S) = \prod_{i=1}^{|W|} W_i(S_i) \tag{1.4.1}$$

Where $|W|$ is the length of the PWM, and W_i is the i'th column of the matrix. The higher the probability of a given sequence under this model, the more similar the sequence is to the “ideal” motif, and therefore the higher the probability that it will function as a binding site. This is the zeroth order model: each position in the sequence is assumed to be independent of all the others. Here, $\overrightarrow{W}$ and $\overleftarrow{W}$ are used to indicate weight-matrix scores for the forward and reverse DNA strands respectively.

Considering a PWM (or other motif description) as a probabilistic model offers an approach for learning optimal PWMs from a set of sequences. A PWM can be seen as a degenerate form of a hidden Markov model [Durbin *et al.* 1998]. HMMs are state machines where each state has an associated emission spectrum over some alphabet of possible observations. In a PWM, the transitions between the states are fixed: after observing the nucleotide at each position, the machine always moves to the next state. But by adding a few additional states to the HMM, it is possible to build a model which emits a variable number of bases of flanking sequence on each side of the motif (figure 1.7). The set of parameters for an HMM which maximize the probability for a set of sequences (often called the maximum likelihood estimate of the parameters) can be found by applying the Baum-Welch algorithm [Durbin *et al.* 1998]. A concrete implementation of motif-learning using this approach is provided by the MEME package [Bailey and Elkan 1994].

It should be pointed out that the assumption of bases being independent in a motif is unlikely to be true in most cases. When proteins bind to DNA, they often significantly deform

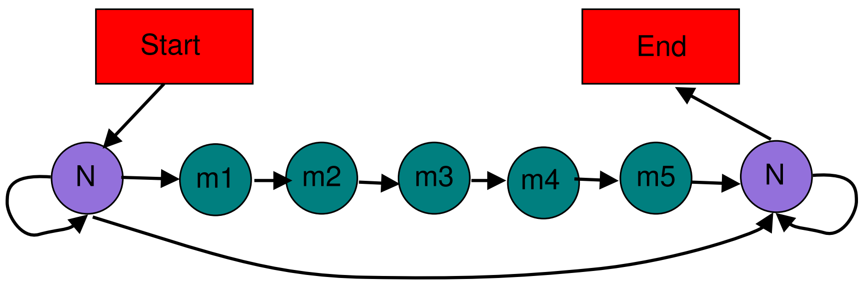

Figure 1.7. Hidden Markov Model of a motif (states m1 to m5) embedded in a longer sequence.

it, so mechanical properties of the DNA other than the actual base sequence will be significant – in particular, the flexibility of the double helix. This property is determined largely by interactions between neighbouring base pairs. Models which take these, and perhaps also longer range interactions, into account can be expected to better predict the binding of a protein to a given sequence [Barash *et al.* 2003]. However, since non-independent models have many more parameters than simple PWMs, they require more example sequences to learn effectively.

Sequence motifs are commonly displayed in logo form – see, for example, figure 1.8. Here, each column of a PWM is rendered as a stack of symbols, with more height given to the most probable symbols. The total height of the stack is proportional to the Shannon information content of the distribution:

$$S = \log_2(|\mathcal{A}|) + \sum_{n \in \mathcal{A}} P(n) \log_2 \frac{1}{P(n)} \tag{1.4.2}$$

Where $\mathcal{A}$ is the alphabet we are considering. This quantity is related to entropy, and is measured in bits. For a four-letter alphabet such as DNA, values can range from 0 (for a completely uniform distribution, *i.e.* $P(n) = 0.25$ for every base) to 2 bits (for the case where only a single base is allowed). Therefore, tall stacks indicate strong constraints on the bases which are acceptable at a given position, while short stacks show positions with only a marginal preference for any particular base. Completely non-informative positions appear blank in the logo.

The TATA box is the best known element of eukaryotic promoter sequences, and was recognized more than 20 years ago in early experiments with eukaryotic gene expression [Corden *et al.* 1980]. The TATA binding protein (TBP) is known to associate with the TATA box, when it is present. However, this association is weak, and TBP (which plays a central role in transcriptional complexes) is actually present even when no TATA box can be found [Rigby 1993]. The definitive TATA box PWM was learned by applying an maximum likelihood algorithm to sequences from EPD, and was published, along with several other motifs, in [Bucher 1990]. The logo form of this is shown in figure 1.8.

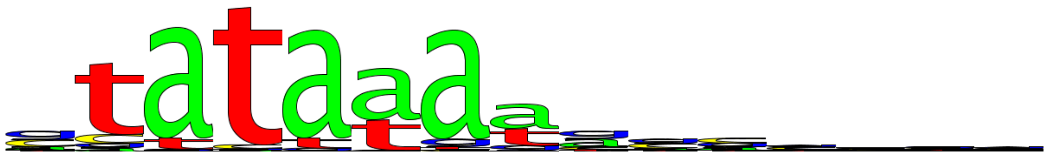

Figure 1.8. Logo view of a Position Weight Matrix model of the TATA box (redrawn from data on the EPD website)

The TATA PWM has been used on its own as a promoter prediction method, and was one of the candidate methods included in [Fickett and Hatzigeorgiou 1997]. Further information about the performance of this method appears in chapter 3.

PWMs are a widely used technique in computational biology, and are by no means specific to promoter research: another common application is to model the sequences around splice-junction sites in gene prediction programs. Profile HMMs are close relatives of PWMs, which allow small insertions and deletions relative to the expected consensus sequence, and are used by the Pfam project to identify families of evolutionarily-related proteins [Bateman *et al.* 2002].

1.4.2. CpG islands

As previously mentioned, the the cytosine in the dinucleotide 5'-cytosine-guanine-3' can be covalently modified with a methyl group. Indeed, in mammalian genomes, this is extremely common and the vast majority of CpGs are methylated. CpG methylation is believed to convey some information, since DNA is methylated in the course of some mechanisms of gene silencing [Razin 1998]. However, storing extra epigenetic information by methylation does not come without a cost. Cytosine bases are prone to deamination. Deamination of normal cytosine yields uracil, which is not normally present in DNA, and can be efficiently recognized and repaired. But the deamination product of methylcytosine is thymine, a normal DNA base. This leaves a thymine-guanine pair in the DNA helix, which may be detected and resolved by mismatch-repair mechanisms. This kind of DNA repair is much less efficient than the rapid detection and elimination of uracil, and a substantial proportion of methylcytosine-to-thymine mutations survive to be replicated. Sequencing of mammalian genomes has shown that CpG dinucleotides are rather rare. If bases in DNA were independent of their neighbours, we would expect the frequency of CpG dinucleotides to be given by:

$$F_{CG} = F_C F_G \tag{1.4.3}$$

In fact, over almost any stretch of mammalian sequence, F_{GC} is vastly lower than this expected value. But some regions of the genome contain much higher CpG levels than average: these are called CpG islands, and are known to be associated with the 5' ends of genes [Larsen *et al* 1992]. It has been further suggested that genes associated with strong CpG islands have "housekeeping" functions – in other words, they encode proteins which are ubiquitous throughout many cell types and developmental stages [Brandeis *et al.* 1993].

The accepted definition of a CpG island, taken from [Gardiner-Garden and Frommer, 1987], is a region of at least 200 bases where:

- At least 50% of bases are G or C
- The observed content of CpG dinucleotides is at least 60% of the "expected" figure from equation 1.4.3 for a random sequence with the same single-nucleotide composition as the region under consideration

A number of programs exist to detect CpG islands in genomic DNA sequences using the criteria given above, and these are commonly used by curators annotating gene structures. When the 5' end of a gene structure annotated primarily on the basis of cDNA or EST evidence falls close to a CpG island, this is a good indication that the structure is not badly truncated. However, CpG islands do not provide strong information about the actual position of the transcription start site. The CpG dinucleotide is palindromic: reading the other DNA strand in the 5'-to-3' direction, you will also see C followed by G. This means that for a given region of sequence, F_{CG} will be the same for both strands, so the predicted CpG islands give no information about the direction of transcription.

1.4.3. PromoterInspector

PromoterInspector is a recent method for predicting promoter regions which works on the basis of detecting multiple motifs [Scherf *et al.* 2000]. It was trained based on sequences extracted from EPD, using a brute force method to determine sets of motifs which are over- or under-represented in promoter regions relative to the genome as a whole. In this case, the motifs are not represented as PWMs, but as simple strings, where some of the positions can be wildcard characters. This is equivalent to a degenerate PWM, where all columns have an information content of either 2 bits (requiring an exact match) or 0 bits (non-informative). This is likely to be less sensitive than allowing all possible PWMs, but has the advantage that it is possible to rapidly enumerate the complete set of patterns up to a given length, and count their occurrences in a set of training data. This makes the brute-force training approach practical.

By design, PromoterInspector does not give direct information about transcription start sites. Instead, it marks "promoter regions" on the genome. However, since the positive training set consisted simply of blocks of sequence 500 bases upstream of EPD transcription start sites, rather than mapped, active, promoter regions, there is no particular reason to believe that the boundaries of the predicted promoter regions correspond to the portion of the sequence which is actually biologically significant. A more conservative description would be "regions likely to contain core promoter elements". Like CpG island predictors, PromoterInspector also gives no information about the direction of transcription.

1.5. Other resources used in this project

The field of bioinformatics is characterized by large, complex data sets. For bioinformaticians concentrating on sequence analysis, the most significant recent developments have been the publication of various genome sequences, with the draft sequences of higher organisms such as human, mouse, and *Fugu* being particularly exciting. The availability of genome data is, of course, vital to this project. But the scale of genome data – plus complications specific to bioinformatics, such as the regularly changing assemblies of draft genome data – mean that they are relatively difficult to manage and work with. The following tools and resources were invaluable in the course of this project.

1.5.1. BioJava

BioJava is an open source library of tools and components for developing bioinformatics applications [The BioJava development group, http://www.biojava.org/]. It draws some inspiration from earlier projects like Bioperl [Stajich *et al.* 2002], but follows rather different design approaches, and places more emphasis on supporting developers working on new analysis methods, rather than manipulating output from existing tools. Most of the code in current versions is aimed at manipulating and analyzing sequence data, but future versions are expected to improve support for other data types, and integration of data sets from disparate sources. In use, BioJava has proved quite scalable: it is quick and convenient to work with small pieces of sequence data, but exactly the same Application Program Interfaces (APIs) can also be applied to much larger sets of data. With extensions like *bj-ensembl*, described below, it is easy to handle and query databases larger than the computer's memory.

One defining pattern in BioJava development has been the use of query languages in preference to defining large sets of *getFooByBar()* accessor methods. The strongest example of this is the handling of annotated sequence, which can be queried using the *FeatureFilter* language:

```
// Locate all gene annotations on the positive strand,
// overlapping the specified region

Sequence seq = loadSequence();
FeatureHolder features = seq.filter(
    new FeatureFilter.And(
        new FeatureFilter.ByType("gene"),
        new FeatureFilter.And(
            new FeatureFilter.OverlapsLocation(
                new RangeLocation(1000, 2000)
            ),
            new FeatureFilter.ByStrand(
                Strand.POSITIVE
            )
        )
    );
```

Simply allowing features to be requested based on a single criterion (type, or location) can be highly inefficient when working on large datasets, since this would require the program above to fetch *all* features in the requested region, then throw away all those which aren't genes on the positive strand. On the other hand, providing accessor methods for all the combinations of criteria which users might wish to access quickly gives unwieldy APIs which are hard to learn, and even harder to maintain and test effectively. By passing in an object which represents a query, users can naturally combine all the available filters operators using the *and*, *or*, and *not* operators, and easily ask complex questions. The underlying implementations can either implement the query system directly, or transform the query into some other language – for example, some BioJava database front ends can directly transform *FeatureFilter* objects into SQL queries. Plans are currently underway for BioJava 2, which will provide a single, more consistent, query language for a wide variety of data types while following a similar spirit to the BioJava 1.x *FeatureFilters.*

I have been involved in BioJava development since the pre-1.0 development which began in late 1999. During this time, I have worked on many areas of the code base, although I have a specific interest in handing of annotated sequence data, and database interfaces such as the BioSQL [OBDA, http://obda.open-bio.org/] and Ensembl (see below) access modules. BioJava code was used extensively in implementing the methods described in this project, and also for the large number of one-off scripts and small programs which were needed to prepare data sets and to analyze and present the results.

1.5.2. Ensembl

Ensembl began as a project to carry out a first-pass automatic annotation of the human genome, and the scope has since expanded to cover a wide range of eukaryotic genomes, and provide sophisticated web-based interfaces for accessing and querying these resources [Hubbard *et al.* 2002, Clamp *et al.* 2003]. The most notable aspect of the Ensembl data is the high-quality set of gene predictions, which are produced by a hybrid method using a range of evidence type, aligned to the genome using the Genewise tool [Birney 1999] and est2genome [Mott 1997], and also some *ab initio* computational predictions. The gene-builder module, which combines these data and produces final predictions of gene structures, is considered to have a very low rate of overprediction, particularly in comparison with the purely *ab initio* methods. Ensembl predictions are all given ID numbers, starting ENSG for predicted genes, ENST for distinct transcriptions, ENSE for individual exons, and ENSP for protein products. These IDs are stable: if a gene predicted on the current assembly exactly matches one on a previous number, the ID is reused. For computational biologists working with large sets of predicted genes, ENS* IDs can be a convenient tool, avoiding traditional arguments about gene nomenclature, and allowing newly predicted genes to be uniquely identified even if no nomenclature has yet been defined.

All Ensembl data – both the raw sequence and the results of the gene build and other analyses – is stored in a relational database. In the course of this project, I developed a module called *bj-ensembl*, which plugs into the main BioJava libraries and provides seamless access to the core Ensembl databases using the standard BioJava interfaces. In the example below, only the first two lines of code are specific to programs working with an Ensembl database. All the querying code is applicable to other types of database, or even (given enough memory to load a whole genome!) objects loaded from normal flat files.

```
// Connect to an Ensembl database

Ensembl ens = new Ensembl(/* database details */);
SequenceDB chromomsomes = ens.getChromosomes();

// Find all CpG islands in a particular region

Sequence chr = chromosomes.getSequence(22);
FeatureHolder cpgs = chr.filter(
    new FeatureFilter.And(
        new FeatureFilter.ByType("cpg"),
        new FeatureFilter.OverlapsLocation(
            new RangeLocation(20000000, 21000000)
        )
    )
);

// Find transcripts of a particular gene

FeatureHolder trans = chromosomes.filter(
    new FeatureFilter.ContainsAnnotation(
        Ensembl.TRANSCRIPT_GENEID,
        "ENSG00000135457"
    )
);
```

Ensembl data was used heavily throughout this project. It proved valuable both as a simple database containing genomic sequences in a more convenient form that flat files, and as a source of high-quality gene predictions.

Ensembl data is presented *via* a sophisticated web interface, which includes a graphical sequence viewer (contigview) as well as many different report pages. It is possible to add extra “tracks” of information to contigview displays using by publishing the data using the DAS protocol [Dowell *et al.* 2001]. I developed a BioJava-based DAS server called Dazzle

[Down and Pocock 2001]. I used Dazzle and contigview to visualize many results from this project in their genomic context.

1.5.3. The Gene Ontology (GO)

The Gene Ontology is a controlled vocabulary for the annotation of genes [The Gene Ontology Consortium 2000]. It is divided into three portions:

- Molecular functions (e.g. “DNA binding”)
- Biological processes (e.g. “Transcription”)
- Cellular components (e.g. “Nucleus”)

Terms in GO are accompanied by human-readable definitions. These aim for rigour, with the hope that data annotated with GO terms by one group will be directly comparable with results from another group, who might be working on another species and come from a quite different background. For a computational biologist who wishes to ask questions along the lines of “are the genes in this (large) set more likely than average to perform some specific function”, statistics based on GO annotation are expected to be far more robust than alternatives, such as lexical analysis of keywords in human-written gene descriptions.

Rather than just offering a flat set of terms, GO also defines relationships between them. These are what mark out GO as a kind of ontology, rather than merely a controlled vocabulary. The relations in GO are either *is-a* or *part-of*. Both types are transitive, i.e. if A *is-a* B and B *is-a* C then A *is-a* C. This property is useful, since it means that it is reasonable to consider the set of all “descendants” of a given term (the transitive closure). This operation makes it possible to take data annotated with a complex vocabulary such as GO and slice it up at an arbitrary level. So a gene annotated with the molecular function “zinc-mediated transcriptional activator” will be pulled out by a specific query fr that particular term, but also for more general queries, such as “transcriptional regulation” or “nucleic acid binding”. Both the relation types are directed,

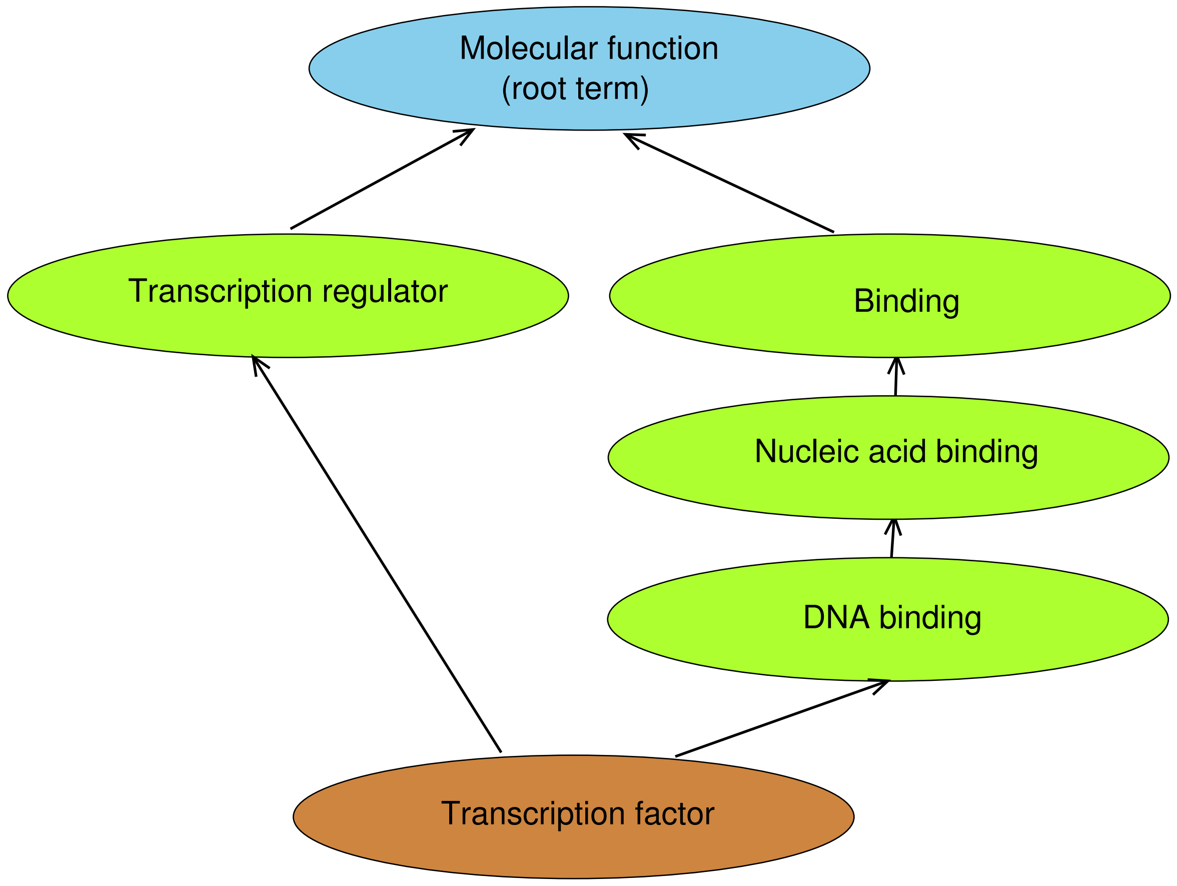

Figure 1.9. Example of a directed acyclic graph of Gene Ontology terms.

so each relationship can be considered to link a parent and a child term. Each term can be in relation to more than one parent, so the ontology forms a directed acyclic graph, as illustrated in figure 1.9.

Of course, a vocabulary is not useful on its own, but becomes relevant when it is used to describe objects of interest. Some genome projects, such as those for *Drosophila* and fission yeast, now use GO terms in all their curated annotation. For mammalian genomes, the current state of the art in GO annotation comes from the GOA project [Camon *et al.* 2003]. The basis of this project is curated annotation of selected Swissprot and Interpro entries with GO terms. Other protein sequences are then annotated on the basis of similarity to entries with curated GO terms. This annotation has been picked up by the Ensembl project, who use it as the basis for GO annotation of their gene predictions. Whenever GO annotation is used in this project, this refers to GOA results which have been mapped to Ensembl genes in this way.